Bioinformatics for plant genome annotation

Mark Fiers

Promoter:

Prof. Dr. W.J. Stiekema
Hoogleraar Genoominformatica
Laboratorium voor Bioinformatica
Wageningen Universiteit

Copromoter:

Dr. Ir. J.P. Nap
Senior onderzoeker
Plant Research International
Wageningen Universiteit en Researchcentrum

Promotiecommissie:

Prof. Dr. Y. van de Peer, Universiteit Gent, België
Prof. Dr. Ir. J.J. van Wijk, Technische Universiteit Eindhoven
Prof. Dr. R.G.F. Visser, Wageningen Universiteit
Prof. Dr. J.A.M. Leunissen, Wageningen Universiteit

Dit onderzoek is uitgevoerd binnen de onderzoeksschool Experimental Plant Sciences

Bioinformatics for plant genome annotation

Mark Fiers

Proefschrift
ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van Wageningen Universiteit
Prof. Dr. M.J. Kropff
in het openbaar te verdedigen
op woensdag 8 november 2006
des namiddags te half twee in de Aula

CIP-DATA Koninklijke Bibliotheek, Den Haag

Bioinformatics for plant genome annotation

Mark Fiers

PhD thesis, Wageningen University, the Netherlands
With references - with summaries in English and Dutch

ISBN 90-8504-519-3

Contents

Chapter 1

Introduction

This thesis is about the role of bioinformatics in the process of genome annotation.

Genome annotation is the process of assigning biological interpretation to a DNA sequence. A DNA sequence, as a string of nucleotides, has limited use in application and research. Various analyses are required to assign biological interpretation to a DNA sequence. The goal of genome annotation is to describe the function of every single nucleotide, in any cell or cell compartment, during the reproduction and the life span of an organism.

The need for bioinformatics in genome annotation became evident upon the completion of the first genomes [49, 156]. Bioinformatics is the multidisciplinary approach that combines, amongst others, molecular biology, information technology, mathematics and statistics in the automated analysis of bio-molecular data. The term 'bioinformatics' appeared in scientific literature somewhere in the 1980's. It has its roots in fields as theoretical and computational biology. Nowadays over 300 genomes have been sequenced [62]. The annotation of a single genome is an intensive task, from both a computational as a biological perspective, confirming the importance of bioinformatics in genome annotation.

A genome is not annotated by bioinformaticians alone, but in close cooperation with biologists. Biologists deliver the raw data and biological context for the annotation of a sequence. Often this results in new hypotheses that lead

to more experiments by both biologists and bioinformaticians and ultimately contribute to the advancement of biological understanding.

The next six chapters describe the different aspects of the application of bioinformatics in genome annotation that were investigated. Chapter 2 presents an overview of the role of bioinformatics in genome annotation. The chapter focuses on computational annotation of both protein-coding and non-protein-coding DNA. Chapter 3 describes the development of an automated system for high-throughput genome annotation. It deals with the information science behind genome annotation: how to organize the data flow and execution of analyses. Different analyses, such as gene finding and similarity searching are organized in a reliable package for automated annotation. Chapter 4 focuses on the second subject from information science: the visualization of annotation data. Visualization of such data helps the biologist in using the data for research purposes. The chapter describes the development of a software package for the interactive visualization of heterologous annotation data. Chapter 5 describes the prediction of the full microRNA potential of the genome of *Arabidopsis thaliana*. MicroRNAs are small RNA genes [38] that were only recently recognized as a gene regulatory mechanism. In an approach that does not depend on conservation of miRNA candidates over multiple species, well over a thousand candidates are predicted. Many of the microRNAs of *Arabidopsis* nowadays validated in laboratory experiments are not in the predicted set. This indicates that the actual number of *Arabidopsis* microRNAs, as predicted in this chapter, may be considerably higher than expected so far. Chapter 6 of this thesis presents the prediction of allergenicity of proteins on level of amino acids. Allergenicity of proteins for human (or animal) consumption is a major issue in the evaluation of genetically modified (GM) food. Chapter 6 describes the development of a website that uses the guidelines from FAO/WHO [45, 46] to evaluate the potential allergenicity of a sequence entered by a user. The discussion in the final chapter 7 places all research work in the context of current genome annotation and future prospects of that field of bioinformatics.

Chapter 2

Bioinformatics for genome annotation

Mark Fiers, Jan Peter Nap, Roeland van Ham

2.1 Introduction

The past 10 years has been the decade of genome sequencing. Since 1995, in which the first complete genome sequence from a free-living organism, the bacterium *Haemophilus influenzae* [49], was published, nearly 300 prokaryotic and 40 eukaryotic genome sequences have been finished and about 1500 more are currently in progress [62]. Among the completed genomes from eukaryotes are those of human and various important model organisms, including mouse, rat, *Drosophila melanogaster* (fruit fly), *Caenorhabditis elegans* (a nematode), *Saccharomyces cerevisiae* (baker's yeast), *Arabidopsis thaliana* (thale cress) and rice (Figure 2.1).

The success of modern genome sequencing is based on several advancements in sequencing technology and strategies made in the nineties. Of great importance was the introduction of the shotgun sequencing strategy in whole genome sequencing of large genomes. The strategy was already proposed and applied successfully to the sequencing of the genome of phage Lambda by

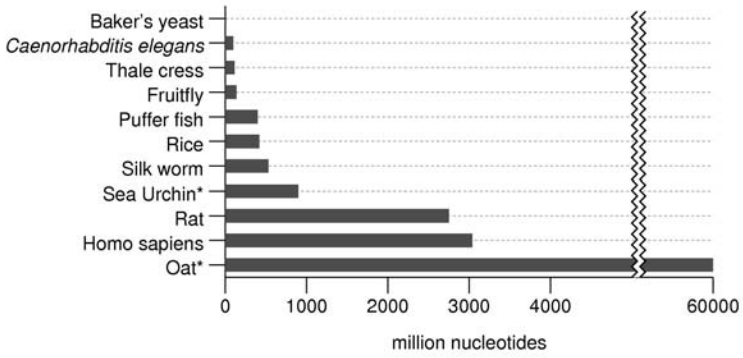

Figure 2.1: A small set of genomes of which the complete DNA sequence is either finished or underway (marked with an *) and the size of that genome [62, 198]

Sanger and co-workers in 1982 [157]. It involves the random and redundant sequencing of large numbers of genome fragments and piecing these together afterwards in a computational process called genome assembly. New algorithms and powerful computers, permitting the assembly of large amounts of sequence data, proved the use of whole-genome shotgun sequencing feasible for megabase-sized genomes in the *Haemophilus* genome project, leading to its application in the sequencing of the human genome [195]. A breakthrough in sequencing technology came with the introduction of automated capillary sequencing equipment in 1998, which tremendously increased sequencing throughput [94]. Thanks to this equipment and the competition between public and private initiatives, the first draft sequences of the human genome were completed considerably ahead of schedule [94, 195].

The biochemical basis of current sequencing technology still relies on the chain-termination method that was developed by Sanger and co-workers as early as in 1975 [158]. In combination with today's sequencing equipment, this method appears to have reached its maximal throughput of two to three megabases per day per machine [86]. It remains a relatively laborious and costly technology. The interest in the scientific community to sequence many more genomes from a great diversity of organisms is calling for innovations that should bring the "$1000 genome" within reach of small research laboratories. Among the more promising new technologies are the single-molecule array approach developed by Solexa [13] and massively parallel pyrosequencing method developed by 454 Life Sciences [109]. It is expected that these new technologies will soon

outperform current capillary DNA sequencers [109] in terms of performance and costs and will bring about a new flood of genome data, at least an order of magnitude larger than produced in the past decade.

Technological advancements in the past 10 years with an impact on bioinformatics for genome annotation are not confined to DNA sequencing alone. Major breakthroughs have also occurred in the development of technologies with which other cellular components can be measured on a large scale. Collectively, these new technologies are known as "omics-technologies". Similar to the way the term "genomics" is used to denote the study of whole genome sequences, the terms transcriptomics, proteomics and metabolomics describe the genome-wide analyses of RNA transcripts, proteins and metabolites, respectively. Transcriptomics refers to the comprehensive genome-wide analysis of gene expression at the mRNA level with the help of micro-array [161], SAGE [194] or MPSS [116] technologies. In proteomics, gene expression is analyzed at the level of proteins, employing, among others, chromatography, large-scale 2D gel electrophoresis and mass spectrometry [2]. Metabolomics is the most recent development in omics technologies that aims at identifying all metabolites in all cells [72].

A mere ten years ago, research was predominantly hypothesis-driven and genes, transcripts, proteins, and metabolites were studied on a one-by-one basis. Today, methodology has shifted to exploratory investigation of larger biological systems. A researcher will now typically analyze a large or even complete set of cellular components in a single experiment. This way, high-throughput omics technologies have made their way into almost every area of research in the life sciences and have had a profound impact on its scientific methodology. Laboratories all over the world have acquired the means to rapidly generate enormous amounts of biomolecular data. The need to efficiently handle and analyze these has called for new computational solutions in the form of databases, user-friendly software and powerful hardware. The development or implementation of such resources is the subject of bioinformatics, the multidisciplinary approach that combines molecular biology, information technology, mathematics and statistics.

Although bioinformatics has grown spectacularly concomitantly with omics research (Figure 2.2), it is not an entirely new scientific discipline. It is rooted

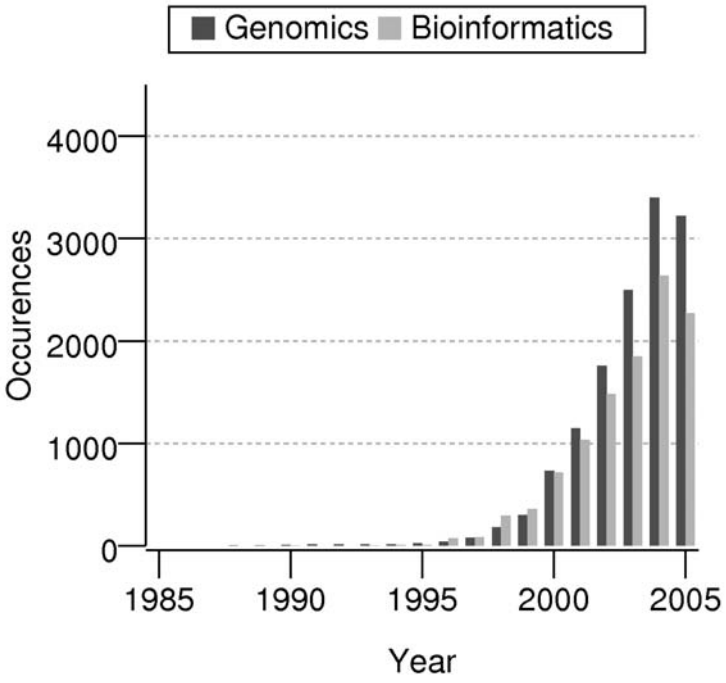

Figure 2.2: Increase in the number of PubMed records in the past two decennia, containing the keywords "bioinformatics" or "genomics".

in the late sixties with the work on molecular sequences by Margaret Dayhoff, Saul Needleman, Christian Wunsch, Walter Fitch and others. It took further shape in the early eighties when computational tools were developed to manage and analyze a growing body of biomolecular sequence data. Among its early achievements are, for instance, the establishment of the first molecular sequence database in 1982, the EMBL Nucleotide Sequence Data Library, and the development of tools for the alignment and database comparison of DNA and protein sequences, including the Smith-Waterman algorithm [170] and the FASTA package [130]. Bioinformatics now facilitates many stages of the high-throughput approach to biological research, from experimental design to the final steps of analysis and web-enabled integration of results with information available from public databases. Without bioinformatics, the practice of modern, large-scale biology research would be unthinkable.

Genome sequence data are being produced at an increasing pace, but a DNA sequence in itself, as a string of nucleotides, has a relatively limited use for

application and further research. Bioinformatics delivers the computational methodology and tools necessary to interpret and assign biological significance to a DNA sequence. The branch of bioinformatics committed to assigning biological meaning to a genome sequence is called "genome annotation". The ultimate goal of genome annotation is to describe for each and every nucleotide its role during the life span and reproduction of an organism. The remainder of this chapter is devoted to genome annotation.

Genome annotation falls into two distinct stages. The first stage is referred to as structural annotation and involves the correct identification and localization of distinct sequence elements such as genes, regulatory elements, transposons, repetitive elements and more. The second stage, termed functional annotation, attempts to predict the biological function for each of those elements and the biological process in which it takes part. In the next two sections we will describe the details of structural and functional genome annotation, followed by a description of key computational aspects of current high throughput genome analysis. It is important to keep in mind that most methods discussed below are fully computational and therefore provide predictions of gene location and structure. In the end, such predictions will have to be validated in the laboratory.

2.2 Structural genome annotation

A genome can be divided into two parts: one comprising the protein and RNA-encoding genes and the other, non-coding DNA. The definition of a "gene" is a somewhat contentious issue [129, 171]. The term is often used to refer to those segments of DNA that are involved in the production of proteins, excluding other transcribed segments that encode functional or structural RNA molecules. We here use the term "gene" for both types of DNA segments, the boundaries of which are defined by the extent of the primary transcript. Consistent with this definition, we use the term "non-coding DNA" as synonymous with "non-transcribed DNA". The amount of DNA thought to "non-coding" currently shrinks as ongoing research discovers more and more of the DNA to be transcribed [129]. The identification of the protein/RNA-coding comple-

ment of a genome remains the first and foremost task in genome annotation and will therefore receive most attention in the following sections.

Non-coding DNA generally makes up the largest fraction of eukaryotic genomes. For example, in *Arabidopsis*, which is considered to have a relatively compact genome, about 70% of the DNA is non-coding (derived from [71]). This fraction is much larger in many other eukaryotes, as an indication, of the human genome less than 2% is believed to be *protein* encoding [26].

The non-coding part of a genome has long been referred to as "junk-DNA", reflecting a notion that it would be devoid of any biological function. This view of the non-coding part of the genome is changing with ongoing research. For example, mapping the complete transcriptome of a cell with "whole genome arrays" revealed much more transcribed regions than assumed so far [207]. Part of these regions may be yet unidentified protein coding genes. Other areas may reflect novel processes, as for example, endogenous silencing as a gene regulatory mechanism by transcription of the opposite strand [139].

2.2.1 Prediction of protein-encoding genes

Computational identification of a protein-coding gene as defined above in a novel sequence is far from trivial. Methods to identify genes in a newly sequenced genome can be divided into three classes: (I) *ab initio* or *de novo* methods, which predict genes solely on the basis of local sequence characteristics; (II) similarity-based methods, which utilize sequence similarity to known genes, and; (III) comparative methods, which employ sequence comparison between multiple related genomes to identify conserved genes.

Ab initio prediction

Ab initio approaches to gene prediction are based on pattern recognition methods to distinguish a gene from its surrounding sequence. Pattern recognition is a generic name for a family of computational methods that recognize defined features within a sequence (or text). The simplest application of pattern recognition in gene prediction is to find an exact match in a string of nucleotides to a given pattern (also called "word"), for example, a start (ATG)

or stop codon (TAA, TGA and TAG) of a gene. Basic gene predictors, the so-called open reading frame (ORF) finders, combine exact word matching with the requirement that the reading frame between a start and a stop codon or between two stop codons contains an exact multiple of three nucleotides. Such ORF finders look for possible ORFs in all six reading frames and report any ORF longer than a predefined length present within a sequence. An example is the application *getorf* from the EMBOSS package [148].

ORF finding performs reasonably well in localizing the uninterrupted genes in viral and prokaryotic genomes, but it is usually inadequate for the identification of genes in eukaryotic genomes. A number of factors complicate eukaryotic gene prediction. First, the complex structure of eukaryotic genes, these genes are often interrupted by transcribed, yet untranslated sequences, the introns. Introns may contain stopcodons or disrupt the reading frame of preceding exons. Secondly, the boundaries between introns and exons, the so-called splice sites junctions, have only weakly conserved signature sequences in comparison to the strictly conserved start and stop codons (Figure 2.3). Thirdly, exons may be very small (¡100 nucleotides) and at the same time be buried in much longer introns. Gene finding is further complicated by the low gene density in eukaryotic genomes. Consequently, the structural features of eukaryotic genes are difficult to recognize and have a low signal-to-noise ratio. To overcome these difficulties, *ab initio* gene prediction methods for eukaryotic genomes depend on probabilistic models for the detection of structural and nucleotide compositional features. Often, these methods are based on pinpointing the boundaries of probability that a given feature (also called "signal") is present. The detection of individual features is done with the help of several methods, called sensors, which all employ pattern matching technologies and include the prediction of transcription and translation start and stop sites, intron splice sites and protein coding capacity. Most *ab initio* gene predictors apply a combination of sensors that each predict a single structural feature of the gene. The gene predictor algorithm evaluates all possible combinations of features that might form a complete gene model and selects the statistically most significant combinations. Figure 2.4 depicts an example of several sequence features used in eukaryotic gene prediction.

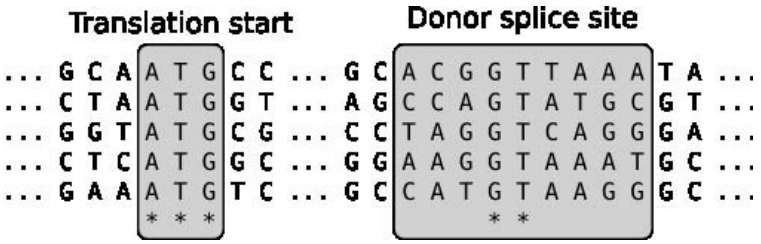

Figure 2.3: Part of a multiple alignment of five DNA regions encoding a gene. The grey boxes indicate gene features used in *de novo* gene prediction. An asterisk under the alignment indicates the perfect alignment on that position. The left box indicates the perfect alignment of the translation start ATG. The right box indicates a donor splice site. Training of a WMM model for the prediction of splice sites based on the right grey box will (approximately) result in the DNA logo depicted in Figure 2.5A.

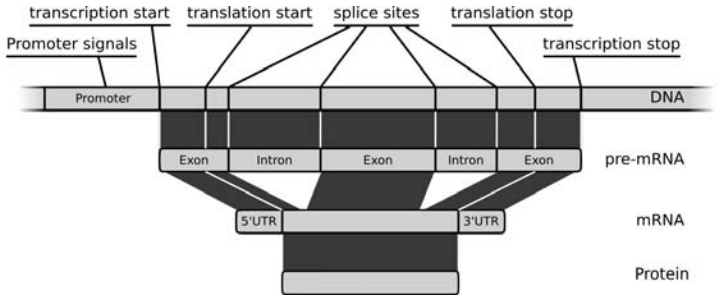

Figure 2.4: Several sequence features used in eukaryotic gene prediction. The features are shown as they are present on the DNA strand (top bar), pre-mRNA, mRNA and the protein (bottom bar).

Popular methods for pattern matching are so-called weight matrix models (WMMs; Figure 2.5A) [176] and Hidden Markov Models (HMMs; Figure 2.5B) [41]. Both methods are able to handle ambiguous nucleotides. For example, the TATA box, involved in the start of translation, is described by the pattern T-A-T-A-[AT]-A-[AT] with ambiguous nucleotides at position 5 and 7. More complex patterns, for instance a pattern that describes variation in indels (insertions and deletions) within a sequence, cannot be described by WMMs and are commonly handled by HMMs. A HMM is a probabilistic model for an ordered series of variables. The state of the variables and all possible transitions between them are unknown and are estimated from observations (Figure 2.5B; [41]). An example of HMMs used in sequence analysis are so-called profile HMMs [40]. These are built from an alignment of a set of sequences and capture all the variation in matches, substitutions and indels

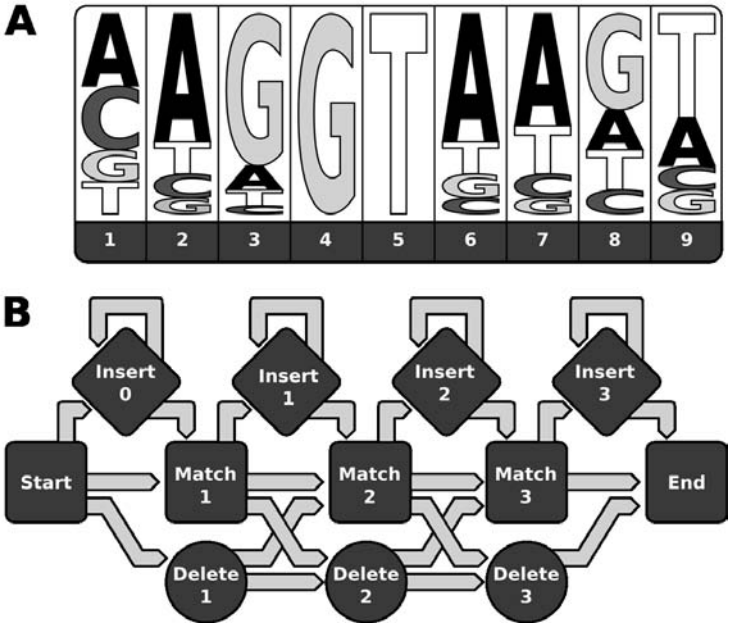

Figure 2.5: A. DNA logo of a *Arabidopsis* splice donor site modelled after Korf [88]. The logo represents the chance of a specific nucleotide occurring at a specific position around the donor site, with the intron starting at nucleotide four. The size of a letter indicates the likelihood that this nucleotide occurs at that position. For example, a G will always occur at position four (according to this model), but at position 1 any nucleotide can be expected. B. An example of a small HMM model (adapted from Eddy [40]) which models a pattern (either nucleotides or amino acids) with a length of three. The nodes are called states, the arrows state transitions. A potential matching sequence is evaluated by the optimal path through the model. For the Insert and Match nodes probabilities are defined (during the training stage) that a certain nucleotide or amino acid appears at this position. As an example, a four letter pattern might take the following path through the model: Start - Match 1 - Insert 1 - Match 2 - Match 3 - End. Multiple inserts at a position are possible through the loop back state transition of the insert states.

present in that multiple alignment as well as all possible transitions between these in the form of a probabilistic model of the sequence. This model can then be used to search a novel genome sequence for the presence of a specific sequence pattern, or profile. For an in-depth description of the use of HMMs in genome annotation we refer to [39, 41, 176]. A widely used implementation of HMMs in DNA analysis is *HMMer* [41, 76].

One of the best known *ab initio* gene prediction tools is GENSCAN [28]. It uses several pattern recognition methods (among which WMMs) to model individual sequence features, that are combined into a complete gene prediction

using a HMM of gene structure. Other implementations of *ab initio* gene predictors are Glimmer [155], Fgenesh [154], Grail [205] and GeneMark [17].

In addition to gene finding tools that predict complete gene models, standalone sensors are implemented in tools that predict a single gene feature. Such tools are useful to consider alternative gene structures. Examples of standalone sensors include NetPlantGene that predicts splice sites [74], NetStart predicts translation start sites [131], or Promoter 2.0 predicting transcription start sites [87].

Both gene predictors and standalone sensors need to be trained. Therefore, the quality of their output depends on the quality and size of the dataset with which they are trained. If insufficient data is available to create an appropriate training set, the prediction program can be trained iteratively. Each iteration is fed with the results from an earlier prediction, usually in a supervised manner in which the initial training is performed by an expert using a dataset from a closely related organism. A recent study, however, indicates that an unsupervised approach that starts with a simplistic gene model may perform as good as the supervised approach [104]. If substantiated, this holds promise for gene prediction and annotation of newly sequenced genomes, for which it is often difficult to compile appropriate training sets.

The performance of most current gene predictors is reasonable, given the complexity of their task. For example, the widely used GlimmerHMM trained with 800 full length *Arabidopsis* cDNAs predicts 70% of all exons correctly and just over 30% of all genes are predicted correctly as complete gene models [108]. This shows that other methods to improve the quality of gene predictions are necessary.

Alignment-based methods

A different approach for gene prediction is to use experimental evidence from transcription to localize genes. It involves for example the comparison of expressed sequence tags (ESTs) or full-length cDNA sequences to the genomic sequence of the same species. Regions where very high quality or perfect local alignments are obtained, represent hypothetical gene locations and can be used to predict complete gene models. (Figure 2.6). As with *ab initio* gene

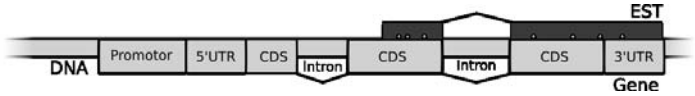

Figure 2.6: The alignment of an EST against the genome. In light grey a gene on the DNA is drawn with its distinct elements. In dark grey the alignment of a gene is depicted, due to the quality of the single read EST sequence, the alignment is usually not perfect but will show some mismatches. If correctly aligned intron/exon boundaries can be identified.

prediction, the quality of the prediction depends largely on the quality of the input data. First, EST sequences must always be treated with caution, as these are single read sequences that often contain a relatively high number of sequencing errors. Secondly, a gene can be a member of a gene family, which implies that an EST or cDNA sequence will align to multiple locations in a genome sequence. Identifying the correct origin of the transcript sequence may be problematic, as it can be difficult (or impossible) to distinguish sequence variation among gene family members from sequencing errors. Thirdly, EST data may only be available from a closely related organism. In this case, it is necessary to be able to distinguish evolutionary differences from sequencing errors.

A common approach to cDNA or EST alignment against genomic sequences is to use a similarity search program such as FastA [130] of BLAST [4]. Both, however, do not consider gene models and may return and connect all genomic regions similar to the input EST, even if these are unlikely to be part of a single gene model, for example, if two parts of a hit are separated by a distance too large to be an intron. These problems are partly circumvented by WU-BLAST [204] and MSPCrunch [172], which are better fitted for this task. These tools have the ability (amongst others) to separate alignments based on the distance between two parts of a hit.

The prediction of a gene by transcript alignment to the genome is improved if gene models are used in the alignment algorithm. Such an algorithm can accommodate gaps in the alignment to coincide with introns and keep the predicted gene in frame. Tools that implement this approach include Grail [206], Procrustes [56], GeneWise [18, 19] and Exonerate [169]. Some tools are able to use protein sequences to align to the genome, as for example GenomeScan [208].

Comparative methods

Coding regions are usually under stronger evolutionary constraint and thus better conserved among different species than their adjacent non-coding sequences. This is exploited in gene prediction by demarcating coding from non-coding sequences in multiple alignments of genomic segments on the basis of sequence conservation. This approach belongs to a class of sequence analysis methods called "phylogenetic footprinting" which seek to identify conserved sequence elements in multiple alignments across evolutionarily distant species [97]. Tools that implement these approaches specifically for the purpose of gene finding include, amongst others, ROSETTA [11], SLAM [3] and TWINSCAN [89].

Discussion

The choice for a particular approach or tools for gene prediction on a new genome sequence will depend strongly on what data are available, in addition to the genome sequence itself. *Ab initio* methods are always employed in structural annotation, but the accuracy of their prediction depends on the availability of suitable gene models or on the (re-)trainability of gene predictors and sensors. Alignment-based methods also depend on the availability of expressed sequence data. But even when these are available, such datasets will never be complete because EST data sets usually miss out on lowly expressed and short genes. Comparative genomics methods require the availability of genomic sequences from closely related species. As the body of genome sequence data continue to grow, it is expected that comparative approaches such as phylogenetic footprinting and shadowing [97] will gain importance and power in structural gene prediction. When additional data from any of the above types is unavailable, however, novel methods such as a self- training algorithm as described by Lomsadze *et al.* [104] provide a resort for automated gene prediction.

All methods discussed above are statistical approaches that attempt to return reliable predictions for each structural feature of a gene. There are, however, exceptional gene structures which remain difficult to predict with automated

procedures, such as for example, very short genes, genes with complex structural features, or genes subject to alternative splicing. Alternative splicing is a post-transcriptional regulatory mechanism that gives rise to multiple mRNAs and proteins through the variable processing of pre-mRNA transcripts (for a review see [113]). Alternative splicing has largely remained unaddressed in computational methods of gene prediction, while it is known as an important mechanism by which organisms expand the functional diversity of their transcriptome and proteome.

In general, the different methods described above will complement each other and a powerful approach to gene finding is therefore to combine the results from different predictions. ExonHunter [25] and EUgene [51], two recently developed systems for gene prediction, take integration of different sources of information to an advanced level. In particular Eugene is very attractive because it is an extensible plug-in system in which the signal predictions from a diverse array of tools are combined in a probabilistic gene model. The advantage of such a system is that novel sensors can be easily integrated.

2.2.2 Prediction of RNA-encoding genes

RNA-encoding genes are genes that are transcribed, but, unlike protein-encoding genes, remain untranslated. The transcripts they produce are functional and perform structural, catalytic or regulatory roles, primarily in translation and mostly in conjunction with proteins in ribonucleoprotein (RNP) complexes. In comparison to protein-encoding genes, RNA genes make up only a small fraction of the coding part of a genome. Some classes can nevertheless have a considerable number of members within a genome, such as the 274 tRNA genes in *Saccharomyces cerevisiae* [105]. Recent discoveries have greatly expanded our knowledge of the diversity of RNA genes beyond that of transfer RNA (tRNA) and ribosomal RNA (rRNA) genes [183]. Due to their divergent structural characteristics, prediction methods are mostly dedicated to one specific class of RNA genes. Excellent tools are available for the prediction of tRNA genes [105], while rRNA genes are easily identified by homology searches [159]. Methods for other, more recently discovered classes have become the subject of intensive research. Because RNA gene function

partly depends on the secondary structure of their transcripts, structure prediction is an important aspect of such methods. Tools for RNA secondary structure prediction are RNAfold [164] and mFold [112]. A comprehensive description of RNA gene classes and methods for their prediction is beyond the scope of this chapter. Recent reviews are given in [27, 183] and a comprehensive catalogue of RNA genes is provided by [67]). However, one specific class of RNA genes, the microRNAs, is the subject of study of chapter 5 of this thesis and will therefore be discussed briefly hereafter. MicroRNA genes (miRNAs) are an exciting new class of small RNA-encoding genes that play a regulatory role in gene expression. The first miRNA was discovered in 1993 in the nematode *Caenorhabditis elegans* [96]. A genetically identified locus in its genome appeared to be expressed, producing a small, untranslated transcript with antisense complementarity to the mRNA of an unrelated protein-encoding gene. After this initial discovery, miRNAs received little attention until interest rekindled in 2001 when a number of reports suggested the presence of numerous miRNA genes in the human genome on the basis of direct cloning and sequencing of expressed RNAs. Since then, a large number of scientific papers have been published on the subject (Figure 2.7)

miRNAs encode small RNA molecules, 21 to 25 nucleotides in length, which are digested out of a precursor molecule (pre-miRNA). The precursor molecule usually contains a long palindromic sequence that causes it to fold into a hairpin-like structure [84]. The mature miRNA binds to a complementary sequence in a mRNA transcript and in this way affects its translation. There are two possible mechanisms through which this can happen, repression of translation or transcript degradation [193]. See [38] for a recent review on miRNAs.

The computational prediction and annotation of miRNA genes is in its infancy, primarily because their structural diversity is largely unknown.

2.2.3 Prediction of cis-regulatory elements

In recent years, the functional significance of the non-coding part of a genome has become an important issue in functional genomics studies and is therefore becoming an integral part of structural genome annotation. *Cis*-regulatory

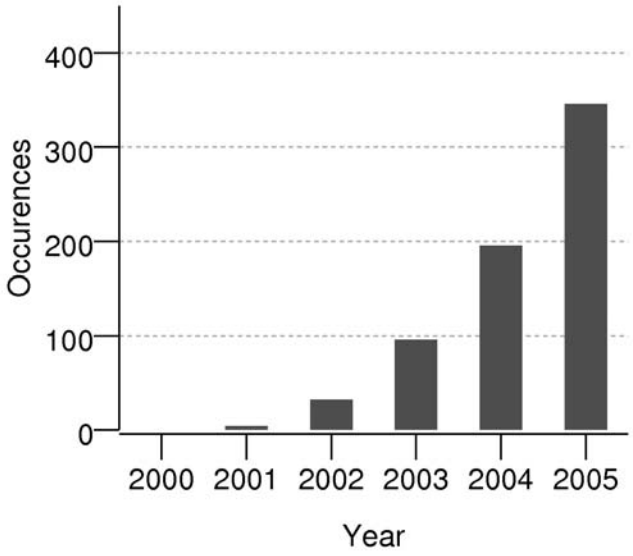

Figure 2.7: The number of records with the term "miRNA" or "microRNA" occurring in Pubmed over the years 2000-2005 illustrate the growing interest in miRNAs over the last years.

elements are DNA sequences that form (part of) the promoter of a gene and are involved in the regulation of expression. An important class of such elements is the transcription factor binding sites (TFBSs), sequence elements that lie in relatively close proximity of the transcription start site and that are the recognition sites for binding by transcription factor proteins. *Cis*-regulatory elements are usually very short, degenerated sequences, that span often less than 10 nucleotides [202]. If a *cis*-regulatory element was previously described, it can be identified in a new genome sequence by searching with a model of those elements (see section 2.3.3). Novel *cis*-regulatory elements, that is, elements that have *not* previously been described, can be discovered by pattern recognition methods (see section 2.2.1). It aims at the identification of overrepresented sequence motifs in sets of promoters that are thought to be regulated by the same transcription factor. Hypothetical co-regulation can be inferred from, for example, expression or orthology data. Many different approaches have been proposed to identify such motifs and to determine what is "over-representation" in terms of statistics and significance, such as by Gibbs sampling [95]. The classical pattern search approach can be enhanced

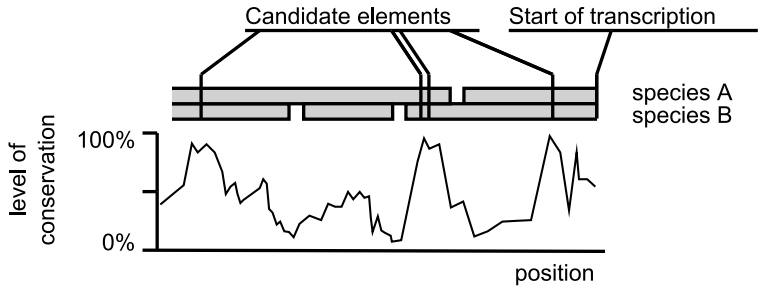

Figure 2.8: The use of phylogenetic footprinting in the discovery of cis-acting regulatory elements. The first step in phylogenetic footprinting is to identify the level of conservation in two related sequences from species A and B, depicted as a graph at the bottom of the image. In the second step, candidate elements are selected from the areas that are highly conserved.

by phylogenetic footprinting [69], in particular phylogenetic shadowing [21], in which sets of promoters from closely related species are compared by multiple alignment. Highly invariant sequences that appear in such alignments are most likely to be evolutionary conserved because of functional constrains and may thus represent regulatory elements (Figure 2.8). However, because of their short length, *cis*- regulatory sequence are likely to occur by chance in any long sequence and their occurrence should therefore be interpreted with caution. Several tools are available for the discovery of novel TFBSs [95, 192], such as MEME [9]. Phylogenetic footprinting is, amongst others, implemented in a tool called Phylofoot [97].

2.2.4 Prediction of other elements

Repetitive elements

A large amount of the non-coding part of a genome is taken up by repetitive elements. They are particularly abundant in the centromeric and telomeric regions of most eukaryotic genomes. The function of those elements is unknown. Repeats come in many varieties and sizes, ranging from short di-, tri- or tetra-nucleotides repeats (micro-satellites) to complex retro-transposons encoding proteins necessary for their own transposition. The occurrence of large numbers of similar repeats in a genome is known to complicate the assembly of such a genome. Two methods are commonly used in structural genome annotation for the identification of repetitive elements. The straightforward

repeat-finding search on the genome works best for finding simple tandem repeats. It is implemented in tools such as tandem repeat finder [14] and vmatch [196]. With a database of known repetitive elements, for example the TIGR plant repeat database [190], alignment-based approaches are used to discover instances of the known elements. This method is more suited for finding complex, interspersed repeated elements such as transposable elements. A widely used tool that incorporates both methods is Repeatmasker [146].

Genetic markers

Approaches to identify a genomic region with a potential gene of interest in the absence of genomic sequences are marker based methods. A genetic marker is a genomic locus that can be used to track a neighboring allele in a population. By assessing many separate markers for inheritance with a phenotypic trait, it is possible to link the trait to a genomic locus. There are many different marker methods available, such as AFLPs [197], SNPs (single nucleotide polymorphisms) and micro-satellites. Today, marker analysis is a valuable tool in the identification of the genomic location of both qualitative and notably also quantitative traits.

Analysis of the frequency by which markers inherit together allows to create a genetic map. Such a map displays the location of each marker along the genome with a position in centimorgan (cM), which is a measure of recombination frequency. A genetic map can be integrated with a genome sequence by *in silico* marker analysis [135]. The combination of a genetic map and a genome sequence allows a researcher to much easier study the genomic loci associated with a trait. When a trait is linked to a marker, a direct sequencing strategy can be applied to identify candidate genes associated with the trait-of-interest. Such a combined map will also give insight into variation of recombination frequencies along a genome and hence the sequence variability.

Sequence variation

Genome sequences that have been determined for an organism are usually derived from a single individual. Such sequences do not reveal the natural variation that exists between individuals or populations. Likewise, there

are over 260.000 plant species known [197] of which only a few have been sequenced, revealing only a small part of the complete scale of sequence variation. The sequence of more species can be invaluable in better understanding the phylogeny and evolution on a molecular scale.

Sequence variation can be assessed by sequencing more genomes of either new species or multiple individuals of a single species. Variation can manifest itself as single nucleotide polymorphisms (SNPs) or inserted, deleted and inverted sequence regions. Complete sequencing of novel genomes gives the best insight into all variation between two or more genomes. This is (still) very expensive, a second best approach is to focus on SNPs. A large-scale project demonstrating this approach, dedicated to the study of variation in genomes, is the human hapmap project [5]. Over five million SNPs (single nucleotide polymorphisms) in 270 individuals have been mapped in the hapmap project and have revealed the occurrence of a varying nucleotide (SNP) in each, on average, 279 nucleotides [5].

2.3 Functional genome annotation

Once the genes and other structural sequence elements in a genome have been identified, the next stage in annotation is to predict the molecular function and biological role of those elements. Similar to structural annotation, the focus in functional annotation is on the function of genes and their products, the proteins. Chapter 5 touches on the functional annotation of a different genomic element, miRNAs.

Evidence for function can be derived from different sources of information and computational inferences on the function of a gene are made by analyzing association within genomic datasets. Using the principle of "guilt by association", evidence from different sources will provide clues as to whether a particular gene is involved in a particular biological process and be assigned a function. A gene can be associated with other genes of known function by its co-expression during specific biochemical, cellular, physiological or developmental processes. Alternatively, a gene can be linked with other genes through a shared genomic neighborhood (especially in prokaryots) or because it shares known regulatory signals. The gene product may be associated with protein

structures of known function, with known protein complexes, or with particular sub-cellular localizations. A gene can be related in terms of its nucleotide sequence to other genes within its own genome or within the genome of other species. In this section we will describe the most important computational methods used in such functional annotation.

The most rigorous approach to functional genome annotation is to determine protein function through experimentation. With the exponential increase of gene sequences available in public databases, however, experimental approaches are lagging far behind the identification of genes with as yet unknown functions. For example, in *Arabidopsis*, which is by far the best studied plant species, well over 40% of its plant specific genes are of unknown function [70]. Computational functional annotation therefore is a challenging task for bioinformatics.

2.3.1 Homology-based prediction of function

Evolutionary associations among genes form the basis of most classical function assignments. The method is commonly referred to as homology-based function prediction. It involves the use of database searches to identify genes that are similar to a query sequence and which have a known, preferably experimentally determined function. Sequences similar to each other are so mostly due to common descent and are inferred to be "homologous". Homology-based function prediction relies on the assumption that homologous proteins in different species are most likely to perform similar functions and that experimentally gained knowledge obtained in one species can be transferred to its homologue in another species.

Although homology-based prediction is very powerful and widely used in functional annotation, it is also fraught with a number of difficulties. Obviously, any new sequence may be short of clear homologues in the sequence databases or it may be homologous to other genes of unknown function. A more fundamental problem is that homologous genes do not necessarily perform the same function. Genes may be related to each other by common descent or by gene duplication events. The terms orthology and paralogy are used to distinguish between these different relationships. Orthologous genes diverged from

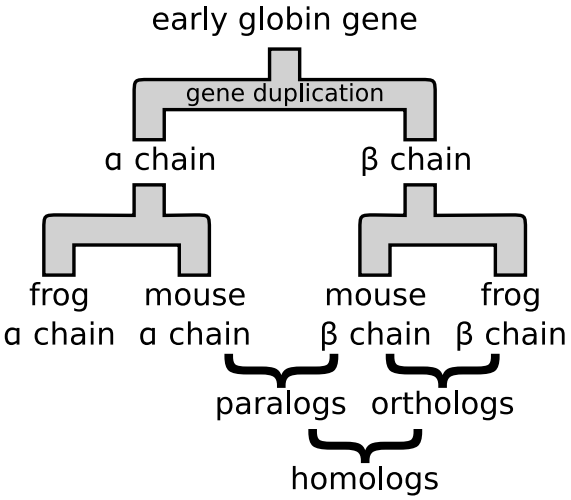

Figure 2.9: A schematic representation of the evolution of the globin gene. The early globin gene at the top underwent a gene duplication at a certain moment that lead to an *alpha* and *beta* chain. Further evolution lead to a copy of these two genes in mouse and human. All copies of both chains are orthologs as they share a common ancestor. Two genes in two different species which that a common ancestor are orthologs, as the globin *beta* chain genes in mouse and human. Two genes with a common ancestor gene in a single species are named paralogs.

each other before speciation and are similar and related to each other due to their descent from the same gene in a shared ancestral species (Figure 2.9). Paralogous genes, on the other hand, are the result of duplication events that occurred after speciation. Both orthologous and paralogous genes are considered homologous. The likelihood, however, that homologous proteins perform similar functions is higher if they are orthologs rather than paralogs of each other [187]. This is reflected in the functional redundancy that arises when copies of a gene are produced in a gene duplication event; one of the two copies may be less (or not) constrained by selection and can diverge by mutation. In the course of evolution this is likely to result in a different function.

Also correct assignment of orthology relationships between genes is difficult [182], particularly in case of genes from large gene families that have disparate rates of molecular evolution among family members. A widely used tool for homology searches in databases is BLAST [4]. BLAST is based on a search and alignment algorithm and is used to compare a nucleotide or protein sequence against a nucleotide or protein database. Similarity scores between the query sequence and database hits reflect their local pairwise alignments.

The method is suitable for initial identification of possible functions for a new sequence, but BLAST results must be interpreted with great caution. Due to the widespread use of BLAST, many of the genes in public databases are annotated using homology-based function prediction. Mis-annotations and other errors are easily propagated. Careful selection of databases and relying only on resources of highest quality and accuracy, such as the manually curated SwissProt database [20] are required to prevent such error propagation as much as possible.

Another drawback of homology-based function prediction, particularly when tools such as BLAST are used, is that the presence of structural domain and motifs that make up a protein are not properly analyzed. BLAST may report most significant database hits for a query sequence that are based solely on the presence of one common, conserved protein domain. Yet, other domains in the sequences may be different and be as indicative of a function. For example, similarity between proteins based on a localization motif gives different information than similarity of a domain containing the catalytic center. Approaches for analyzing protein domains and extracting functional information from such domains are described in the next paragraph.

2.3.2 Protein domains

Proteins are complex three-dimensional structures built up from sequence elements that fold into distinct sub-structures, called domains. Domains, in turn, can be composed of one or more motifs. Motifs are smaller substructures that generally have highly specific molecular functions, subordinate to the more general function of the protein in which they occur. For example, many enzymes contain an "ATPase binding domain", that is responsible for the hydrolysis of ATP. The energy that is released in ATP hydrolysis can be used in another domain of the protein to catalyze specific chemical reactions. An important factor in protein evolution is domain shuffling, a recombination process that involves the exchange of functional modules between genes [32]. As a consequence of domain shuffling, many protein motifs and domains appear not to be confined to one gene but to occur in different combinations among gene families and thus to have led to the evolution of chimeric proteins. It is obvious that such proteins further complicate homology assignment between

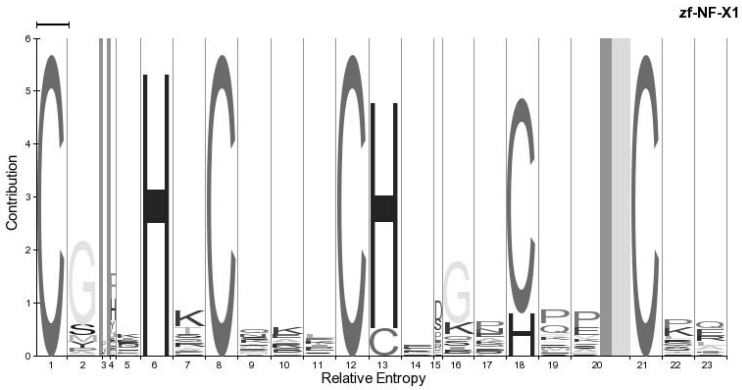

Figure 2.10: A zinc finger protein motif derived from the PFAM database. The motif is modelled as an HMM and represented in the form of a DNA logo [165]. The size of a letter represents the likelihood of the corresponding amino acid occurring at that position.

proteins discussed in the previous section. A more sophisticated approach in assessing the function of a predicted protein is therefore to analyze its structural composition in terms of the presence of protein motifs and domains with known functions. Protein motifs and domains can be discovered and extracted as profiles from multiple alignments of proteins with known similar functions using HMMs (explained in section 2.2.1). The compilation of such profiles has been a major activity of bioinformatics in the past years and has led to the development of a number of secondary protein sequence databases, such as PFAM, PRINTS or SMART [141]. An example of an HMM profile as available from the PFAM database is that of the well known "zinc-finger domain", the profile of which is depicted in Figure 2.10. Protein localization signals can be predicted by for example the SignalP [12] and TargetP [42] web-servers.

New protein sequences are analyzed with respect to their functional domain composition by searching various domain databases. A very useful application for this analysis is InterProScan [141], a tool designed to take full advantage of InterPro, the resource of protein families, domains and functional sites that integrates ten of the major protein signature databases [7, 141]. Protein profile searches and the analysis of domain composition can yield very specific information on protein function. It is therefore becoming the most important and powerful approach in genome-scale functional annotation.

2.3.3 Cis-regulatory elements

An entirely different type of evidence for gene function is derived from data on the regulation of gene expression. One of the best studied regulatory mechanisms is that involving transcription factors; proteins that co-operate with the RNA polymerase in multi-protein complexes at the start of transcription. An important class of *cis*-regulatory elements are the transcription factor binding sites (TFBSs; see section 2.2.3), short, conserved sequence motifs in close proximity of the transcription start site that act as recognition sites for transcription factors. Identification of such elements can give insight in the function of a protein. For example, the presence of the TFBS "evening element" (AAAATATCT) [117] in an *Arabidopsis* promoter region indicates regulation of the gene by the circadian clock. A useful resource on known TFBSs is the Transfac database [191, 201]. Transfac stores recognition sites as weight matrix models which can be used to search the promoter region of a new sequence. The major pitfall in attempts to use TFBSs in gene function prediction is the extremely short size of these elements, sometimes only a few nucleotides long, as a result of which they frequently arise purely by chance. If a binding site is biologically relevant, however, it is likely to be better conserved during evolution than its surrounding sequence. This type of evidence is exploited in the phylogenetic footprinting approaches outlined in section 2.2.3.

2.3.4 Gene Ontology

The previous sections have described data sources and methods that help to infer the function of a gene. For a small set of genes, the results of all these analyses are interpretable for human beings, but as the number of genes grows, interpretation becomes more and more difficult. Each gene can have many different pieces of evidence associated with it. Some of the evidence might (or is likely to) be erroneous. Apart from errors in the data, a researcher will have to search through data and interpret the results. Such a laborious exercise in annotation can be improved with the help of a standardized description of data. A framework for the description of annotation data is provided by the Gene Ontology (GO). The GO paper [8] describes an ontology as follows:

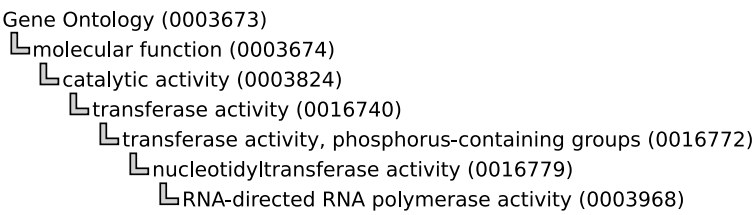

Figure 2.11: A part of the Gene Ontology ontology tree representing the molecular function branch leading to an RNA-directed RNA polymerase.

> "*An ontology comprises a set of well-defined terms with well-defined relationships. The structure itself reflects the current representation of biological knowledge as well as serving as a guide for organizing new data.*"

The gene ontology is used to map biological knowledge to genes. The GO is subdivided into three classifications, *i.e.* (1) biological process, (2) molecular function, and, (3) cellular component. The ontology is structured like a tree with increasingly specific terms occurring towards the smaller branches. A gene obtains GO-term assignments on the basis of underlying biological evidence as outlined in the previous sections. If a gene is active in multiple biological processes, it can be assigned multiple terms. As an example, the molecular function annotation of an RNA-dependent RNA polymerase in GO is given in Figure 2.11.

In addition to the functional annotation, the gene ontology also uses a code to indicate the type of evidence with which GO terms have been assigned. If the assignment is based on experimental evidence a gene will be marked with the code "IDA", meaning "inferred from direct assay". If it concerns homology-based assignment the code "IEA" is used, which stands for "inferred from electronic annotation". This provides the researcher using genome annotation data a rough indication of the reliability of the assignment.

2.3.5 Expression and protein interaction data

Transcriptomics and proteomics generate a wealth of genome-wide expression data. It is possible to mine these data for possible clues to gene function. Straightforward evidence can be derived from the observation in what tissues

and under what circumstances a gene is expressed. Such evidence, however, tends to be global and does not provide detailed and accurate information on gene function and biological processes. More indirect evidence can be inferred from co-expression analysis. For example, if a gene of interest is always expressed together with a set of genes of known function, it is possibly active in the same process. This type of evidence can provide very detailed information on gene function, but is dependent on the availability of reliable annotation of other (co-expressed) genes A prime example of a project in which large scale -omics data have been used for the prediction of gene function is the construction of a genome-wide protein complex map in yeast, identifying all proteins forming a complex [55]. The authors followed the "guilt by association" principle in function assignment, which suggests that if two genes physically form part of the same protein complex they are most likely active in the same cellular process.

2.3.6 Discussion

Various methods are available for functional annotation. Again, none of the methods described stands out or stands alone to render another method superfluous. Each of the methods described adds different information to the annotation that will allow an expert to make a better assessment of the function. Caution should be taken in the interpretation of a functional assignment. Transfer of function based on sequence homology is error prone, even when the annotation of the known homologous sequence is correct. A very high level of similarity would seem to be necessary to reliably transfer function [151].

2.4 Computational genome annotation

Complete and in-depth annotation of a genome requires the application of many different software tools. The number of separate computational executions that needs to be performed can be extremely high, generating complex data flows and requiring large amounts of CPU-time as well as data storage capacity. Data must always be in a form that can intelligibly be presented to

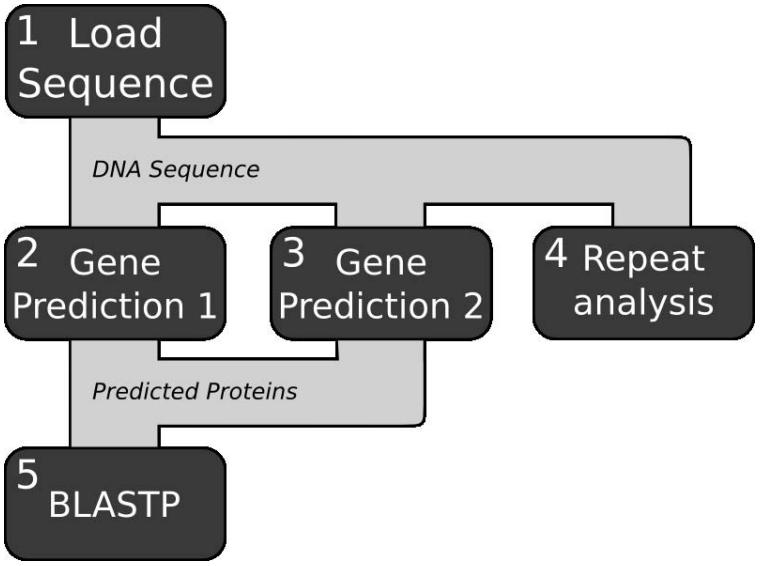

Figure 2.12: Schematic representation of a small annotation project in the form of a pipeline. The input sequences (1) are analyzed by two gene prediction tools (2&3) and an analysis of the repetitive regions (4). The predicted proteins are subsequently compared against a protein database by BLASTP (5).

and used by researchers. In this section, the various computational aspects of high- throughput genome annotation are discussed.

2.4.1 Workflow management

A simple genome annotation project will consist of several gene predictors, a tool to consolidate gene predictions, a repeat-finding tool, and a BLAST analysis on predicted genes (Figure 2.12). This set of tools and the order in which they should be executed is called a "pipeline". Execution of the pipeline will involve a large number of separate jobs in which the output of one job serves as input for a subsequent job. The results of all analyses need to be stored in a database and made available to an end user.

On the scale of a complete genome analysis, it will quickly become impossible to perform all analyses manually. Computational pipelines require the use of specialized software that schedules and keeps track of jobs as well as of the creation and storage of data and results. Several systems are available for this kind of pipeline or workflow management in a genome annotation environment

[124, 138, 167]. Chapter three of this thesis describes the development of a novel, generic workflow management system.

Any pipeline system that is able to handle complex, elaborate and configurable pipelines requires extensive computing. There are several possibilities for the scaling of computing capacity. One is to use a single multiprocessor system but these tend to be expensive and difficult to scale. A second, more scalable, solution is to use a cluster of distributed small to midrange systems. In 1993, the Beowulf project [16] was the first to implement such a cluster using commodity hardware and brought such systems into the reach of many. Many different systems have since been developed. These can be divided roughly into two types; clusters that operate as a single multiprocessor computer and clusters in which separate computers (nodes) are directed by a central control unit.

Clusters of the first type simulate a multiprocessor computer by using a middle layer that handles communication between the nodes. This type of cluster is particularly designed to handle large, computationally demanding jobs but requires specially adapted software. Two well known examples are PVM [127] and MPI [114]. HMMer [76] and BLAST [120] are examples of bioinformatics applications that are able to work on PVM or MPI clusters, respectively. In the second type of clusters, parallel computing is achieved through distribution of separate jobs by the central control unit (or "master") over independent nodes (or "slaves") that execute the job and return the results back to the master. This type of clusters does not require specialy adapted software and are well-suited for the execution of large numbers of jobs that require relatively little computing power. Common job management software for cluster computing includes Sun Grid Engine (SGE) [184], openPBS [125] and Condor [33]. The latter is aimed at heterologous, non-dedicated, hardware and is able to run on, for example, idle office desktops. A complete Linux distribution that includes different job management is Rocks [150]. If computational facilities are distributed over different physical locations, it is commonly named a "grid". Implementation of a grid can provide a level of throughput which is not achievable for a single cluster of super-computer. A common used toolkit for developing grids is Globus [61] but Condor and SGE also contain grid-like features.

The choice for a system depends strongly on the requirements of an application. As most genome annotation pipelines need to execute large numbers of separate applications the second type is in most cases better suited. Especially as several of the mentioned solutions are able to run a "sub-cluster" of the first type. For example, SGE can reserve a group of nodes to execute a PVM job.

2.4.2 Data management and data exchange

A second aspect in automated execution of genome annotation pipelines is concerned with data management: exchange and storage. The various bioinformatics tools available use a wide variety of data input and output formats. This hampers communication between separate analyses. One tool may deliver an output that can not be used as input for the next tool without a translation. A notorious example is the output of a BLAST analysis [4] which is a human readable text document that has undergone many changes. Small changes which are easy to understand for a human usually breaks software attempting to automate a task. It is extremely difficult to design a single, generic data format, due to the inherent diversity of biomolecular data types (reviewed in Stein [179]). However, a number of solutions have been proposed for the standardization of genome annotation data, including the General Feature Format (GFF) [57] and XML based BioMOBY [199, 200]. GFF is widely used in the bioinformatics community, although it is limited in scope and difficult to extend. BioMOBY provides a more flexible solution for the exchange of biomolecular data. BioMOBY distinguishes itself by not attempting to describe data, but describes *how* to describe data, BioMOBY is a meta-format. This makes the exchange of data *formats* much easier and as a direct result, the exchange of data itself. Before data can be exchanged however, a decision on the BioMOBY must still be made.

Similar problems apply to data storage. Raw output from can be stored as such but this renders the data inaccessible for subsequent inspection and integration. The database underlying the Generic Genome Browser [180] stores GFF data and is thus limited to what GFF can describe. Again, the use of a meta-format such as BioMOBY can help in storing data and making it more accessible.

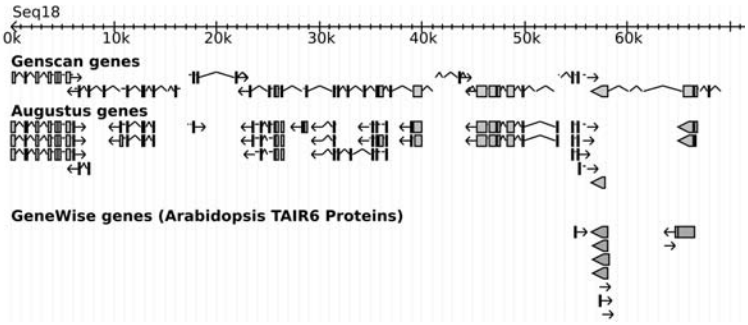

Figure 2.13: A screenshot of a part of the popular generic genome browser. The image represents a DNA sequence of little more than 90 kb and shows Genscan, GlimmerHMM and BLASTX annotations.

2.4.3 Visualization

The final step of an annotation program is to present the results of all analyses and biological interpretations in an intelligible and easily accessible form to the biologist. This is a not a trivial task as the amount of data generated is often enormous. Several strategies can be taken to help explore and understand genome annotation data, the best among which is through visualization ("a picture is worth a thousand words"). Among the widely used tools available for the visualization and exploration of an annotated genome are the Generic Genome Browser (Gbrowse) [180] and Ensembl [78]. Figure 2.13 gives an example of a Gbrowse screenshot. Most visualization tools have been developed as web interfaces to underlying annotation databases. Therefore, they generally perform poorly with respect to interactivity. For example, step-less zooming and panning, two powerful visualization techniques for user interaction, are difficult to implement in web interfaces. In Chapter four we describe the development of a novel tool (DNAvis) for the visualization of genome annotation data, which implements several modern visualization concepts and technologies.

2.5 Future of genome annotation

Annotation of genome sequences is a complex process for which there is no true end point. Biological data volumes will continue to rise and without au-

tomatic annotation, it will be impossible to translate these data in biological knowledge. On one hand, annotation depends on the interpretation and presumed understanding of biological systems and processes. On the other hand, it aims to contribute to that understanding. The biological interpretation of genomes will therefore be subject to continuous revision as new discoveries are made and new interpretations are necessary.

Future annotation should also be prepared to deal with an increasing dimensionality of data and acknowledge and use the natural variation within an organism, between organisms of the same species and between organisms of different species. Variation between genomes is present in many forms, including single-nucleotide polymorphisms, insertion-deletion polymorphisms, variable numbers of repeats as well as various structural variations. Genome structures may be much more dynamic than currently shown in genome data, which is usually a snapshot from a single individual. Heterogeneity within a single organism, for example between tissues or different environments is largely due to epigenetic variation. The presence of such a code "on top of" the DNA code (the epi-code) [65] is an issue that future genome annotation should incorporate. To deal with variation and dynamics in genomes and genome annotation is a future challenge for both biological and computer science and scientists.

We are likely to witness a dramatic growth of the quantity, quality and availability of genome data and models related to any biological phenomenon over the coming years. The value of these data and models depends largely on the quality of the annotation provided. This applies to experimental data generated in the laboratory, to data analysis and processing, as well as to manual or automatic annotations. The more annotation is going to be automated, the more vulnerable it will become to error generation and error propagation. Quality will therefore be a major issue for future annotation, requiring standards for the indication of quality and reliability, and easy exchange of data, tools and annotations between platforms, humans and machines. Annotation pipelines (Chapter 3) will continue to be developed and will evolve, integrating data sources, data types and annotation types. More attention should be given to increase the utility of annotation data for the non-specialists. When all this is organized and funded properly, genome annotation will allow to create a

unique knowledge base for future biological research, the whole of which is likely to be far greater than the sum of its parts.

Chapter 3

High throughput analysis in bioinformatics with the Cyrille2 pipeline software

Mark Fiers,Ate van der Burgt,Erwin Datema,Joost de Groot and Roeland van Ham

Abstract

High throughput sequencing must be matched by high throughput annotation. Given the large number of annotation tools available, a multitude of interdependent analyses are required for an in-depth annotation of even a single BAC sequence. Special annotation pipeline software is required to make such annotation processes feasible in an automated fashion. In terms of functionality, such software should meet the key requirements of enabling high throughput data analysis while providing an easy-to-use, configurable and extensible workflow management system. In the public domain, there is currently no tool available that truly meets all these requirements.

Therefore we have developed a generic pipeline system called Cyrille2. The software consists of three, functionally distinct parts that can be executed independently so as to ensure modularity and extensibility. These parts include: 1) the Graphical User Interface (*GUI*), a web-based interface to the pipeline management system for the pipeline operator and administrator, 2) the *Scheduler*, the functional core of the pipeline management system that schedules jobs for execution and 3) the *Executor*

that searches for jobs prepared by the scheduler and executes those on a dedicated computational cluster. Cyrille2 enables easy employment of high throughput, extensible pipelines.

3.1 Introduction

Large-scale computational analysis of biomolecular data often requires many different, sequential operations on the data and integration of the results of a diverse array of tools. Such a chain of computational operations is often called a "pipeline". A pipeline can be viewed as a program that describes the exact order in which analyses are to be performed on an input dataset and what the relationships between in- and output datasets are. In a formal representation of a pipeline, an operation performed by a computational tool on input data is represented by a "node". The connection between nodes are represented by streams, defining the data-flow in-between operations. An example of a simple bioinformatics pipeline that could be part of a basic genome annotation is depicted in Figure 3.1.

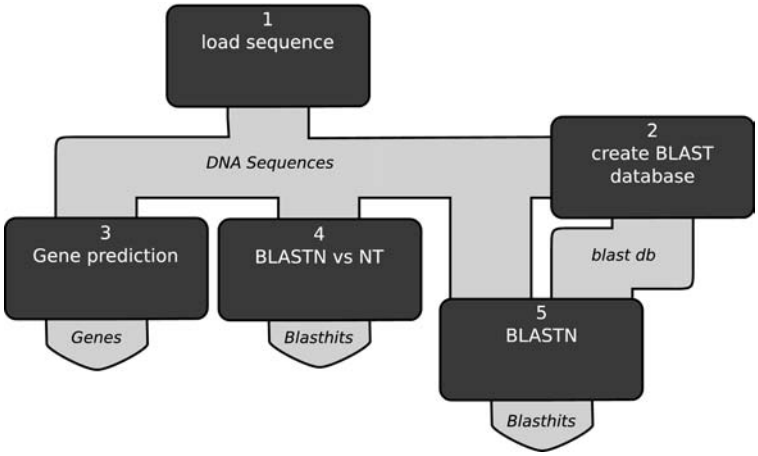

Figure 3.1: An example of a simple annotation pipeline. The pipeline describes execution of a gene predictor (3) and two BLAST analyses (4&5) [4] on an a set of input DNA sequences (1). The BLAST analysis of node 4 compares the incoming sequences against the NCBI NT database. Node 5 uses a BLAST database created by node 2 from the same set of sequences.

Even for a small bioinformatics project, for example, the phylogenetic analysis of a small number of related genes, it is cumbersome to run all the necessary

analyses manually. For larger projects, this quickly becomes impossible. For example, annotation of a single DNA sequence such as a Bacterial Artificial Chromosome (BAC), which is often used in sequencing projects and about 100 kilo-bases long, may require thousands of interdependent analyses including gene prediction, homology searches, protein domain analyses and repeat discovery. This problem can be resolved by the use of specific "pipeline software" that runs all required analyses and manages all data.

Adequate pipeline software must be able to integrate a wide variety of bioinformatics tools and be able to transport, transform and store the data in between the nodes. The practice of modern day genomic research and data production also requires it to be able to handle very large amounts of data, to execute pipeline analyses over extended periods of time, and to recover data from computer calamities, specifically in a high-throughput production environment.

With having primarily the development of a computational environment for large-scale genome annotation in mind, we have defined the following criteria to be important in the design of pipeline software:

Ease of use In a production environment, it is important to have a pipeline system in place that is easy to use by non-expert end-users. This can be achieved through a well-designed graphical user interface (*GUI*) that allows easy and intuitive programming, adaptation, monitoring and administration of a pipeline.

High throughput For annotation of a complete eukaryotic genome sequence, a pipeline system should be capable of handling large datasets and complex pipeline structures. The system must be able to schedule and execute very large numbers of analyses and be able to distribute jobs over (multiple) computational clusters, possibly taking days of processing time on a Linux cluster. An important requirement for high-throughput analysis is data persistence and data tracking. This allows easy resumption after a system failure or recalculation of parts of the pipeline.

Flexible New or upgraded bioinformatics tools appear frequently. For pipeline execution to remain up-to-date, it is essential that implementation of

new tools or upgrading existing tools in the pipeline software should be straightforward. The system should be flexible and modular to allow complex rearrangements of data required by some tools. Use of an open communication standard is important to allow the system to communicate with third party annotation service providers.

Updating In ongoing projects it is often undesirable to postpone annotation until all sequences are finished. Initial annotations are therefore repeated on a regular basis, for example, when genome assemblies are updated or new reference data (i.e. BLAST databases) become available. Therefore, the pipeline software should be able to reschedule and execute the affected parts of an annotation with a minimum of redundant effort.

3.1.1 Cyrille2

This chapter describes the development of a new pipeline software system, Cyrille2, which fully complies with the aforementioned criteria. Cyrille2 is the successor of Cyrille1, which is an in-house set of static scripts used for genome annotation. There are a number of pipeline software systems publicly available, including Ensembl [138], Pegasys [167], GPIPE [54], Taverna [124], Wildfire [186] and MOWserv [123]. Systems which are not publicly available (as the NCBI pipeline) will not be discussed in this chapter. An obvious question is why we would want to develop yet another system? The answer is that none of the available systems sufficiently complies with requirements outlined above and at the same time can be distributed and implemented easily and robustly at new sites. We implemented Cyrille2 to provide this distinct set of features, which no other tool, to our opinion, combines. The most important one is that it is a system that is truly tailored for high-throughput and large-scale bioinformatics analysis, a feature that no other publicly available system really implements, with the exception of the Ensembl pipeline. The Ensembl system, however, appears not to be developed to be deployed at other sites and appears much less flexible and harder to extend than the Cyrille2 system.

Table 3.1: Summary of the most important pipeline terminology

Pipeline	A pipeline is the definition of a series of computational analyses that are to be performed on a set of data. A pipeline can be described as a set of nodes representing the individual analyses.
Node	A node defines a single analysis in the context of a pipeline. A node is associated with a tool and is responsible for the execution of one to many jobs. A node also specifies how the data from a preceding node is shuttled into the current node.
Tool	A single, pipeline embedded application, for example BLAST [4].
Tool wrapper	A tool wrapper is a script that frames and embeds a tool within the pipeline. It enables execution of the tool through communication with the pipeline software, from which it receives the tools' parameter settings (for example, which BLAST database to use). It translates in- and outgoing data from the pipeline in the format required by the tool.
Job	A job is a single execution of an analysis. For example, a single gene prediction performed on a DNA sequence loaded into the pipeline.
Object	Objects are created by the analysis tools and represent the (smallest) units of analysis traversing a pipeline. Examples of objects are a DNA sequence, a predicted exon or a protein motif.
Stream	A stream connects two nodes and describes the data-flow between those nodes. To allow complex pipelines any node can define multiple in- and output streams.

3.2 Implementation

For a detailed description of the structural design and operation of the Cyrille2 system several key terms must be defined. Table 3.1 provides the definitions of the most important terms used. Hereafter, we will first give a general overview of the Cyrille2 design followed by an explanation of how the data-flow through the system is standardized. The next section will explain how the system standardizes communication with external tools. Then the operation of a node will be explained. A node is the functional core of the system and embodies most of the application logic conceived. The last section combines all previous information and describes how a complete pipeline manages execution.

3.2.1 System overview

The Cyrille2 software is written in Python [140] and uses amongst others: an Apache2 web server [6], a MySQL database [122] and mod_python [119]. Software and database management run on a Linux server and make use of a Rocks Linux cluster [150] to distribute computational analyses. The system architecture is composed of multiple layers (Figure 3.2). The functional

components (Layer 1) consist of the graphical user interface (*GUI*), used by the pipeline operator and administrator, the *Scheduler*, and the *Executor*. These three core components make extensive use of the well-documented, modular, application programming interface (API) (Layer 2). The API allows unified access of the system databases (Layer 3). The *biological database* and end user interface are third party systems and adopted by the Cyrille2 system (Layer 4).

To allow easy tracking and debugging of a pipeline run, an advanced status and logging system is implemented. This provides a pipeline operator access to detailed information on the status of a pipeline run and errors that might have occurred.

A pipeline system needs to manage and store large amounts of diverse information. To keep different types of data separated, the system employs four databases (Figure 3.2, Layer 3 & 4): (1) a *pipeline database* stores data on defined pipeline structures, node settings and parameters associated with wrapped tools; (2) a *status database* stores the state of execution of a pipeline at any given time, keeping track of jobs waiting to be executed or that have run successfully, and logging messages; (3) a *biological database* that stores and provides access to the results of all analyses that the end-users are interested in, and; (4) a *failover database* that, automatically and in a generic fashion, parses and stores temporary objects that do not need to be stored in the *biological database*. This database is not strictly required by the system but it relieves developers from writing specific code for intermediate object types and so ensures data persistence.

Consider, as an illustration, the "gene prediction" node (no. 3) from the example pipeline in Figure 3.1. The *pipeline database* stores the information that defines node 3 as a gene prediction node and instructs the gene prediction tool to run an analysis on each of the input DNA sequences from the preceding "Load sequence" node (node 1). The *pipeline database* stores parameters specific for the gene prediction tool, for example, which gene model to use. The *status database* stores information on which of the sequences the gene prediction has been completed, what genes have been predicted and keeps track of all objects held in the *biological database* by storing a unique identifier for each object.

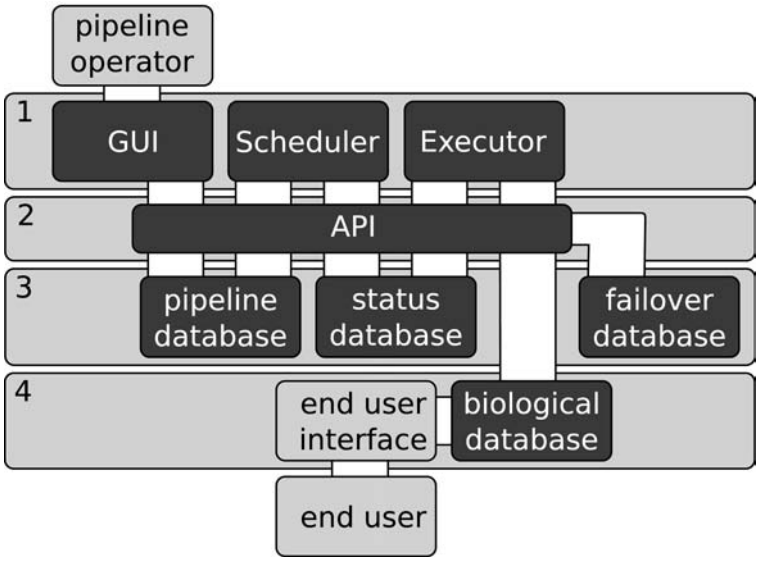

Figure 3.2: A schematic representation of the Cyrille2 system. See the text for an in depth description

Equivalent to the functional division of the databases, the software can be divided into three distinct functional parts: the Graphical User Interface (*GUI*), the *Scheduler* and the *Executor* (Figure 3.2). The *GUI* allows pipeline operators and administrators to create, adapt, start and stop pipeline runs and fine-tune pipeline and tool settings (Figure 3.3). The *Scheduler* represents the core of the Cyrille2 system. Using a pipeline from the *pipeline database*, it schedules all jobs to be executed in the correct order, accounting for dependencies among analysis. *Scheduler* operation is described in more detail in a later section. The *Executor* loops through the *status database* and passes scheduled jobs and associated data on to compute nodes for execution. The final results of a job execution are stored in the *biological database* and tracked in the *status database*. As large amounts of analyses need to be distributed over one or more computing clusters to keep total computing time in bounds, the *Executor* runs on a computer cluster and acts as a broker between the Cyrille2 system and standard cluster scheduling software such as for example Sun Grid Engine [184]. It is possible to employ multiple clusters, of different type, by running multiple instances of the *Executor*.

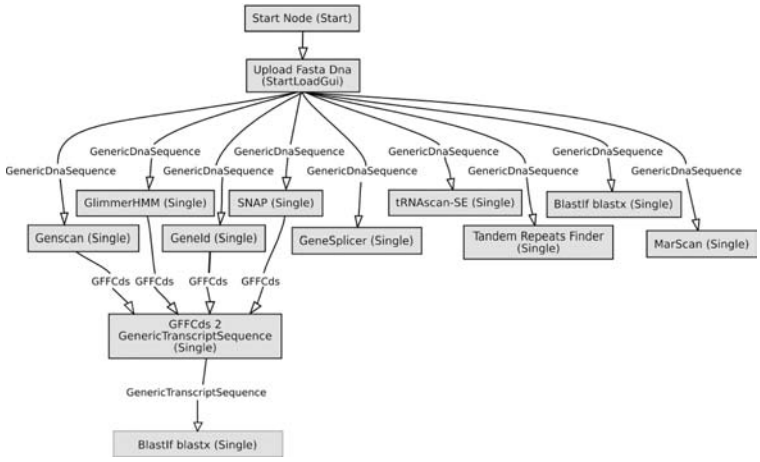

Figure 3.3: An image of a pipeline as generated by Cyrille2 *Graphical User Interface* (adapted to optimize space usage) showing a simple genome annotation pipeline consisting of four gene prediction analyses (GlimmerHMM, Genscan, GeneID and SNAP), a intron-exon splice-site prediction (GeneSplicer), a tRNA analysis (tRNAscan-SE), a MAR element scan (Marscan), a repeat analysis (Tandem Repeat Finder) and a BLASTX analysis. The predicted genes (stored as coordinates on the original sequence; GFFCds) are converted into a sequence object (GenericTranscriptSequence) and subsequently subjected to a BLASTX analysis against the NCBI NT database.

3.2.2 Nodes

From a functional point of view, the Cyrille2 pipeline software is comprised of three separate parts, the *GUI*, *Scheduler* and *Executor*. From the point of view of software implementation, however, most of the fundamental application logic of Cyrille2 is implemented in so called "node classes". A "node class" is a class as defined in object oriented programming, which means that all functionality that is semantically attributed to a node is implemented in the node class. For example, the node class has a function called "schedule" and "execute". Both the *Scheduler* and *Executor* employ node classes for their functionality.

The system distinguishes between several different node types (Figure 3.4), each of which is implemented as a separate class. The creation of a new node type is possible by inheriting from an existing node class. By choosing an existing node class that resembles the required functionality, the new class needs only to implement those features that differ from the parent class.

3.2.3 Job scheduling

The various tools used in an analysis pipeline require different arrangements of incoming data. For example, in node 2 in Figure 3.1, in which a blast database is to be created, all sequences retrieved in node 1 need to be combined in a single input file whereas in node 4 the sequences from the same set are processed one by one. This is further illustrated in Figure 3.4, in which for the same pipeline shown in Figure 3.1 detailed information is provided on what objects are created and what jobs are scheduled. A node of the type "single" (Figure 3.4, node 3) schedules a computational job for each of the incoming objects separately, whereas a node of type "all" (Figure 3.4, node 2) takes all incoming objects together and schedules these for a single computational job. The modular implementation of a node allows many more, arbitrarily complex, scheduling strategies.

3.2.4 Data flow

A major challenge for any pipeline system is to devise a robust and fast way to conduct data through a pipeline. This is not trivial, given that even a pipeline for a relatively simple analysis of a single DNA sequence (Figure 3.3) may actually encompass many thousands of separate jobs that need to be scheduled and executed correctly, resulting in possibly millions of predicted objects.

Without uniform data management, implementation of this pipeline would be very difficult. Each different node would have to analyze the incoming data stream separately and in a different manner for each data type traversing the pipeline. Standardized identification of an object, on the other hand, allows generic implementation of different node types, description of interfaces and facilitates uniform tracking of data between nodes.

An appropriate data exchange format identifies and communicates objects in a uniform and unambiguous manner. Such a format must have a unique data type identifier and classification (for example, an object is of type "cDNA sequence"). The format must also be extensible to accommodate future incorporation of novel data types. An important additional feature of a communication standard is interoperability with third party servers offering specialized

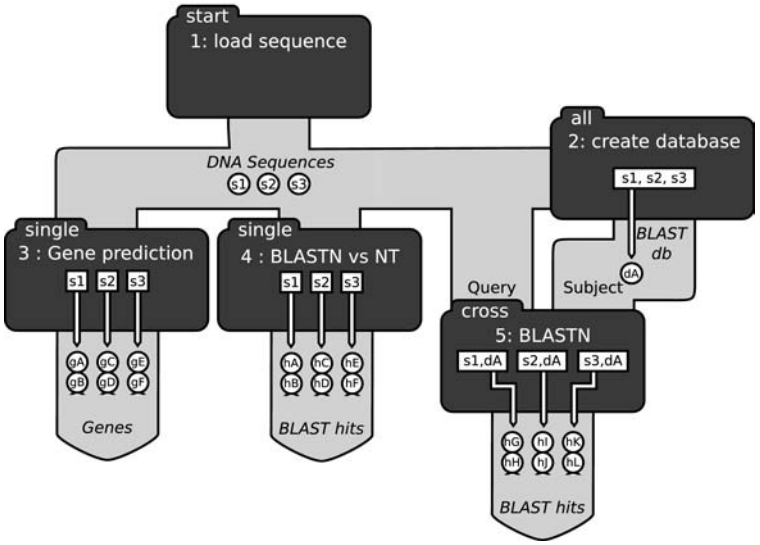

Figure 3.4: Detailed illustration of the data flow and scheduling strategies of the node types from the pipeline shown in Figure 3.1. Node types are indicated in the top left of each node. In this example, three DNA sequences are uploaded into the system (s1, s2 and s3) which are subsequently processed by the different nodes. The white circles in-between the nodes indicate objects traversing the pipeline. The white boxes inside the nodes represent the jobs that are scheduled. For example; sequences uploaded (s1, s2 and s3) are, amongst others, scheduled by node 4 for a BLASTN analysis against the NT database. This BLAST analysis results in the BLAST-hits indicated by objects hA-hF. See text for more details.

analysis tools and computational facilities . In the current era distributed computing, the ability to communicate with systems worldwide is becoming ever more important (see also chapter 2.4).

Several data exchange formats, with varying scope, have been devised and proposed for the handling and communication of biomolecular data, including XML-based formats such as GAME (used by Apollo [98]) and BioMOBY [199, 200, chapter 2.4.2] and flat-file formats such as GFF [57]. We have chosen to implement BioMOBY [199, 200], which is emerging as the most promising and widely used standard in bioinformatics and has already been implemented in systems like Taverna [124] and MOWserv [123]. Another motivation for opting for BioMOBY in Cyrille2 is that it was chosen as the data interchange format for the worldwide Solanaceae sequencing project [121] in which our group participates and for which we will employ the Cyrille2 system for genome annotation.

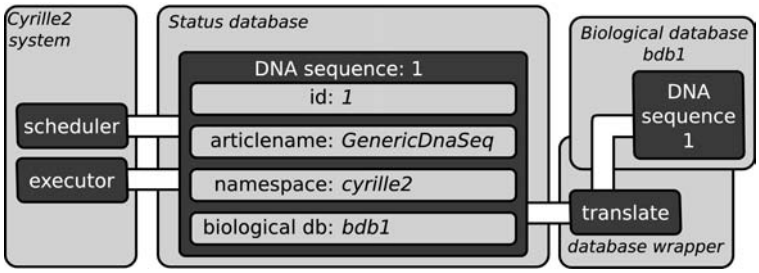

Figure 3.5: Relationship between the *status* and the *biological database*. The *status database* employs BioMOBY identification ("id", "articlename" and "namespace") and holds information on the *biological database* in which the object is stored. The *biological database* is accessed through a database specific wrapper that provides a generic interface to retrieve an object based on the information in the schedule database. The database wrapper is accessed through a wrapper script (number 1 & 3 in Figure 3.6).

The XML-based BioMOBY standard contains a specification how to describe data and web services. BioMOBY is a meta-data format, which does not attempt to describe data but defines how to describe data. It employs a method of object identification and classification, in which controlled vocabularies are used to define each BioMOBY object with an (1) identification string ("id"), (2) an object type (called "articlename") and (3) a "namespace". The Cyrille2 system adheres to both the specifications of data format and web services. To reduce computational overhead, the service provider specification is implemented, but not used for local execution of analysis tools

3.2.5 Data storage

Standardized object identification can also be used in standardized data storage. If a BioMOBY object is properly stored with a unique "id", "articlename" and "namespace", this information is sufficient to uniquely retrieve the object from a database. Because the Cyrille2 system is designed to allow for the use of different databases to ensure flexibility, a Cyrille2 database wrapper functions as an intermediate between uniform, BioMOBY, object identification and database specific storage of these objects (Figure 3.5). The database wrapper contains specific instructions to store and retrieve each different object type in the *biological database*. This solution combines unique identification of any object with the freedom to use any database required.

Another important feature of the Cyrille2 system is data persistence. All intermediate data is stored in a database. This procedure is particularly important for a high-throughput system in which a pipeline may be running for extended periods of time and therefore requires a mechanism to restore execution after a system failure. With no intermediate data stored, the only solution would be to rerun the complete pipeline. A further advantage of intermediate data storage is that expansion of a pipeline with additional nodes or partial recalculations can easily be performed, for example, re-execution of a BLAST analysis with an updated database. Data persistence is guaranteed by the "failover" database. If an object cannot be stored in the *biological database*, it is automatically stored in the *failover database*. An object may not be stored in the *biological database* for two reasons: firstly, when it concerns an intermediate object of no importance to the end-user or, secondly, when the database wrapper to the *biological database* fails to store the object for an unexpected reason. In either case no data is lost and stored as raw BioMOBY XML.

3.2.6 Tool wrappers

A tool wrapper is responsible for execution of external applications and generic interaction with the Cyrille2 system. Tool wrappers are implemented in such a way that they can easily run standalone, be deployed as a BioMOBY web service or function as components of the Cyrille2 system. A tool wrapper is equivalent to a "Runnable" in the Ensembl system [138]. Upon execution, the tool wrapper receives input data and parameter settings of the application. The tool wrapper starts with conversion of the incoming BioMOBY data into a format required by the tool. It then executes the application. Operation of the tool wrapper is finalized by reading of the output of the application, converting this output back to BioMOBY and subsequently returning it to the system. For example, upon execution, a BLASTP tool wrapper will receive the sequence to be analyzed and the parameters applying to the BLASTP run, including which BLAST protein database and expectation value to use. The sequence is extracted from the BioMOBY XML and temporarily saved as a FASTA file. Subsequently, BLASTP is executed and upon completion the output is read by the wrapper and converted into a BioMOBY XML representation.

A further task of the tool wrapper is to register itself in the Cyrille2 system. Registration implies that the tool becomes available through the *GUI*, allowing a pipeline operator to integrate it in a pipeline and the *Scheduler* to correctly schedule jobs for that tool. The process communicates what objects are required as input (e.g. protein sequences for BLASTP), what parameters are accepted (e.g. specification of a BLASTP database) and with what node type it must be associated. This is implemented in a generic registration method where the wrapper registers all required information in the *pipeline database*.

In a rapidly evolving field like bioinformatics, it is of great importance that new tools can be implemented quickly. In the Cyrille2 system this requirement is met by a generic design method in which programming is facilitated through a base tool wrapper class from which novel tool wrappers inherit their code (as in object oriented programming) and most of the required functionality. In brief, implementation of a novel tool in the Cyrille2 system involves the following steps: (1) installation and configuration of the new tool on the execution cluster; (2) writing of the BioMOBY-compatible tool wrapper; (3) definition of new BioMOBY objects if required by the tool; (4) confirmation of compatibility between object types and the *biological database* in use, and (5) registration of the tool in the *pipeline database*.

3.2.7 Pipeline execution

Execution of a pipeline can be considered at two levels: execution of a single job and that of the entire pipeline. A single job is executed by running three scripts (Figure 3.6): (1) data is retrieved from the database; (2) the tool is executed, and; (3) the results are stored back in the database. Communication with the database is handled by the database connection scripts ("database get" and "database store"), which are equivalent to the Ensembl "RunnableDB" [138]) These two scripts access the database wrapper from Figure 3.5 and provide generic communication with any database of choice.

The data flow in the execution of a single tool wrapper starts with sending the object identifiers describing what objects are to be retrieved from the database as input for this specific job (A). The "database get" script retrieves this data from the *biological database* (B), converts it to BioMOBY format (C) and

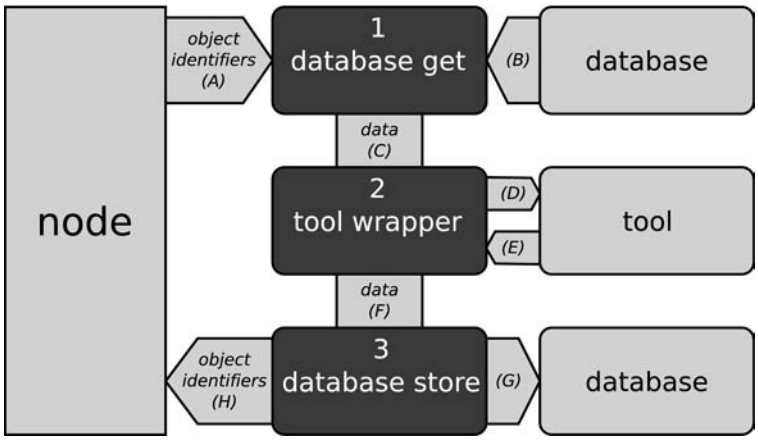

Figure 3.6: The standardized execution and data flow of single job within a node in the Cyrille2 system. See the text for an in depth description.

sends it to the tool wrapper. The tool wrapper prepares the data for the tool (D), executes the tool, interprets its output (E) and converts it to BioMOBY (F). The "database store" script stores the data in the *biological database* (G) and returns the newly created object identifiers (H) back to the system.

A complete pipeline operates iteratively by iteratively running the *Scheduler* and *Executor*. Results produced by a tool under control of the *Executor* will result in more jobs to be scheduled by the next *Scheduler* run. If there are no more jobs to be executed for a node and its parents, it is flagged as "finished". A complete pipeline is finished if all nodes are in the "finished" state. Pipeline iteration can be resumed after new data is uploaded into the pipeline, a database has been updated or the pipeline has changed. Resumption is accomplished by unflagging the "finished" state of one or more nodes in a pipeline. This is either done manually (through the *GUI*) or automatically, for example after a BLAST database update.

3.3 Results

Our local implementation of the Cyrille2 system runs on a dedicated server (dual AMD Opteron 850, 4 Gb memory, 300 Gb disk) and has a 26 node SGE [184] based Linux cluster at its disposal. The tools currently wrapped in the Cyrille2 system are listed in Table 3.2.

Table 3.2: The tools wrapped for use inside Cyrille2 at the moment of writing this paper

Application	Reference	Application	Reference
BLAST	Altschul *et al.* [4]	BLASTIF	in house developed
Clustalw	Thompson *et al.* [189]	InterPRO	Quevillon *et al.* [141]
Genscan	Burge and Karlin [28]	GlimmerHMM	Majoros *et al.* [108]
GeneID	Guig *et al.* [68]	SNAP	Korf [88]
GeneWise	Birney and Durbin [19]	Sim4	Florea *et al.* [50]
GeneSplicer	Pertea *et al.* [134]	Tandem Repeat Finder	Benson [14]
tRNAscan-SE	Lowe and Eddy [105]	Marscan (Emboss)	Rice *et al.* [148]
RNAfold	Hofacker *et al.* [77]	TribeMCL	Enright *et al.* [43]
Inparanoid	Remm *et al.* [144]		

In a test run the Cyrille2 system analyzed 50 *Arabidopsis* BAC sequences (4.8Mb) randomly downloaded from NCBI using the pipeline shown in Figure 3.3. The results of the analyses and the numbers of objects created are summarized in Table 3.3. Measurement of the pipeline execution time is not relevant as the bulk of execution time results from executing the actual tools. As an illustration, however, the analysis of a single BAC as detailed in Table 3.3 typically takes a few hours on a further unoccupied linux cluster with 26 calculation nodes.

The Cyrille2 system is, parallel to the genome annotation project, successfully employed in two other projects. A comparative genomics project aims at predicting orthologs between several fungal genomes and a second project predicts miRNAs in several plant and animal genomes. Both projects successfully use the Cyrille2 system to execute computationally demanding pipelines.

3.4 Discussion

The Cyrille2 system was developed with the aim of providing an easy to use, automated, high-throughput, and flexible and extensible bioinformatics pipeline system. Among its most notable features are the implementation of a powerful job scheduler module, storage of intermediate data, compatibility with different database types for storage of biological data, a generic tool wrapper module, and uniform data transport and data tracking.

Ease of use is achieved through implementation of an intuitive user interface with two distinct layers of complexity. A pipeline administrator is fully authorized to construct and fine-tune bioinformatics analysis pipelines while a

Table 3.3: The results from a test run with the pipeline from Figure 3.3 with 50 *Arabidopsis* BAC sequences (4.8Mb)

Node	No jobs	Object Type	No Generated
Upload Fasta Dna	1	BAC	50
Genscan	50	Exon	6,342
		PolyA	979
		Promotor	926
		Cds	995
		Gene	995
GlimmerHMM	50	Exon	6,493
		Cds	1346
		Gene	1346
GeneId	50	Exon	5,521
		Cds	1,096
		Gene	1,096
SNAP	50	Exon	7,029
		Cds	1,713
		Gene	1,713
GeneSplicer	50	Donor site	59,541
		Acceptor site	69,911
tRNAscan-SE	50	Exon	20
		tRNA	19
MarScan	50	Mar	858
Tandem Repeats Finder	50	Repeat	1,036
BlastIf blastx	50	Blast Hsp	41,451
		Blast hit	16,094
		Raw blast output	50
GFFCds2Transcript	5,150	Transcript Seq	5,150
BlastIf blastx	5,150	Blast Hsp	279,295
		Blast hit	218,978
		Raw blast out	5,150
total	**10,751**		**735,184**

pipeline operator can select from a predefined set of pipelines and nodes to perform complex data analysis tasks. Other pipeline management systems either lack a user interface (Ensembl [138]) or have implemented complex interfaces that can easily overwhelm the non-expert user with choices (Taverna [124]).

The *Scheduler* was designed for flexible and high throughput computational operation. It assesses the pipeline status and determines which new jobs are to be scheduled. In a high throughput data analysis environment it is important to execute jobs in a parallel fashion to optimally use the computational facilities and hence, make pipeline calculation time as short as possible. This requires specific scheduling logic for different node types. In the Cyrille2 *Scheduler*, we employ the "single" type node for analyses in which a job is scheduled immediately after an input object becomes available, and the "all" type node for analyses in which job scheduling requires completion of multiple parent nodes. Each different node type can implement its own scheduling logic. Most pipeline systems implement a scheduling engine able to schedule jobs in parallel [54, 124, 138, 167, 186]. A distinguishing feature of the Cyrille2 system is the embedding of the scheduling logic in the Node class.

Parallel scheduling requires parallel execution, which is controlled by the *Executor*. There are many advanced solutions available for the distribution of jobs over a calculation cluster, amongst others: Sun Grid Engine [SGE, 184], Condor [33], OpenPBS [125] and LSF [106]. For the Cyrille2 *Executor*, we have chosen to employ SGE for distribution of jobs over a cluster and a Condor port is under development. We opted for SGE because it is a stable and flexible system able to handle the high loads necessary. A Condor port is being developed to allow the Cyrille2 system to employ idle Windows desktops. The use of a cluster solution within a pipeline system is a common solution, Ensembl [138] makes use of LSF [106].

Another important aspect in high throughput pipeline analysis is the storage of intermediate results. If this is implemented, the pipeline system will be able to resume calculations close to the point where it may have stopped after a system failure. This feature becomes important when a pipeline requires a long execution time and hence, the chance of a failure, somewhere in the system, increases.

If storage of intermediate data is undesirable, for example because of disproportional usage of storage capacity, it is straightforward to either develop a node-type which embeds two or more other nodes and directly transfers the data between the nodes in a single executor run, or to develop a single tool

wrapper which executes both steps and behaves as a single tool in the system. In both cases, intermediate data storage is by-passed.

A further advantage of intermediate data storage is that each part of a pipeline can be re-executed when necessary. This is essential when only part of a pipeline needs to be repeated with either different parameter settings, after a database update, or upon the addition of extra nodes to the pipeline. In the current implementation of Cyrille2, the system will remove, prior to a rerun, all data that is affected by the update from both the *schedule* and *biological databases* and rerun the necessary analyses. For example, if a new version of a BAC is uploaded, the system will delete all predictions associated with that BAC and repeat the necessary analyses. A planned feature of Cyrille2 is to further refine the update function. For example, if a gene prediction tool is updated with a new gene model matrix, the system will repeat only the analyses for those genes that have changed. This is opposed to a repeated analysis for all genes as is the case in the current implementation.

In the rapidly evolving field of genome annotation, it is critical that a pipeline management system is easily extensible. The Cyrille2 system was designed to cope with possible future requirements of incorporating novel tools, data types, or databases in a generic fashion. For example, for a present-day genome annotation project, it is mostly sufficient to store all relevant data in a biological database such as the Generic Genome Browser database [180]. However, if one would require the inclusion of data such as multiple alignments or 3D protein structures, a different database is required. The Cyrille2 system is designed to make the addition of a novel object type or the complete change of the biological database as easy as possible. This is achieved by implementation of the database wrapper as a separate module. Addition of a novel data type can be done by adding a "get" and "store" function for this type of data to the database module. To connect to a new database, a new module must be written with a storage and retrieval function for each object type.

This mode of integration of a third party database with a pipeline system is unique for the Cyrille2 system. Many alternative systems do not use a database for storage of intermediate results (Taverna [124], GPIPE [54]and Wildfire [186]) but either transport the output of one program directly to the next program or store intermediate results as flat files. Such an approach has its

drawback for a high throughput system, as a database is better suited to keep track of many, possibly millions, of intermediate objects and better adapted to distribute data in a heterogeneous environment. Other systems employ a database for data storage (Ensembl [138], Pegasys [167] and MOWserv [123]) but these systems are strongly linked to one specific database, thus limiting their flexibility. In our current implementation of Cyrille2, the Generic Genome Browser database [180] is used for storage of the biological data and for viewing by end users.

The use of BioMOBY as a communication standard combined with the storage of standardized object identifiers by the Cyrille2 system ensures that any object can be handled and tracked by the system, including binary objects such as images [199, 200]. Other advantages of using BioMOBY are that it ensures easy integration with the growing body of external BioMOBY web services and optimal inter-connectivity between nodes. Several systems employ specific embedded scripts to translate the output of one tool to the input of the next (Wildfire [186], FIGENIX [63] and GPIPE [54]). Most analysis tools have a unique in- and output format and thus the number of unique translation steps grows quickly with the number of tools wrapped. This can be mitigated by using uniform (BioMOBY) data transport as implemented in Cyrille2, Taverna [124] and MOWserv [123]. The Ensembl system employs a uniform Perl data structure to the same end [138, 175].

The Cyrille2 system has been developed to operate in a high throughput sequencing facility with a need for robust, automated and high throughput genome analysis, and easy creation, adaptation and running of pipelines by non-expert users.

Most of the pipeline systems recently released are developed as a workbench for bioinformaticians. Some systems excel in the way they allow for complex pipelines to be built through a visually appealing but sometimes complex *GUI* (Taverna). On the other hand, most systems are not suited for automated, high throughput operation. Ensembl [138] is an obvious exception in this respect, but it is very complex to deploy the system to other sites, it is far from flexible, difficult to extend, strongly tied to the Ensembl database, and it is difficult to adapt a pipeline for non-expert users.

In view of the distinctive functionality and combination of features implemented in the Cyrille2 system we believe that it is a valuable addition to the array of pipeline systems available and particularly useful in environments that require high throughput data analysis.

Chapter 4

Interactive visualization of comparative genome annotations

Mark Fiers, Huub van de Wetering, Tim Peeters, Jack van Wijk and Jan Peter Nap

A modified version is published in *Bioinformatics*, 22(3):354-5, 2006.

Abstract

Visual exploration of DNA sequences and their multiple annotations is an important help in assigning function to sequence. Both the increasing volume of sequence data and the highly variable scale of annotation (from nucleotide to chromosome) challenge the proper display of annotation data, especially when comparing different annotations within or between genomes. In this context, the advances and advantages of information visualization have to be investigated and implemented.

Several new methods for interactive and real-time visualization of DNA sequences and their comparative genome annotations are presented in a software package called DNAVis. Modern PC graphics hardware in combination with concepts and methods of information visualization such as linked views, focus+context, perspective walls,

semantic zooming and dot plot-like matrix views results in novel approaches for obtaining better insight in large datasets containing multiple and comparative genome annotations.

The software is freely available at `http://www.win.tue.nl/dnavis`

4.1 Introduction

Genomics research is resulting in massive volumes of DNA and notably annotation data is still rapidly expanding. The approximately 130 million nucleotides of the *Arabidopsis thaliana* genome contain over 26000 separate genes in the latest TIGR annotation [71], and a wealth of associated annotations, such as splice sites, homologs and paralogs. More annotations will be added through ongoing research, for example markers, expression, possible relationships with miRNA, transcription factor binding sites and many more. The goal is to understand the function of DNA in action. Therefore, the analysis of annotation should now be considering the individual nucleotide (the lowest level), up to complete chromosome organization. The scale and size of current sequence and annotation datasets require appropriate and novel tools to explore and retrieve biological relevance from such data. Visual exploration tackles such a challenge by presenting information interactively and in real-time, by appealing to the intuition of biologically skilled user and by exploiting man's natural abilities to build mental maps of visually presented data.

In continuous interactions between (plant) bio-informaticians, genome biologists and visualization specialists, we have defined [132] and further refined the features that visual exploration of comparative annotation should offer. The need to visualize different scales (nucleotide to chromosome), the increasingly heterologous nature of annotations, as well as the need for comparisons of such annotations is demanding. We have identified the visualization concepts and technologies that we think are most appropriate for the implementation of such desired features: semantic zooming, perspective walls, linked windows and dot plot-like matrix views. Current genome viewers, such as the Generic Genome Browser [180], Apollo [98], Artemis [152], Artemis Comparison Tool [29], Ensembl [34], Entrez [188] or the Microbial Genome Viewer [83] can generate excellent visualization of genome data, but generally lack one or more

of the features here defined as desirable. Viewers operating through a web interface [79, 180, 188] do not offer smooth scrolling and zooming, but operate stepwise with distinct pauses between views when new data is downloaded from their servers. Others offer continuous panning [98, 152], but lack continuous zooming. The display of comparative genomics data is, if possible, a major challenge for all viewers. More advanced in this area is the Artemis Comparison Tool [29] drawing colored areas depicting a similar area between two linear representations of a sequence and its annotations.

We here present new approaches to the visualization of comparative genomics data implemented in software called DNAVis. DNAVis implements new methods for improved, real-time interaction with and visualization of genomes for the exploration and comparison of annotated DNA sequences. In Peeters *et al.* [132] we have described a first version of this system and discussed it from the point of view of information visualization. Meanwhile, we have extended DNAVis with a number of additions and here we give an overview of the system from the perspective of prospective users from the bioinformatics research community.

4.2 Methods

4.2.1 Design issues

In visual genome exploration, continuous, real-time interaction with the data is considered essential for getting a feel for the dataset at hand. It is necessary to be able to view and relate multiple and different genomic regions simultaneously using any measure of similarity. Moreover, the display of such genomic regions should be connected in such a way that operations in one view translate immediately to equivalent operations in other views. In all views it should be an option to display context. Equally essential is high flexibility of scale. It should be possible to evaluate complete chromosomes with an arbitrary number of annotations and to explore them from the lowest scale (individual nucleotide) to whole genomes. To meet these requirements, the software uses modern graphics hardware to allow smooth and real-time interaction with datasets as large as complete genomes. Information visualization

technology such as linked views [53], perspective walls [107] and semantic zooming [133] supply the visual means to realize these requirements.

The ability to visually explore comparative annotations within or between genomes was a particular challenge. Comparative genomics data can be displayed in several ways. A (multiple) sequence alignment [128, 189] is relatively simple, but is essentially restricted in scale. Such sequence alignments do not allow the addition of other annotations such as expression data. Also, it is difficult to get an overview of large sets of aligned sequences. An improvement over such alignments is the use of two or more bar-like representations of a region of DNA with lines or colored areas between the bars indicating similarity [29, 78, 98]. In such a visualization colors can distinguish several types of comparative information but the view becomes chaotic when many similarities are displayed. Another well-known method to compare genomic data is the dot plot [173]. In a dot plot, two sequences are shown perpendicular to each other and each similar attribute is shown in-between these sequences with a dot. Even though a dot plot is restricted to two sequences, this approach was developed into what we call the "matrix view". We deem this is the most powerful for the display of complex heterologous data in comparative annotations.

4.2.2 Implementation, downloads and pre-computed datasets

DNAVis is written in C++ and employs the widely used OpenGL library for visualization. The program runs on both Linux and Microsoft Windows operating systems and can be downloaded (`http://www.win.tue.nl/dnavis`). A detailed manual describing how to operate the software is included with the software download. To comfortably use the software, notably with larger datasets, a computer with at least 512 Mb memory and modern accelerated 3D graphics hardware is recommended. Sequence data should be in the FASTA file format, annotation data require the GFF format [57].

Displaying a whole genome with comparative data in a dot plot-like matrix is computationally not trivial. Therefore, comparative datasets were pre-computed, except for the classical nucleotide versus nucleotide dot plot that is generated on the fly. The use of pre-computed comparative data makes it

easy for a user to decide what is being displayed since it allows one to create and fine-tune private datasets. The user can define several datasets containing a FASTA file and related GFF annotation files which the user can import in DNAVis.

Several example datasets can be downloaded from the DNAVis website. All data is based on the latest TIGR (Ath1-v5) annotation of *Arabidopsis thaliana* [71]. The most basic dataset (not pre-computed but a direct translation) describes the TIGR Ath1 annotation and is called "DvAth1". The other datasets are pre-computed and contain comparative information for matrix views. The dataset named "DvBlast" contains similarity information derived from a BLAST [4] analysis of all *Arabidopsis* genes against a database with all *Arabidopsis* genes. All genes with a BLAST score over 200 are recorded as similar and will in DNAVis result in a dot. The dataset named "DvMPSS" contains similarity information about all genes co-expressing in 14 *Arabidopsis* MPSS libraries [116] as described by Ren *et al.* [145]. The third set named "DvVMatch" contains the locations on the *Arabidopsis* chromosomes which are exactly identical on a stretch of 200 nucleotides with at most two mismatches or inserts. This dataset is generated by Vmatch (`http://www.vmatch.de`). The scripts to generate these datasets are written in Python and are available for download at the DNAVis website.

4.2.3 Details of the software

After having started the program, the first step is to load data. As many datasets as desired can be loaded. After data import, DNAVis can display two types of views, a bar view and a matrix view. Both are described in more detail below. Different and multiple types of views can be displayed simultaneously and any view of interest can be stored for later inspection.

The bar view

A bar view consists of a stack of bars. See Figure 4.1 for an example. Each bar displays a sequential, linear representation of a DNA strand. This approach is roughly equivalent to genome views as drawn by several other genome viewers

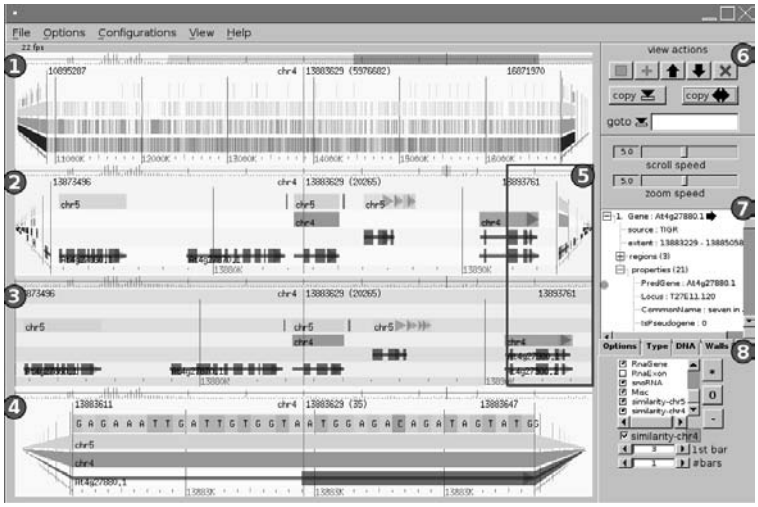

Figure 4.1: The DNAVis interface shows four bar-views all displaying a part of chromosome 4 of the TIGR Ath1v5 annotation of *Arabidopsis thaliana* [71] at different zoom levels. The main part of the interface is occupied by the four bar views (boxes 1 to 4). The views are linked together and center on the same position on the chromosome. In all views, the same annotations have the same colors. Genes are blue, parts that are similar (based on a user defined criterion) are green for chromosome 4 and yellow for chromosome 5. Purple annotations are RNA genes. The background color is blue for selected views, green for views linked to the currently selected view and yellow for unselected views. The top part of each view shows a histogram overview of annotation density of the complete chromosome. The red highlighted part of this histogram is the part currently displayed with the lighter parts of this being those parts displayed on the perspective walls. The top view shows a large part (approximately 30%) of chromosome 4. At this zoom level it is possible to see that genes are distributed over the chromosome with small fluctuations. Boxes 2 and 3 both show a much smaller area (of approximately 20k). These two views zoom in on an area containing 5 genes and display several areas of similarity. The difference between box 2 and 3 is the presence of perspective walls in box 2. Box 5 highlights the difference between a view with and without perspective walls. The bottom bar view (box 4) shows the same area in close-up, now showing nucleotides. The user interface offers possibilities to manage the display of the bar views. The top right part (indicated with 6) allows management of the different views, moving them around, creating new ones or deleting views. The two copy buttons allow synchronization of linked interfaces. The two sliders below regulate scroll and zoom speed. In 7 a tree will appear upon selection of an annotation describing the information about that annotation. In the right lower part (8), several tab pages allow detailed control of the currently selected view. Here it is possible, amongst other options, to turn annotation types on or off to determine their position. A colour version of this image is included as an insert.

(Hubbard *et al.* [78], Rutherford *et al.* [152], Stein *et al.* [180]), but several improvements were implemented. In each bar, an annotation is shown as a rectangular glyph aligned to a representation of a DNA strand to show its position on the DNA. The orientation of the annotation is shown by a small

transparent triangle. Gene annotations are displayed as a group of small boxes (exons) connected by a thick line. Annotations have types, for example a gene, an RNA gene or a microRNA. The visualization and layout in a bar view are configurable per type, for example, the number and position of bars used, the visibility and the color. The total number of bars is also configurable. The coordinates at the top and the bottom of a bar view show the position of the currently displayed area. The overview bar displayed on top shows in red which area of the chromosome is displayed and a histogram which shows the annotation density over the whole DNA sequence. The bar view can be fluently panned by dragging it. More information is being displayed in the user interface upon clicking an annotation. There can be multiple annotations selected in case their glyphs overlap. Upon clicking on a name in the collapsed tree, more information is revealed. Zooming in or out is done smoothly by dragging the view vertically (Figure 4.2). Semantic zooming [133] determines the visual appearance of glyphs depending on the zoom level. Gene structure, gene names and ultimately nucleotides are shown when zooming in. Another novel feature for genome visualization software is the application of perspective walls [107] at both the right and the left side of the bar. These walls show the neighboring areas (the context) of the currently displayed region (the focus). These perspective walls are the implementation of a well-established concept in information visualization called focus+context [174] that prevents users from getting lost when examining a relatively small area of a much larger body of information. The overview bar at the top of a bar view shows which areas are visible on the perspective walls by a lighter shade of red.

In addition, it is possible to create an arbitrary number of views of the same sequence, only limited by the size of the screen and the speed of the graphics hardware. Each view can show different zoom levels and a user specified selection of annotation types. Multiple bar views can be linked together so that these views pan and zoom simultaneously [53]. This makes it easy to examine for example the same area at different zoom levels or to compare orthologous areas of chromosomes in different organisms. Linked bar views can be synchronized on location and zoom level.

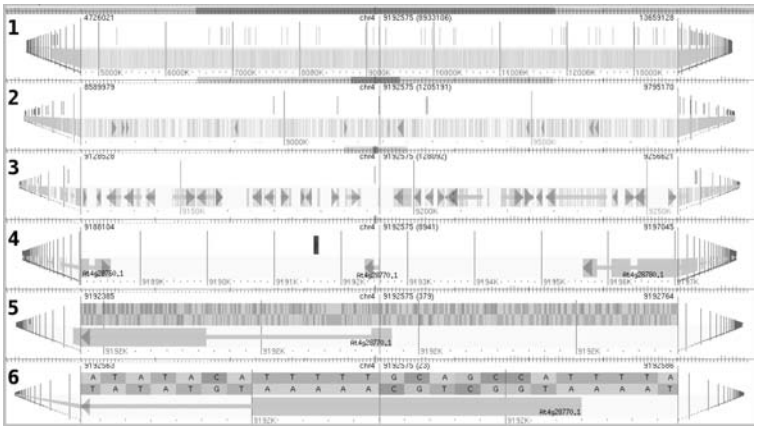

Figure 4.2: An illustration of zooming capabilities of the DNAVis linear view. From the top to the bottom are six views (1-6) of chromosome 4 of *Arabidopsis*, displaying different levels of detail. Each view displays two features, RNA coding genes (dark grey) in the top bar and protein coding genes (light grey) in the bottom bar. The top view shows $\pm 50\%$ of the chromosome whereas the bottom view is zoomed in to 23 bases at the start of a gene. Semantic zooming reveals the names of displayed genes only in the bottom three views where there is sufficient display space.

The matrix view

In the classical dot plot [59, 173] the axes represent the sequences compared and the dot in between these axes represents the shared feature (often that is identity). In DNAVis, comparative genomics information is displayed in a dot plot-like matrix view. This view consists of two axes of perpendicularly placed bar views as described above that define the dot plot-like matrix area. All features described above for a bar view apply to both axes in the dot plot-like matrix view. Annotations defining two areas as similar are displayed as a rectangle in the matrix area. The definition of what is similar is fully defined by the user. DNAVis allows multiple similarity datasets to be displayed in the same matrix in different and configurable colors. For example, it is possible to visualize gene similarity (the DvBlast dataset) in red and gene co-expression score (the DvMPSS dataset) in the same matrix in green (Figure 4.3). Fluent zooming and scrolling is also possible in a matrix view. The two bar views in the matrix view can be linked with each other or with other bar views (for example, to have a very detailed bar view of a part of the matrix view) as described above. Data on the user-defined similarity of annotations must be pre-computed and supplied in the GFF file format. Zooming in to the

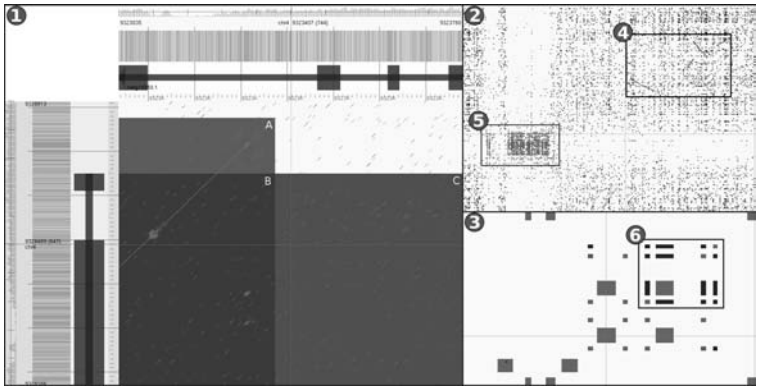

Figure 4.3: The DNAVis interface displays three dot-plot-like matrix views as generated by the DNAVis software package. All data is derived from the ATH1v5 *Arabidopsis thaliana* annotation of TIGR [71]. Red blocks represent an area designated to be similar based on a BLASTP [4] analysis of the predicted proteins (data set name: DvBlast; see accompanying website `http://www.win.tue.nl/dnavis`). Each pair of genes that generated a hit with a score of at least 150 results in a red block. Dark blue dots represent pairs of genes that have a Pearson correlation coefficient of more than 0.98 in the MPSS expression data (data set name: DvMpss [115]), purple areas show strong similarity on chromosome DNA level as calculated by Vmatch (dataset name: DvVmatch, [196]). The green lines are dot plots generated on the fly, that show any window of 7 nucleotides that is similar to another area on the other axis, with one mismatch allowed. Part 1 shows a close-up of a self-comparison of chromosome 4 with only the matrix view shown. The rest of the user interface is omitted as it is equivalent to Figure 4.1. Two linear views are visible to the top and left part of the image. In this case, the linear views only show annotated genes (in blue) and a bar representing nucleotides (designated by color). In the center there are three annotations visible, a large purple block from the DvVmatch data set showing a large area that is almost perfectly similar (1 mismatch every 200 nucleotides allowed). The red block marks the two genes in the top and left bar as having a blast score over 200 against each other. The small green lines are a dot plot that verifies the purple perfect hit block and shows further that blast similarity between these two genes is likely to reflect the perfect hit, because the dot-plot shows not much similarity in the rest of the gene. It also does not seem to follow intron/exon boundaries. Part 2 shows a much larger comparison of chromosome 4 with chromosome 5. The view shows approximately 70% of both chromosomes. The view clearly hints at several ancient duplications in red (box 4), showing the regions α_{21} and α_{22} as described by Bowers *et al.* [23]. The area with highly similar repeated elements depicted as purple dots shows the centromere (box 5). Part 3 is a close-up of a so-called quadruplet (box 6) of co-expressing genes (with a cutoff of 0.70) as described by Ren *et al.* [145]. A colour version of this image is included as an insert.

lowest scale of the individual nucleotide yields classical dot plots for nucleotide comparison. Such dot plots are generated on-the-fly and it is possible to configure color, window size and mismatch tolerance.

4.3 Discussion

To get a good feel for and therefore a proper analysis of a genomics dataset continuity and real-time behavior of the visualization are considered to be very important. DNAVis uses advanced methods for visualization depending on modern PC graphics hardware. The advanced technology results in smooth and real-time interaction with datasets as large as complete chromosomes with large numbers of different annotations. With semantic zooming, it is possible to seamlessly zoom from annotations on the nucleotide scale to annotations on the chromosome and whole genome level. To see where the currently viewed area is on the genome, the current position is highlighted. If so desired, views can be linked to scroll and zoom together. Perspective walls provide context for the current area of interest and help a user to place any detailed analysis in the context of information from a substantially larger area. For exploring heterologous annotations within or between genomes or genomic regions, a dot plot-like matrix view offers all possibilities of the linear view in two dimensions. For the current implementation of DNAVis several improvements are conceivable. Improvements could include interactive database connections and the possibility to edit annotations in a view, more extensive search facilities and/or more freedom in glyph shapes.

The addition of a third dimension in our visualization would result in more space to display data. This could be of use with yet larger or more dimensional datasets. However, 3D visualizations inherently introduce occlusion and consequently require (interactive) definitions of appropriate view points. The associated problems with interpretation for a user add to the complexity of using 3D in genome visualization. In view of the way DNAVis manages to visualize large datasets without the drawbacks of 3D, it has been our deliberate choice not to implement any 3D visualization.

The future of the visualization of genome information is likely to focus on the display of comparative heterologous data. Both heterologous data and the desire to compare multiple datasets create additional challenges. The dot plot-like matrix view offers a comparison of not more than two sequences. We have demonstrated that an integrated display of multiple heterologous datasets is feasible in such a matrix. Future applications will have to invent

ways to display heterologous comparative information for many more genomes. For example the ability to interactively and in real-time compute and visualize complex relationships between multiple annotations will be a major challenge for the future of genome visualization. This is likely to lead to new insights in genome structure and organization. The careful consideration and use of information visualization technology has already resulted in an efficient and effective approach for modern genome exploration. The team involved in creating this novel approach has recognized that in retrospect it was rather surprising how little of the standard techniques of visualization science had yet found a place in genome visualization. This opens up many future promises for comparative display and exploration of genomes.

Chapter 5

Predicting the micro-RNA potential of the Arabidopsis genome

Mark Fiers, Ludmila Mlynarova, Willem Stiekema and Jan-Peter Nap

Abstract

Micro-RNAs (miRNAs) constitute a new level in the circuitry of gene regulation. The numbers of active miRNAs in plants and the scope of miRNA-based gene regulation are currently an issue of debate. We here present an *in silico* analysis of the miRNA potential of the *Arabidopsis thaliana* genome. We have identified all genomic sequences able to form a predefined hairpin structure as potential miRNA precursors with the help of the repeat-finder program REPuter. The stem sequences identified were subsequently used in low-stringency gapped BLAST analyses to identify potential target genes. MiRNA candidates are those hairpin forming structures that have a putative target gene. In this way, we predict no less than 2,427 possibly biological relevant miRNA candidates in the *Arabidopsis* genome. Out of seven randomly picked predicted miRNA candidates, five showed the presence of small RNA *in vivo*. This laboratory confirmation indicates that a major part of the set of predicted miRNA candidates is likely to exist *in vivo* and may have biological relevance. These results

indicate that many more miRNAs are likely to be present in the *Arabidopsis* genome than predictions of numbers implied by current genomics studies.

5.1 Introduction

The discovery that small, ∼22 nucleotide (nt) short RNA species, known as microRNAs (miRNAs), are involved in gene regulation, expands the role of RNA beyond the regulatory role of other types of RNA, such as tRNA, snRNA and snoRNA [73, 209, 210]. Well over 700 miRNAs have now been identified in 8 plant species of which 117 in *Arabidopsis* and over 2,500 in 27 animal species (miRBase [67]). MiRNAs act through complementary pairing with an mRNA. If the miRNA binds with high complementarity the mRNA is degraded. If the hybridization is less specific, mRNA will not be degraded but translation is inhibited. The former is the usual mode of operation in plants, the latter in animal systems, but neither appears exclusive [48, 84]. The miRNA biogenesis starts with transcription of a pri-miRNA which is digested into the precursor miRNA (pre-miRNA). The pre-miRNA has the characteristic hairpin that is digested by the enzyme Dicer (animals) or a Dicer-Like enzyme (DCL, in plants) into the mature ∼22 nucleotide miRNA. The process is illustrated in Figure 5.1, for a more in-depth description see Kim [84], Millar and Waterhouse [118] or Filipowicz [48]. MiRNAs share at least part of their biosynthesis pathway with small RNAs involved in gene silencing, known as siRNAs. Both are small RNA molecules of ∼22 nucleotides, generated by the activity of the ribonuclease complex known as DICER-LIKE1 [80, 160] and their function can be interchanged [37].

An indication of miRNA function can be inferred from discovery of its target. This can be done by identifying a position on a mRNA complementary to the miRNA [147]. Candidate targets are easier to detect in plants than in animals due to strict miRNA-mRNA paring [84]. More extensive proof of miRNA presence is necessary with as a first, good, indication the observation of the mature miRNA in, for example, an RNA blot. The few miRNAs that have been studied in more detail in animals and plants (see Table 5.1 for some examples) indicate their involvement in basal developmental processes. For example, the 'archetype' miRNA, *let7* from *Caenorhabditis elegans* is involved in regulation

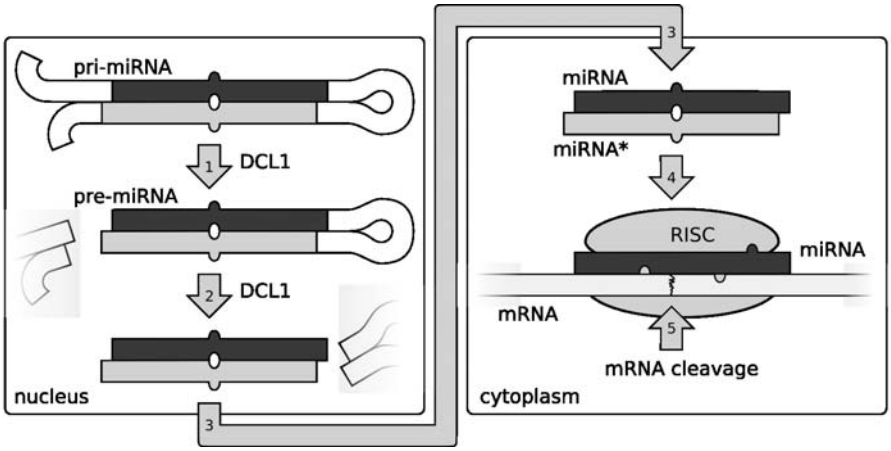

Figure 5.1: A schematic representation of the miRNA generation pathway in plants. In the nucleus the pri-miRNA is transcribed which is digested by DCL1 (DiCer Like, step 1) into the precursor miRNA. In animal systems this step is performed by DROSHA. The precursor is the digested by DCL1 into the mature miRNA (step 2). One of the strand of the hairpin stem will become the miRNA. The other strand, called miRNA* will be degraded. The miRNA is transported to the cytoplasm (step 3) where it is incorporated into the RISC (step 4). In the RISC complex the miRNA binds to a complementary mRNA which is subsequently cleaved by the complex (step 5). Figure adapted from Millar and Waterhouse [118].

of early embryonic development [96, 142]. It is still the question if all miRNAs are involved in such basal and conserved processes. The majority of predicted miRNAs have been found through a comparative approach and it has long been assumed that all miRNAs were conserved among species (for example, Reinhart *et al.* [143]). Recent research clarified that many miRNAs may be unique to a single species [15, 101].

A major challenge in whole genome annotation nowadays is thus to determine how many miRNAs are hidden in a given genome and what their function is. The structural characteristics of known miRNAs allows the *in silico* prediction of potential miRNA molecules. Several analyses [1, 93, 99–101] present a variety of approaches. Most methods rely heavily on an evolutionary approach which will obviously fail to recognize species-specific miRNAs. In this paper we opt for a single genome approach using the repeat finder program REPuter [90]. The use of REPuter allows ignoring of complex RNA folding patterns and helps to focus on short-spaced hairpin structures. Analysis of a single genome generates insight in the complete miRNA potential of the genome and may indicate to what extent miRNAs could be involved in gene regulation.

Table 5.1: A selection of miRNA of which the function is known.

miRNA	**Organism**	**Target gene(s)**	**Involved in**
lin-4	CE	*lin-14&lin-28*	Early development
let-7	CE	*lin-41&hbl-1*	Early development
lsy-6	CE	*cog-1*	Chemoreceptor expression
bantam	DM	*hid*	Apoptosis and growth control
miR-14	DM	unknown	Apoptosis and fat metabolism
miR-181	MM	unknown	Hematopoietic differentiation
miR165/166	AT	*REV*	Axial meristem initiation and leaf development
miR172	AT	*AP2*	Flower development
miR-JAW	AT	*TCP4*	Leaf development, embryonic patterning
miR159	AT	*MYB33*	Leaf development

This table is derived from Bartel [10] and references therein. The organism abbreviations are: CE - *Caenorhabditis elegans*, DM : *Drosophila melanogaster*, MM : *Mus musculus* and AT : *Arabidopsis thaliana*

Any approach will identify false positives as relatively short hairpin structures can occur randomly. To improve the relevance of the prediction, the prediction of potential precursors is followed by a low-stringency gapped BLASTN [4] of the stem sequence against all known coding sequences of *Arabidopsis* from the May 2002 MIPS annotation [163]. Each potential miRNA precursor of which the stem sequence has at least one potential target gene is scored as a predicted miRNA candidate. In this way, no less than 2,427 predicted miRNA candidates are identified. To validate this prediction experimentally, from seven randomly chosen predicted miRNA candidates, five were confirmed to generate small RNA of the predicted size in a total RNA fraction of wild-type *Arabidopsis*. This implies that the number of active miRNA molecules in *Arabidopsis* might be 1,733 or higher and is with 95% confidence higher than 828. Based on various biological considerations and additional data, the latter is likely to be a lower estimate. These analyses support previous suggestions [101, 103, 153] that the number of miRNA/target gene combinations present in (plant) genomes may be much larger than hitherto shown. If so, miRNA-mediated gene regulation may represent an even more widespread and important mechanism for gene regulation than is now assumed.

5.2 Materials and Methods

5.2.1 In silico identification of potential miRNA precursors

The full MIPS *Arabidopsis* annotation from May 11, 2002 [111, 163] was stored in an object-oriented database (ZODB [212]) build to contain various indices for easy searching and retrieval as well as a web-based front-end (in Python [140]) for visualization of annotated genome parts (up to whole chromosomes). This local database was used for all subsequent analyses. Palindromic repeats, defined as repeats in reverse complement, were detected with the repeat-detecting software REPuter [90]. The parameter settings used were: only palindromic repeats, allmax on, length 20, error rate 10% (*i.e.* two mismatches, gaps or deletions allowed in a string of 20 nucleotides). For performance reasons, the *Arabidopsis* genome was analyzed in sliding windows of 10 kb with an overlap of 500 bp. Under these conditions, a genome-wide analysis required approximately three hours CPU time on a dual processor machine running Linux. Scripts in Python subsequently parsed the output list of repeats generated to identify and discard duplicated repeats due to the overlap in the sliding windows. Potential miRNA precursors were then selected on the basis of a maximal length of 100, 200 or 350 nucleotides (including repeats) of the miRNA precursor molecule. All resulting potential miRNA precursors were stored in a separate MySQL [122] database.

5.2.2 In silico prediction of miRNA targets

To identify the putative target genes of the potential miRNA candidates, all *Arabidopsis* coding sequences (CDS) plus 250 bases 5' and 3' untranslated region (UTR) were extracted from the database to create a second, CDS/UTR database. Each potential miRNA precursor sequence identified was blasted against the CDS/UTR database, using a stand-alone version of the NCBI implementation of BLASTN. BLASTN parameter settings used were: gapped, mismatch -1 (compared to the default value of -3), low complexity filter off. Potential target genes were defined as those genes with a high scoring segment pair (HSP) of minimal 20 nucleotides with maximal 20% (four out of

Table 5.2: Number of potential miRNA precursors identified in the *Arabidopsis* genome.

		Maximal precursor length		
Chromosome	**Length 10^6 nt**[a]	**100 nt**	**200 nt**	**350 nt**
1	30.15	674	1,482	2,288
2	19.84	520	1,123	1,842
3	23.77	469	1,122	1,858
4	17.79	429	953	1,573
5	26.99	585	1,396	2,309
Chloroplast	0.15	16	28	29
Mitochondrion	0.36	0	0	3
Total		2,693	6,105	9,902

[a]nt: nucleotides

twenty) mismatches with a potential miRNA precursor molecule. All resulting candidate target genes were stored in the MySQL database as well.

5.2.3 Hybridization with small RNA in Arabidopsis total RNA preparations.

Total RNA was isolated from flowers, leaves, stems, and siliques of *Arabidopsis* using the TRIZOL reagens according to the instructions of the manufacturer (Invitrogen). The RNA was quantified and separated on 8% polyacrylamide gels as described [81] previously. For each chosen predicted miRNA candidate, primers were designed to generate a 200 to 300 nucleotide fragment from *Arabidopsis* genomic DNA by PCR containing the full miRNA precursor and some neighboring DNA. This PCR fragment was labelled by random prime labeling and used for RNA blot hybridization at 50°C as described [81]. When hybridization with small, 21-24 nucleotides short RNA was obtained, the prediction was considered validated. In individual cases, primers were designed on both strands of the predicted stem-loop sequence. These were end-labeled with polynucleotide kinase and [-32P] ATP (3,000 Ci/mmol, Amersham) and used for RNA blot hybridization at 42°C in formaldehyde buffer. Washing was done as for the PCR probes, but at 42°C. Each blot was checked for RNA loading by subsequent hybridization with a 100 bp PCR fragment carrying the DNA sequence of the 90 nt U6 snRNA from *Arabidopsis*. Autoradiographs were visualized on a BASReader 2000 and analyzed with TINA software.

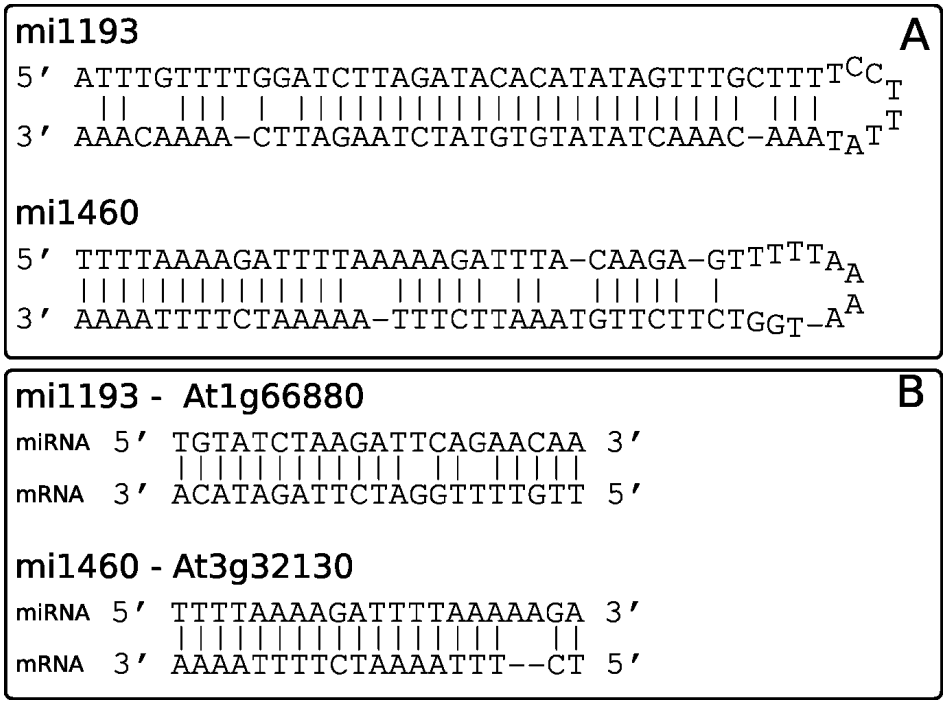

Figure 5.2: Two predicted miRNA candidates; mi1193 and mi1460. A) The predicted precursor hairpins and B) The alignments of these candidates with one of their hypothetical target genes. mi1193 targets 21 genes of which At1g66880.1 was selected, a member of the serine/threonine protein kinase family. mi1460 targets 15 genes, of these the reverse transcriptase (At3g32130) described in the text is shown.

5.3 Results

The program REPuter [90] was used to analyze the May 2002 MIPS annotation of the *Arabidopsis* genome [163] for the presence of potential miRNA precursors. The presence of a palindromic, *i.e.* reverse complementary repeated, sequence of 20 nucleotides or longer with a maximum of two errors (either mismatches and/or gaps) was taken as selection criterium for the identification of such potential miRNA precursors. This configuration is able to form the hairpin characteristic for the precursor of active miRNA species [153]. The total length allowed for the palindromic sequence and interrupting spacer was varied between 100, 200 and 350 nucleotides. The results obtained are given in Table 5.2 and two predicted hairpins are shown in Figure 5.2. Depending on the total length allowed, large numbers of potential miRNA precursors are identified, up 9,902 in case of the 350 nucleotides length setting (Table 5.2). Potential

Table 5.3: The miRNAs in miRBase [66] corresponding to the ones identified in this study.

miRBase id	**this study**	**miRBase id**	**this study**
ath-MIR156a	mi1051	ath-MIR390a	mi1145
ath-MIR156b	mi2045	ath-MIR390b	mi2642
ath-MIR156c	mi2049	ath-MIR400	mi184
ath-MIR156d	mi2137	ath-MIR403	mi1190
ath-MIR156e	mi2140	ath-MIR404	mi169
ath-MIR156f	mi2217	ath-MIR405a	mi1019
ath-MIR156h	mi2633	ath-MIR405b	mi2593
ath-MIR158a	mi1214	ath-MIR405d	mi1747
ath-MIR172a	mi1081	ath-MIR407	mi1111
ath-MIR172c	mi1219		

miRNA precursors also occur in the genomes of chloroplasts and mitochondria (Table 5.2). Although plant miRNAs precursors may be longer in length than 100 nucleotides [143], we will present all subsequent analyses for the set with a maximal length of 100 nucleotides. This will generate a lower limit estimate for the actual number of potential miRNA precursors present in the *Arabidopsis* genome. Given the size class of 100 nucleotides, a total of 2,693 potential miRNA precursors are present in the *Arabidopsis* genome, including sixteen in the chloroplast genome, but none in the mitochondrial genome (Table 5.2). Determining the chance occurrence of a potential miRNA precursor with the configuration as defined here is not trivial. The chance occurrence of two inverted repeats on a given distance with possibilities of inserts, deletions and mismatches cannot be calculated exactly, but was approached by repeated sampling [91]. In a randomized genome of the size and composition of *Arabidopsis*, the chance occurrence of potential miRNA candidates is estimated to be about 80 (binomial distribution; Kurtz and Myers [91]). The occurrence of 2,693 potential miRNA precursors therefore indicates that the *Arabidopsis* genome has a much higher propensity to generate and/or maintain potential miRNA candidates than to be expected by chance alone (standard Chi-square test, P¡0.001). The set of 2,693 miRNA precursors contains 19 of the 117 *Arabidopsis* miRNAs in miRBase (Table 5.3, Griffiths-Jones [66]).

Detailed analyses of the characteristics of the potential miRNA precursors reveal a range of 20 (the lower limit set in the analysis) to 45 nucleotides for the stem-loop length. The spacer size ranges between zero to 60 nucleotides (the upper limit set by the 100 nt total length criterion). The class of putative

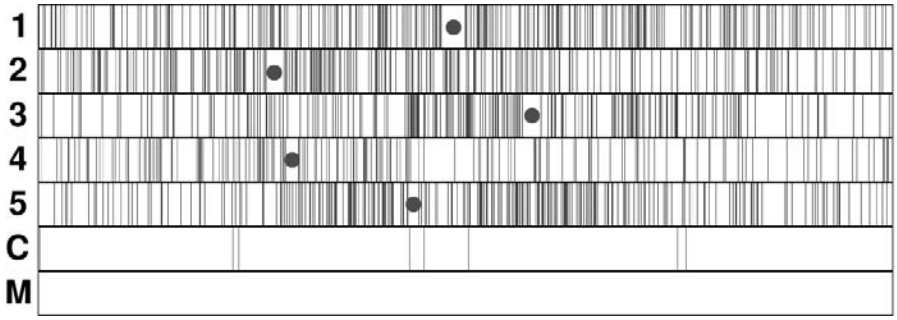

Figure 5.3: Distribution of the identified potential miRNA precursors over the five *Arabidopsis* chromosomes and organelle genomes. Each vertical bar represents a potential miRNA precursor (total: 2,693). Each black dot depicts the centromere region (where generally sequences are lacking). The horizontal bars labelled 1 to 5 represent the respective chromosomes, the bars marked C and M give the chloroplast and mitochondrial genome, respectively.

miRNA precursors with a palindromic repeat of just 20 nucleotides is the largest class with 825 members. The distribution of all 2,693 potential miRNA candidates over the *Arabidopsis* chromosomes and chloroplast genome is given in Figure 5.3. Apart from still missing sequences in the centromeric regions (given as dot in the bar), the different chromosomes show that areas more rich in potential miRNA precursors than other areas tend to be closer to the centromeres. More detailed inspection of all potential precursors shows that the set identified does not contain very simple repeats or micro-satellites. For example, only four percent contains the stretch $[AT]_4$ (data not shown).

Obviously, the set of 2,693 potential miRNA precursors identified may contain families and/or duplicates. To assess the amount of relatedness in the 2,693 potential precursor set, the sequences were analyzed using the assembly software Gap4 [22]. Sequences were considered related if they had at least 95% similarity over a stretch of at least 15 nucleotides. In Figure 5.4, the resulting family distribution of the set of potential miRNA precursors is given. The 2,693 potential miRNA precursors establish 1,928 families, that vary in size from 1 to 39 members. Of all families, there are 1,721 unique miRNAs in one-member families. As relatedness appears not to be a major issue, further analyses were carried out on the basis of the complete set of 2,693 potential miRNA precursors. When the genomic position of these potential miRNA precursors was analyzed in relationship to the position of all annotated genes in the *Arabidopsis* genome, only 117 showed an overlap of five nucleotides or

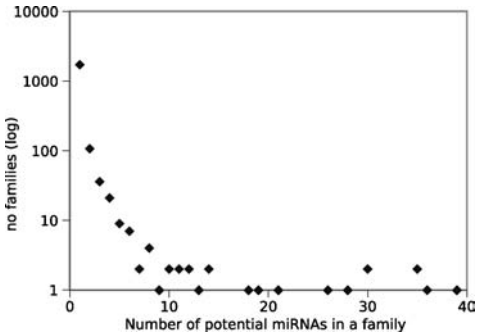

Figure 5.4: Distribution of potential miRNA precursors in families. The 2,693 potential miRNA precursors were assembled with Gap4 [22] with a setting of 95% similarity in a stretch of 15 nucleotides. The number of one-member families, identifying 1,721 unique potential miRNA precursors, is given in the graph.

more with an annotated gene. All of these 117 potential miRNA precursors overlapped at least partly with an annotated intron and none was fully contained within an intron. Only 11 precursors (partly) overlapped with an exon sequence (data not shown).

To remove putative false positives from the potential precursor set and to begin to classify these precursors, we investigated the precursors that target genes in the *Arabidopsis* genome. Coding sequences and 250 nucleotides of the 5' and 3' untranslated region were extracted from the MIPS-annotated *Arabidopsis* genome sequence. Putative miRNA target genes were identified by gapped BLASTN analysis of the stem-loop sequence with low stringency parameter settings. This approach allows for gaps between miRNA and target sequence and is equivalent to previous attempts to predict plant miRNA targets [147]. Figure 5.2 presents two examples. Each gene with at least 80% similarity to at least 20 nucleotides of the palindromic stem-loop sequence of a potential miRNA precursor was classified as a putative target gene for the miRNA. The combined likelihood of a potential miRNA precursor with a putative target sequence in the *Arabidopsis* genome is taken to indicate potential biological relevance and regulatory function in that genome. In this way, no less than 12,729 genes were classified as putative miRNA target genes. Each potential miRNA precursor with at least one putative target gene was subsequently classified as a predicted miRNA candidate. Using this criterion 2,427 of the 2,693 potential miRNA precursors (90%) has one or more putative target genes

Table 5.4: Number of potential miRNA precursors with at least one putative target gene in the *Arabidopsis* genome for different selections from all annotated genes in the *Arabidopsis* genome.

Coding	**3' UTR**[a]	**5' UTR**	**80% similarity**[b]		**90% similarity**[c]	
+	-	-	2,056		516	
+	+	-	2,369	$\Delta = 313$[d]	1,201	$\Delta = 685$
+	-	+	2,114	$\Delta = 58$	692	$\Delta = 176$
+	+	+	2,427	$\Delta = 371$	1,377	$\Delta = 861$

[a]UTR: UnTranslated Region. [b]80% setting of gapped BLASTN. [c]90% setting of gapped BLASTN. [d]Δ increase in the number of predicted miRNA candidates relative to the coding sequence.

in the *Arabidopsis* genome (Table 5.4). When the target sequence is limited to only the coding region, the analysis results in 2,056 predicted miRNAs (Table 5.4). In 313 cases, the similarity is based on similarity to the 3' untranslated region (UTR) of a gene only. The 80% similarity criterion is chosen arbitrarily. When the similarity between the stem-loop sequence of the potential miRNA precursor and putative target gene is raised to 90% of at least 20 nucleotides, the number of predicted miRNA candidates reduces to 1,377 (Table 5.4). Assuming biological relevance of the set of potential miRNA precursors by the presence of a putative target gene, depending on the similarity score used, the number of predicted miRNA candidates is therefore well over one to two-thousand.

With the 80% similarity criterion, the total number of target genes putatively subject to miRNA regulation is no less than 12,729, close to half of all the currently annotated genes in the *Arabidopsis* genome. This is much higher than we anticipated. Given 2,427 predicted miRNA candidates and 12,729 putative target genes, multiple predicted miRNA candidates will target a given gene (Figure 5.5A), whereas also given miRNA candidates can target multiple genes (Figure 5.5B). There are only 64 predicted miRNA candidates that target a single gene in the *Arabidopsis* genome (Figure 5.5B), whereas 140 predicted miRNA candidates each target 7 different genes (Figure 5.5B). Vice versa, genes can be (putative) target for multiple, either related or independent predicted miRNA candidates. Whereas the majority of the putative target genes (7,019) are the target for a single miRNA, the remaining 5,710 genes are targets for more than one miRNA (Figure 5.5A). In an intriguing case, a single gene is the putative target for as many as 274 predicted miRNA

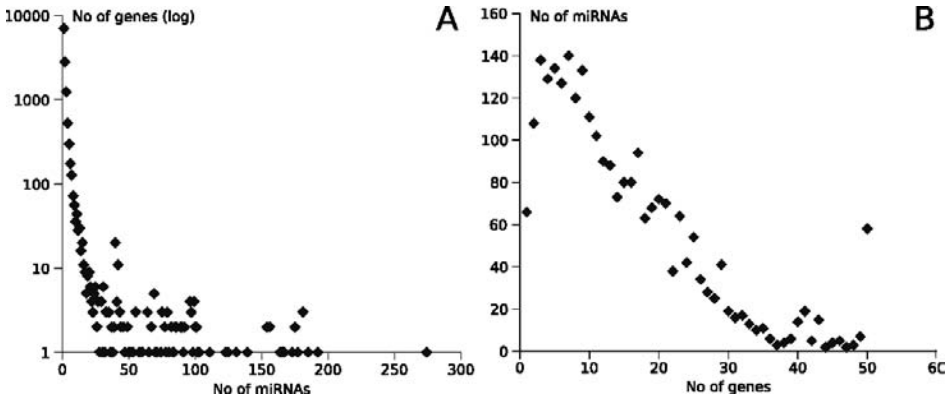

Figure 5.5: Relationships between predicted miRNA candidates and identified putative target genes. (A) The number of predicted miRNA candidates that target a given gene. The graph shows on the left that 7,019 genes are potential target for a single predicted miRNA candidate. Multiple miRNA candidates can also target a single gene. The graph shows as extreme a single gene that is targeted by 274 different predicted miRNA candidates (see also text). (B) The number of genes targeted by a single miRNA candidate. A predicted miRNA candidate can target multiple genes. The graph shows on the left that there are only 64 predicted miRNAs that target a single gene in the *Arabidopis* genome; on the right that 58 predicted miRNA candidates target 50 genes. The extreme is 140 predicted miRNA candidates each targeting seven different genes.

candidates. This gene is annotated as a putative non-LTR reverse transcriptase (At3g32130). Alignment of all the 274 predicted miRNA candidates targeting this gene reveals two hot-spots for potential interaction, one in the 3' UTR of the gene and one in the 5' part of the coding sequence (Figure 5.6).

Using the MIPS ontology for gene classification [52, 163], only 2,546 (20%) of the 12,729 putative target genes are classified, showing that the genes potentially regulated by miRNAs are for a major part in the functionally unknown categories. The classification of the known genes that are putative miRNA targets shows that basically all classes of genes in the MIPS ontology classification have members that may be target for a miRNA. When the numbers are compared to the classification of all annotated *Arabidopsis* genes, notably the class of genes encoding transposable elements, viral and plasmid proteins is more represented in the set of putative miRNA target genes here identified.

A requirement for any predicted miRNA candidate to be valid is the actual presence of a small RNA of 21-24 nt length in a total RNA fraction of *Arabidopsis*. Seven randomly chosen predicted miRNA candidates were taken for laboratory validation (Table 5.5): mi248, mi381, mi445, mi1022, mi1193,

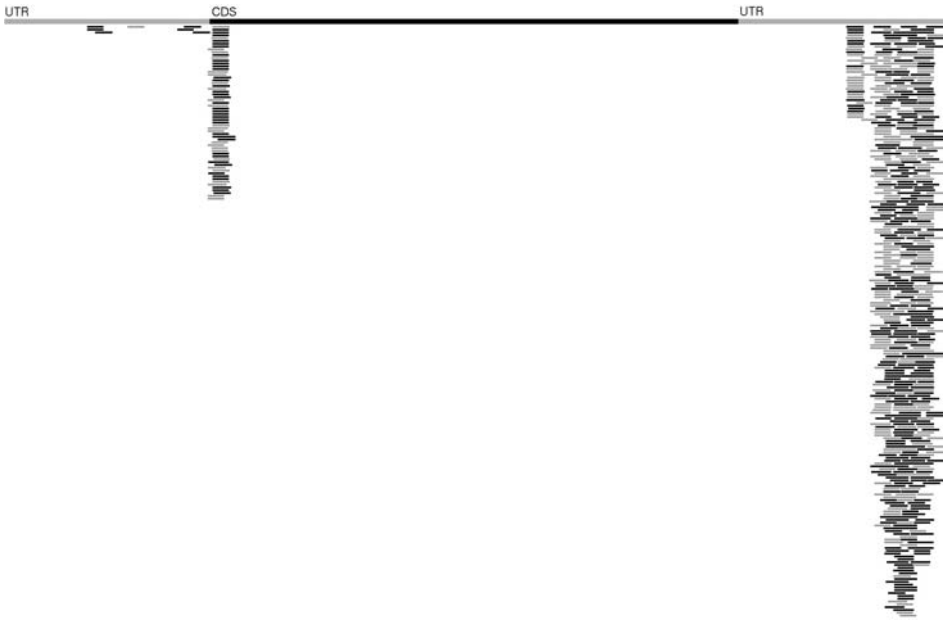

Figure 5.6: One gene can be target for multiple miRNA candidates. At3g32130 is a putative non-LTR reverse transcriptase gene that is target for 274 different predicted miRNA candidates. The thick black bar represents the coding sequence of the gene. The grey parts of the bar are the introns and untranslated regions. The potential positions of the predicted miRNA candidates in this target sequence are given as lines below the bar, grey depicts a predicted miRNA in forward orientation, black in reverse orientation. The predicted miRNA candidates target this gene most notably in the 3' untranslated region.

mi1460 and mi2512. Of these, mi1460 is among the 274 predicted miRNA candidates putatively targeting the non-LTR reverse transcriptase (At3g32130; Figure 5.6). The number of putative target genes for these predicted miRNA candidates ranges from 1 (mi381) to 21 (mi1193) (Table 5.5). Total *Arabidopsis* RNA from flowers was analyzed for the presence of small RNA hybridizing to the predicted miRNA candidate by the method described previously [81]. The analysis was first performed by hybridizing an RNA blot with a 250 bp PCR fragment containing the complete predicted miRNA precursor. If the PCR fragment gave a positive result, the prediction was considered validated. The results of these hybridizations are summarized in Table 5.5. In five out of seven cases (mi381, mi1022, mi1193, mi1460, mi2512), a small RNA of approximately 21 nt could be detected with a labelled PCR probe. The other two miRNAs (mi248, mi445) may be wrongly predicted, occur in tissues or cells not sampled, or occur in levels below detection limits. In three cases

Table 5.5: Characteristics of seven randomly selected predicted miRNA candidates. Summary of the number of putative target genes for each selected candidate, as well as results of all hybridization experiments. Probes are either PCR fragments or labeled oligonucleotides.

miRNA	**No target genes**	**Hybridization probes on small RNA blots**								
		PCR	**Oligonucleotide**							
			Forward				**Reverse**			
		Fl[a]	**Fl**	**Le**[b]	**St**[c]	**Si**[d]	**Fl**	**Le**	**St**	**Si**
mi248	20	-	NA[e]	NA	NA	NA	NA	NA	NA	NA
mi381	1	+	+	-	ND[f]	ND	-	-	ND	ND
mi445	9	-	NA	NA	NA	NA	NA	NA	NA	NA
mi1022	14	+	ND	ND	ND	ND	ND	ND	ND	ND
mi1193	21	+	-	-	-	-	+	+	+	++
mi1460	15	+	ND	ND	ND	ND	ND	ND	ND	ND
mi2512	17	+	+	+	+	ND	-	-	-	ND

[a]Fl: Flower,[b]Le: Leaf,[c]St: Stem,[d]Si: Silique,[e]NA: not applicable,[f]ND: not determined

(mi381, mi1193 and mi2512), a small RNA blot with different total RNA preparations (flowers, leaves, stems, and siliques) was hybridized with an end-labeled oligonucleotide representing either strand of the stem-loop structure of the predicted miRNA candidate. In all three cases, hybridization showed that only one of the two oligonucleotides of the stem-loop sequence could be detected in the small RNA fraction (Table 5.5). It is furthermore noteworthy that the expression patterns differ between the different miRNAs (Table 5.5). All three miRNA probes hybridize to small RNA in flowers. mi381 is not hybridizing to small RNA in leaves, whereas mi2512 is hybridizing to small RNA in leaves and stems. In addition, mi1193 is detecting small RNA in all four tissues analyzed, but the amount of small RNA is notably enhanced in siliques (Figure 5.7). The oligonucleotide specificity as well as the differences in tissue-specific presence of these predicted miRNA candidates both support the potential biological relevance of these predicted miRNA candidates. Five positives out of seven randomly chosen candidates from a total population of 2,427 predicted miRNA candidates indicate that a major fraction of the whole population of potential miRNA candidates may generate small RNAs. The 95% lower confidence limit of the expected number of predicted miRNA candidates can be calculated with the appropriate binomial statistics [91, 211]. This shows that there are at least 828 potentially active miRNA candidates in the *Arabidopsis* genome.

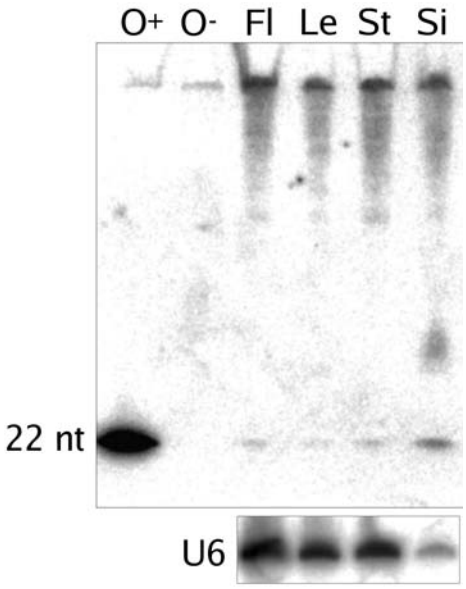

Figure 5.7: Tissue specific occurrence of mi1193 in *Arabidopsis* total RNA. (Top) Autoradiograph of an RNA blot hybridized with the end-labeled reverse oligonucleotide of the stem-loop sequence of predicted miRNA candidate mi1193. The lane labeled O+ contains 50ng of the forward oligonucleotide and serves as positive control for hybridization. The lane labeled O- contains 50ng of the reverse oligonucleotide used as a probe on this blot and is a negative control for (cross)hybridization. The lanes labeled Fl (flower), Le (leaf), and St (stem) contain approximately 80 μg of total RNA isolated from the tissues indicated. The lane labeled Si (silique) contains less RNA. (Bottom) Autoradiograph of the same RNA blot hybridized with the *Arabidopsis* snRNA U6. After the first hybridization the blot was re-hybridized without stripping with a probe encoding the *Arabidopsis* small nuclear RNA U6 to confirm the actual amount of RNA applied to the gel. The result shows that silique RNA was underrepresented on this blot, which is in agreement with our laboratory observations that RNA from siliques is more difficult to isolate and comes out less clean. The software package TINA was used to calculate the ratio of the hybridizing signal by comparing the mi1193 hybridization with the U6 hybridization. For the four tissues represented on this RNA blot, the mi1193/U6 ratios are: flowers 0.01, leaves and stems 0.02 and siliques 0.22. Relative to U6 RNA, silique total RNA thus shows a tenfold increase in amount of the small RNA hybridizing to the mi1193 miRNA candidate compared to the other three tissues.

5.4 Discussion

A whole-genome analysis for the presence of potential miRNA precursors based on the distinguishing structural features of miRNA-generating DNA indicates the presence of 2,693 potential miRNA precursors. By chance one would expect only 80 candidates (see results), showing that the *Arabidopsis* genome contains considerably higher numbers of potential miRNA precursors than ex-

pected on the basis of chance alone. This may indicate the high functional importance of the class of potential miRNA precursors. The approach here taken for identification of potential miRNA precursors differs from previous analyses in setup and software used for identifying palindromic repeats in the *Arabidopsis* genome. Additional assumptions based on conservation of miRNAs over multiple genomes [93, 99, 100] and miRNA/mRNA pairing restrictions [178] could be useful in further classifying and ranking of the predicted miRNA precursors. However such assumptions are based on a relatively small set of known miRNAs and may cause an underestimation of the number of miRNAs in *Arabidopsis*.

To single out false positives, the presence of a putative target gene in the same genome was taken as additional criterion. The identification of putative target genes by low stringency gapped BLASTN with a small segment high scoring pair (HSP) of minimal 20 nucleotides is difficult, due to the high chance that a miRNA targets a gene [147, 178]. This does however not mean that the candidates upon transcription are not biologically relevant [82]. The combined presence of a potential miRNA precursor and putative target gene thus increases credibility to the miRNA predictions, in a way similar to comparative genomics approaches [64, 100]. In total, 12,729 genes may be target for miRNA regulation. This is a major part of the currently known genes in the *Arabidopsis* genome. Only 20% of the potential target genes now identified are classified in the ontology used [52]. Therefore, the genes potentially regulated by miRNAs reside for a major part in the functionally unknown genes. Given that putative target genes distribute evenly over all functional categories, miRNA-mediated regulation may be broader than previously suggested. It will be highly interesting to see what future functional gene analyses will reveal for the function of these genes and their regulation.

Single miRNAs may target different genes, and, vice versa, a single gene may be targeted by different, not necessarily sequence related, miRNAs. The extreme case identified is a putative reverse transcriptase that may be target for 274 different miRNAs of which one was shown to exist *in vivo*. Such a transcriptase is likely to be associated with transposon activity. It seems reasonable to assume that the more miRNAs target a given gene, the lower the likelihood that the gene escapes miRNA mediated (down) regulation. The

large number of predicted miRNA might therefore indicate that the *Arabidopsis* genome exerts a great effort for down regulating this reverse transcriptase.

The combined criteria of palindromic sequence and putative target gene results in the presence of 2,427 predicted miRNA candidates in the *Arabidopsis* genome. This set does not contain micro-satellites or abundantly repetitive palindromic elements. In this population families of apparently related predicted miRNA candidates can be identified. The *Arabidopsis* genome is characterized by no less than 1,928 families of distinct predicted miRNA candidates. The analysis indicates that also the chloroplast genome contains predicted miRNA candidates (Table 5.2). The presence of miRNA candidates in the chloroplast genome is also indicated in other laboratory experiments [110].

Our *in silico* prediction of large numbers of miRNA candidates was further validated in the laboratory by small-scale sampling and detection of the predicted small RNA in a total RNA fraction of *Arabidopsis*. Five out of seven randomly chosen potential miRNA candidates hybridize to small RNA in the small, 21-24 nucleotide fraction of flower RNA (see Table 5.5 and Figure 5.7). This result does not necessarily disqualify the two negative cases: these may exist in concentrations below detection levels and/or in tissues/cells not sampled such as the root or the highly specialized microspore mother cell. In the three cases fully evaluated, only one of the two strands forming the hairpin is hybridizing to small RNA in the total RNA fraction, as is observed in other cases [80]. Moreover, the small RNA molecules detected by hybridization shows tissue specific expression. One of the mi1193 oligonucleotides hybridizes to small RNA in all tissue samples analyzed, but the hybridization in siliques relative to U6 RNA is markedly stronger (Figure 5.7). The random set selected out of a much larger set of predicted miRNA candidates shows all the signs of genuine and possibly regulatory miRNAs, although determination of a precise biological function of these miRNAs requires more analyses. In the framework of a genome-wide inventory of the miRNA potential of the *Arabidopsis* genome, the five randomly chosen candidates in a sample of seven from a population of 2,427 are likely to represent active miRNAs. This sampling result indicates the presence of an average of 1,733 miRNA candidates in *Arabidopsis*. The lower (95%) confidence limit of the number of miRNA

candidates in the *Arabidopsis* genome is 828. It is noteworthy to point out, however, that this may be a severe underestimation. The number of predicted miRNA candidates presented is limited by the size exclusion limit of 100 nt for the miRNA precursor molecule, and a relatively simple stem-loop structure. Both assumptions may be overly conservative in case of plants [103, 143]. Among our set of 2,427 predicted miRNA candidates, 19 miRNAs are also in miRBase (Table 5.3; Griffiths-Jones [66]). This shows that the criteria used in this paper are restrictive and the actual number of miRNAs could be much higher. Additional miRNA candidates have either a larger precursor structure or fewer constraints in the secondary structure of the miRNA precursor. The sequence characteristics for the identification of potential miRNA precursors used here is stricter. Therefore, the estimates given should be interpreted as a lower estimate of the true miRNA potential of the *Arabidopsis* genome.

Yet, even the lower estimates given suggest a much higher miRNA potential than concluded in previous studies. The upper limit of the number of miRNAs in the *Arabidopsis* genome [1, 101] was suggested to be 600 at most. Our predictions do not rely on the assumption that miRNAs should be conserved in evolution. Species-specific mRNAs shows that such conservation is an unnecessarily limiting assumption [1, 15, 101]. Moreover, much more of the genome is transcribed than indicated by the current genome annotations [71, 163]. Many small RNAs are identified in intergenic regions [103]. The MPSS *Arabidopsis* database identifies numerous tags (class 3&4) outside of the current annotations [115] . This supports the likelihood that many of the predicted miRNA precursors are transcribed and may have a biological function.

More research will be necessary to decide on the true miRNA potential of the *Arabidopsis* genome. Possibly co-evolution of miRNA and target gene may be as good as, if not a better, criterion for miRNA identification and classification. It may therefore be a valuable exercise to repeat the approach here presented for the human, rice or *Caenorhabditis elegans* genome. Such future studies may allow to investigate whether plants differ from human and/or invertebrates in the numbers and/or extent of miRNA-based gene regulation.

Chapter 6

In silico prediction of protein allergenicity using Allermatch

Mark Fiers, Gijs Kleter, Ad Peijnenburg, Herman Nijland, Jan Peter Nap and Roeland van Ham

A modified version is published in *BMC Bioinformatics*, 5:133, 2004.

Abstract

Allermatch™ `http://allermatch.org` is a novel webtool for the efficient and standardized prediction of potential allergenicity of proteins according to the current recommendations of the FAO/WHO Expert Consultation, as outlined in the Codex alimentarius. A query amino acid sequence is compared against the Allermatch Allergen database based on current SwissProt and WHO-IUIS allergen lists. The webtool uses a sliding window to identify stretches of 80 amino acids with more than 35% similarity, or identical small stretches of at least six amino acids. The outcome of the analyses is presented in a concise format. Allermatch is likely to contribute to improved, transparent and more consistent analyses of potential allergenicity of genetically modified food prior to market release. In the future, the FAO/WHO guidelines may be improved upon. Different methods that could enhance the predictive value of allergen prediction are discussed.

6.1 Introduction

The safety of genetically modified foods must be assessed before authorities in most nations will consider granting market approval. An important issue in the food safety assessment is the evaluation of the potential allergenicity of food derived from biotechnology. Food allergy is an immunogobulin E mediated response to food components and is part of a wider group of adverse reactions to food termed "food sensitivity". Food allergy may have symptoms that vary from itching, vomiting, diarrhea to life threatening anaphylaxis. As all known food allergens are proteins, the introduction of a new ("foreign") protein in food by genetic engineering can cause allergic reactions in a "worst case" scenario. Potential allergenicity of a protein is a complex issue and various tests are used to predict potential allergenicity, including bioinformatics, *in vitro* digestibility of the protein, and binding to antisera of allergic patients [46, 181] The FAO/WHO's Codex alimentarius and an Expert Consultation group have established guidelines to assess potential allergenicity of proteins with bioinformatics in a step-by-step procedure [45, 46]. Eventually, these guidelines will have to be incorporated into law by all FAO/WHO member states. The guidelines aim to assess whether a given primary protein sequence is sufficiently similar to sequences of known allergenic proteins to cause reason for concern. The recommended procedure to establish the potential for allergenicity is as follows [45]:

1. Obtain the amino acids sequences of known allergens in public protein databases in FASTA-format (using the amino acids from the mature proteins only, disregarding the leader sequences, if any are annotated)

2. Prepare a complete set of 80-amino acid length sequences derived from the expressed protein (again disregarding the leader sequence, if any).

3. Compare each of the sequences of (2) with all sequences of (1), using the program FASTA [130] with default settings for gap penalty and width.

According to the Codex alimentarius potential allergenicity should be considered [46], when there is

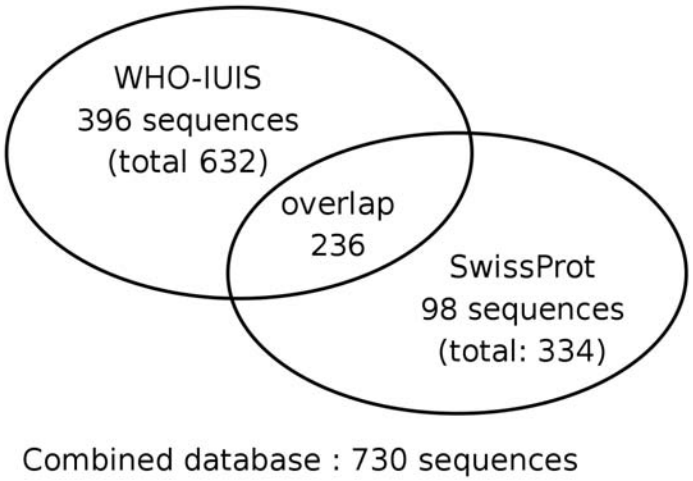

Figure 6.1: Venn diagram showing the relationship between the two databases used by Allermatch

a. More than 35% similarity over a window of 80 amino acids in the amino acid sequence of the query protein (without the leader sequence, if any) with an entry known as allergen or

b. A stretch of identity of 6 to 8 contiguous amino acids.

If either analysis points to possible allergenicity, the allergenicity of the protein should be verified using serum-binding tests and/or *in vivo* methods such as patient panels, skin prick tests or animal exposure tests [181]

6.2 Features of the Allermatch webtool

The Allermatch webtool complies with the FAO/WHO criteria given above. The first step was to create databases for the analysis. These databases were established in three steps. First, a SwissProt allergen database was created by extracting all 334 proteins from SwissProt [20] annotated as an allergen [102, SwissProt version 44.1, July 5 2004]. Leader sequences were, if annotated, trimmed and the mature protein sequences were stored in the Allermatch SwissProt allergen database. Secondly, all 632 entries (excluding some duplicates) from the WHO-IUIS allergen list [85] were extracted from

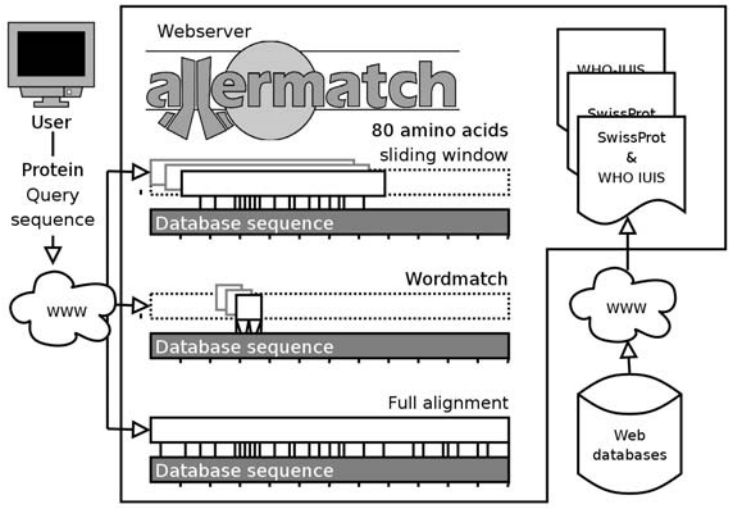

Figure 6.2: Schematic representation of the Allermatch^{TM}webtool.

the public databases SwissProt, PIR [203] and GenPept [58]. It should be noted that the WHO-IUIS list contains SwissProt sequences which are not on the SwissProt allergen list and that the SwissProt allergen list contains sequences which are not on the WHO-IUIS list. Annotated leader sequences were trimmed and the sequences were stored in the Allermatch WHO/IUIS allergen database. Joining the above databases and removing redundancies created the Allermatch combined allergen database. The combined SwissProt and WHO-IUIS allergen databases contained 236 duplicate sequences (Figure 6.1). The resulting non-redundant Allermatch allergen database contains 730 allergen sequences. The current version of the Allermatch webtool allows analysis of a given query protein with any of the three databases created but uses the combined database per default. In the future, it will be possible to upload local sequences to be used as database.

For protein sequence alignment, Allermatch uses the FASTA program [130] version 3.4t2 with default settings (ktup = 2, matrix = Blosum50, Gap open = -10, Gap extend = -2). All other software is written in Python and runs on a Suse Linux Enterprise Server 8 using mod_python and an Apache webserver (version 1.3.26). Allermatch provides three different search modes to assess and visualize the potential allergenicity of proteins (Figure 6.2). These three search modes are described in the next paragraphs.

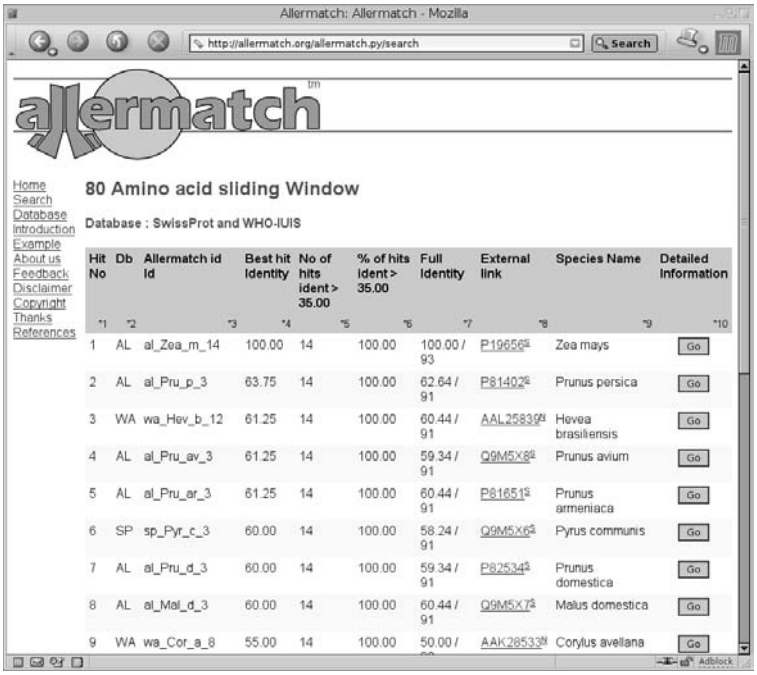

Figure 6.3: Screenshot of a results screen of an 80 amino acid alignment. This figure shows an overview of all matches found with the 80 amino acid sliding window method on an pollen allergen sequence from *Zea mays* (Zea m 14). The columns represent: 1) The number of the hit, sorted on column 4. 2) The database from which the sequences was derived. 3) The allergen identifier. 4) The best 80 amino acid similarity of all matched windows. 5 and 6) The number and percentage of windows with a similarity above 35%. 7) The percentage similarity and the number of similar amino-acids in a full alignment of the query sequence with this database allergen. 8) The SwissProt identifier and a link to the SwissProt website. 9) The species name from which the allergen sequence derives and 10) a link to a page with more details on this specific hit.

6.2.1 Mode 1, 80 amino acids sliding window

The query protein sequence is divided into windows of 80 amino acids using a sliding window with steps of a single amino acid. Each of these windows is compared with all sequences in the Allermatch allergen database. All database entries showing a similarity higher than a given threshold percentage (default is 35%; a user can adjust the threshold percentage if desired) to any of all 80 amino acids query sequence windows, are identified. Upon completion of the analysis, a table is generated that shows all database entries identified (Figure 6.3). For each database entry, the highest similarity score is given, as well as the number of 80 amino acids windows having a similarity above the

Figure 6.4: Screenshot of a detailed view of a single wordmatch analysis on the same sequence as used for figure 6.3. The image shows detailed information on the sequence from the Allermatch^{TM} allergen database matched. Below two alignments can be seen, the first alignment shows which parts of the input sequence have a 6 amino acid exact match with the database sequence (marked with #). The second alignment displays the same for the allergen database sequence.

cut-off percentage. For each entry identified, more detailed information can be retrieved on the similarity between the allergen and query sequence. For example, the areas of both proteins within all 80 amino acids windows that score above the threshold percentage.

If the similarity score calculated by FASTA applies to stretches smaller than 80 amino acids, Allermatch converts such a similarity score to an 80 amino acids window in a linear fashion. For example, a 40% similarity on a stretch of 40 amino acids, converts to 20% similarity in an 80 amino acids window.

This criterion implies that sequences shorter than 80 amino acids need to have higher similarity in order to be identified as a potential allergen.

6.2.2 Mode 2, Wordmatch

The second method looks for short sub-sequences (words) that have a perfect match with a database entry. The wordsize is configurable (default is 6 amino acids). The resulting output is similar to the output given by Mode 1. All database entries with at least one hit are listed and for each entry more detailed information can be retrieved upon request (Figure 6.4).

6.2.3 Mode 3, Full alignment

The Allermatch webtool offers the full alignment of the query sequence with the Allermatch allergen database entries. A FASTA alignment of the entire input protein allows one to obtain a global view of the query's protein similarities with a known allergen and may help to position regions of potential allergenicity in the primary structure of the protein. Upon parsing of the FASTA output, information from the Allermatch Allergen database is added and presented. This full alignment is not part of the recommendations of the FAO/WHO guidelines. It is added as an additional useful tool for further research.

6.3 Validation of the Allermatch webtool

A major issue in the prediction of potential allergenicity of a protein from a biosafety point of view is the likelihood of error. The algorithms used for prediction should be as accurate as possible and have as low an error rate as possible. One can identify two types of errors: a query protein that is identified as a potential allergen, while in fact it is not (i.e., a false positive) and a query protein that is excluded from the possibility of being an allergen, while in fact it is (i.e., a false negative). Both error types are estimated for each of the three databases evaluated. For the sliding window approach an 80 amino acid

Table 6.1: Prediction quality of the FAO/WHO methods. The number and percentage of false negative and false positive hits is shown here for all FAO/WHO recommended method/database combinations. Result set 1 describes the number of false negative hits observed in a leave-one-out method. The next result set (2) shows the same results but corrected for those sequences that were not able to generate a hit against itself due to the short length of the sequence. The last (3) result set shows the observed number of false positives when testing 12 non-allergenic sequences (see table 2) against the Allermatch^{TM}webtool. Each of the result sets consists of two columns; the first column shows the number of false hits and the total number of sequences in this set. The second column shows the percentage of false hits.

Database	Method	ws [1]	Result Set 1 False negatives		Result Set 2 False Negatives (corrected)		Result Set 3 False Positives	
			Number	%	Number	%	Number	%
SwissProt	Window	n.a.	71/334	21.3	57/320	17.8	3/12	25.0
	Wordmatch	6	54/334	16.2	n.a.	n.a.	7/12	58.3
		7	69/334	20.7	n.a.	n.a.	6/12	50.0
		8	78/334	23.4	n.a.	n.a.	3/12	25.0
WHO-IUIS	Window	n.a.	99/632	15.7	78/611	12.8	4/12	33.3
	Wordmatch	6	58/632	9.2	n.a.	n.a.	9/12	75.0
		7	98/632	15.5	n.a.	n.a.	8/12	66.7
		8	117/632	18.5	n.a.	n.a.	3/12	25.0
Both	Window	n.a.	101/730	13.8	77/706	10.9	5/12	41.7
	Wordmatch	6	55/730	7.5	n.a.	n.a.	9/12	75.0
		7	95/730	13.0	n.a.	n.a.	8/12	66.7
		8	115/730	15.8	n.a.	n.a.	3/12	25.0

window with a 35% similarity cutoff is used and for the wordmatch approach 6, 7 and 8 amino acids word sizes are tested.

6.3.1 Estimation of the error rate of false negatives

It is not easy to investigate the tion rate of false negatives by the algorithms employed in the Allermatch webtool, as there are no proteins known as allergen while they are not represented in the databases used (as there should be none). As an estimation we have determined the number of "orphan" entries in the Allermatch Allergen database. An orphan entry is an entry that, according to the Allermatch analysis, is not predicted to be an allergen by similarity to any other entry in the database except itself. Such an orphan entry would represent a false negative relative to that database if this sequence were not present in this database. This approach is also called "jackknifing". The number of orphan entries in a database is an approximation of the false negative rate. The results of performing this analysis on all three databases and using the first two analysis methods are summarized in Table 6.1.

In examining the false negative results, various sequences were observed that did not produce a hit against itself (data not shown). On closer inspection,

this was found to be due to the short length of these protein sequences. If a sequence is shorter than 28 amino acids, even 100% similarity will convert the similarity to less than 35% after conversion to an 80 amino acid window. This may overestimate the error rate. Therefore, we also determined the false negative rates with those sequences not able to generate a hit against itself excluded. Even after this correction the wordmatch method, with a 6 amino acids word length, gives a lower percentage of false negatives than the sliding window approach.

6.3.2 Estimation of the error rate of false positives

The second control examines 12 proven non-allergenic sequences against the Allermatch databases. Non-allergenicity can, for example, be based on non-reactivity of these proteins towards IgE-sera of allergy patients or the inability to cause IgE-responses in experimental animals (see Table 6.2). It should be noted that such data exists only for a limited number of proteins, which also accounts for the size of this dataset. A non-allergenic sequence is not supposed to generate a hit; therefore we consider each hit a false positive. Results are summarized in Table 6.1, result set 3.

6.4 Discussion

Prediction of allergenicity can broadly be done in two ways: one can look for linear or conformational epitopes [24]. The first method tries to assess whether two proteins share similarities in the primary sequence, whereas the second method looks at similarities in 3D structure. The Codex guidelines recommend a combination of both approaches. Short exact word-matches and positive hits in the sliding 80-amino acid window may indicate potential linear epitopes and similar 3D structures, respectively.

Examination of the false negative rate (see table 6.1, result set 1) shows a link between the database size and the false negative hit rate. This is to be expected because a larger database increases the probability that a similar allergen present. Another possible part of the explanation is that a larger database is likely to have more isoallergen families (a group of allergens

Table 6.2: Sequences used for the negative control.

Protein	Host organism	Evidence for non-allergenicity	Accession	Reference
Amaranth seed albumin	*Amaranthus hypochondriacus*	IgG-response, but no raised IgE-levels, after administration (intranasal and intraperitoneal) of amaranth seed albumin to mice	GenPept CAA77664	[30]
T1	*Catharanthus roseus*	No reaction of recombinant T1 in IgE-sera binding, basophile histamine release, and skin prick testing using patients allergic to the related birch pollen allergen Bet v 1	Not applicable	[92]
Mite ferritin heavy chain	*Dermatophagoides pteronyssinus*	Reaction of mite ferritin with IgG, but not with IgE, of sera from patients allergic to house dust mite	GenPept AAG02250	[44]
Maltose binding protein	*Escherichia coli*	No reaction with IgE-sera from patients allergic to natural rubber latex (maltose binding protein used as part of fusion proteins with latex allergens)	SwissProt P02928	[149]
Human serum albumin	*Homo sapiens*	No reaction of human serum albumin with IgE-sera of patients allergic to cat- and porcine-serum albumin	SwissProt P02768	[75]
Human heat shock protein 70	*Homo sapiens*	No reaction of human heat shock protein 70 with IgE-sera of patients allergic to heat shock protein 70 from Echinococcus granulosus	SwissProt P08107	[126]
Human beta-2-glycoprotein I	*Homo sapiens*	Presence of IgM antibodies, but not of IgE antibodies, directed against human beta-2-glycoprotein I in sera from atopic eczema/dermatitis patients	SwissProt P02749	[185]
Guayule rubber particle protein	*Parthenium argentatum*	No cross-reactivity between proteins from guayule and latex using IgE-sera from patients allergic to latex	SwissProt Q40778	[168]
Purle acid phosphatase 1	*Solanum tuberosum*	Stimulation of IgG-, but no or only low stimulation of IgE-antibodies following administration of potato acid phosphatase to mice (oral and intraperitoneal)	TrEMBL Q6J5M7	[35]
Purle acid phosphatase 2	*Solanum tuberosum*	See above	TrEMBL Q6J5M9	[35]
Purle acid phosphatase 3	*Solanum tuberosum*	See above	TrEMBL Q6J5M8	[35]
Potato lectin	*Solanum tuberosum*	Stimulation of IgG-, but no or only low stimulation of IgE-antibodies following administration of potato lectin to mice (intraperitoneal)	TrEMBL Q9S8M0	[36]

with minor sequence differences) present. This diminishes the chance of false negatives since fewer sequences will be an "orphan". A third factor influencing the false negative hit rate is bad protein annotation. Signal peptides still present in the database might generated a positive hit for an orphan protein.

When evaluating the false positive hits we see a similar trend; the number of false positives grows with the database size, as is to also be expected since the chance of a random hit increases with a larger database. In contrast to the false negative hit rates however, the sliding window method gives a lower percentage of erroneous hits here. The results of this test might overestimate the number of false positives, since a number of these non-allergens are related to and display similarities with their allergenic counterparts, i.e. T1 is related to Bet v 1 [92], human serum albumin is related to animal serum albumins [75] and human heat shock protein 70 is similar to heat shock proteins from

fungi and other allergens [126] (Table 6.2). A true selection of unrelated, non-allergenic proteins is therefore likely to give a lower false positive rate.

These results show that by choosing a database and algorithm one can influence the error rates towards either a higher rate of false positives or towards more false negatives. A too high detection rate of false positives would generate an unnecessary and undesirable burden of additional testing of proteins used in genetic engineering. On the other hand, a too high detection rate of false negatives would generate undesirable potential health risks for consumers. Either error is undesirable, but because this bioinformatics analysis identifies proteins for further testing of true allergenicity, a "better safe than sorry" strategy could be opted for. Such a strategy would obviously strive to minimize the detection rate of false negatives by using the results of both the sliding window and the six amino acid wordmatch against the largest Allermatch combined allergen database. Positive results from these analyses should first be analyzed in depth by checking medical literature on these proteins. Ultimately all valid predictions will, as suggested by FAO/WHO, have to be tested further with methods as skin prick tests or animal models. Even after these tests there is no absolute certainty that the protein in question will never elicit an allergenic reaction. In time, people might still become sensitized to the protein as a novel allergen, or only a very small part of the population is sensitive to cross-reacting allergens, too small to have been noticed in the tests.

In general, one should keep in mind that performance of these algorithms is far from perfect. This is in agreement with other literature where similar results for the FAO/WHO methods are shown and other algorithms proven to give better results KP02,SZG+04,ZGH02. These supplementary methods include, for example, advanced motif discovery methods where a complete allergen database is scanned for highly represented motifs. These motifs are then used to identify possible allergenicity [177]. In addition, a machine-learning approach was described using FASTA and a neural network to compare query proteins with allergens [213].

In the public domain, several other websites have emerged that assess potential allergenicity of proteins based on their primary sequence. For example:

- SDAP: `http://fermi.utmb.edu/SDAP/`

- FARRP: `www.allergenonline.com`
- AllerPredict: `research.i2r.a-star.edu.sg/Templar/DB/Allergen/`

These websites are also able to perform complete FASTA alignments (SDAP, Farrp), 80 amino acids sliding window (SDAP, AllerPredict) and 6 to 8 amino acids exact matches (SDAP, AllerPredict).

Allermatch will greatly enhance and improve the prediction of allergenicity according to current guidelines in the Codex by combining all recommended algorithms in a single website. In the future, the Allermatch webtool will stay updated with the public allergen databases on a regular basis and the requirements by law on assessing allergenicity. To increase the predictive power, supplementary bioinformatics facilities will be added. Such additional facilities may include, among others, the possibility to do batch analyses, to upload users' own databases, and to use supplementary tools such as the examples described above. Feedback from users will help us to identify particular issues that address their needs.

Chapter 7

General discussion

This thesis describes a broad scope of topics with respect to the application of bioinformatics in genome annotation. Nowadays, for only a relatively small part of genomes known, appropriate biological functions have been assigned to genomic sequences. In future research, the role of bioinformatics in guiding biological interpretation and integration of existing data will have to grow to reach the ultimate goal of genome annotation: to describe the function of every nucleotide in every cell, tissue and stage of development, under every condition during reproduction and entire life span of the organism in question.

Current bioinformatics involves large scale computational integration (Chapter 3) and intuitive data visualization (Chapter 4) to achieve high quality predictions of individual features in genomes, such as microRNAs (Chapter 5) and allergenicity of proteins (Chapter 6). Yet, the future challenges of all bioinformatics approaches in genome annotation undertaken in this thesis are related to two issues most researchers will not immediately associate with bioinformatics: communication and the assessment of quality.

Communication in bioinformatics needs to be improved on at least three levels: communication between computers, communication between computer and researcher and communication between researchers. The continued specialization of biological research motivates the continued development of specific databases where specialists remain in control of data and updated annotations. Such databases must be able to communicate. A major challenge in the development of an automated system for genome annotation (Chapter 3)

has been the communication between separate steps of genome annotation. The use of BioMOBY [200, chapter 2.4.2] as a standard allows uniform description of data for communication. However, BioMOBY allows easy creation of data types and services, with as result a labyrinth of (partly) overlapping data types and services. Therefore, future developments should aim at either unifying or interlinking these data types and services. The first option requires an ontology based approach that forces BioMOBY users to adhere to a predefined set of objects and service types. The second approach is probably easier to achieve and ensures a greater flexibility of the system. An example of the latter approach is the promising semantic MOBY project [162].

Improvement of the communication between computer and researcher has much to do with intuitive use and attractive presentation. Biologists can improve the interpretation of biological data by increasing their knowledge of computer assisted data analysis. A dot-plot [173], for example, takes some training to understand, but is, once mastered, a very powerful tool to visualize and explore the relationship between two sequences. Bioinformaticians must continue to develop novel ways of data presentation that facilitate analysis, interpretation and mining of genome data (Chapter 4). Visualization is one of the most powerful methods to allow easy access to large and inherently complex datasets. In communication with visualization scientists to develop the tool described in Chapter 4, it was surprising to discover that what was new and exciting for biologists and bioinformaticians, was considered almost outdated by visualization scientists. This implies that new visualization approaches may further appeal to the desires and needs of genome annotation. The more multi-dimensional and more comparative genome data will become, the more need there will be for advanced and interactive visualization. A crucial factor in achieving such developments will be the communication between biologist, bioinformatician and visualization scientist. Inital development should aim on an application able to display data produced by all current omics technologies and their mutual relationships. Such an application should be extensible, easy to use and run on a researchers desktop. The latter criterium is important as it lowers the treshold for a researcher to "play" with the data and hence, start to understand it.

For the further development of the bioinformatics approaches as investigated in this thesis, improved communication between researchers in the various disciplines should be considered the most important of all. Genome annotation is about giving biological relevance to the biological molecule DNA. Close cooperation with experimental biologists is required to give useful and applicable results that can be translated in new experiments and improved annotations.

Yet another topic of communication that is currently underestimated is the second issue identified during the preparation of this thesis: communication of the quality of data and the consequences thereof. Experimental biologists often complain that the annotations delivered by bioinformatics are not very precise (or plain wrong), because their laboratory experiments do not comply with the results predicted from annotations. Bioinformatics would be better off when admitting (and warning) clearly that this is unavoidable, due to various reasons, among which the presence of biological variability will appeal best to the laboratory researcher. All stakeholders involved need to become more aware of the methods and limitations of the methods that bioinformatics is using and developing for annotation.

Good annotation is no easy task. Whatever algorithm is chosen, it is an approximation that has assumptions and requirements that may not be met in all cases considered. For example, the annotation of protein coding genes is still far from satisfactory [26, 108, Chapter 2] although it has been a subject of research for well over two decades [see for example 47]. Therefore, it is generally considered a valid approach to use algorithms based on different computational principles and consider predictions that different algorithms share as the better predictions.

Biological variation is a major issue in the accuracy of predictions. Even within a species sequence variation is considerable. The hapmap consortium has shown that two individual human beings are likely to differ no less than 1 nucleotide per 279 [5], implying a much larger source of variation in the human genome than previously thought.

Most methods in bioinformatics depend on "learning algorithms" where known annotations are used to train the prediction algorithm. Such an algorithm is usually trained with data from one individual and is therefore biased towards that individual. This is generally unknown to the experimental biologist. It

should be explained better that all learning depends on input and when supplied with biased or unrepresentative input data, output will be equally biased. More attention for the accuracy of computational predictions is warranted in annotation and communication on annotation.

Equally essential for a good annotation is good data. An algorithm cannot be better than the data it gets. Unfortunately, there are many sources of error conceivable in the data used for annotation. Older sequences may be less reliable than sequences determined with later-generation equipment and approaches. Assembly may also give rise to errors, notably in the case of highly repeated sequences. New approaches are still developed to improve this [136]. The overall error rate in genome sequences may approach 0.01% (1 nucleotide per 10 thousand), but this is an average that may fail on individual areas of sequence. A more hidden source of error in annotation is the process of annotation itself. As outlined in Chapter 2, many methods in genome annotation rely on previous annotations. Gene function is usually assigned based on similarity with known genes, a notoriously difficult task [151]. It is not uncommon that a gene is assigned a function based on similarity with a gene that also was assigned a function based on similarity. This way, the process of annotation is recursive and prone to error propagation [60]. Such error propagation is a serious and underestimated issue in current biological databases.

Several initiatives are put in place to try to prevent error propagation in automated annotation. An example of such precaution is the use of highly curated databases, such as SwissProt [20]. Manual curation of data is a good way to reduce errors in annotation, but it is intrinsically slow and expensive. As a result, the contents of curated databases are lagging behind considerably. Also, curated databases are inherently conservative and may refuse and therefore miss new developments for quite a while. A lot of biological insight and innovation has come from exceptions to rules [137] that were only accepted hesitantly, such as for example gene silencing [137] and microRNA genes (Chapter 4). In general, a combination of both curated and not-curated databases is recommended to have access to the latest information and data for an annotation, provided the quality of the annotation is given as well. More attention

should be given to systems and methods that allow assessing the quality of an annotation in an easy and intuitive way.

The Gene Ontology consortium [8] employs the system of so called "evidence codes" to indicate how an annotation is obtained, This system allows clear distinction between, for example, an annotation based on laboratory evidence as opposed to a BLAST analysis. This system is a good beginning and deserves to be extended beyond the assignment of protein function. A uniform scoring system to allow quick and intuitive insight in the quality of an annotation will be essential for future bioinformatics and laboratory biology. Such a system should be able to handle recursive annotation. A BLAST hit against a gene given a function based on laboratory data should score better than a BLAST hit against a gene given a function on the basis of a previous BLAST analysis. Obviously, also laboratory data can be erroneous and possibly also the quality and number of independent confirmations should be included in the assessment. A Bayesian approach that implements prior knowledge in assigning confidence appears well suited to this task and will generate a quality score that can be easily interpreted and used in database searching.

It is anticipated that the issues and needs identified above will become more important in the near future. Data production speed will continue to go up, while costs per nucleotide will continue to go down. Current developments in sequencing technology [31, 109, 166] bring the $1000 genome in sight. With the rise of sequence data of individuals, the importance of comparative genomics in assessing differences between genomes and assigning biological function will grow tremendously. As already indicated by the human hapmap project [5], this is likely to show much more variability than assumed. Future genome annotation should be able to contribute to and incorporate such variation. It is also becoming increasingly clear that next to the DNA code there exist an informative epigenetic code [65]. This epi-code, consisting of methylation and histone modifications, is influencing expression and regulation of expression. and should obviously be incorporated in future annotations. Moreover, biological phenomena are triggered and regulated by large complexes. Such units consist of multiple proteins, protein-DNA complexes, protein-RNA complexes and/or complexes with non-protein metabolites. Unraveling the resulting "interactome" presents on of the next challenges of genome annotation.

Ultimately, understanding of the genome helps in understanding the function of a cell and an organism. This is the aim of the emerging field of systems biology of which genome annotation will be a pillar. Given all challenges and developments ahead, genome annotation will stay one of the most exciting fields of research for decades to come. In view of such future foreseen, it is unavoidable that this thesis will be outdated relatively soon.

Bibliography

[1] Adai *et al.* Computational prediction of miRNAs in Arabidopsis thaliana. *Genome Res*, 15(1):78–91, 2005.

[2] Agrawal *et al.* System, trends and perspectives of proteomics in dicot plants Part I: Technologies in proteome establishment. *J Chromatogr B Analyt Technol Biomed Life Sci*, 815(1-2):109–23, Feb 2005.

[3] Alexandersson *et al.* SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res*, 13(3): 496–502, 2003.

[4] Altschul *et al.* Basic local alignment search tool. *J Mol Biol*, 215(3): 403–10, 1990.

[5] Altshuler *et al.* A haplotype map of the human genome. *Nature*, 437 (7063):1299–320, 2005.

[6] Apache website. URL `http://apache.org/`.

[7] Apweiler *et al.* InterPro–an integrated documentation resource for protein families, domains and functional sites. *Bioinformatics*, 16(12): 1145–50, 2000.

[8] Ashburner *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–9, 2000.

[9] Bailey and Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol*, 2:28–36, 1994.

[10] Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, 2004.

[11] Batzoglou *et al.* Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res*, 10(7):950–8, 2000.

[12] Bendtsen *et al.* Improved prediction of signal peptides: SignalP 3.0. *J Mol Biol*, 340(4):783–95, 2004.

[13] Bennett *et al.* Toward the $1000 human genome. *Pharmacogenomics*, 6(4):373–82, 2005.

[14] Benson. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res*, 27(2):573–80, 1999.

[15] Bentwich *et al.* Identification of hundreds of conserved and nonconserved human microRNAs. *Nat Genet*, 37(7):766–70, 2005.

[16] Beowulf website. URL `http://www.beowulf.org/`.

[17] Besemer and Borodovsky. GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Res*, 33(Web Server issue):W451–4, 2005.

[18] Birney and Durbin. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Proc Int Conf Intell Syst Mol Biol*, 5:56–64, 1997.

[19] Birney and Durbin. Using GeneWise in the Drosophila annotation experiment. *Genome Res*, 10(4):547–8, 2000.

[20] Boeckmann *et al.* The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res*, 31(1):365–70, 2003.

[21] Boffelli *et al.* Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science*, 299(5611):1391–4, 2003.

[22] Bonfield *et al.* A new DNA sequence assembly program. *Nucleic Acids Res*, 23(24):4992–9, 1995.

[23] Bowers *et al.* Unravelling angiosperm genome evolution by phylogenetic analysis of chromosomal duplication events. *Nature*, 422 (6930):433–8, 2003.

[24] Bredehorst and David. What establishes a protein as an allergen? *J Chromatogr B Biomed Sci Appl*, 756(1-2):33–40, 2001.

[25] Brejova *et al.* ExonHunter: a comprehensive approach to gene finding. *Bioinformatics*, 21 Suppl 1:i57–65, 2005.

[26] Brent. Genome annotation past, present, and future: how to define an ORF at each locus. *Genome Res*, 15(12):1777–86, 2005.

[27] Brown *et al.* Plant snoRNAs: functional evolution and new modes of gene expression. *Trends Plant Sci*, 8(1):42–9, 2003.

[28] Burge and Karlin. Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94, 1997.

[29] Carver *et al.* ACT: the Artemis comparison tool. *Bioinformatics*, 21 (16):3422–3, 2005.

[30] Chakraborty *et al.* Increased nutritive value of transgenic potato by expressing a nonallergenic seed albumin gene from Amaranthus hypochondriacus. *Proc Natl Acad Sci U S A*, 97(7):3724–9, 2000.

[31] Chan. Advances in sequencing technology. *Mutat Res*, 573(1-2): 13–40, 2005.

[32] Chervitz *et al.* Comparison of the complete protein sets of worm and yeast: orthology and divergence. *Science*, 282(5396):2022–8, 1998.

[33] Condor website. URL `http://www.cs.wisc.edu/condor/`.

[34] Curwen *et al.* The Ensembl automatic gene annotation system. *Genome Res*, 14(5):942–50, 2004.

[35] Dearman and Kimber. Determination of protein allergenicity: studies in mice. *Toxicol Lett*, 120(1-3):181–6, 2001.

[36] Dearman *et al.* Evaluation of protein allergenic potential in mice: dose-response analyses. *Clin Exp Allergy*, 33(11):1586–94, 2003.

[37] Doench *et al.* siRNAs can function as miRNAs. *Genes Dev*, 17(4): 438–42, 2003.

[38] Du and Zamore. microPrimer: the biogenesis and function of microRNA. *Development*, 132(21):4645–52, 2005.

[39] Durbin *et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[40] Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–63, 1998.

[41] Eddy. What is a hidden Markov model? *Nat Biotechnol*, 22(10): 1315–6, 2004.

[42] Emanuelsson *et al.* Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J Mol Biol*, 300(4):1005–16, 2000.

[43] Enright *et al.* Protein families and TRIBES in genome sequence space. *Nucleic Acids Res*, 31(15):4632–8, Aug 2003.

[44] Epton *et al.* Non-allergenic antigen in allergic sensitization: responses to the mite ferritin heavy chain antigen by allergic and non-allergic subjects. *Clin Exp Allergy*, 32(9):1341–7, 2002.

[45] FAO/WHO. Allergenicity of Genetically Modified Foods. Technical report, FAO/WHO, Rome, Italy, 2001. http://www.who.int/foodsafety/publications/biotech/en/ec_jan2001.pdf.

[46] FAO/WHO. Codex Principles and Guidelines on Foods Derived from Biotechnology. Technical report, Joint FAO/WHO Food Standards Programme, Rome, Italy, 2003. ftp://ftp.fao.org/codex/standard/en/CodexTextsBiotechFoods.pdf.

[47] Fickett. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res*, 10(17):5303–18, 1982.

[48] Filipowicz. RNAi: the nuts and bolts of the RISC machine. *Cell*, 122 (1):17–20, 2005.

[49] Fleischmann *et al.* Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269(5223):496–512, 1995.

[50] Florea *et al.* A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res*, 8(9):967–74, Sep 1998.

[51] Foissac *et al.* EUGENE'HOM: A generic similarity-based gene finder using multiple homologous sequences. *Nucleic Acids Res*, 31(13): 3742–5, 2003.

[52] Frishman *et al.* Functional and structural genomics using PEDANT. *Bioinformatics*, 17(1):44–57, 2001.

[53] Furnas and Bederson. Space-scale diagrams: understanding multiscale interfaces. In *Proc. of chi-95*, pages 234–41, Denver, CO, USA, 1995.

[54] Garcia Castro *et al.* Workflows in bioinformatics: meta-analysis and prototype implementation of a workflow generator. *BMC Bioinformatics*, 6:87, 2005.

[55] Gavin *et al.* Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–7, 2002.

[56] Gelfand *et al.* Gene recognition via spliced sequence alignment. *Proc Natl Acad Sci U S A*, 93(17):9061–6, 1996.

[57] Generic Feature Format website. URL `http://www.sanger.ac.uk/Software/formats/GFF/`.

[58] Genpept website. URL `http://www.ncbi.nlm.nih.gov`.

[59] Gibbs and McIntyre. The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur J Biochem*, 16 (1):1–11, 1970.

[60] Gilks *et al.* Modeling the percolation of annotation errors in a database of protein sequences. *Bioinformatics*, 18(12):1641–1649, Dec 2002.

[61] Globus toolkit website. URL `http://www.globus.org/`.

[62] Genomes online database (gold) website. URL `http://genomesonline.org`.

[63] Gouret *et al.* FIGENIX: intelligent automation of genomic annotation: expertise integration in a new software platform. *BMC Bioinformatics*, 6:198, Aug 2005.

[64] Grad *et al.* Computational and experimental identification of C. elegans microRNAs. *Mol Cell*, 11(5):1253–63, 2003.

[65] Grant-Downton and Dickinson. Epigenetics and its implications for plant biology. 1. The epigenetic network in plants. *Ann Bot (Lond)*, 96 (7):1143–64, Dec 2005.

[66] Griffiths-Jones. The microRNA Registry. *Nucleic Acids Res*, 32 (Database issue):D109–11, 2004.

[67] Griffiths-Jones *et al.* Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*, 33(Database issue):D121–4, 2005.

[68] Guig *et al.* Prediction of gene structure. *J Mol Biol*, 226(1):141–57, Jul 1992.

[69] Gumucio *et al.* Phylogenetic footprinting reveals a nuclear protein which binds to silencer sequences in the human gamma and epsilon globin genes. *Mol Cell Biol*, 12(11):4919–29, 1992.

[70] Gutirrez *et al.* Phylogenetic profiling of the Arabidopsis thaliana proteome: what proteins distinguish plants from other organisms? *Genome Biol*, 5(8):R53, 2004.

[71] Haas *et al.* Complete reannotation of the Arabidopsis genome: methods, tools, protocols and the final release. *BMC Biol*, 3:7, 2005.

[72] Hall. Plant metabolomics: from holistic hope, to hype, to hot topic. *New Phytol*, 169(3):453–68, 2006.

[73] Hamilton and Baulcombe. A species of small antisense RNA in posttranscriptional gene silencing in plants. *Science*, 286(5441):950–2, 1999.

[74] Hebsgaard *et al.* Splice site prediction in Arabidopsis thaliana pre-mRNA by combining local and global sequence information. *Nucleic Acids Res*, 24(17):3439–52, 1996.

[75] Hilger *et al.* Allergic cross-reactions between cat and pig serum albumin. Study at the protein and DNA levels. *Allergy*, 52(2):179–87, 1997.

[76] HMMer website. URL `http://hmmer.wustl.edu`.

[77] Hofacker *et al.* Fast folding and comparison of rna secondary structures. *Monatshefte f. Chemie*, 125:167–88, 1994.

[78] Hubbard *et al.* Ensembl 2005. *Nucleic Acids Res*, 33(Database issue): D447–53, 2005.

[79] Hubbard *et al.* The Ensembl genome database project. *Nucleic Acids Res*, 30(1):38–41, 2002.

[80] Hutvagner *et al.* A cellular function for the RNA-interference enzyme Dicer in the maturation of the let-7 small temporal RNA. *Science*, 293 (5531):834–8, 2001.

[81] Hutvagner *et al.* Detailed characterization of the posttranscriptional gene-silencing-related small RNA in a GUS gene-silenced tobacco. *RNA*, 6(10):1445–54, 2000.

[82] Jackson *et al.* Expression profiling reveals off-target gene regulation by RNAi. *Nat Biotechnol*, 21(6):635–7, 2003.

[83] Kerkhoven *et al.* Visualization for genomics: the Microbial Genome Viewer. *Bioinformatics*, 20(11):1812–4, 2004.

[84] Kim. MicroRNA biogenesis: coordinated cropping and dicing. *Nat Rev Mol Cell Biol*, 6(5):376–85, 2005.

[85] King *et al.* Allergen nomenclature. WHO/IUIS Allergen Nomenclature Subcommittee. *Int Arch Allergy Immunol*, 105(3):224–33, 1994.

[86] Kling. Ultrafast DNA sequencing. *Nat Biotechnol*, 21(12):1425–7, Dec 2003.

[87] Knudsen. Promoter2.0: for the recognition of PolII promoter sequences. *Bioinformatics*, 15(5):356–61, 1999.

[88] Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 5:59, 2004.

[89] Korf *et al.* Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17 Suppl 1:S140–8, 2001.

[90] Kurtz *et al.* REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res*, 29(22):4633–42, 2001.

[91] Kurtz and Myers. Estimating the probability of Approximate Matches. In *Proc. of the Annual Symposium on Combinatorial Pattern Matching*, volume 8, pages 52–64, 1997.

[92] Laffer *et al.* Molecular characterization of recombinant T1, a non-allergenic periwinkle (Catharanthus roseus) protein, with sequence similarity to the Bet v 1 plant allergen family. *Biochem J*, 373(Pt 1): 261–9, 2003.

[93] Lai *et al.* Computational identification of Drosophila microRNA genes. *Genome Biol*, 4(7):R42, 2003.

[94] Lander *et al.* Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[95] Lawrence *et al.* Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–14, 1993.

[96] Lee *et al.* The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75(5):843–54, 1993.

[97] Lenhard *et al.* Identification of conserved regulatory elements by comparative genome analysis. *J Biol*, 2(2):13, 2003.

[98] Lewis *et al.* Apollo: a sequence annotation editor. *Genome Biol*, 3 (12):RESEARCH0082.1–14, 2002.

[99] Lim *et al.* Vertebrate microRNA genes. *Science*, 299(5612):1540, 2003.

[100] Lim *et al.* The microRNAs of Caenorhabditis elegans. *Genes Dev*, 17 (8):991–1008, 2003.

[101] Lindow and Krogh. Computational evidence for hundreds of non-conserved plant microRNAs. *BMC Genomics*, 6:119, 2005.

[102] Swissprot allergen list website. URL `http://www.expasy.org/cgi-bin/lists?allergen.txt`.

[103] Llave *et al.* Endogenous and silencing-associated small RNAs in plants. *Plant Cell*, 14(7):1605–19, 2002.

[104] Lomsadze *et al.* Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Res*, 33(20):6494–506, 2005.

[105] Lowe and Eddy. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*, 25(5): 955–64, 1997.

[106] Load sharing facility (lsf) website. URL `http://www.platform.com/Products/Platform.LSF.Family/`.

[107] Mackinlay *et al.* The perspective wall: detail and context smoothly integrated. In *Proceedings of acm chi'91*, pages 173–79, 1991.

[108] Majoros *et al.* TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, 20(16):2878–9, 2004.

[109] Margulies *et al.* Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–80, 2005.

[110] Martinez de Alba *et al.* Two chloroplastic viroids induce the accumulation of small RNAs associated with posttranscriptional gene silencing. *J Virol*, 76(24):13094–6, 2002.

[111] MATDB, Mips Arabidopsis thaliana database website. URL `http://mips.gsf.de/proj/thal/db/index.html`.

[112] Mathews *et al.* Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J Mol Biol*, 288(5):911–40, 1999.

[113] Matlin *et al.* Understanding alternative splicing: towards a cellular code. *Nat Rev Mol Cell Biol*, 6(5):386–98, 2005.

[114] Message Passing Interface (mpi) website. URL `http://www-unix.mcs.anl.gov/mpi/`.

[115] Meyers *et al.* Arabidopsis MPSS. An online resource for quantitative expression analysis. *Plant Physiol*, 135(2):801–13, 2004.

[116] Meyers *et al.* The use of MPSS for whole-genome transcriptional analysis in Arabidopsis. *Genome Res*, 14(8):1641–53, 2004.

[117] Michael and McClung. Phase-specific circadian clock regulatory elements in Arabidopsis. *Plant Physiol*, 130(2):627–38, Oct 2002.

[118] Millar and Waterhouse. Plant and animal microRNAs: similarities and differences. *Funct Integr Genomics*, 5(3):129–35, 2005.

[119] Mod python website. URL `http://www.modpython.org/`.

[120] mpiBLAST website. URL `http://mpiblast.lanl.gov/`.

[121] Mueller *et al.* The SOL Genomics Network: a comparative resource for Solanaceae biology and beyond. *Plant Physiol*, 138(3):1310–7, Jul 2005.

[122] Mysql website. URL `http://www.mysql.com/`.

[123] Navas-Delgado *et al.* Intelligent client for integrating bioinformatics services. *Bioinformatics*, 22(1):106–11, Jan 2006.

[124] Oinn *et al.* Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54, 2004.

[125] OpenPBS: Portable Batch System website. URL `http://www.openpbs.org/`.

[126] Ortona *et al.* Molecular and immunological characterization of the C-terminal region of a new Echinococcus granulosus Heat Shock Protein 70. *Parasite Immunol*, 25(3):119–26, 2003.

[127] Parallel Virtual Machine website. URL `http://www.csm.ornl.gov/pvm/pvm_home.html`.

[128] Parry-Smith *et al.* CINEMA–a novel colour INteractive editor for multiple alignments. *Gene*, 221(1):GC57–63, 1998.

[129] Pearson. Genetics: What is a gene? *Nature*, 441(7092):398–401, May 2006. ISSN 0028-0836.

[130] Pearson and Lipman. Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, 85(8):2444–8, 1988.

[131] Pedersen and Nielsen. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. *Proc Int Conf Intell Syst Mol Biol*, 5:226–33, 1997.

[132] Peeters *et al.* Case study: visualization of annotated dna sequences. In Oliver Deussen, Charles D. Hansen, Daniel A. Keim, and Dietmar Saupe, editors, *Vissym*, pages 109–14. Eurographics Association, 2004. ISBN 3-905673-07-X.

[133] Perlin and Fox. Pad: an alternative approach to the computer interface. *Computer Graphics*, 27(Annual Conference Series):57–72, 1993.

[134] Pertea *et al.* GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res*, 29(5):1185–90, Mar 2001.

[135] Peters *et al.* A physical amplified fragment-length polymorphism map of Arabidopsis. *Plant Physiol*, 127(4):1579–89, 2001.

[136] Peters *et al.* TOPAAS, a Tomato and Potato Assembly Assistance System for Selection and Finishing of Bacterial Artificial Chromosomes. *Plant Physiol*, 140(3):805–817, Mar 2006.

[137] Phillips *et al.* Treasure your exceptions. *Plant Cell*, 7(10):1522–7, 1995.

[138] Potter *et al.* The Ensembl analysis pipeline. *Genome Res*, 14(5): 934–41, 2004.

[139] Prasanth *et al.* Regulating gene expression through RNA nuclear retention. *Cell*, 123(2):249–63, Oct 2005.

[140] Python website. URL `http://www.python.org`.

[141] Quevillon *et al.* InterProScan: protein domains identifier. *Nucleic Acids Res*, 33(Web Server issue):W116–20, 2005.

[142] Reinhart *et al.* The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. *Nature*, 403(6772):901–6, 2000.

[143] Reinhart *et al.* MicroRNAs in plants. *Genes Dev*, 16(13):1616–26, 2002.

[144] Remm *et al.* Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J Mol Biol*, 314(5):1041–52, Dec 2001.

[145] Ren *et al.* Local coexpression domains of two to four genes in the genome of Arabidopsis. *Plant Physiol*, 138(2):923–34, 2005.

[146] Repeatmasker website. URL `http://www.repeatmasker.org/`.

[147] Rhoades *et al.* Prediction of plant microRNA targets. *Cell*, 110(4): 513–20, 2002.

[148] Rice *et al.* EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet*, 16(6):276–7, Jun 2000.

[149] Rihs *et al.* Molecular cloning, purification, and IgE-binding of a recombinant class I chitinase from Hevea brasiliensis leaves (rHev b 11.0102). *Allergy*, 58(3):246–51, 2003.

[150] Rocks Cluster Distributrion website. URL `http://www.rocksclusters.org/`.

[151] Rost *et al.* Automatic prediction of protein function. *Cell Mol Life Sci*, 60(12):2637–50, 2003.

[152] Rutherford *et al.* Artemis: sequence visualization and annotation. *Bioinformatics*, 16(10):944–5, 2000.

[153] Ruvkun. Molecular biology. Glimpses of a tiny RNA world. *Science*, 294(5543):797–9, 2001.

[154] Salamov and Solovyev. Ab initio gene finding in Drosophila genomic DNA. *Genome Res*, 10(4):516–22, 2000.

[155] Salzberg *et al.* Microbial gene identification using interpolated Markov models. *Nucleic Acids Res*, 26(2):544–8, 1998.

[156] Sanger *et al.* Nucliotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596):687–95, 1977.

[157] Sanger *et al.* Nucleotide sequence of bacteriophage lambda DNA. *J Mol Biol*, 162(4):729–73, Dec 1982.

[158] Sanger and Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J Mol Biol*, 94(3): 441–8, 1975.

[159] Schattner *et al.* The tRNAscan-SE, snoscan and snoGPS web servers for the detection of tRNAs and snoRNAs. *Nucleic Acids Res*, 33(Web Server issue):W686–9, 2005.

[160] Schauer *et al.* DICER-LIKE1: blind men and elephants in Arabidopsis development. *Trends Plant Sci*, 7(11):487–91, 2002.

[161] Schena *et al.* Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–70, 1995.

[162] Schiltz *et al.* Semantic MOBY. In *Position Paper for the W3C Workshop on Semantic Web for Life Sciences*, 2004.

[163] Schoof *et al.* MIPS Arabidopsis thaliana Database (MAtDB): an integrated biological knowledge resource based on the first complete plant genome. *Nucleic Acids Res*, 30(1):91–3, 2002.

[164] Schuster *et al.* From sequences to shapes and back: a case study in RNA secondary structures. *Proc Biol Sci*, 255(1344):279–84, 1994.

[165] Schuster-Bockler *et al.* HMM Logos for visualization of protein families. *BMC Bioinformatics*, 5:7, 2004.

[166] Service. The race for the 1000 dollar genome. *Science*, 311(5767): 1544–6, 2006.

[167] Shah *et al.* Pegasys: software for executing and integrating analyses of biological sequences. *BMC Bioinformatics*, 5:40, 2004.

[168] Siler *et al.* Absence of cross-reactivity of IgE antibodies from subjects allergic to Hevea brasiliensis latex with a new source of natural rubber latex from guayule (Parthenium argentatum). *J Allergy Clin Immunol*, 98(5 Pt 1):895–902, 1996.

[169] Slater and Birney. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, 6:31, 2005.

[170] Smith and Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7, 1981.

[171] Snyder and Gerstein. Genomics. Defining genes in the genomics era. *Science*, 300(5617):258–60, Apr 2003.

[172] Sonnhammer and Durbin. A workbench for large-scale sequence homology analysis. *Comput Appl Biosci*, 10(3):301–307, Jun 1994.

[173] Sonnhammer and Durbin. A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. *Gene*, 167(1-2):GC1–10, 1995.

[174] Spence. *Information Visualization*. Addison Wesley, 15 December 2000.

[175] Stabenau *et al.* The Ensembl core software libraries. *Genome Res*, 14 (5):929–33, May 2004.

[176] Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, 12(1 Pt 2):505–19, 1984.

[177] Stadler and Stadler. Allergenicity prediction by protein sequence. *FASEB J*, 17(9):1141–3, 2003.

[178] Stark *et al.* Identification of Drosophila MicroRNA targets. *PLoS Biol*, 1(3):E60, 2003.

[179] Stein. Integrating biological databases. *Nat Rev Genet*, 4(5):337–45, 2003.

[180] Stein *et al.* The generic genome browser: a building block for a model organism system database. *Genome Res*, 12(10):1599–610, 2002.

[181] Stiekema and Nap. *Genomics for Biosafety in Plant Biotechnology*, volume 359 of *NATO Science Series*, pages 98–114. IOS Press, Amsterdam, The Netherlands, 2004.

[182] Storm and Sonnhammer. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics*, 18(1):92–9, 2002.

[183] Storz *et al.* An abundance of RNA regulators. *Annu Rev Biochem*, 74: 199–217, 2005.

[184] Sun Grid Engine website. URL `http://gridengine.sunsource.net/`.

[185] Szakos *et al.* Association between the occurrence of the anticardiolipin IgM and mite allergen-specific IgE antibodies in children with extrinsic type of atopic eczema/dermatitis syndrome. *Allergy*, 59(2):164–7, 2004.

[186] Tang *et al.* Wildfire: distributed, Grid-enabled workflow construction and execution. *BMC Bioinformatics*, 6:69, 2005.

[187] Tatusov *et al.* A genomic perspective on protein families. *Science*, 278 (5338):631–637, Oct 1997.

[188] Tatusova *et al.* Complete genomes in WWW Entrez: data representation and analysis. *Bioinformatics*, 15(7-8):536–43, 1999.

[189] Thompson *et al.* CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80, 1994.

[190] TIGR Plant Repeats Database website. URL `http://www.tigr.org/tdb/e2k1/plant.repeats/`.

[191] Transfac website. URL `http://www.gene-regulation.com/pub/databases.html`.

[192] Trindade *et al.* PRECISE: Software for Prediction of cis-Acting Regulatory Elements. *J Hered*, 96(5):618–22, 2005.

[193] Valencia-Sanchez *et al.* Control of translation and mRNA degradation by miRNAs and siRNAs. *Genes Dev*, 20(5):515–24, Mar 2006.

[194] Velculescu *et al.* Serial analysis of gene expression. *Science*, 270 (5235):484–7, 1995.

[195] Venter *et al.* The sequence of the human genome. *Science*, 291(5507): 1304–51, Feb 2001.

[196] Vmatch website. URL `http://vmatch.de/`.

[197] Vos *et al.* AFLP: a new technique for DNA fingerprinting. *Nucleic Acids Res*, 23(21):4407–14, 1995.

[198] Wallis *et al.* A physical map of the chicken genome. *Nature*, 432 (7018):761–4, Dec 2004.

[199] Wilkinson *et al.* BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiol*, 138(1):5–17, 2005.

[200] Wilkinson and Links. BioMOBY: an open source biological web services proposal. *Brief Bioinform*, 3(4):331–41, 2002.

[201] Wingender. Recognition of regulatory regions in genomic sequences. *J Biotechnol*, 35(2-3):273–80, 1994.

[202] Wingender *et al.* TRANSFAC: a database on transcription factors and their DNA binding sites. *Nucleic Acids Res*, 24(1):238–41, 1996.

[203] Wu *et al.* The Protein Information Resource. *Nucleic Acids Res*, 31 (1):345–7, 2003.

[204] WU-BLAST website. URL `http://blast.wustl.edu`.

[205] Xu *et al.* An improved system for exon recognition and gene modeling in human DNA sequences. *Proc Int Conf Intell Syst Mol Biol*, 2: 376–84, 1994.

[206] Xu and Uberbacher. Automated gene identification in large-scale genomic sequences. *J Comput Biol*, 4(3):325–38, 1997.

[207] Yamada *et al.* Empirical analysis of transcriptional activity in the Arabidopsis genome. *Science*, 302(5646):842–6, Oct 2003.

[208] Yeh *et al.* Computational inference of homologous gene structures in the human genome. *Genome Res*, 11(5):803–16, 2001.

[209] Zamore. Ancient pathways programmed by small RNAs. *Science*, 296 (5571):1265–9, 2002.

[210] Zamore *et al.* RNAi: double-stranded RNA directs the ATP-dependent cleavage of mRNA at 21 to 23 nucleotide intervals. *Cell*, 101(1): 25–33, 2000.

[211] Zar. *Biostatistical analysis*. Prentice Hall, 3rd edition, 1996.

[212] ZODB Object Oriented Database website. URL `http://www.zope.org/Products/StandaloneZODB`.

[213] Zorzet *et al.* Prediction of food protein allergenicity: a bioinformatic learning systems approach. *In Silico Biol*, 2(4):525–34, 2002.

List of publications

- Aarts and Fiers. What drives plant stress genes? *Trends Plant Sci*, 8 (3):99–102, 2003.

- Fiers, van den Bosch, Debets and Hoekstra. The dynamics of a senescence plasmid in fungal populations *Genet Res*, 74(1):13–22, 1999.

- Fiers, van de Wetering, Peeters, van Wijk, and Nap. DNAVis: interactive visualization of comparative genome annotations. *Bioinformatics*, 22(3):354–5, Feb 2006.

- Fiers, Kleter, Nijland, Peijnenburg, Nap, and van Ham. Allermatch, a webtool for the prediction of potential allergenicity according to current FAO/WHO Codex alimentarius guidelines. *BMC Bioinformatics*, 5:133, 2004.

- Fiers, Kleter, Peijnenburg, Nijland, Nap, and van Ham. *In Silico prediction of potential allergenicity of proteins according to the FAO/WHO guidelines with the help of Allermatch*, volume 10 of *Wageningen UR Frontis*, chapter 12, pages 109–20. Springer, Dordrecht, The Netherlands, 2006.

- Kleerebezem *et al.*(2003)Kleerebezem, Boekhorst, van Kranenburg, Molenaar, Kuipers, Leer, Tarchini, Peters, Sandbrink, Fiers, Stiekema, Lankhorst, Bron, Hoffer, Groot, Kerkhoven, de Vries, Ursing, de Vos, and Siezen. Complete genome sequence of Lactobacillus plantarum WCFS1. *Proc Natl Acad Sci U S A*, 100(4):1990–5, 2003.

- Peeters, van de Wetering, Fiers, and van Wijk. Case study: visualization of annotated dna sequences. In Oliver Deussen, Charles D. Hansen, Daniel A. Keim, and Dietmar Saupe, editors, *Vissym*, pages 109–14. Eurographics Association, 2004. ISBN 3-905673-07-X.

- Ren, Vorst, Fiers, Stiekema, and Nap. In plants, highly expressed genes are the least compact. *Trends Genet*, 22(10):528–32, 2006.

- Ren, Fiers, Stiekema, and Nap. Local coexpression domains of two to four genes in the genome of Arabidopsis. *Plant Physiol*, 138(2): 923–34, 2005.

- Rigola, Fiers, Vurro, and Aarts. The heavy metal hyperaccumulator thlaspi caerulescens expresses many species-specific genes, as identified by comparative expressed sequence tag analysis. *New Phytol*, 170(4): 753–765, 2006.

- Trindade, van Berloo, Fiers, and Visser. PRECISE: Software for Prediction of cis-Acting Regulatory Elements. *J Hered*, 96(5):618–22, 2005.

- van Hulten, Witteveldt, Peters, Kloosterboer, Tarchini, Fiers, Sandbrink, Lankhorst, and Vlak. The white spot syndrome virus DNA genome sequence. *Virology*, 286(1):7–22, 2001.

Summary

Large amounts of genome sequence data are available and much more will become available in the near future. A DNA sequence alone has, however, limited use. Genome annotation is required to assign biological interpretation to the DNA sequence. The aim of genome annotation is to describe the biological function of every single nucleotide during the life span of an organism. This requires the help of bioinformatics. Bioinformatics is a multidisciplinary approach that combines several areas of expertise in the automated analysis of bio-molecular data. To achieve the goal of proper annotation of a genome, close cooperation between bioinformaticians and (genome) biologists is required at several levels.

This thesis describes a variety of research topics for bioinformatics in the context of genome annotation. A review of the various topics and issues in the science of genome annotation is given in Chapter 2. The research performed has focused on large scale computational efforts in Chapters 3 and 4 and focuses on the prediction of detailed features of genomes in Chapters 5 and 6.

Chapter 3 describes the development of an extensive system for automated genome annotation, called Cyrille2. The major challenges encountered during development of this system have been the extensibility with desired features and the necessary communication between calculation nodes and the databases. Extensibility of the system was achieved by a highly modular structure which allows easy implementation of new analytical approaches and features. Communication between the separate analyses steps of genome annotation is performed with the help of BioMOBY. BioMOBY is an emerging XML standard allowing flexible and uniform description of data and web ser-

vices. The resulting genome annotation system is a flexible, user friendly and high-throughput system able to annotate large amounts of data.

Chapter 4 focuses on visualization of different types of annotation data. Visualization is one of the more powerful methods to evaluate large datasets such as annotated genomes. The research has resulted in the development of a software package, called DNAvis, which applies modern visualization concepts and know-how to forward bioinformatics. DNAvis implements for example fluent zooming and panning to help biologists in the exploration of annotated genomes.

Chapter 5 describes new approaches for the prediction of potential micro-RNAs (miRNAs) in a plant genome. A miRNA is a small RNA gene that downregulates gene expression via pairing with an mRNA that results in degradation or inhibition of translation of that mRNA. A miRNA candidate is identified by its characteristic hairpin shaped precursor and its ability to target an mRNA. This research investigates the complete miRNA potential of the Arabidopsis genome. The results indicate that there are likely to be more miRNAs and more affected mRNAs than predicted.

Chapter 6 of this thesis focuses on proteins and the prediction of a particular protein feature, allergenicity. This has resulted in a novel webtool, called Allermatch, for the prediction of protein allergenicity according to the current guidelines from the World Health Organization (WHO) and the Food and Agriculture Organization of the United Nations (FAO). The potential allergenicity is predicted by comparing the sequence of the protein of interest to a database of known allergens. The tool implements three methods of comparing: a sliding window approach, a wordmatch approach and a full sequence alignment.

The overall scope of this thesis is fairly broad and many of the challenges for bioinformatics in relationship to genome annotation are encountered. The primary challenges are reviewed in the final discussion of this thesis in Chapter 7. It is concluded that these challenges fall into two main categories: communication and quality.

Communication in bioinformatics for genome annotation is a major challenge on several levels: communication between computers and communication be-

tween researchers are both at stake, as well as the communication between computers and human beings. The global bioinformatics community is moving towards a (web) service-based infrastructure. Therefore, communication between computers must be uniform and standardized as discussed in more detail in Chapter 3 of this thesis. Communication between researchers is far from trivial, especially in a multidisciplinary field as bioinformatics. This was particularly an issue during the development of the visualization tool of Chapter 4 that was accomplished in close cooperation between biologists, bioinformaticians and visualization scientists. Visualization also deals with the communication between computers and human beings to communicate the results of genome annotation.

The second major issue in bioinformatics and genome annotation is the quality of annotation data, as is discussed in Chapter 2 of this thesis. Most annotation depends in some way or another on previous annotations. For example, inference of gene function is often done by comparing a new gene to a database of known genes. Obviously the quality of such prediction relies on the quality of the underlying data. The issue of error propagation is an important issue in the field of genome annotation and needs much future attention. Ample room and importance should be given to proper validation and experimentation in the laboratory to back up any annotation based on bioinformatics. An easy quality indication system that covers the type and number of data behind any annotation would seem essential for the future of genome annotation.

Current developments in notably sequencing technology indicate dramatic increases in data production speed for genomics and related 'omics' areas in the near future. A major challenge of that future will be to manage and annotate the growing data stream. It should be made sure that more data help to improve the quality of data and help to improve the quality of genome annotations.

Currently only a small percentage of any genome is understood. To reach the goal of genome annotation and correctly describe the function of every single nucleotide, a lot of work remains to be done. This is by itself a major challenge. It is not unlikely that along the way evidence for new mechanisms of gene regulation and organization will be discovered. Genome annotation will therefore

remain one of the more exciting and challenging fields of bioinformatics for decades to come.

Samenvatting

De basenvolgorde (sequentie) van een enorme hoeveelheid DNA is beschikbaar en op korte termijn zal nog veel meer ter beschikking komen. De DNA sequentie van een genoom (alle unieke chromosomen van een organisme bij elkaar) is op zichzelf beperkt bruikbaar. Genoomannotatie is het proces dat een interpretatie (annotatie) toekent aan een (deel van een) genoomsequentie. Het doel van genoomannotatie is het beschrijven van de biologische functie van iedere nucleotide op ieder moment van de levensduur van een organisme. Dit doel vereis de hulp van bio-informatica. Bio-informatica verenigt een aantal wetenschappelijke disciplines in zich om biomoleculaire informatie automatisch te analyseren. Succesvolle genoomannotatie is afhankelijk van nauwe samenwerking, op verscheidene niveaus, tussen (bio-)informatici en (genoom)biologen.

Dit proefschrift beschrijft verschillende onderwerpen binnen de bioinformatica die zijn gericht op genoomannotatie. Een samenvatting van het vakgebied is te vinden in hoofdstuk 2. Het onderzoek beschreven in de rest van het proefschrift varieert van genoombrede,grootschalige analyses in de hoofdstukken 3 en 4, tot de voorspelling van specifieke elementen in de hoofdstukken 5 en 6. Tenslotte wordt de toekomst van het vakgebied besproken in hoofdstuk 7.

Hoofdstuk 3 beschrijft de ontwikkeling van een uitgebreid systeem, Cyrille2 genaamd, om genoomannotatie te automatiseren. De belangrijkste uitdagingen in de ontwikkeling van een dergelijk systeem liggen op het gebied van uitbreidbaarheid en communicatie. Uitbreidbaarheid is noodzakelijk in een snel ontwikkelend veld als genoomannotatie. Het is gewaarborgd door een modulaire structuur, waardoor snelle implementatie van nieuwe eigenschappen mogelijk is. Voor de communicatie tussen de individuele stappen van het annotatieproces wordt gebruik gemaakt van BioMOBY. BioMOBY is een recente

XML standaard die een flexibele, uniforme en automatiseerbare beschrijving van informatie en analyses (eventueel over internet) mogelijk maakt. Het resulterende genoomannotatiesysteem is flexibel, gebruikersvriendelijk en heeft de capaciteit om grote hoeveelheden informatie (tegelijkertijd) te verwerken.

Hoofdstuk 4 behandelt de visualisatie van verschillende soorten annotaties. Visualisatie is een van de meest krachtige hulpmiddelen om grote hoeveelheden informatie te interpreteren en evalueren. Onderzoek beschreven in dit hoofdstuk heeft geresulteerd in de ontwikkeling van een software pakket, genaamd DNAvis, dat moderne visualisatieconcepten en -kennis toepast op genoomannotatie. DNAvis gebruikt bij voorbeeld vloeiend "scrollen" en "zoomen" ter ondersteuning van het onderzoek van een geannoteerd genoom.

Het volgende hoofdstuk (5) beschrijft een nieuwe aanpak voor de voorspelling van micro-RNAs (miRNAs) in een plantengenoom. Een miRNA is een klein RNA molecule dat de expressie van een gen kan remmen door te plakken aan het messenger-RNA (mRNA, boodschapper RNA) van dat gen. Een miRNA kandidaat kan herkend worden aan de typische haarspeldstructuur waaruit het miRNA gevormd wordt, gekoppeld aan de mogelijkheid om te plakken aan een mRNA. De resultaten beschreven in dit hoofdstuk maken het aannemelijk dat het totale aantal miRNAs en in het *Arabidopsis thaliana* genoom vele malen groter is dan tot nu toe werd aangenomen.

Hoofdstuk 6 van dit proefschrift richt zich op de annotatie van een specifieke eigenschap van eiwitten, namelijk allergeniciteit. Allergeniciteit is het vermogen van een eiwit om een allergische reactie te veroorzaken. Dit hoofdstuk beschrijft de ontwikkeling van een webgebaseerde applicatie, Allermatch, die allergeniciteit voorspelt op basis van de criteria opgesteld door de Wereldgezondheidsorganisatie (WHO) en de "Food and Agriculture Organization" (FAO) van de Verenigde Naties. Potentiële allergeniciteit wordt voorspeld door de sequentie van een te onderzoeken eiwit te vergelijken met een database van bekende allergenen (eiwitten die een allergische reactie kunnen veroorzaken). Allermatch gebruikt hiervoor drie methoden: (1) volledige sequentievergelijking, (2) het flexibelvergelijken van alle sequentiefragmenten van een bepaalde lengte en (3) de stringente vergelijking van hele korte stukjes van de sequentie (zogenaamde woorden).

Dit proefschrift beschrijft een breed scala aan onderwerpen en veel van de huidige uitdagingen in de bioinformatica zijn onderwerp van discussie in dit proefschrift. De belangrijkste kwesties worden besproken in hoofdstuk 7. De conclusie is dat de belangrijkste uitdagingen in twee categorieën uiteenvallen: communicatie en kwaliteit.

Communicatie in de bioinformatica voor genoomannotatie is een uitdaging op onderscheidbare niveaus: communicatie tussen computers onderling en tussen onderzoekers onderling is van belang, evenals communicatie tussen computers en onderzoekers. De bioinformatica gemeenschap beweegt richting een op internetdiensten (webservices) gebaseerde infrastructuur. Dit vereist een gestandaardiseerde communicatie tussen computers, zoals besproken in hoofdstuk 3. De ontwikkeling van het visualisatiepakket uit hoofdstuk 4 is het resultaat van een nauwe samenwerking tussen biologen, bioinformatici en visualisatiedeskundigen. Tijdens dit project bleek het belang van een goede communicatie tussen onderzoekers. Visualisatie is eveneens een belangrijk hulpmiddel in de communicatie tussen computers en onderzoekers.

Het tweede belangrijke struikelblok in de bioinformatica voor genoomannotatie is de kwaliteit van annotaties. Dit wordt onder andere besproken in hoofdstuk 2 van het proefschrift. Het annoteren van een eigenschap van een deel van het genoom is in bijna alle gevallen direct afhankelijk van eerdere annotaties. Het toekennen van een genfunctie wordt bijvoorbeeld vaak gedaan door het nieuwe gen te vergelijken met een set genen waarvan de functie bekend is. De kwaliteit van dergelijke annotaties is daarom direct afhankelijk van de kwaliteit van de onderliggende dataset en/of annotaties. Het propageren van fouten is een belangrijk onderwerp binnen genoomannotatie en dient in de toekomst verder onderzocht te worden. Veel meer aandacht dan nu gebruikelijk zal moeten worden besteed aan bevestiging van de voorspellingen in het laboratorium om de door de bioinformatica voorspelde annotaties te ondersteunen en te verbeteren. Een eenduidig indicatiesysteem voor de kwaliteit van een annotatie dat ook de onderliggende informatie van een annotatie omvat, lijkt essentieel voor de toekomst van genoomannotatie.

De huidige ontwikkelingen in onder andere sequentietechnologie wijzen op een dramatische stijging in de productiesnelheid van informatie binnen de genomica en gerelateerde “omics” vakgebieden in de zeer nabije toekomst. De uitdaging

is om die informatie in goede banen te leiden en goed te annoteren. Het is eveneens van belang deze datastroom in te zetten om de kwaliteit de bestaande van genoomannotaties te verbeteren.

Momenteel wordt de biologische betekenis van slechts een klein percentage van een genoom begrepen. Het uiteindelijke doel van genoomannotatie, het beschrijven van de biologische functie van iedere nucleotide op ieder moment van de levensduur van een organisme, is nog lang niet bereikt. Het is waarschijnlijk dat aanwijzingen voor nieuwe mechanismen van genregulatie en organisatie ontdekt zullen worden. Genoomannotatie zal daardoor een van de meest spannende en uitdagende vakgebieden van de bio-informatica blijven voor de komende decennia.

Curriculum vitae

Mark Fiers werd op 9 september 1970 geboren te Eindhoven. Na het doorlopen van de MAVO, MLO microbiologie en HLO plantenbiotechnologie is hij gestart met de opleiding biologie, vrije oriëntatie, aan de universiteit van Wageningen. Binnen de vrije orientatie heeft hij zich gespecialiseerd in de mathematische biologie, mede door een afstudeervak waarin een populatiemodel is ontwikkeld dat het gedrag van een verouderingsgerelateerde plasmide in een schimmel beschrijft. In 2007 is Mark afgestudeerd en na enkele baantjes in de IT begonnen als junior onderzoeker bij de bioinformaticagroep van het toenmalige CPRO, nu Plant Research International. Tijdens deze periode heeft hij een half jaar doorgebracht bij het NCGR in Santa Fe (VS). In 2001 is hij gestart met een promotieonderzoek bij Willem Stiekema. Het onderzoek heeft zich gericht op verscheidene onderwerpen binnen de bioinformatica voor genoomannotatie. In de nabije toekomst zal Mark werkzaam blijven op de afdeling applied bioinformatics van Plant Research International.

ACAGAGCGACGCTGGTCTCCCMIRTEGTTGCTAGAAGAAGACGGCGTCGACGTCTGACTG
GACTCGCGGCGACTTACCTTTCAGTCGTGCGCTCCTGATCCGGCGCTCGGAATTTGTCCC
CGGCTTCAGGGCTGCGGGGCCTEPCOGGAGGCGTATOMAGGCGGCTCGAAAACGATCCAG
GGGAGCCGAGGCGCTCCTCTTGTCATCCCACTCAGCGCCATGTCCTGGATGTTCAAGAGG
GATCCAGTTJEANNETACTTGCAGACJOOSTAGTATGGAGTTCATGGAAATRICHARDGC
CTCTCATATCCAACTTTCTTTCCACGTTTTGAATTCCAAGATGTTATCCCTCCAGATGAC
TTTCTAACTAGTGATGAAGAAGTAGATTCCLIESBETHTGGAAGTIMGAGAGGTCATGTG
GTTGGACTACGJUDITHCACGGGAGTAGTTAATAATAATGAAATGELLENATTACAACGA
GATCCTAATAACCCTTTOMATAAGAATGCAATTAAAGTROELANDGTGAATADAAATCAA
GTTGGCCATTTAAAGAAAGAGCDAPHNEGTGCTTTGGCCTATATCATGGACAACAAATTG
GCACVELITCHKAGGGTAGTTCCTTTTGGTGCAAACAATGCTTTTACCATGCCTCTGCAT
ATGACTTTTTGGXINYINGAAGAAAATAGAAAAGCGGSANDERATCAGTTGAAGAAACAT
GGATTTAAATTGGGTCCTGCACCAAAAACTTTAGGATTCAATTTGGAAAGTGGTTGGGGC
TCTGGAAGAGCTGGACRENSKEATAGTATGCCAGCGCATGCTGCAGTKOOSATGACAACT
GAACPACTTAAAACAGACGGTGACAAATTGTTTGAAGATTTAAAAGAAGATGATAAAACC
CATGAAATJANACCAGCTGAGMAICOTGAAACACCACTGCTTMAAIKECAAAAACAAGCT
CTAGCTTGGATGGTGTCACGGGAAAACAGCAAAGATETTCCACCATTCTGGGAACAGCGA
AATGACTTATACTATTCGCAAATAACAAATTTTTCTGAGAAGGACCGACCAGAAAATGTC
CATGGATHEOTTTTAGCTOSCARTATGGWILLEMGTAAAACTCTTACAGCCATTGCAGTA
ATCCTTACCAACTTCCATGATGGCAGACCTCTTCCTATTGAAAGAGTTAAGIJSAATCTA
CTGAAGAAGGAATGTAATGTTAACGATGASASKIAGAAACHUUBAGGAAACAATACCAGT
GAAAAGGCJONNAGACTAAGCAAAGAAGCATCTEDOUARDGTGAACAACCCAGTATTTCA
GATATCAAGGAGAAGAGTAAGTTTCGCATGTCAGAATTGJANAGCTCCCGCCCCAAAAGA
AGAAAAACTGCTGTCFERRYCATAGAAAGCAGTGATTCAGAGGAAATTGAAACAAGTGAA
TTGCCGCAGAAAATGAAAGGCAAACTGAAAAATGTACCYRILLEAACTAAAGGCAGGGCG
AAAGCAGGMARIONTAAGGTEZRIGAAGATGTGGCATTTGCAJPTGCATTLUDMILATCC
GTTCCTACAACAAAAAAGAAAATGTTGAAAAAGGGAGCTTGTGCAGTGGAGGGGTCAAAG
AAAACTGATERWINGGAGAGACCBASAACAACACTGATCATCTGTCCBERLINDATGTTA
AGCAACTGGATTGACCAGTTTGGACAACATATAAAAHERMANGTACACTTGAATTTTTAT
GTTTATTATBARTCTGATJACKTTAGAGAACCGGCCTTACTTTCAAAACAGGATATTGTT
TTGACTHANSATAATATTTTAACTCATGACTATGGAACTAAAGGAD&DERSCCATTACAT
AGCATAVIOLETTACAGAGTGATCCTGGDAANAGGACATGCCPAULGAAATCCAAATGCT
CATGAAATGGAACCAGCTGAGGCTATTGAAACACCACTGCTTCCACATCAAAAACAAGCT
AATGACTTATACTATAACACAATAACAAATTTTTCTGAGAAGGACCGACCAGAAAATGTC

Nawoord

Aan velen ben ik dank verschuldigd! Veel heb ik geleerd en velen hebben me gesteund in de afgelopen jaren.

Als eerste, Hans, die me binnen heeft gehaald in de bioinformatica en waarvan ik veel heb geleerd, en niet alleen als wetenschapper. Helaas overleed hij veel te vroeg.

Evenzeer ben ik JP en Roeland erg dankbaar, voor het voortzetten van de begeleiding, vele discussies en geduld. Willem die de promotie mogelijk heeft gemaakt. Alle co-auteurs! Collega's, een vreemd doch vriendelijk volkje. Theo, Saskia, Ferry en Cyrille voor hun hulp bij het tot stand komen van dit boekje.

Ook dank aan vele van mijn, allezins aardige, vrienden! Vele uren van diepe gesprekken, ouwehoeren, drinken, muziek luisteren, wandelen. Jullie weten wie jullie zijn! Vind jezelf terug op de tegenoverliggend pagina, dan hebben jullie tenminste iets te doen tijdens de promotie.

Mijn familie! Sjan, sFer en 'sSas, en ook de schoonfamilie: Tom, Liesbeth, Maaike, Bart en Daan, zijn allemaal, op hun eigen wijze, erg goed in familie zijn (fijn).

Edoch, het is Cyrille die ik het meest dankbaar ben. Zonder haar steun, geduld, vertrouwen en bovenal liefde, was het ondenkbaar dat ik zou zijn waar ik nu ben. Dank, meisje. En Mirte heeft niet direct geholpen, maar wat ze mist aan subtiliteit met mijn toetsenbord, maakt ze ruim en breed goed met onbevangenheid, enthousiasme, liefde en brabbelen!

dank allen!

en, Pa, wat had ik graag gehad dat je hier nog bij was geweest.

Education Statement of the Graduate School Experimental Plant Sciences

Issued to: Mark Fiers
Date: 8 November 2006
Group: Applied Bioinformatics, PRI, Wageningen University & Research Centre

1) Start-up phase	*date*
▶ **First presentation of your project**	
Mining of the arabidopsis genome	Feb 2002
▶ **Writing or rewriting a project proposal**	Feb 2002
▶ **Writing a review or book chapter**	Feb 2006
▶ **MSc courses**	
▶ **Laboratory use of isotopes**	
Subtotal Start-up Phase	*13.5 credits**

2) Scientific Exposure	*date*
▶ **EPS PhD student days**	
EPS PhD student day, Wageningen University	Sep 19, 2006
▶ **EPS theme symposia**	
EPS theme 4 symposium 'Genome Plasticity'	Dec 20, 2002
EPS theme 4 symposium 'Genome Plasticity'	Dec 9, 2004
▶ **NWO Lunteren days and other National Platforms**	
Spring and autumn meetings Lunteren	2001-2005
▶ **Seminars (series), workshops and symposia**	
Bioinformatics seminars (CBSG & PRI)	2001-2005
EPS seminars	2001-2005
Isabelle Carr, The molecular mechanism of...	Mar 21, 2002
Karin van Haren, bioASP	Apr 15, 2004
Dr. Elena R. Alvarez-Buylla, Molecular mechanisms of floral organ ...	May 18, 2005
▶ **Seminar plus**	
▶ **International symposia and congresses**	
ISMB, Heidelberg, Germany.	Aug 6-10, 1999
ISMB, San Diego, USA.	Aug 19-23, 2000
Genome Informatics, Hinxton, UK.	Aug 8-12, 2001
Solanaceae genome workshop, Wageningen, NL.	Sep 19-21, 2004
Netherlands Conference on Bioinformatics 2004 "Images of Life", Groningen, NL.	Oct 7-8, 2004
▶ **Presentations**	
Poster at Genome Informatics Hinxton, UK.	Aug 2001
Poster at Solanaceae genome workshop, Wageningen, NL.	Sep 2004
Poster at Images of Life, Groningen, NL.	Oct 2004
Presentation at the Springschool Bioinformatics, Wageningen, NL. .	Mar-Apr 2004
Subtotal Scientific Exposure	*13.0 credits**

3) In-Depth Studies	*date*
► **EPS courses or other PhD courses**	
Winterschool Bioinformatics	Dec 11-15, 2000
Mathematics & Biology Winterschool	Dec 17-19, 2001
Springschool Bioinformatics	Mar 31-Apr 1-2, 2004
► **Journal club**	
Bi-weekly literature discussions of applied bioinformatics (PRI) & Bioinformatics (WUR)	2001-2005
► **Individual research training**	
0.5 year practical period at NCGR, Santa Fe, USA	2000
Subtotal In-Depth Studies	*9.3 credits**
4) Personal development	*date*
► **Skill training courses**	
Projectmatig werken, Kern Konsult (one week)	2005
► **Organisation of PhD students day, course or conference**	
► **Membership of Board, Committee or PhD council**	
Subtotal Personal Development	*1.5 credits**
TOTAL NUMBER OF CREDIT POINTS*	**37.3**

Herewith the Graduate School declares that the PhD candidate has complied with the educational requirements set by the Educational Committee of EPS which comprises of a minimum total of 30 credits (ECTS)

* A credit represents a normative study load of 28 hours of study

Work presented in this thesis has received funding from the Centre for Biosystems Genomics (CBSG)

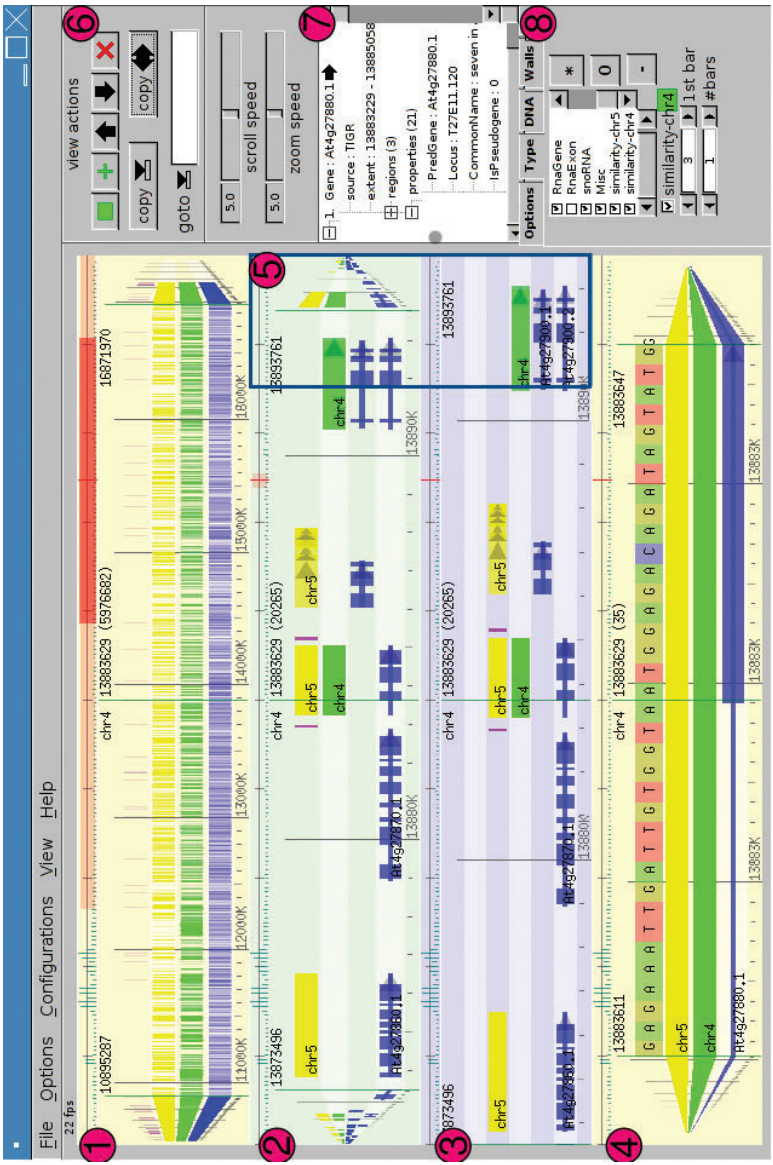

Figure 4.1: behorende bij Bioinformatics for plant genome annotation

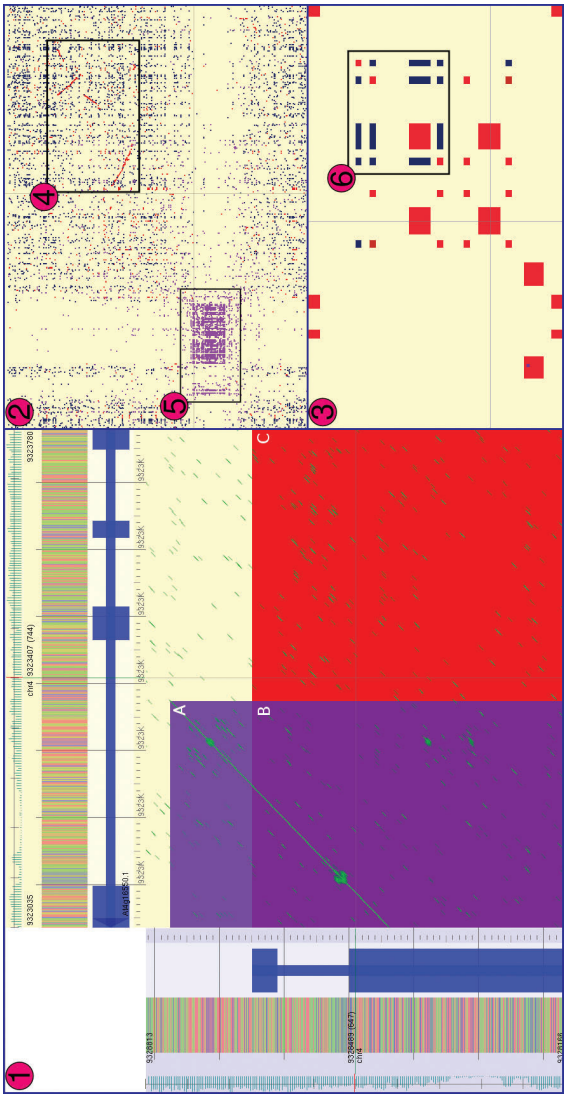

Figure 4.3: behorende bij Bioinformatics for plant genome annotation

Stellingen

1. De rol van kleine RNAs in de cel is veel omvangrijker dan nu wordt aangenomen.
   Dit proefschrift en Kim. Genes Dev, 20(15):1993-7, 2006.

2. Communicatie tussen bioinformatici en biologen zal verbeterd worden door een uniform indicatiesysteem dat de betrouwbaarheid van een annotatie weergeeft.
   Dit proefschrift.

3. Genregulatie is geen hiërarchisch systeem.
   Kauffman. The origins of order: Self-Organization and Selection in Evolution. Oxford University Press, 1993.

4. Studies die eetgedrag koppelen aan een geringe verhoging van de kans op kanker dienen de stress van vele goedbedoelde adviezen in acht te nemen.
   Mariken et al. *Cancer Epidemiol Biomarkers Prev. 14(12):2943-51, 2005.*

5. Het minimale verschil tussen mens en chimpansee op genoomniveau sluit niet uit dat het wanhopige (menselijke) zoeken naar dat verschil veroorzaakt wordt door een gen voor antropocentrisme.
   Khaitovich et al. *Nat Rev Genet. 7(9):693-702, 2006.*

6. Nu computers een steeds grotere rol krijgen in wetenschap en het dagelijkse leven wordt het hoog tijd dat ook de leek er een gezond wantrouwen tegen ontwikkelt.

7. Het belangrijkste wapenfeit van vier jaar debat over normen en waarden is dat de buschauffeur nu consequent gegroet wordt bij het verlaten van de bus.

Stellingen behorende bij het proefschrift "Bioinformatics for plant genome annotation". Te verdedigen op 8 november 2006 door Mark Fiers.