



INSTITUTO  
SUPERIOR  
TÉCNICO

# **A system for automated genome annotation**

**André Filipe Magalhães Gomes**

Dissertation to obtain the Master's Degree in  
**Biomedical Engineering**

## **Jury**

President:	Prof. Paulo Freitas
Supervisors:	Prof. Ana Teresa Correia Freitas Prof. Dr. João André Carriço Prof. Dr. Paulo Soares da Fonseca
Member:	Prof. Dr. Pedro Monteiro

**October 2010**



---

## Acknowledgments

I would like to begin by thanking to Mauro Santos for the lessons that he gave me during the realization of my thesis.

I also would like to thank my supervisors, Prof. Ana Teresa Freitas, Dr. Paulo Soares da Fonseca and Dr. João André Carriço that guided me through my thesis.

Finally, I would like to thank my family and friends for all the support.



# Abstract

*Eucalyptus globulus* is a tree used to produce paper pulp. With the aim to study the genes responsible for wood formation in *E. globulus*, which are not yet fully understood although known to have an impact on the quality of paper pulp, GenEglobwq project was defined.

One of the project goals is the development of an annotation module, to automatically annotate the gene sequences obtained. Therefore, the aim of this thesis was to extend a web information system, GEDI, used to manage the data from GenEglobwq project, in order to provide a tool for automated genome annotation. In this sense the annotation module in the GEDI system was created, which makes use of an automatic genome annotation pipeline, named MAKER, which can be used to annotate any type of organism. One way to annotate a genome is through the use of markers to identify specific areas of the genome. Thus, it was also created a marker module to manage marker data.

In order to validate the annotation module operation, the chloroplast genomes of *Eucalyptus globulus* and *Eucalyptus grandis* was annotated. However, like any automatic genome annotation pipeline, not all the genes were annotated, being necessary a manual curation of the results.

## Keywords

Genome, Annotation, Biomarkers, Chloroplast, *Eucalyptus globulus*.



# Resumo

O *Eucalyptus globulus* é uma árvore utilizada na produção de pasta de papel. Com o objectivo de estudar os genes responsáveis pela formação de madeira do *E. globulus*, que ainda não são totalmente compreendidos mas sabe-se que tem um impacto na qualidade da pasta papel, surgiu o projecto GenEglobwq.

Um dos objectivos deste projecto é a criação de um módulo de anotação de sequências de ADN. Neste sentido, o objectivo desta tese consistiu em estender o sistema de informação, GEDI, utilizado para gerir os dados do projecto GenEglobwq, a fim de fornecer uma ferramenta para a anotação automática de genomas. Para isso, foi criado um módulo de anotação no sistema GEDI, que faz uso de uma pipeline de anotação automática MAKER, que permite a anotação de qualquer tipo de organismo. Uma das possíveis formas de anotar um genoma é através do uso de marcadores, que permite identificar zonas específicas do genoma. Deste modo, foi também criado um módulo de marcadores para gerir este tipo de dados.

De modo a testar o funcionamento do módulo de anotação automática, anotou-se os genomas dos cloroplastos do *Eucalyptus globulus* e *Eucalyptus grandis*. No entanto, como qualquer outra ferramenta de anotação automática, não foram anotados todos os genes, sendo posteriormente necessária uma curação manual dos resultados.

## Palavras Chave

Genoma, Anotação, Biomarcadores, Cloroplasto, *Eucalyptus globulus*.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Context . . . . .	2
1.2	Aim and Motivation . . . . .	2
1.3	Layout of this Thesis . . . . .	2
<b>2</b>	<b>Genome Sequencing and Annotation</b>	<b>5</b>
2.1	Introduction . . . . .	6
2.2	Tools and Methods for Genome Annotation . . . . .	8
<b>3</b>	<b>GEDI - Genomic Data Information System</b>	<b>11</b>
3.1	System Architecture . . . . .	14
3.2	Database . . . . .	17
<b>4</b>	<b>New Modules</b>	<b>21</b>
4.1	Annotation Module . . . . .	22
4.1.1	Use Cases and Pipeline . . . . .	25
4.1.2	Annotation Runs List . . . . .	27
4.1.3	Annotation Run Configuration . . . . .	27
4.1.4	Results List . . . . .	29
4.2	Markers Module . . . . .	30
4.2.1	Use Cases . . . . .	30
4.2.2	Markers List . . . . .	32
4.2.3	Project Markers List . . . . .	35
4.2.4	Clone Markers List . . . . .	36
<b>5</b>	<b>Case Study</b>	<b>39</b>
5.1	<i>Eucalyptus globulus</i> . . . . .	40
5.1.1	Pipeline Configuration . . . . .	40
5.1.2	Measures of performance . . . . .	41
5.1.3	Results . . . . .	41
5.2	<i>Eucalyptus grandis</i> . . . . .	44

## Contents

---

5.2.1 Results . . . . .	44
<b>6 Conclusions and Future Work</b>	<b>47</b>
<b>Bibliography</b>	<b>51</b>

# List of Figures

2.1	Simple work flow of a sequence and annotation project [1]. . . . .	9
3.1	GEDI administration interface. . . . .	14
3.2	System Architecture. . . . .	15
3.3	JavaEE multitiered platform . . . . .	16
3.4	JavaEE Server and Containers. . . . .	16
3.5	Entity-relationship model (ERM) of Geneglob database. . . . .	18
3.6	Some Chado Modules and Their Dependencies. . . . .	19
4.1	MAKER pipeline [2]. . . . .	23
4.2	Run management use case diagram. . . . .	25
4.3	Curation management use case diagram. . . . .	26
4.4	GEDI annotation pipeline. . . . .	27
4.5	Annotation runs list page. . . . .	28
4.6	Annotation run configuration page - top. . . . .	28
4.7	Annotation run configuration page - bottom. . . . .	29
4.8	Annotation Run Results Page. . . . .	30
4.9	Marker module use case. . . . .	31
4.10	Markers list page – top. . . . .	33
4.11	Markers list page – bottom. . . . .	33
4.12	Project Markers Page. . . . .	35
4.13	Clone Markers Page – top. . . . .	36
4.14	Clone Markers Page — bottom. . . . .	37
5.1	MAKER results for the chloroplast genome of <i>E. globulus</i> using Apollo. . . . .	43
5.2	NCBI annotation for the chloroplast genome of <i>E. globulus</i> . . . . .	43
5.3	MAKER results for the chloroplast genome of <i>E. grandis</i> using Apollo. . . . .	45
5.4	NCBI annotation for the chloroplast genome of <i>E. grandis</i> . . . . .	45



# List of Tables

- 5.1 Performance of the annotation pipeline (AP) and GeneMark on the chloroplast genome of *E. globulus*. . . . . 41
- 5.2 Performance of the annotation pipeline on the chloroplast genome of *E. Grandis*. . 44



# List of Abbreviations <sup>1</sup>

<b>BAC</b>	Bacterial Artificial Chromosome
<b>BLAST</b>	Basic Local Alignment Search Tool
<b>CSV</b>	Comma Separated Values
<b>DAS</b>	Distributed Annotation System
<b>DAO</b>	Data Access Object
<b>DDBJ</b>	DNA Data Bank of Japan
<b>ddNTP</b>	dideoxyNucleotide TriPhosphate
<b>DNA</b>	DeoxyriboNucleic Acid
<b>DTO</b>	Data Transfer Object
<b>EAR</b>	Enterprise Archive
<b>EJB</b>	Enterprise JavaBean
<b>EMBL-Bank</b>	European Molecular Biology Laboratory Nucleotide Sequence Database
<b>ERM</b>	Entity-Relationship Model
<b>EST</b>	Expressed Sequence Tag
<b>GBrowse</b>	Generic Genome Browser
<b>GEDI</b>	GEnomic Data Information system
<b>GFF3</b>	Generic Feature Format version 3
<b>Glimmer</b>	Gene Locator and Interpolated Markov ModelER
<b>GMOD</b>	Generic Model Organism Database
<b>GO</b>	Gene Ontology
<b>HMM</b>	Hidden Markov Models
<b>IBET</b>	Instituto de Biologia Experimental e Tecnológica
<b>IDE</b>	Integrated Development Environment
<b>INESC-ID</b>	Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento
<b>INSDC</b>	International Nucleotide Sequence Database Collaboration

---

<sup>1</sup>Note: This list is in alphabetic order

## List of Abbreviations

---

<b>J2EE</b>	Java 2 Platform Enterprise Edition
<b>JBrowse</b>	JavaScript-based genome browser
<b>JSF</b>	JavaServer Faces
<b>JSP</b>	JavaServer Pages
<b>KDBIO</b>	Knowledge Discovery and BIOinformatics
<b>KEGG</b>	Kyoto Encyclopedia of Genes and Genomes
<b>mRNA</b>	Messenger RNA
<b>NCBI</b>	National Center for Biotechnology Information
<b>ORF</b>	Open Reading Frame
<b>QTL</b>	Quantitative Trait Loci
<b>RAID</b>	Redundant Array of Independent Disk
<b>RNA</b>	RiboNucleic Acid
<b>rRNA</b>	Ribosomal RNA
<b>SABIA</b>	System for Automated Bacterial Integrated Annotation
<b>SGN</b>	Sol Genomics Network
<b>SNAP</b>	Semi-HMM-based Nucleic Acid Parser
<b>STRP</b>	Short Tandem Repeat Polymorphism
<b>SVN</b>	Subversion
<b>tRNA</b>	Transfer RNA
<b>VNTR</b>	Variable Number Tandem Repeat
<b>WAR</b>	Web Archive



# 1

## Introduction

### Contents

---

1.1 Context . . . . .	2
1.2 Aim and Motivation . . . . .	2
1.3 Layout of this Thesis . . . . .	2

---

### 1.1 Context

Portugal is a major producer of eucalypt (*Eucalyptus globulus*), which provides most of the raw material used to produce paper pulp. The final pulp and paper quality is highly affected by the wood properties. It is accepted that variations of wood properties are regulated by a high number of genes and proteins during xylem differentiation. However the xylogenesis (wood formation) is not yet fully understood. Sequencing, mapping and annotating the *E. globulus* genome will help better understand this process. With that objective in mind, the GenEglobwq [3] project was created. The GenEglobwq project is a collaboration between IBET, RAIZ, IST and INESC-ID and aims to identify and characterize the genomic regions that underlies strong effect QTL (Quantitative Trait Loci) for pulp yield, in *E. globulus*, combining an array of genomic tools and transcriptomic approaches. As specific objectives, this project intends to identify and create new markers for the region of interest in genetic linkage maps, annotate the genomic sequence that underlies a strong effect QTL, present a list of genes associated to pulp yield determination, analyze the expression of the supposed "pulp yield genes" in different genetic backgrounds, and finally, improve bioinformatics tools.

The GenEglobwq project requires dealing with a huge amount of data, ranging from sequencer raw data to annotation data. The main role of KDBIO group in this project is to provide an easy way to collect, manage and process these huge amount of data. Therefore KDBIO group developed a web information system, GEDI - GENomic Data Information system (that will be described later), to fulfill this demand.

### 1.2 Aim and Motivation

In order to study the xylogenesis of *Eucalyptus globulus*, it is necessary to annotate the regions of interest. Since a purely manual annotation is a very laborious, time-consuming and error prone task, an automatic annotation pipeline is desired. Therefore the main goal of this thesis is to provide an automatic annotation pipeline to the GenEglobwq project by extending the GEDI system with a tool for automated genome annotation. Besides that, one method to annotate a genome is through the use of markers to identify specific areas of the genome. Thus, a marker module was also created to manage marker data.

This work includes also the study of several automatic annotation tools in order to select the one that integrates better with the GEDI system and test it with a real case.

### 1.3 Layout of this Thesis

The layout of this thesis is as follows. In Chapter 1, we present a brief description of the context and motivate our work. Chapter 2 introduce the tasks of genome sequencing and annotation. A

set of genome sequencing and annotation methods are described. In Chapter 3 we describe the architecture and implementation of the GEDI information system. Chapter 4 describes the main work developed in this project. The new annotation and marker modules are detailed. The reasons that lead to the choice of the pipeline MAKER, for automatic genome annotation, are also presented and justified. Chapter 5 includes two test cases that validates the automatic annotation module. Both chloroplast genomes of *Eucalyptus globulus* and *Eucalyptus grandis* were used to illustrate the use of the annotation pipeline. Conclusions and future work are presented in chapter 6.



# 2

## Genome Sequencing and Annotation

### Contents

---

2.1 Introduction . . . . .	6
2.2 Tools and Methods for Genome Annotation . . . . .	8

---

### 2.1 Introduction

In our days, the importance of the DNA molecule on the development and functioning of all known living organisms and its role in heredity is acknowledged. But it was not always like that, since its first isolation in 1869 by Friedrich Miescher until the 1930s, DNA was largely ignored. It was only with experiments such as Griffith's Experiment [4] and the Avery-MacLeod-McCarty experiment [5] in the 1930s and the 1940s that indicated that DNA perhaps carries genetic information. And it was only confirmed in 1952 by Alfred Hershey and Martha Chase in the Hershey-Chase experiment [6]. With these new discoveries and with the Watson and Crick double-helical DNA model for the structure of DNA and the genetic code cracked in the 1960s, many of DNA's secrets had been revealed. However, it was only with the beginning of DNA sequencing in the 1970s which accelerated biological research and discovery.

DNA sequencing is the process of identifying the order of the nucleotide bases in a molecule of DNA. One of the initial methods used and still used and the most popular for DNA sequencing is the Sanger method, due to its efficiency and ease automation. In this method, the DNA sequence is added in four reaction tubes. In each tube a DNA primer which serves as starting point synthesis of a new DNA strand, a DNA polymerase to extend the primer into a full-length DNA chain, the four nucleotides and a radioactive label, usually a radioactive deoxynucleotide are added. The core of this method is the addition, in each tube, of a single type of dideoxynucleotide triphosphates (ddNTPs) as DNA chain terminators. The ddNTPs are nucleotides that lacks the site at which other bases can attach during chain growth. Therefore after DNA synthesis we will have a set of sequences with different lengths that end with the same nucleotide. Then, each reaction is loaded in its own gel lane and separated by size via gel electrophoresis. Since the DNA strand is labeled with a radioactive deoxynucleotide, the DNA bands are visible by autoradiography or ultraviolet light. Then, we can easily read (from bottom to top) the DNA sequence through the relative positions of the different bands among the four lanes, taking in consideration that what we are reading is the complementary sequence of the target sequence.

A similar method to the previous one, is pyrosequencing based on the "sequencing by synthesis" principle. This method differs from the preceding one by detecting pyrophosphate release on nucleotide incorporation instead of relying on chain termination with ddNTPs. In this process the target sequence is fixed and separately exposed to each nucleotide. When there is an incorporation of nucleotides a light is produced and detected, being its intensity proportional to that number of nucleotides incorporated. Thus, allowing the determination of the complementary sequence [7].

The previous methods are commonly used to sequence DNA strands with a maximum of 700-900 bases, since after this number the sequencing quality deteriorates. So in order to sequence a whole genome which have a lot more than 900 bases, other approaches are necessary. There

are two classical procedures to sequence a genome: the BAC-to-BAC, or hierarchical method, and the whole genome shotgun sequencing. These were the methods used to sequence the first draft of the human genome.

In the BAC-to-BAC sequencing process, the DNA is fragmented into pieces of about 150 kbp and cloned into bacterial artificial chromosomes (BACs), which are plasmids (small piece of DNA that can replicate in bacteria) containing foreign DNA. The whole set of BACs containing the whole genome is called a BAC library. Before sequencing the fragments, a rough physical map is built by figuring out the relative order of each BAC. This is done by a technique called fingerprint, which consists of determining the location of each BAC along the chromosome by cutting each BAC fragment with a single enzyme and finding common sequence landmarks in overlapping fragments. Afterwards, each BAC is broken into pieces of 1500 bp long and their ends of about 500 bp are sequenced and fed into computers algorithms that look for common sequences that join two fragments together [7]. This method is commonly used to sequence small genomes or whenever someone wants to study only parts of a genome. Since this method does not require any pre-existing information is also referred as *de novo* sequencing.

The whole genome shotgun sequencing is a sequencing process in which the whole DNA is sheared into randomly small pieces about 2, 10 kbp long and inserted into a plasmid. Then, for each piece, the sequence of 500 bp from each end, also called reads, is sequenced. After that, computers algorithms assemble the sequenced fragments into a contiguous sequence, also called contig, and the full genome assembled from this contigs. This method does not require the construction of a physical map, and therefore is usually faster [7], although less accurate than the BAC-to-BAC approach. This method is commonly used for larger genomes.

In the last five years a huge advance on sequencing technologies was observed. With high-throughput sequencing technologies is now possible to improve the throughput/cost ratio by miniaturization and parallelization the sequencing process. As for example, one of these technologies is the 454 pyrosequencing, which use a parallelized version of pyrosequencing. In this technique millions of copies of a unique single-stranded DNA molecule attached to a bead are sequencing in parallel with hundreds of thousands of others beads. And each time nucleotide are incorporated, a light signal is generate and recorded, allowing the determination of the complementary sequence.

However, in the study of a particular organism, the complete genome sequence provides only partial and raw information. More importantly, scientists need to find out where the genes are, what they do, how they are related, etc. This is where the annotation process intervenes to attach this information to the genome sequence. Genome annotation is thus the process of attaching biological information to the genome sequences and starts by identifying the positions of structural genomic elements, like genes, exons, introns, repeated regions, promoters, etc. This process can be defined as structural annotation. After identifying these elements a secondary annotation to

## 2. Genome Sequencing and Annotation

---

provide biochemical and biological function information to these elements is necessary and this process is called functional annotation [8].

The quality of the annotation is very important for future experiments. If an annotation is correct, then experiments, such as, RNAi, PCR, gene expression arrays, targeted gene knockout, or CHIP [8], that need information from these annotation, are greatly facilitated. The quality of the annotation, on its turn, depends, among other things, on the annotation pipeline that is employed, some more detailed and accurate than others, and also on the skills and experience of those operating the pipeline. It is also important for a correct annotation to have a curation and review process with an expert on the biology of each genome.

Sequencing and annotation process are not the only processes on a genome project. As mentioned before the BAC-to-BAC sequencing process needs a physical map in order to sequence a genome. Physical map is also a type of genome map. Genome map consists of locating a specific gene to a particular region of a chromosome and determining the relative distances between genes. It can be of two types: physical mapping, which consists of determining the physical distances, in nucleotide bases, from one landmark to another; and genetic-linkage mapping, which consists of determining the relative distances between genes and the order of those genes along the chromosome. Mapping a genome can be done with the use of so-called *markers*. Depending on its type, a marker can easily identify a gene on a chromosome and any other feature. Example of markers are: variable number tandem repeats (VNTRs), also know by minisatellites. These markers with 10-100 bp, can be repeated a variable number of times; short tandem repeat polymorphisms (STRPs), also know by microsatellites. These markers with 2-5 bp are also repeated many times.

The benefits that come from sequencing, annotate and mapping a genome are vast. In medicine, for instance, they can help improve diagnosis of disease, earlier detect genetic predispositions to a disease, and can be applied to gene therapy and drug control systems. In agriculture, they can be used to create crops more nutritious and resistant to diseases, insects and drought. With many other applications in different areas.

### 2.2 Tools and Methods for Genome Annotation

The advance on DNA sequencing technologies, culminating at the recent high-throughput sequencing-by-synthesis techniques such as 454 pyrosequencing or Illumina sequencing enable researches to efficiently sequencing an entire genome in a very cost effective way. The quest for the \$1000 USD genome, that seemed so distance, now clearly within reach of these new technologies. In contrast, genome annotation has not followed these developments, and therefore genomes are being sequenced at higher rate than they can be properly annotated.

There are several methods and pipelines for genome annotation, with different degrees of



automation versus manual intervention. A very simple annotation pipeline is represented in Figure 2.1, from step d) to e). First, a gene prediction software is used, such as Glimmer [9] or GeneMark [10], which produces a set of predicted protein-coding genes. Then, using the previous predictions, BLAST [11] searches are performed against large protein databases in order to assign name and function to the protein based on homology.

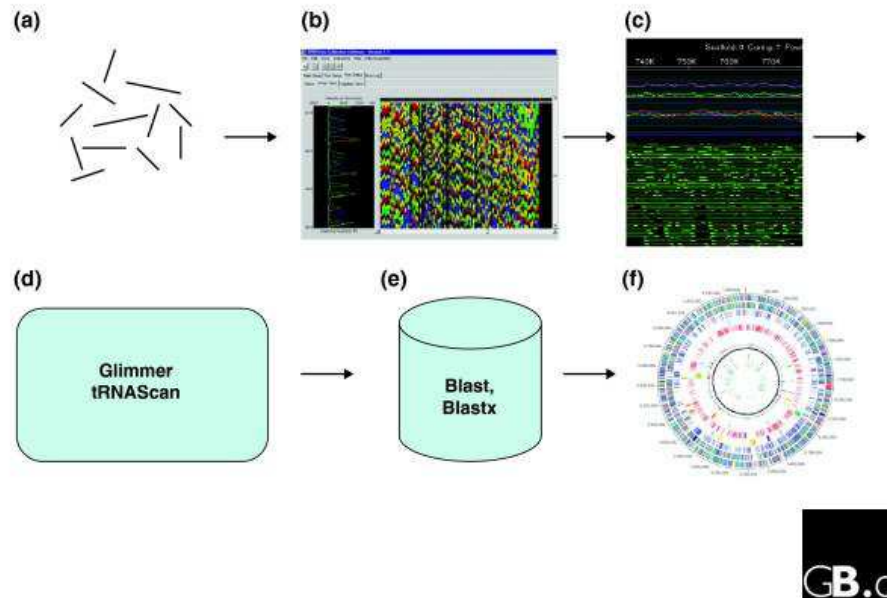


Figure 2.1: Simple work flow of a sequence and annotation project [1].

Depending on the adopted pipeline and the intended results, several tools can be used to help in the annotation process. One of such tools is, as mentioned before, an *ab initio* gene predictor, such as Glimmer or GeneMark, for prokaryotic organisms, and SNAP [12] or Augustus [13] for eukaryotic. Most gene predictors are based on probabilistic models, such as Hidden Markov Models, used to recognize specific sequence patterns occurring within the genes or at their vicinities. These tools usually give more accurate results for prokaryotic genomes than for eukaryotic genomes, because the later are typically far more complex than the former.

Although gene predictors can be used to find protein-coding genes, if, for example, a functional annotation is desired, one should resort to a different tool, like Blast2GO [14]. Blast2GO initially performs BLAST against National Center for Biotechnology Information (NCBI) database or another custom databases, and then does the mapping of GO terms to the BLAST results. Gene Ontology [15], the GO consortium is a effort that aims to normalize the representation of gene and gene product attributes across species and databases. The second task of Blast2GO is to perform annotation. At this stage, it makes use of an annotation rule to find out the most precise annotations. An optional step allows to analyze GO term distribution differences between groups of sequences. The final step is the visualization where the annotations and statistics results can be verified.

## 2. Genome Sequencing and Annotation

---

Another genome annotation pipeline, with focus on structural annotation, is provided by MAKER [2]. This pipeline uses four external executables: RepeatMasker [16], BLAST, Exonerate [17] and a gene predictor. RepeatMasker is used to identify repeats, BLAST and Exonerate are used to align ESTs and proteins to a genome and the gene predictor is used to produce initial gene predictions. At the end of the pipeline, the results of these tools are combined, leading to the final gene annotations. The results are in GFF3 format [18] and can be easily visualized in a genome browser.

A useful tool to exchange biological information is the Distributed Annotation System (DAS) [19]. This project is based on the idea that annotation should be spread across multiple sites and not kept in centralized databases. DAS is a client-server system that allows a client to collate genome annotation information from multiple servers and display it in a single view.

Once a genome is annotated, the data must be made available in order to be useful. The typical solution is to submit the annotation to a public database such as one of the three databanks of the International Nucleotide Sequence Database Collaboration, namely the DNA Data Bank of Japan (DDBJ) [20], the European Molecular Biology Laboratory Nucleotide Sequence Database (EMBL-Bank) [21], and the U.S. National Center for Biotechnology Information GenBank [22]. These databases are the largest public genome databanks, working together in exchanging data. Alternatively, the annotation data can be saved in a local database, in which case a generic genome database schema, such as Chado [23], should be used.

Apart from annotating a genome accurately, one needs an easy way to visualize (and optionally modify) the obtained results. The popular visualization tool GBrowse [24] allows the user to navigate through the annotated features at different detail levels through a convenient graphical user interface. Mostly identical features are provided by JBrowse [25], a browser with an AJAX-based interface, regarded as the official successor to GBrowse. A stand-alone tool that not only allows to visualize but also to edit annotation is Apollo [26], a genome editor based in Java.

Many of the aforementioned tools, including MAKER, Apollo, Chado and GBrowse are maintained through the Generic Model Organism Database project, GMOD [27]. GMOD is a federation of open source software components aimed at providing the functionality that is needed by all organism databases.

# 3

## **GEDI - Genomic Data Information System**

### **Contents**

---

<b>3.1 System Architecture</b>	<b>14</b>
<b>3.2 Database</b>	<b>17</b>

---

### 3. GEDI - Genomic Data Information System

---

GEDI is a web information system designed to store and manage data from BAC-to-BAC sequencing and pyrosequencing projects. It was originally built within the GenEglobwq Project [3], which main goal is the study of the *Eucalyptus globulus* species, but its development has always been thought in a way that it could be used to cover other organism.

Prior to developing the system, some research was done on similar systems in order to define the appropriate architecture. One of the system analyzed was the Sol Genomics Network (SGN) [28]. This system is a clade oriented database dedicated to the biology of the Solanaceae family, containing genomic, genetic, phenotypic and taxonomic information about this family. It is used by one of the bioinformatics centers involved in the genome sequencing of the tomato. The tomato genome project has begun in 2004 and its initial approach was to sequence the genome using a BAC-by-BAC approach. The main advantages of this software are the visualization tools to analyze the contigs, the genetic maps, the physical maps, and the database structure that is linked with these tools and is organized to integrate all the related data. This system is open source and uses the Chado database to store their data.

Another important system that was analyzed, was the system named SABIA [29]. This tool is a genome project management information system maintained by the Bioinformatics Group at LNCC, in Petrópolis, Brazil. It is a new tool developed for the assembly and annotation of prokaryotic genomes. The main advantages of this software come from the fact that it allows managing the assembling phase of the sequencing project through automatic assembling scripts. Using this scripts it is possible to generate reports about the sequencing status in terms of DNA contamination, number of contigs built by number of reads inserted in the system, number of gaps still to be closed, etc. In order to annotate the contigs, the annotation module makes use of Glimmer [9], GeneMark [10] and tRNAScan-SE [30] programs to identify ORFs and tRNAs and it does the BLAST [11] against reference biological databases (Interpro [31], KEGG [32], COG [33], etc) to identify metabolic pathways. It also allows the identification of new ORFs in the genome using the “pick a sequence” tool and manual identification of mRNAs, rRNAs and frameshifts. It makes a final report on all the matches that were found and finally allows the user to validate the annotated contig or to reject it.

The automatic scripts in the assembling module of SABIA are linked with Phred/Phrap scripts [34], which parse chromatograms into nucleotide sequence and base-call quality files, and finally assemble reads. Along with Phred and Phrap, a tool called Consed is also provided for viewing and finishing the assemblies produced with Phrap. The Phred/Phrap software suite is currently maintained by Phil Green's group at the University of Washington.

As mentioned above, the GEDI system was initially developed to handle data from the sequencing project of the *E. globulus* organism. Here we briefly present the methodology adopted in this project, in order to clarify the decisions made during the information system development.

At the earliest stage of the project a set of BAC libraries were created containing specific

---

regions of the genome. Once these libraries were created, eucalyptus genes of interest were screened by hybridization with genetic markers/probes of other eucalyptus species. The clones with a positive hybridization signal were selected for further clone validation tests, since they have a higher probability of carrying the sequences of interest.

To determine the minimum number of clones to be sequenced, two possible strategies could be used: fingerprinting and pair end sequencing. In the first case, fingerprint band patterns of the selected BACs are compared using the Image/FPC [35, 36] software stack, and the minimum BAC tiling path is obtained. In the second case, BAC end sequences, together with sequences obtained in the validation tests, are used with sequence assembler tools like Phred/Phrap with the purpose of finding the minimum number of BACs required to sequence the genes of interest. In this project the sequencing was done using 454 pyrosequencing technology.

Once sequenced, the contigs are then allocated to the related Eucalyptus species genetic maps through the genetic markers (DNA probes), allowing for the *E. globulus* physical map construction. Probes from *Eucalyptus grandis* can be used to support the physical map construction because of synteny between these two species. Finally, the contigs are then analyzed with a gene finder software, and compared against other biological databases in order to find biological features with which they are to be annotated after validation by a human expert.

In the context of this project, the GEDI system has been used to manage and store BAC libraries characteristics and screened clones, DNA markers, fingerprint results and pair end sequences, reads from sequencing technologies and contigs assembled with these reads, physical maps, and biological annotation. For the automatic annotation, an external tool was used, and for the visualization of these annotations the choice was to use JBrowse, a standard visualization tool in this context. The system uses two databases: the Geneglob database to store most of the data and the Chado database to store the annotated data.

GEDI has two different interfaces: a public interface where any user can visualize the projects and the BACs that are made public; and an administration interface where an administrator user can manage all the data in the system. These interfaces were designed with two frameworks, RichFaces [37] and PrimeFaces [38], which allows intuitive visualization and manipulation of big volumes of data. Figure 3.1 shows a screen capture of the administration interface.

At this interface, it is possible to find the following items: Figure 3.1, 1- User information, and logout action - Displays information about the user and allows session termination. 2- Top Navigation Menu - Provides a centralized and intuitive way to navigate through all the application pages. 3- List Content Area - Is the area where all the database lines for a specific database entity, that has to be administrated, are displayed. It has a data scroller on the top of the table that allows table pagination. The number of rows showed in each page is 20. 4- Edit Content Area - It has a form that allows to create a new entity that is added to table in the List Content Area.

The GEDI system was developed using the Integrated Development Environment (IDE) Eclipse

### 3. GEDI - Genomic Data Information System

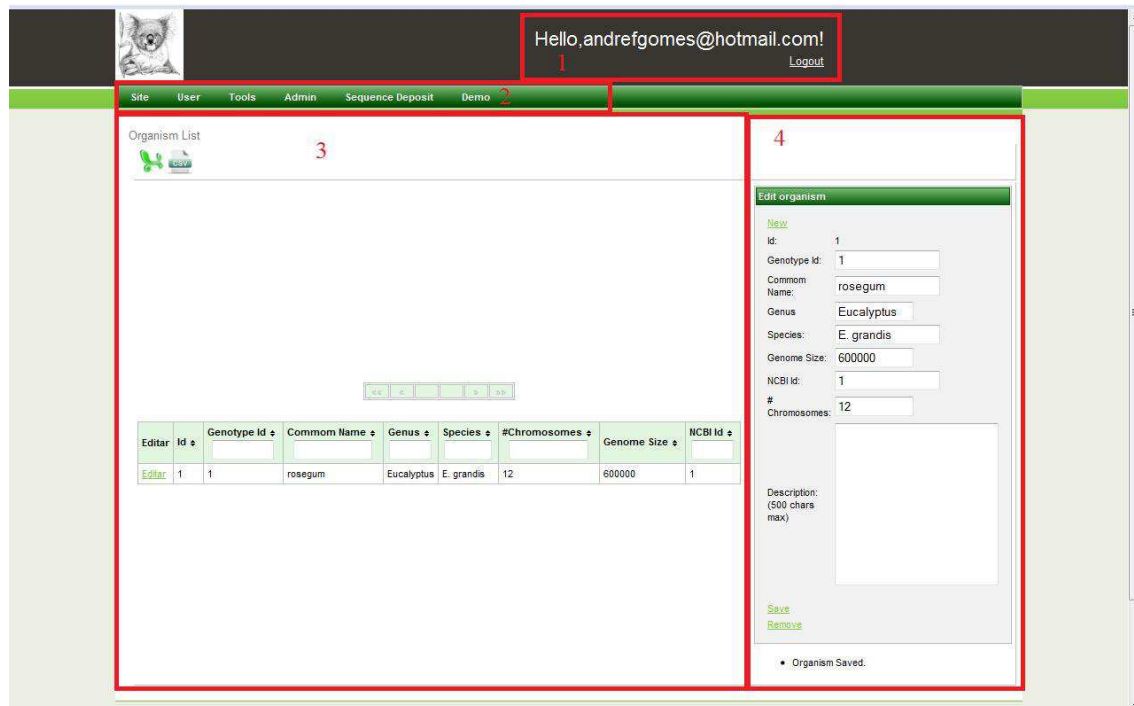


Figure 3.1: GEDI administration interface.

Galileo (3.5). The application server used was Glassfish v2.1, a Java application server that follows the reference Java 2 Platform Enterprise Edition (J2EE) [39] implementation. The Database Management System was PostgreSQL 8.3, a robust open source relational database management system. The Bioinformatics Software Phred was used to obtain the quality values from the trace files base calls. Java Frameworks Biojava 1.7 was used to manipulate chromatograms, phred quality files, and DNA and protein sequences. For building, management and versioning the project Maven2 was used. SVN was used to do code conversioning. To build the user interfaces, the user interface framework for building web applications JavaServer Faces (JSF) 1.2 was used.

The system should be available even in case of disk failure. A RAID 1 solution was used to solve this problem, meaning that the information system disk is copied in real time to a parallel disk. Also the information system will be linked to a secured periodic backup system so the data can be recovered in case of an unpredictable data loss.

### 3.1 System Architecture

GEDI has been implemented as an online information system, which is a standard approach to store and manipulate this kind of data. The user (Figure 3.2) has access to the web server through an ordinary web browser. The web server will allow authorized users to visualize, insert, delete and export data from the database. The information system's server is based on a Linux

distribution, and open source or academic licensed software for data analysis is preferred.

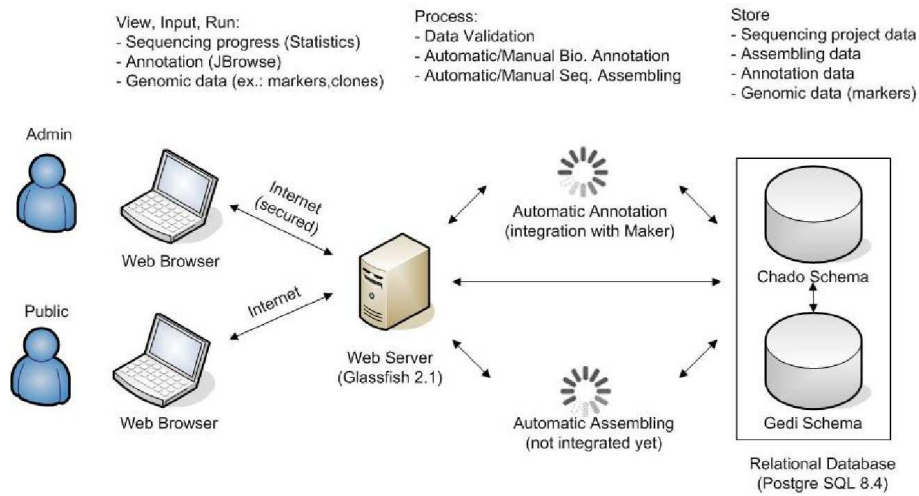


Figure 3.2: System Architecture.

The software development has been based on Java SUN technology, more precisely the J2EE, and as so it is compatible with most operating systems.

The Sun J2EE software stack provides a standard platform supporting services that communicate data in an interoperable way to both desktop and web clients. This is achieved by implementing the server side and the data communication APIs according to the J2EE specifications. The Java bioinformatics community is progressively adopting these J2EE standards, where applicable. For instance, this has already been observed in the Biojava 3.0 library [40], by encapsulation the business logic code in the JavaBeans Specification.

The Java EE platform provides a multitier architecture for enterprise applications. Each Java EE application is composed by various components and each component belongs to a tier in the multitiered Java EE environment, so they can be installed on different machines. In Figure 3.3 is presented two multitiered Java EE applications.

Java EE multitiered applications can have three or four tiers, as shown in Figure 3.3, distributed in three locations: client machines, the Java EE server machine, and the database, and because of that they are considered as three-tiered applications.

Due to the component-based and platform-independent Java EE architecture and to the structure in reusable components of business logic, it makes easy to write Java EE applications. Also, services in a form of containers for every component type are supplied, which makes the user free to focus on their own problem. In Figure 3.4 is a representation of the JavaEE server and containers.

The components of Figure 3.4 can be described as follows:

1. Java EE server: The runtime portion of a Java EE product. A Java EE server provides EJB and web containers.

### 3. GEDI - Genomic Data Information System

---

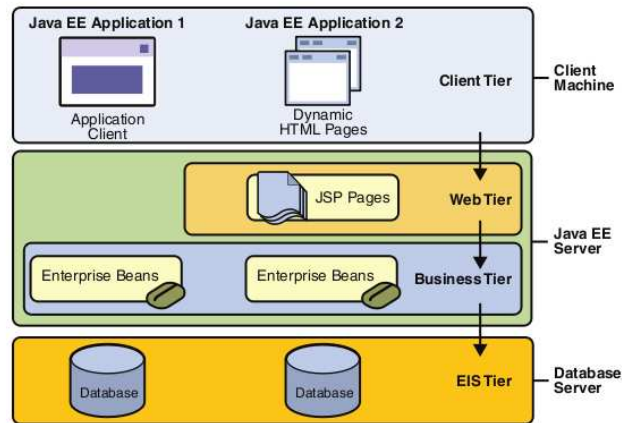


Figure 3.3: JavaEE multitiered platform. 1. Client-tier components run on the client machine. 2. Web-tier components run on the Java EE server. 3. Business-tier components run on the Java EE server. 4. Enterprise information system (EIS)-tier software runs on the EIS server.

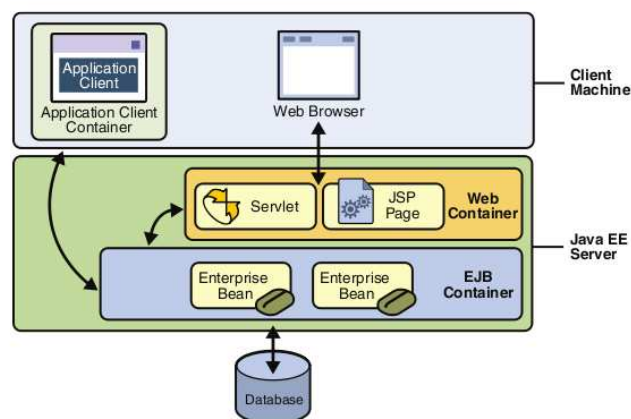


Figure 3.4: JavaEE Server and Containers.



2. Enterprise JavaBeans (EJB) container: Manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.
3. Web container: Manages the execution of JavaServer Pages (JSP) and servlet components for Java EE applications. Web components and their container run on the Java EE server.
4. Application client container: Manages the execution of application client components. Application clients and their container run on the client.
5. Applet container: Manages the execution of applets. Consists of a web browser and Java Plug-in running on the client together.

As for the J2EE APIs used in this system they are: Java Persistence API, a Java standards-based solution for persistence. Persistence uses an object-relational mapping approach to bridge the gap between an object oriented model and a relational database. This solution allows Java code to access data in different databases; Java Authentication and Authorization Service, provides a way of authentication and authorization for a specific user or group of users to run a Java EE application; EJB, to encapsulate the business logic code.

GEDI has a modular structure comprising the following modules:

**geneglob-ear** This is the Enterprise Archive (EAR) module that is deployed to the Java application server and that holds all the other application modules.

**geneglob-webapp** This is the Web Archive (WAR) that holds the dynamic JSF web pages.

**geneglob-dao-x** This module contains the Data Access Objects (DAO) that are encapsulated in the Enterprise JavaBeans (EJB) specification, holding the business logic that manages the module persistence. EJB specification divides the module in a client part and a server part. The client part provides interfaces with methods that can be consumed by the client applications. The server part holds the implementation of the interface methods. The client methods transfer data to the client applications through Data Transfer Objects (DTOs).

**geneglob-facade-x** This module serves as an unified interface to a set of modules in the system, in order to maintain the modular structure.

The remaining modules, `geneglob-util-persistence`, `geneglob-configuration`, have utility methods that support the application code.

## 3.2 Database

GEDI has two databases, the Geneglob and Chado databases. The Geneglob database defines a schema specifically made for this project whose design was partially inspired by the Ensembl database schema [41]. Ensembl is a joint project between EMBL - EBI and the Wellcome

### 3. GEDI - Genomic Data Information System

Trust Sanger Institute whose goals include the development of a software system which produces and maintains automatic annotation on selected eukaryotic genomes. It was launched in 1999 in response to the imminent completion of the Human Genome Project.

The Geneglob database model, Figure 3.5, represents most of all the important data, which must be stored in the information system database. The model organizes data in entities and defines the relationships between these entities.

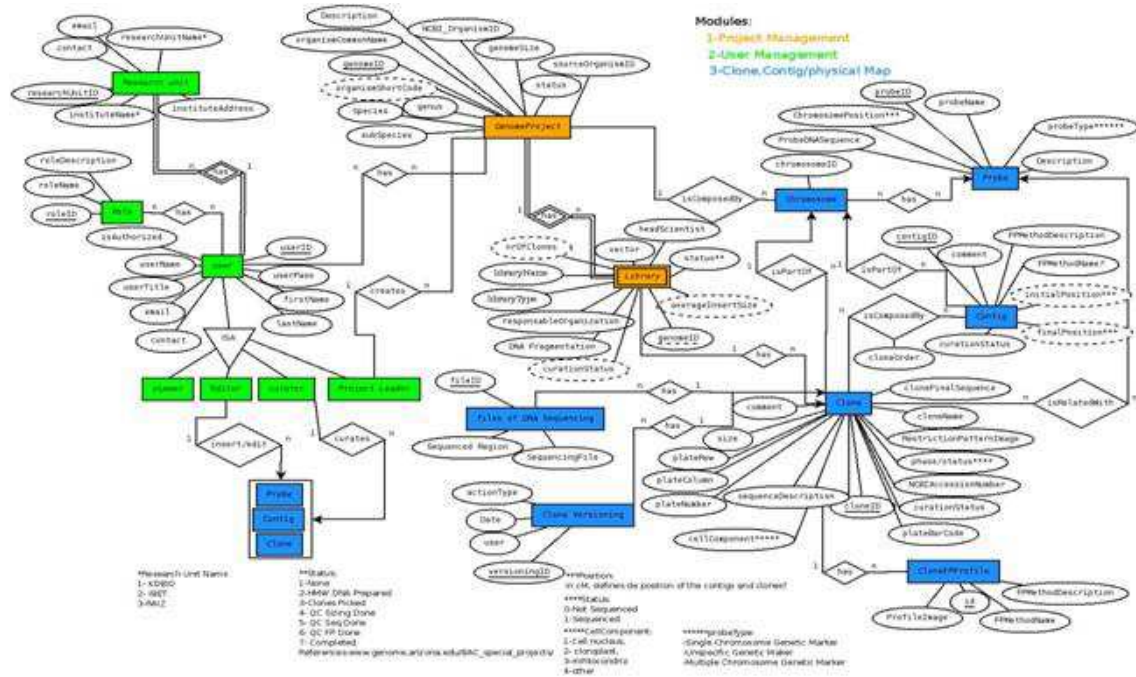


Figure 3.5: Entity-relationship model (ERM) of Geneglob database.

The main entities are:

**User** The users that are allowed to use the system.

**Genome Project** The definition of the genome project.

**Library** The BAC libraries to store the clones.

**Clones** The hosts of the DNA inserts that will be sequenced.

**Contigs** The consensus sequence resulted from the assemble of different BACs.

**Marker** The DNA markers used in the project.

**Chromosomes** The importance of this entity is better understood when linked with the Markers, becoming the Species Genetic Map; or when linked with the Contigs, becoming the Species Physical Map.

The other database used, Chado, is a database schema of the GMOD project which was developed by FlyBase [42], a database of *Drosophila* Genes and Genomes carried out by a consortium of *Drosophila* researchers and computer scientists. However, its development has always been thought to be generic, extensible and available as open source, so that it could apply for any model organism data repository. At present, Chado is being used on several research projects such as FlyBase, Sol Genomics Network, VectorBase [43], XenBase [44], among others.

Chado is a successful and flexible database schema mainly because of the use of ontologies (or controlled vocabularies) as a way to describe and organize data, and also because it has a modular schema. The use of ontologies was motivated by the need to describe the significant diversity of living things. At the same time they served to categorize and classify the contents of databases. The Chado modular schema made it a highly extensible database, allowing the introduction of new modules covering new domains of knowledge. Some Chado modules are represented in Figure 3.6.

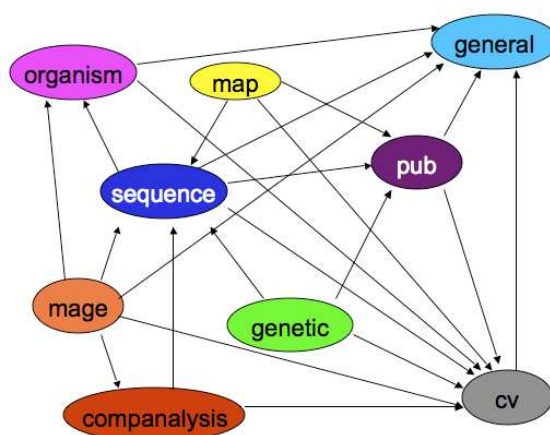


Figure 3.6: Some Chado Modules and Their Dependencies. General - Provides data entities with stable, global, unique identifiers. Controlled Vocabulary (cv) - Stores controlled vocabularies (CVs) and ontologies. Publication (pub) - Describes references to sources of knowledge, for example, published articles, conferences, Web pages, and so on. Organism - Holds information about a given species. Sequence - Describes genomic features and things that can be tied to or descend from genomic features.

The use of ontologies and its modular structure make Chado a good option for saving the annotated data. GEDI system mainly uses the sequence module to manage the annotated data.

The main table in the sequence module is the *feature* table, for describing biological sequence features. For Chado a feature is a region of a biological polymer or a set of regions on this polymer. The features stored are typed according to an ontology, through the *cvterm* table in the cv module, and these can be gene, exon, transcript, regulatory region, chromosome, etc. They may contain others features, for example a chromosome contains a gene, this relation is managed by the *featureloc* table. Also, a feature may have a certain relation with other features, for example an exon is part of a transcript, this relation is managed by the *feature\_relationship* table.

### 3. GEDI - Genomic Data Information System

---

# 4

## New Modules

### Contents

---

4.1 Annotation Module . . . . .	22
4.2 Markers Module . . . . .	30

---

## 4. New Modules

---

In order to study the xylogenesi s of *Eucalyptus globulus*, it is necessary to annotate the regions of interest. For an easy and fast way to annotate the sequences stored in the GEDI system, an annotation module was created. Besides that, one method to annotate a genome is through the use of markers to identify specific areas of the genome. Thus, a marker module was also created to manage marker data. These two modules are described in the following sections.

### 4.1 Annotation Module

In order to develop the annotation module, it was necessary to decide which automatic annotation tool should be used in the GEDI system. The software should fulfill certain requirements, such as, to be open source, to be preferably written in Java, to be well documented and to present good results for any type of organism.

During the process of choosing the software, several tools were tested. Special attention was given to Blast2GO [14], DAS [19] and MAKER [2]. The first options were Blast2GO and DAS, mainly because they are written in Java, which would facilitate integration. DAS is not really a annotation tool but a client-server system to exchange biological annotation. In case we would like to annotate a new genomic sequence not in any DAS server, this system would be useless. Blast2GO is a good tool for functional annotation but we wanted a tool for structural annotation. Another option would be to use gene predictors, but they are usually specific to one type of prokaryotic or eukaryotic organism. Therefore, MAKER became a good choice because it is a structural annotation pipeline that can be used to annotate equally prokaryotic and eukaryotic organisms, and can also be used to re-annotate a genome sequence as well as to annotate a new one. The only drawback of this tool is that it is written in Perl. However, during the development of the module, this has proven to be a minor issue.

MAKER is a structural annotation pipeline, being one of the Generic Model Organism Database (GMOD) [27] components. It was developed to allow researchers to easily annotate eukaryotic and prokaryotic genome sequences and to create genome databases. This system makes use of existing software tools that can produce *ab initio* gene predictions, align ESTs and proteins to a genome and identify and mask repeat elements, combining their output and produce what it believes to be the best possible annotation [8].

The installation of MAKER requires a computer running Linux or Mac OS X. The list of external software used includes BioPerl [45] a toolkit of perl modules useful in building bioinformatics solutions in Perl, SNAP [12] a gene finder for eukaryotic organism, RepeatMasker [16] a program that identifies repeats and low complexity DNA sequences, Exonerate [17] a tool for pairwise sequence comparison, and NCBI BLAST [11] a tool for sequence comparison. Some optional components, such as, Augustus [13] a gene finder for eukaryotic organism, GeneMark [10] a gene finder for prokaryotic organism, and FGENESH [46] a gene finder for eukaryotic organism,

can also be installed.

In order to annotate a sequence, MAKER needs some minimal input: a genomic sequence, of any length, in FASTA format; and three configuration files containing the path to the executables and various computation parameters. The user can also give an optional RepeatMasker database file, a proteins file, and an ESTs/mRNAs file, all of them in FASTA format. RepeatMasker file is a file containing organism specific repeats. The proteins file contains proteins to be aligned to the genome, and the ESTs/mRNAs file contains EST and/or mRNA sequences from the organism being annotated [8]. After putting the sequence to annotate, it will pass through five discrete steps that are easily interoperable: compute, filter/cluster, polish, synthesize, and annotate (Figure 4.1).

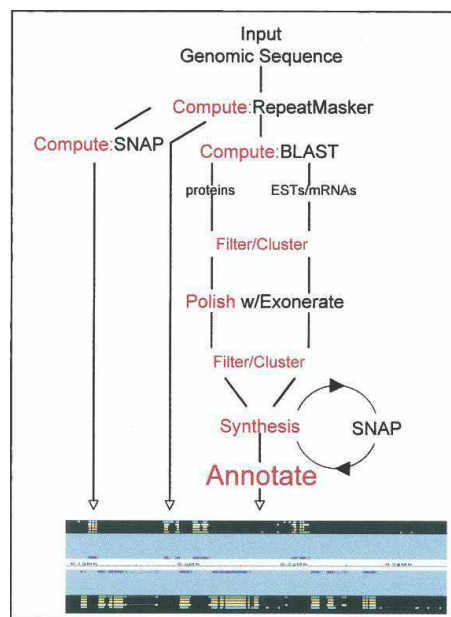


Figure 4.1: MAKER pipeline [2].

In the first step of the annotation process (compute phase), repeat elements are identified and masked. These elements constitute a significant portion of the genome and can be of two types: high-complexity repeats, which often encode viral proteins; and low-complexity repeats, which are segments of C's or G's or repeated dinucleotides. Unless these elements are effectively masked, annotation will have portions of transposons and viruses. The external program used to identify and mask these repeats is RepeatMasker, which align, through the use of another external program BLAST, those repeats with transposon and viral sequences that encodes proteins from an internal library, or any other repeat library provided by the user. The high-complexity repeats are hard-masked, which means that the nucleotide sequence is replaced with the letter N, which prohibits, in the next steps, the alignment of that region. The low-complexity repeats are soft-masked, which consists in replacing the uppercase nucleotide sequence by a lowercase sequence, so that they don't lose any sequence information. Still in the compute phase, the input proteins and ESTs/mRNAs sequences are aligned with the genomic sequence, through the

#### 4. New Modules

---

use of BLAST. Then in the filter/cluster phase these alignments are filtered based on their scores and percent identities and then clustered against the genomic sequence. After that, in the polish phase, BLAST alignments are realigned using Exonerate, which takes into account splice-sites, providing a useful information about splice donors and acceptors.

In the next step, synthesis phase, the data from the previous steps are synthesized in order to generate evidences for annotations. For that, ESTs presumably from the same alternatively spliced transcript are identified by comparing the coordinates of each sequence alignment on the genomic sequence. Then, the protein alignments whose coordinates are coincident with each EST splice form are identified. When these set of EST and protein alignments are identified, the upstream and downstream region of the alignment are labeled as possible intergenic regions, those between exons as putative introns and those that are aligned with proteins as putative translated sequence. Afterwards, a score for these labels based upon the percentage of similarity and type of alignment is generated and passed to SNAP, that will modify its internal Hidden Markov Model (HMM) to generate gene predictions. If these alignments are not available, SNAP uses its *ab initio* prediction. In the final phase, annotation phase, all the data from the previous steps is processed in order to produce the annotations. This is done by comparing all the SNAP prediction against all the ESTs and mRNAs that are coherent with the prediction and then altering them to include these regions.

Relatively to the output, it is a GFF3 format file [18], which can be directly imported into genome browsers and databases that adhere to Sequence Ontology and GMOD standards, such as Apollo, GBrowse, JBrowse or Chado. In this file, gene annotations and evidence alignments supporting those annotations are presented. As for the evidence alignments, these can have different sources, which include:

- BLASTN alignments of EST evidence, the *BLASTN* source.
- BLASTX alignments of protein evidence, the *BLASTX* source.
- TBLASTX alignments of EST evidence from closely related organisms, the *TBLASTX* source.
- Polished EST alignments from Exonerate, the *EST2Genome* source
- Polished protein alignments from Exonerate, the *Protein2Genome* source.
- SNAP *ab initio* gene predictions, the *SNAP*, the source.
- GeneMark *ab initio* gene predictions, the *GENEMARK* source.
- Augustus *ab initio* gene predictions, the *Augustus* source.
- FGENESH *ab initio* gene predictions, the *FgenesH* source.
- RepeatMasker identified repeats, the *Repeatmasker* source.



Other outputs are the transcripts and protein sequence files in FASTA format for both annotations and gene predictions.

After the annotation, the question how to store the results was raised. Instead of creating new tables on the existing database to manage annotation information, we decided to use the Chado database, which is already prepared to save the annotation from MAKER.

### 4.1.1 Use Cases and Pipeline

A *use case* [47] is a means of specifying required usages of a system. It consists of a subject, upon which the action unfolds; an actor, which is a user or any other system that interacts with the subject; and a specification of the required behavior of the subject in a scenario of interaction with the actor. For the annotation module, two use cases were defined. The diagrams presenting in Figure 4.2 and Figure 4.2 indicate the use cases concerning the *annotation run manager* and the *curation manager*, respectively.

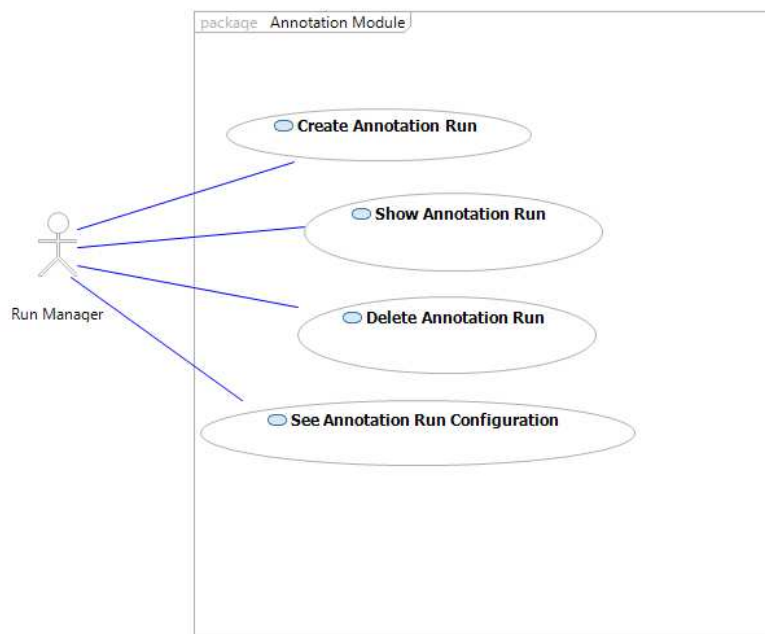


Figure 4.2: Run management use case diagram.

Concerning the management of the annotation procedures (Figure 4.2), we need to distinguish the following interaction scenarios:

- Show Annotation Run: In this scenario the user wants to see all the annotation runs that have been executed, along with their status. The system displays all the runs, one per line.
- Create Annotation Run: In this scenario the user wants to create an annotation run. The system displays a form to be filled with basic information on the annotation run. After fulfilled the system can create an annotation run.

## 4. New Modules

---

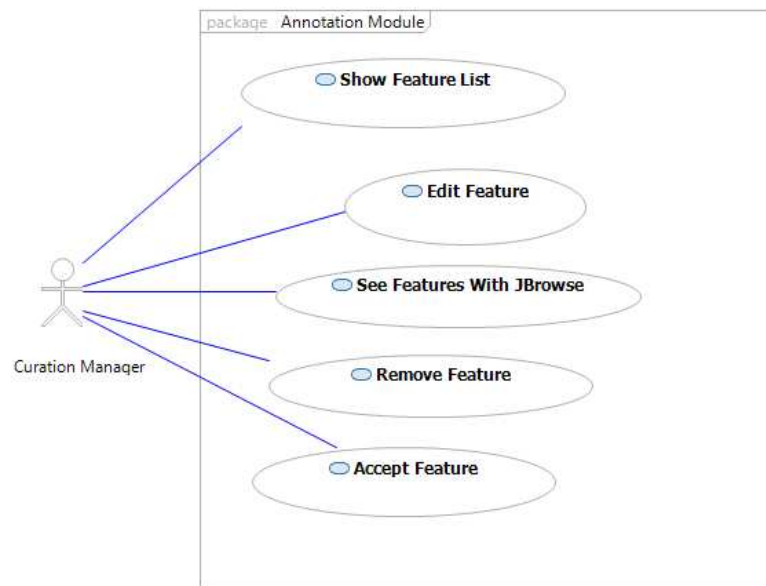


Figure 4.3: Curation management use case diagram.

- See Annotation Run Configuration: In this scenario the user wants to see configuration of an existing annotation run. The system displays the annotation run configuration.
- Delete Annotation Run: In this scenario the user wants to remove an annotation run. The system removes the annotation run after confirmation.

As for the annotation curation, we need to distinguish the following interaction scenarios:

- Show Feature List: In this scenario the user wants to see all the features of an annotation run. The system displays the list of features.
- Edit Feature: In this scenario the user wants to edit a existing feature. The system displays a form with the feature data to be edited. After edited the system updates the feature.
- Remove Feature: In this scenario the user wants to remove a feature. The system removes the feature.
- Accept Feature: In this scenario the user wants to validate a feature. The system associates effectively the feature to the respective sequence.
- See Features with JBrowse: In this scenario the user wants to see the features with the visualization tool JBrowse. The system displays the features on JBrowse.

Taking into consideration the use cases above, an annotation pipeline was created for the GEDI system, which is represented in Figure 4.4. This pipeline represents a recipe of what a user has to do to annotate a sequence. First, the user needs to set up the annotation configuration,

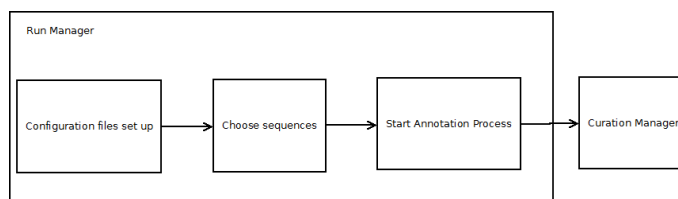


Figure 4.4: GEDI annotation pipeline.

next it needs to choose the sequences that he wishes to annotate and then the MAKER software is executed. After the annotation run is finished the user proceeds to the curation process.

The implementation of the use cases described above required the development of three web interfaces: an interface to visualize the list of annotation runs, an interface to create a annotation run, and another to visualize the annotation run results. These interfaces will be described in more detail in the next sections.

### 4.1.2 Annotation Runs List

In this section, the “Annotation Runs List” web page is described (Figure 4.5). This page is used to visualize the list of annotation runs.

This page is composed by a table with a list of annotation runs and a button on top of the table to create a new annotation. The table is composed by the columns: “Run Id”, with the unique identifier of the run; “Run Name”, with a human-readable name for the run; “Configuration”, with a link to the configuration page of the annotation run; “Status”, with an indication of the progress status of the run, which can be *completed*, *in progress* or *failed*; “Results”, with a link to the results page, in the case of completed annotation runs; and finally, “Remove”, where the user can mark runs to be deleted.

When an annotation run is launched, its status is *in progress*, and so there is not a link for the results in the “Results” column. After being completed, the “Status” column needs to be changed. In order to do that, the system checks the progress and automatically refreshes the page every minute.

### 4.1.3 Annotation Run Configuration

In this section, the “Annotation Run Configuration” web page is described . This page serves to create an annotation run or see an annotation run configuration. In order to visualize this page, the user needs to press the “create new annotation” button in the “Annotation Run List” page or to press the “Configuration” link in the table from the “Annotation Run List” page.

When creating a new annotation run, several fields need to be filled. These fields are: “name”, the name a user wants to give to an annotation run; “organism type”, the type of organism to be annotated, prokaryotic or eukaryotic; “EST file”, an existing or newly uploaded EST file in FASTA

## 4. New Modules

Annotation Run List

Create new annotation

Run Id	Name	Configuration	Status	Results	Remove
26	Egrandis	<a href="#">Configuration</a>	completed	<a href="#">Result</a>	✗
27	Eglobulus	<a href="#">Configuration</a>	completed	<a href="#">Result</a>	✗
28	Eglobulus2	<a href="#">Configuration</a>	completed	<a href="#">Result</a>	✗
29	Egrandis2	<a href="#">Configuration</a>	completed	<a href="#">Result</a>	✗
30	Eglobuluse	<a href="#">Configuration</a>	completed	<a href="#">Result</a>	✗

Figure 4.5: Annotation runs list page.

Annotation Run Configuration

[Back to Annotation Run List](#)

Name:

Organism type:

EST file:

Or

Select file:

Protein file:

Or

Select file:

Predictor:  est2genome  protein2genome  snap (Eukaryotic)  Genemark (Prokaryotic)

snap hmm file:

Or

Upload snap hmm file:

Select Project:

Figure 4.6: Annotation run configuration page - top.

format, only necessary if the *est2genome* option is selected in predictor field; “protein file”, an existing or newly uploaded protein file in FASTA format, only necessary if the *protein2genome* option is selected in predictor field; and “predictor”, the methods that can be used to generate the annotations. The annotation methods currently available are the *est2genome* (a MAKER program which allows to EST alignments to become gene annotations), *protein2genome* (a MAKER program which tries to construct gene models directly from protein alignments), *SNAP* (a gene predictor for eukaryotic organisms), or *GeneMark* (a gene predictor to prokaryotic organisms). If the user chooses the SNAP program, there is a field where he can upload a HMM file or select one already added to the database.

At the bottom of this page, Figure 4.7, the user needs to choose the sequence to be annotated. For that, the user needs to select a project and a clone, to retrieve its sequence.

Figure 4.7: Annotation run configuration page - bottom.

After completing all the required fields, the user simply needs to press the “run MAKER” button to create an annotation run. If all the fields were correctly filled, then the user is redirected to the “Annotation Run List” page, otherwise an error message will appear.

#### 4.1.4 Results List

In this section, the “Results List” web page is described. This page serves to visualize the annotation run results. For that, the run needs to be successfully finished and the user needs to press the “Result” link in the annotation runs table from the “Annotation Runs List” page.

This page presents the results of an annotation run in the form of a table and it is where here

## 4. New Modules

The screenshot displays the 'Annotation Run Results Page'. At the top, the user's email 'Hello.andrefgomes@hotmail.com!' and a 'Logout' link are visible. Below the navigation bar, there is a 'Result List' section with a 'Back to Annotation Run List' link. The main content is a table of annotation results. The table has the following columns: Id (with an 'Edit' link), Name, Source, Type, Start, End, Score, Strand, Phase, and Status (with an 'Invert' link). The table contains several rows of data, including features like 'match\_part', 'polypeptide', 'exon', and 'mRNA'. Below the table are 'Remove Select' and 'Accept Select' buttons. On the right side, there is an 'Edit Annotation' form with fields for Name, Source, Type, Start, End, Score, Strand, and Phase, and a 'Save' button. The form also includes a 'Description' field with a 500-character limit.

Id (Edit)	Name	Source	Type	Start	End	Score	Strand	Phase	Status	All Invert
55708 (Edit)	9616	blastn	match_part	158314	158595	282.0	1			<input type="checkbox"/>
55807 (Edit)	polypeptide-auto55897	maker	polypeptide	158620	159012	0.0	1			<input type="checkbox"/>
55896 (Edit)	maker-18-1-BLASTX-gene-1.17-mRNA-1.cds:78	maker	exon	158620	159012	0.0	1	0		<input type="checkbox"/>
55713 (Edit)	maker-18-1-BLASTX-gene-1.17-mRNA-1.exon:78	maker	exon	158620	159012	393.0	1			<input type="checkbox"/>
55826 (Edit)	maker-18-1-BLASTX-gene-1.17-mRNA-1	maker	mRNA	158620	159012	0.0	1			<input type="checkbox"/>
55192 (Edit)	maker-18-1-BLASTX-gene-1.17	maker	gene	158620	159012	0.0	1			<input type="checkbox"/>

Figure 4.8: Annotation Run Results Page.

the curation process begins. In the bottom of the page, there are two buttons, “Accept Selected” button and “Remove Selected”, here a user can save or remove a selected feature, respectively. Saving a feature will effectively associate it to the respective sequence in the database. Still on this page, the user can edit a feature by pressing the “Edit” button in the table and then editing the feature information on the displayed form. In order to save the edited feature, the user needs to press the “save” button on the form.

## 4.2 Markers Module

Markers are used to build the genome map by identifying where the genes are in the chromosome and by identifying other features. Attaching a marker to a sequence can be viewed as a form of annotation, so a marker is another feature of the annotation process and the reason why this module had been developed.

Markers are not only used in the annotation process. For genome projects that use the BAC-to-BAC sequencing method and are only interested to study parts of a genome, markers can be used to identify the clones that have the regions of interest.

### 4.2.1 Use Cases

In the case of the markers module, the use case diagram shown in Figure 4.9 indicates the use cases (interaction scenarios) concerning the *marker manager*.

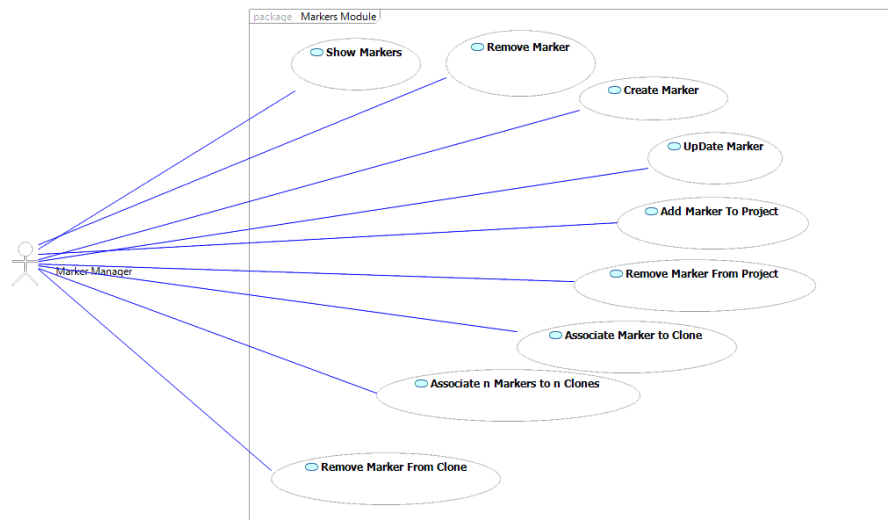


Figure 4.9: Marker module use case.

Regarding the management of markers, we need to distinguish the following interaction scenarios:

- **Show Markers:** In this scenario the user wants to see all the markers in the system. The system displays a list of markers in the system.
- **Create Marker:** In this scenario the user wants to create an marker. The system displays a form to be filled with basic information on the marker. After fulfilled the system can create an marker.
- **Remove Marker:** In this scenario the user wants to remove a marker from the system. The system removes the marker after confirmation.
- **Update Marker:** In this scenario the user wants to edit a marker information. The system displays a form with the marker data to be edited. After edited the system updates the marker.
- **Associate Marker to Project:** In this scenario the user wants to associate a marker to a project. The system associates a marker to a project after a project and a marker being selected.
- **Remove Marker from Project:** In this scenario the user wants to dissociate a marker from a project. The system dissociates a marker to a project after a project and a marker being selected.
- **Associate Marker to Clone:** In this scenario the user wants to associate a marker to a clone. The system associates a marker to a clone after a clone and a marker being selected.

## 4. New Modules

---

- Associate  $n$  Markers to  $n$  Clones: In this scenario the user wants to associate  $n$  clones to  $m$  markers. The system associates  $m$  marker to  $n$  clone after a csv file being provided.
- Remove Marker from Clone: In this scenario the user wants to dissociate a marker from a clone. The system dissociates a marker to a project after a clone and a marker being selected.

The realization of the use cases above required the development of three web interfaces: an interface to visualize the list of markers, an interface to manage marker-project associations, and another to manage marker-clone associations. These interfaces will be described in more detail in the next sections.

### 4.2.2 Markers List

In this section, the “Marker List” web page is described. This page serves to visualize the list of markers. In order to reach this page, an administration user needs to select, in the menu “Admin”, the “Edit” option in the group “Markers”.

The “Markers list” page, represented in Figure 4.10, is composed of a table showing the list of markers that are in the system, and three forms. Each form is associated with a table in the system. The top form, “Edit Marker Type”, serves to edit a marker type and is associated to the *marker\_type* table in the database. The one in the middle, “Edit Marker”, serves to edit a marker and is associated to the *marker* database table. The one in the bottom, “Edit Marker Synonyms”, serves to edit a marker synonym, and is associated to the *marker\_synonyms* table in the database.

In the top form, the user can create, remove or update a marker type. This form has two fields: “Id”, the identifier of the marker type, which cannot be edited; and “Marker Type Name”, which corresponds to the name of the marker type.

To create a marker type, if no marker type is selected, the user only needs to fill the field “Marker Type Name” and press the “Save” button. Then the system will generate an automatic id number for the marker type and save it in the database. If a marker type is already selected, the user needs to press the “New” button to clear the form and then fill it with the new marker type information. The system does not allow to create a marker type with a name already in the database or a blank name.

To update an existing marker type, the user first needs to select it from the existing combo box by choosing it and pressing the “Select” button. Then the system will display the marker type information in the form that can be edited by the user. After that, the user needs to press the “Save” button in order to update it.

In order to remove a marker type, the user needs to proceed just like the case before, but instead of pressing the “Save” button he should press the “Remove” button. The system will then



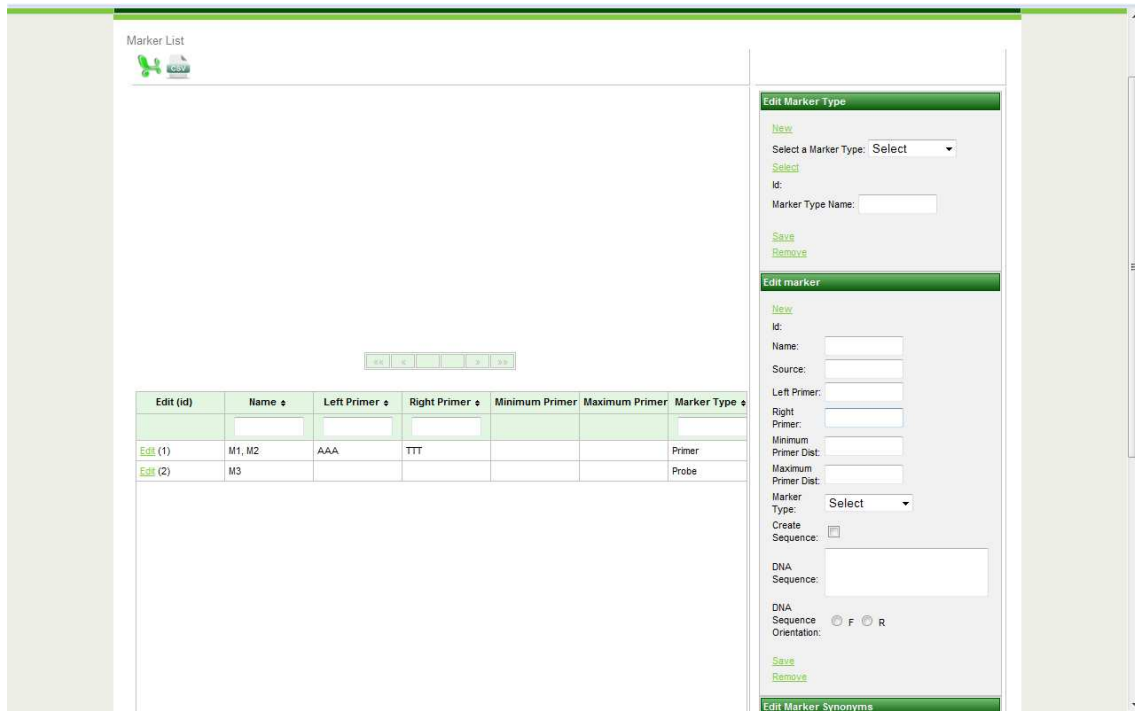


Figure 4.10: Markers list page – top.

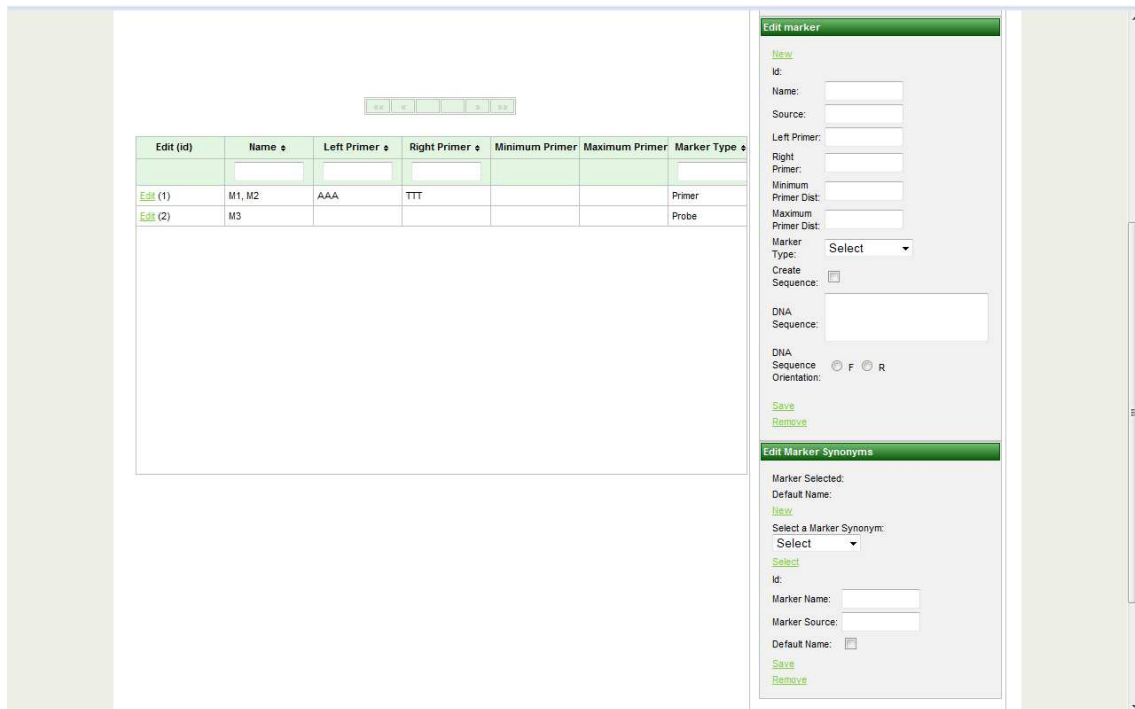


Figure 4.11: Markers list page – bottom.

## 4. New Modules

---

ask for removal confirmation and once the user confirms the deletion, the system will remove it. The system will only remove the marker type if it is not associated to any marker.

In the middle form, there are eleven fields: "Id", the identifier of a marker, which can not be edited; "Name", the name of a marker; "Source", the name of the source that built/discovered the marker; "Left Primer", the sequence of the left primer of a marker; "Right Primer", the sequence of the right primer of a marker; "Minimum Primer Dist", the minimum separation distance between the left and right primers; "Maximal Primer Dist", the maximal separation distance between the left and right primers; "Marker Type", the marker type of a marker; "Create Sequence", a check box that indicates if a sequence is to be added to the marker; "DNA sequence", the marker sequence; and "DNA sequence orientation", which indicates if the sequence is in the sense or anti-sense orientation.

To create a marker, whenever no marker is selected, the user only needs to fill the field "Name" and the field "Marker Type", being the others optional, and press the "Save" button. If there is already a marker with the same name, the system does not allow saving it. If the user wants to create a marker but there is already a marker selected, he must first press the "New" button to clear the form.

To update a marker, the user needs to select a marker by pressing the "Edit" button in the table. Then the fields "Name" and "Source" from the "Edit Marker" form will disappear but the remaining information will be displayed and the user is allowed to edit the information. After editing the marker data, the user needs to press the "Save" button in order to update the marker in the database. And once again, if there is already a marker with the same name, the system does not allow the update.

To remove a marker, the user needs to proceed similarly to the update process, but instead of pressing the "Save" button, he should press the "Remove" button. The system will then ask for removal confirmation and once the user confirms the deletion, the system will remove it. If the marker is associated with a project, the system will not allow its removal.

In the bottom form, the user can edit the synonyms of a marker by selecting it in the table. In this form, there are seven fields: "Marker Selected", the id of the selected marker which cannot be edited; "Default Name", the default name of the marker which cannot be edited; a combo box with the marker synonyms list; "Id", the identifier of the marker synonym which cannot be edited; "Marker Name", the name of the marker; "Marker Source", the name of the source that built/discovered the marker; and the "Default Name" check box, where the user can choose the default name of a marker.

To create a new name for a marker, the user first needs to select a marker from the table and then fill the "Marker Name" field and, optionally, the "Marker Source" field. If the user wishes to set the name as the default name, he needs to check the "Default Name" check box. The created name cannot be in the database, otherwise the system does not allow the creation of the new

marker name.

To edit a marker synonym, the user first needs to select a marker by pressing the “Edit” button in the table and then select a marker name from the combo box by pressing the “select” button. Afterwards, the user can edit the marker synonym information and save it by pressing the “Save” button. Once again the changed name cannot be in the system, if so, the system will not save the changes.

To remove a marker synonym, the user needs to proceed similarly to the edit process, but instead of pressing the “Save” button, he should press the “Remove” button. The system will then ask for removal confirmation and once the user confirms the deletion, the system will remove it. If the selected *marker synonym* is the default name, the system does not allow to remove it.

### 4.2.3 Project Markers List

In this section, the “Project Markers List” web page is described. This page serves to manage project–markers associations, and it was created because different types of organism can have different markers, so it was necessary to associate the markers to an organism. In order to visualize this page, an administration user needs to select the “Projects Markers” option of the group “Markers” in the menu “Admin”.

This page, shown in Figure 4.12, is composed of a table with the list of markers per project, and a form. The form represents the *markers\_projects* table of the database.

The screenshot displays the 'Project Markers List' interface. At the top, there is a navigation menu with options: Site, User, Tools, Admin, Sequence Deposit, and Demo. The user is logged in as 'Hello, andrefgomes@hotmail.com!' with a 'Logout' link. Below the menu, there is a 'Marker List' section with a 'Select Project: Select All' dropdown and navigation buttons. The main table has the following data:

Edit (id)	Name	Marker Type	Marker Projects
<a href="#">Edit (1)</a>	M1	Primer	pro_test, h
<a href="#">Edit (2)</a>	M3	Probe	

On the right side, there is a 'Marker' sidebar with a 'New' button, 'Marker Selected:', 'Marker Name:', 'Select Projects:', and checkboxes for 'pro\_test' and 'h', followed by a 'Save' button. The footer contains the text 'Copyright Gedi 2009' and logos for 'inescid', 'tsboa', and 'IBET Instituto de Biologia Experimental e Tecnológica 1883'.

Figure 4.12: Project Markers Page.

In the table, there are four columns: the “Edit” column, where the user can select a marker

## 4. New Modules

to edit in the “Marker” form; the “Default Name” column which presents the default names of the markers; the “Marker Type” column which presents the marker type of the markers; and the “Projects” column which contains the projects to which each marker is associated. Above the table, there is a combo box that allows filtering the table by projects.

The “Marker” form is composed of three fields: “Selected Marker”, the marker select from the table; “Marker Name”, the default name of the marker selected; and the list of projects with check boxes. When the user wishes to associate a marker to a project, he first needs to select a marker from the table, and then system will display the information relative to the marker in the form. The user can select the projects that he wishes to associate or dissociate to the selected marker by checking or unchecking the check boxes. In order to save the changes, the user needs to press the “Save” button. If a marker is associated to a clone from a determined project and the user wishes to remove the marker from that project, the system will not allow it.

### 4.2.4 Clone Markers List

In this section, the “Clone Markers List” web page is described. This page serves to manage marker–clone associations. In order to visualize this page, an administration user needs to select the “Clone Markers” option of the group “Markers” in the menu “Admin”.

This page, shown in Figure 4.13, is composed of a table with the list of associations between clones and markers, and two forms. The forms manage the *clone\_marker* table from the database.

Clone Marker List

Select Project: Select  
Select Library: Select  
Select Plate: Select  
Select Clone: Select  
Clone Selected:

Select Marker Type: Select  
Select Marker: Select  
Marker Selected:

[Remove Selected](#)

Select	Edit (id)	Clone Id	Clone Name	Marker Id	Marker Name	Technique	Description
<input type="checkbox"/>	<a href="#">Edit (4)</a>	1	Eglobulus	1	M1		

[Upload Clone Markers](#)

[Download Template](#)

Upload File:

[Add](#)

[clear all](#)

[Upload csv File](#)

[Edit clone Marker](#)

[New](#)

Clone Marker Id: Select  
Select Project: Select  
Select Library: Select  
Select Plate: Select  
Select Clone: Select

Figure 4.13: Clone Markers Page – top.

Remove Selected

Select	Edit (id)	Clone Id	Clone Name	Marker Id	Marker Name	Technique	Description
<input type="checkbox"/>	Edit (4)	1	Eglobulus	1	M1		

Copyright Gedi 2009






Figure 4.14: Clone Markers Page — bottom.

The table is composed of eight columns, the first one is a column with check boxes used to select multiple lines for deletion. The second column contains “Edit” buttons used to select a line from the table and pass its information to the “Edit Clone Marker” form. The third and fourth columns are the “Clone Id” and the “Clone Name”, respectively. The fifth and sixth columns are the “Marker Id” and the “Marker Name”, respectively. The seventh column indicates the technique that was used to associate a marker to a clone, and the last column contains a brief description or comment on the association. The user can filter the displayed information by clone or by marker, through the combo boxes on top of the page.

The top form, “Upload Clone Markers”, is used to associate  $n$  clones to  $m$  markers through a comma separated values (CSV) spreadsheet file. The user can download a template spreadsheet (Excel) file by pressing the “Download Template” button. This spreadsheet is composed of three columns, the first of which contains marker names, the second contains clone standard names, and the third the techniques used to associate a marker to a clone. The template contains an example line which needs to be removed. Prior to submitting, the user needs to save the file as CSV, and then upload it by pressing the “Add” button followed by the “Upload CSV File” button.

In the “Edit Clone Marker” form, the user can associate a single marker to a single clone. In this form, there are nine fields. The first field is “Clone Marker Id”, which cannot be edited. The following fields are “Select Project”, “Select Library”, “Select Plate” and “Marker Type”. these are filtering fields that can optionally be used to easily select a marker or a clone to associate. Next, there are the mandatory fields to create an association, namely “Select Clone” and “Marker

#### 4. New Modules

---

Id”, and finally “Technique” and “Description fields”, which are optional. After selecting a clone and a marker, the user presses the Save button to create an association between them, if that association does not already exist. To remove an association, the user must select it from the table and then press the “Remove” button. As usual, the system will ask for confirmation prior to removal. To clear the form, the user just needs to press the “New” button.

# 5

## Case Study

### Contents

---

5.1 <i>Eucalyptus globulus</i> . . . . .	40
5.2 <i>Eucalyptus grandis</i> . . . . .	44

---

### 5.1 *Eucalyptus globulus*

In this chapter, we report on the results of our case study, which consisted in using the implemented annotation pipeline on the chloroplast genome of the *Eucalyptus globulus*. To assess the quality of our findings, we compare them to those obtained with the GeneMark annotation suite [10].

The chloroplast genome of *E. globulus* was released in 2005. It is a genome with 160 286 bp with an inverted repeat (IR) region of 26 393 bp separated by a large single copy (LSC) region of 89 012 bp and a small single copy (SSC) region of 18 488 bp with a GC-content of 36.9%. The initial release contained 128 documented genes coding for 30 transfer RNAs, 4 ribosomal RNAs and 78 proteins. Currently, there are 135 documented genes, coding for 45 structural RNAs, with 8 of them containing one intron, which gives a total of 53 exons, and 90 genes coding for proteins, with 11 of them containing one intron and 4 containing two introns, which gives a total of 108 exons (one exon is shared by two genes). The genome is available from GenBank with accession id AY780259.

#### 5.1.1 Pipeline Configuration

The input provided to the annotation pipeline consisted of the chloroplast genome sequence, retrieved from GenBank, as well as an EST and a protein file. The EST file consisted of 90 CDS, 37 tRNA and 8 rRNA of the *E. globulus*. It was obtained from the Chloroplast Genome Database [48]. The protein file consisted of 90 proteins from the *E. globulus*, also obtained from the Chloroplast Genome Database.

The parameters for the algorithm, mainly for MAKER, were set to their default values, except for those modifiable through the “Annotation Run Configuration” web page (chapter 4), namely the predictor and organism type parameters. For the predictor parameter, the “est2genome” and “protein2genome” options were used. For the organism type, we used the “prokaryotic” option.

The “prokaryotic” option was chosen as organism type because plastids, in particular chloroplasts, are commonly thought to have prokaryotic origin. Indeed, according to the endosymbiotic theory, these organelles evolved from a prokaryotic cell that was originally ingested as food by the eukaryotic cell, and somehow these prokaryotic cells were not digested and became part of their hosts. Since they have prokaryotic origin, their structure remains similar to the prokaryotic cells; they have their own DNA, which is circular and with a single strain, they don't have a nucleus and they have their own population of ribosomes similar to the prokaryotic ribosomes [49]. However, with evolution, the chloroplast gene expression machinery changed, and it became distinct from prokaryotic, eukaryotic and phage [50]. Nevertheless, it still more similar to the prokaryotic gene machinery, hence choice of prokaryotic organism type.

The “est2genome” and “protein2genome” option were chosen because, since we supply a



ESTs and proteins file from *E. globulus* and we wanted these programs generate gene annotations from the ESTs and proteins alignments.

The GeneMark suite used in our evaluation was run with a heuristic model retrieved from the software web page corresponding to the GC-content of the chloroplast genome.

### 5.1.2 Measures of performance

The metrics used to analyze the annotation pipeline performance were the sensitivity and the specificity [51]. The sensitivity is given by

$$SN = \frac{TE}{AE}, \quad (5.1)$$

and specificity is defines as

$$SP = \frac{TE}{PE}, \quad (5.2)$$

where  $AE$  is the number of annotated exons that are documented,  $PE$  is the number of predicted exons by the gene predictor, and  $TE$  is the number of true exons, i.e. the number of predicted exons that are correctly annotated, meaning that both boundaries of the predicted exon match with an annotated exon.

### 5.1.3 Results

The results were divided into two main categories, “RNA” prediction and “Protein” prediction. This was done because heuristic models, for the GeneMark, do not include parameters for structural RNAs binding sites.

The results from the performance of the annotation pipeline, as well as GeneMark are represented in Table 5.1.

Table 5.1:  
Performance of the annotation pipeline (AP) and GeneMark on the chloroplast genome of *E. globulus*.

	AE	PE	TE	PCE	OE	NDE	FE	SN	SP
AP RNAs	53	37	35	2	0	16	0	0.66	0.95
AP Protein	108	89	59	30	0	19	0	0.55	0.66
GeneMark Protein	108	115	36	33	10	26	36	0.33	0.31
AP	161	89	59	30	0	72	0	0.37	0.66
AP with evidence	161	136	103	33	0	25	0	0.64	0.76

AE stands for annotated exons, PE stands for predicted exons and TE stands for true exons. Partially corrected exons (PCE) is the number of predicted exons with only one boundary matching that of an annotated exon. Overlap exon (OE) is the number of predicted exons with no boundary exactly matching that of an annotated exon, but still having some overlap. Non-detected exons (NDE) is the number of annotated exons that were not annotated by any predicted exon. False exons (FE) is the number of predicted exons that don't overlap any annotated exon. SN stands for sensitivity and SP stands for specificity.

## 5. Case Study

---

The first line, corresponds to the genes coding for structural RNAs. Our annotation pipeline did not identify any gene coding for structural RNAs, the results obtained were from evidence alignments that it produced through *BLASTN* source as an *expressed\_sequence\_match* (a match to an EST sequence) type. So for the RNAs results we considered a *expressed\_sequence\_match* evidence, as a PE, predicted exon.

By analyzing the first line, we notice that 16 exons were not detected, which made the SN value low. These exons were all from the 8 genes containing one intron and they were not detected because it was considered the input sequence from a prokaryotic organism, therefore it didn't take splice sites into account.

The second line, corresponds to the genes coding for proteins. Here, the exons predicted by the annotation pipeline are considered as PE. We notice that it didn't identify 19 exons, with 9 of them being from genes with one intron or more. Once again, the main reason for the annotation pipeline being unable to identify these exons was because prokaryotic organisms don't have introns. Regarding the other 10 exons, 9 of them had some information from *BLASTN* source or *BLASTX* source or both. The reason why it didn't annotate these exons is possibly because it gives a low score to these evidences. If some filtering parameters were changed, it could have detected these exons. The low values of SN and SP are mainly because of the high number of partially correct exons that weren't correctly annotated.

The third line is relative to the results obtained with GeneMark. If we compare these results to those of the second line, we can easily see that MAKER outperforms GeneMark, with a sensitivity of 0.55 against 0.33 and a specificity of 0.66 against 0.31, suggesting that our annotation pipeline, using MAKER, is indeed a better tool to annotate the chloroplast genome of *E. globulus*.

The fourth line considers the RNAs as undetected exons, which decreases, as expected, the sensitivity from 0.55 to 0.37.

The fifth line considers the evidences from *BLASTN* and *BLASTX* source as PE. This was done to demonstrate the results after a fast curation of the results, which raise the sensitivity value of 0.37 to 0.64 and the specificity value of 0.66 to 0.76.

For a visual comparison of the results, a preview of the results obtained by the annotation pipeline and NCBI annotation is presented in Figure 5.1 and Figure 5.2, respectively.

In Figure 5.1 is possible to visualize the annotation retrieve from the annotation pipeline by the genome visualization tool Apollo. The annotations are in the blue area and the evidences alignments are in dark area, the sequence is represented in the white area. In Figure 5.2 we can see in the top panel a brief glance of the entire sequence and in the bottom panel a more detail view of the annotations, here a gene is represented by the color green, the mRNAs by the color blue and the coding regions by the color red.

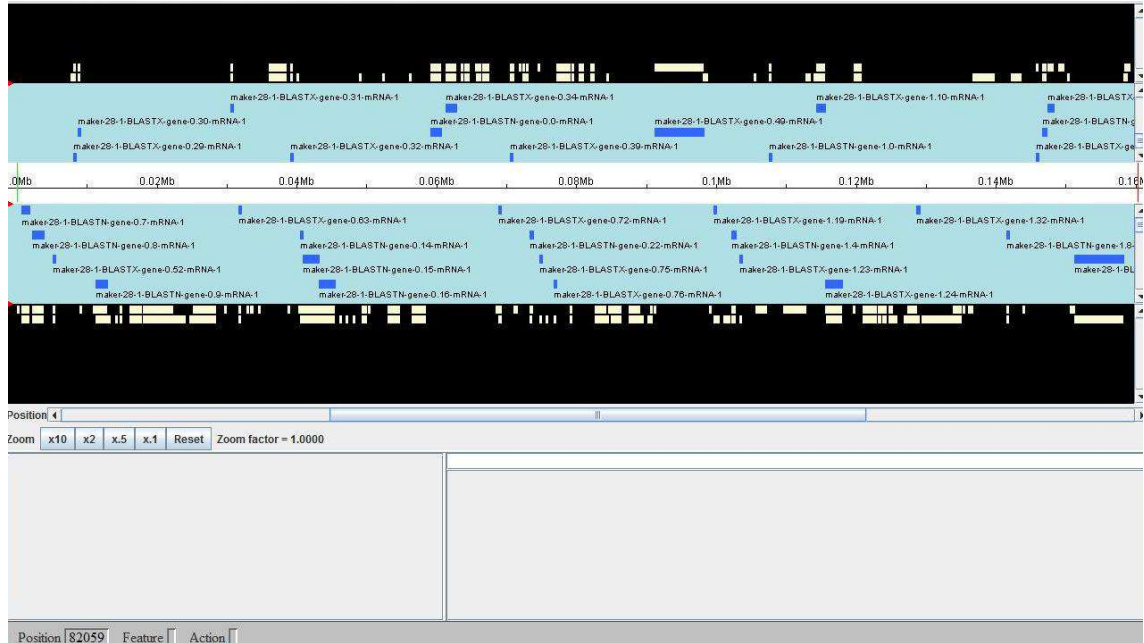


Figure 5.1: MAKER results for the chloroplast genome of *E. globulus* using Apollo.



Figure 5.2: NCBI annotation for the chloroplast genome of *E. globulus*.

### 5.2 *Eucalyptus grandis*

A further test of the annotation pipeline was done with the chloroplast genome of the *Eucalyptus grandis*. In particular, we were interested in assessing the impact of annotating a genome with ESTs and proteins from a different species, albeit from the same genus.

The chloroplast genome of *E. grandis* was sequenced and annotated by Paiva and colleagues [52] and available from the GenBank database under the accession id HM347959. It is a genome with 160 142 bp with an inverted repeat region of 26 390 bp separated by a large single copy region of 88 883 bp and a small single copy region of 18 457 bp and with a GC-content of 36.9%. This genome has 138 genes coding for 50 structural RNAs and 88 for proteins, with 8 of them containing one intron and 4 containing two introns, what gives a total of 103 exons (an exon is shared by two genes).

For this test it was used the same parameter values from the test before.

#### 5.2.1 Results

The obtained results are presented in Table 5.2. If we compare them to the results from Table 5.1, we verify that there is not much difference between them, except for the RNAs results where the sensitivity and specificity is better for the *E. globulus*. The main reason why the results are similar, even using ESTs and proteins from other species, is because the species belong to the same *genus*, what makes their genomes very similar.

Table 5.2: Performance of the annotation pipeline on the chloroplast genome of *E. Grandis*.

	AE	PE	TE	PCE	OE	NDE	FE	SN	SP
AP RNAs	50	37	31	6	0	13	0	0.62	0.84
AP Protein	103	84	57	25	2	19	4	0.55	0.68
AP	153	84	57	25	2	69	4	0.37	0.68
AP with evidence	153	133	99	31	3	32	4	0.65	0.74

For a visual comparison of the results, a preview of the results obtained by the annotation pipeline and NCBI annotation is presented in Figure 5.3 and Figure 5.4, respectively.

In Figure 5.3 is possible to visualize the annotation retrieve from the annotation pipeline by the genome visualization tool Apollo. The annotations are in the blue area and the evidences alignments are in dark area, the sequence is represented in the white area. In Figure 5.2 we have on the top bar a representation of the sequence and in the below areas we have the genes represented by the blue color, the coding regions by the rose color, the tRNA by the green color, the rRNAs by the black color and the repeat regions by the gray color.

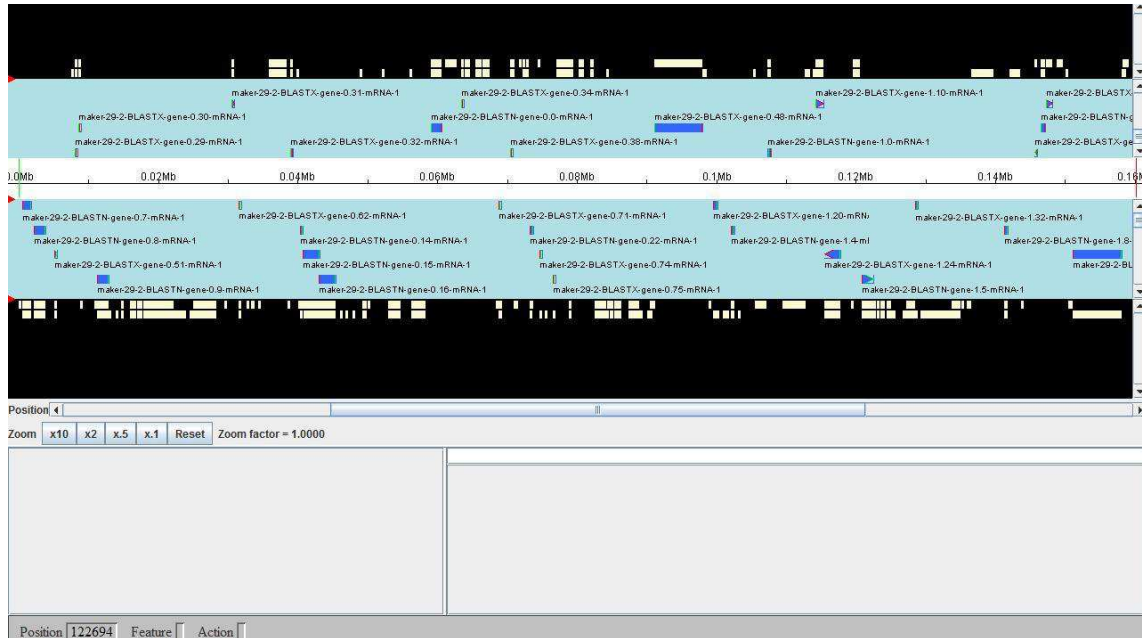


Figure 5.3: MAKER results for the chloroplast genome of *E. grandis* using Apollo.

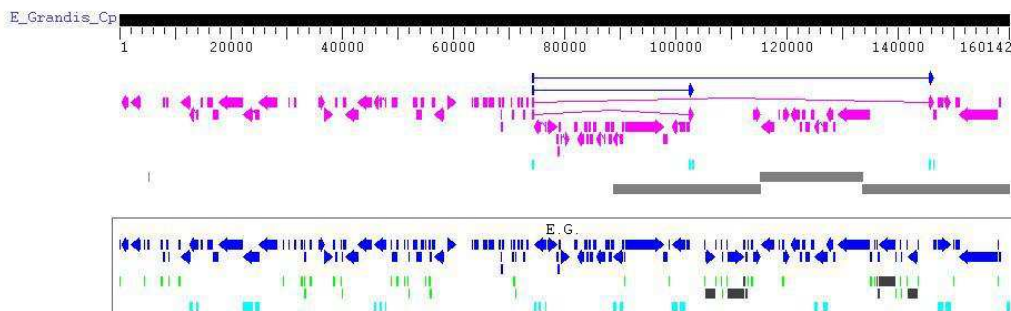


Figure 5.4: NCBI annotation for the chloroplast genome of *E. grandis*.



# 6

## **Conclusions and Future Work**

## 6. Conclusions and Future Work

---

This thesis consisted of extending a web information system, GEDI, so that an automatic annotation tool could be integrated in the system. The automatic annotation tool chosen was MAKER.

According to the literature, the performance of MAKER on eukaryotic organisms was proven to be almost at the same level as that of other gene finders, being Augustus [13] the one that presented better results, which is good because Augustus is one of the optional components of MAKER [53].

It was also our intention to test even further the performance of MAKER through the annotation pipeline, and to prove its versatility by testing it on the chloroplast genome of *E. globulus* and *E. grandis*. The results showed that our annotation pipeline obtained better results compared to GeneMark, a gene finder for all prokaryotic organisms, suggesting it is a better annotation tool. Regarding the performance on the chloroplast genome of *E. grandis*, the results were similar to those obtained with the chloroplast of *E. globulus*, showing that ESTs and protein sequences from sufficiently close species can be used without reducing significantly the performance.

According to the literature and the results obtained in this thesis, it can be concluded that MAKER is indeed a good option to be integrated into the annotation module, since it can yield good annotation results for prokaryotic organisms and it can be used as well to annotate any other type of organism. In addition, the ease to run and configure the tool, reinforces its choice. One setback is that it requires EST and protein database from the organism being annotated or from a related organism for more complete and sensible results. Moreover it is unable to identify genes coding for structural RNAs as genes, even with some evidence from the BLASTN.

With the choice of MAKER as an automatic annotation tool, it was possible to create a simple but effective annotation pipeline for the annotation module, making it a easy way to annotate the stored sequences in the GEDI system. The choice of Chado database to store the annotations also proved to be a good option, by facilitating the process of storing and managing the annotations results.

Since an automatic genome annotation is not 100% successful a manual curation of the results is required. The annotation module provides a initial process of curation by allowing the user to remove or accept the annotations results.

As future work, Augustus can be integrated into the annotation module in order to have better results for eukaryotic organisms. Other aspect that can possibly be extended, is the list of parameters made available for modification from the GEDI user interface. Since there were some missed exons in our case studies which could have been annotated if some parameters such as the filtering parameters were properly set. Nevertheless, the parameters that are currently available are sufficient to annotate a sequence.

An extra feature that can be implemented is to send a notification email to the user whenever an annotation run is finished. For a visualization tool of annotations it was installed JBrowse



---

which still in underdevelopment. An alternative visualization tool, that can be installed, for the administration interface, could be Apollo, which also allows the edition of annotation.

Finally, other complementary annotation tools can be integrated into the GEDI system, with the aim of providing more accurate and more complete information. For example, Blast2GO, can be integrated to provide functional annotation to the sequences, and DAS, to share and collate genomic annotation information.



# Bibliography

- [1] GenomeBiology, "Genome biology," Online Website, BioMed Central Ltd, July 2010. [Online]. Available: <http://genomebiology.com/2007/8/1/102>
- [2] B. L. Cantarel, I. Korf, S. M. C. Robb, G. Parra, E. Ross, B. Moore, C. Holt, A. Sánchez Alvarado, and M. Yandell, "Maker: An easy-to-use annotation pipeline designed for emerging model organism genomes," Genome Research, vol. 18, no. 1, pp. 188–196, January 2008. [Online]. Available: <http://dx.doi.org/10.1101/gr.6743907>
- [3] "Geneglobwq - scanning for candidate genes underlying a pulp yield qtl in eucalyptus globulus." [Online]. Available: <http://geneglob.inesc-id.pt/>
- [4] F. Griffith, "The significance of pneumococcal types," Journal of Hygiene, vol. 27, pp. 113–159, 1928.
- [5] O. T. Avery, C. M. MacLeod, and M. McCarty, "Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from Pneumococcus type III," Journal of Experimental Medecine, vol. 79, no. 2, pp. 137–158, February 1944. [Online]. Available: <http://view.ncbi.nlm.nih.gov/pubmed/11039746>
- [6] A. D. Hershey and M. Chase, "Independent functions of viral protein and nucleic acid in growth of bacteriophage," J. Gen. Physiol., vol. 36, no. 1, pp. 39–56, September 1952. [Online]. Available: <http://dx.doi.org/10.1085/jgp.36.1.39>
- [7] A. M. Lesk, Introduction to Genomics, 1st ed., O. U. P. Inc., Ed. Oxford University Press Inc., 2007.
- [8] GMOD, "Maker tutorial," Online Website, GMOD, August 2010. [Online]. Available: [http://gmod.org/wiki/MAKER\\_Tutorial](http://gmod.org/wiki/MAKER_Tutorial)
- [9] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with glimmer." Nucleic acids research, vol. 27, no. 23, pp. 4636–4641, December 1999. [Online]. Available: <http://dx.doi.org/10.1093/nar/27.23.4636>

## BIBLIOGRAPHY

---

- [10] A. V. Lukashin and M. Borodovsky, "Genemark.hmm: new solutions for gene finding," *Nucl. Acids Res.*, vol. 26, no. 4, pp. 1107–1115, February 1998. [Online]. Available: <http://nar.oxfordjournals.org/cgi/content/abstract/26/4/1107>
- [11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, October 1990. [Online]. Available: <http://dx.doi.org/10.1006/jmbi.1990.9999>
- [12] I. Korf, "Gene finding in novel genomes," *BMC Bioinformatics*, vol. 5, no. 1, pp. 59+, May 2004. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-5-59>
- [13] M. Stanke and S. Waack, "Gene prediction with a hidden markov model and a new intron submodel." *Bioinformatics (Oxford, England)*, vol. 19 Suppl 2, October 2003. [Online]. Available: <http://view.ncbi.nlm.nih.gov/pubmed/14534192>
- [14] Blast2GO, "Blast2go," Online Website, Bioinformatics and Genomics Department, August 2010. [Online]. Available: <http://blast2go.bioinfo.cipf.es/>
- [15] T. G. O. Consortium, "The gene ontology project in 2008," *Nucl. Acids Res.*, vol. 36, no. suppl.1, pp. D440–444, January 2008. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkm883>
- [16] "A.f.a. smit, r. hubley & p. green repeatmasker." [Online]. Available: <http://repeatmasker.org>
- [17] G. S. S. Slater and E. Birney, "Automated generation of heuristics for biological sequence comparison." *BMC bioinformatics*, vol. 6, no. 1, pp. 31+, 2005. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-6-31>
- [18] "So the sequence ontology project." [Online]. Available: <http://www.sequenceontology.org/gff3.shtml>
- [19] R. Dowell, R. Jokerst, A. Day, S. Eddy, and L. Stein, "The distributed annotation system," *BMC Bioinformatics*, vol. 2, no. 1, pp. 7+, 2001. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-2-7>
- [20] M. S. F.-K. K. S. N. S. H. G. T. Tateno Y, Imanishi T, "Dna data bank of japan (ddbj) for genome scale research in life science," *Nucleic Acids Res.*, vol. 30, pp. 27–30, 2002.
- [21] Kulikova, Tamara, Akhtar, Ruth, Aldebert, Philippe, Althorpe, Nicola, Andersson, Mikael, Baldwin, Alastair, Bates, Kirsty, Bhattacharyya, Sumit, Bower, Lawrence, Browne, Paul, Castro, Matias, Cochrane, Guy, Duggan, Karyn, Eberhardt, Ruth, Faruque, Nadeem, Hoad, Gemma, Kanz, Carola, Lee, Charles, Leinonen, Rasko, Lin, Quan, Lombard, Vincent, Lopez, Rodrigo, Lorenc, Dariusz, Mcwilliam, Hamish, Mukherjee, Gaurab, Nardone, Francesco, Pastor, M. P. Garcia, Plaister, Sheila, Sobhany, Siamak, Stoehr, Peter, Vaughan,

- Robert, Wu, Dan, Zhu, Weimin, Apweiler, and Rolf, "Embl nucleotide sequence database in 2006," Nucleic Acids Research, vol. 35, no. Supplement 1, pp. D16–D20, January 2007. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkl913>
- [22] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "Genbank." Nucleic acids research, vol. 36, no. Database issue, January 2008. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkm929>
- [23] C. J. Mungall, D. B. Emmert, and T. F. Consortium, "A chado case study: an ontology-based modular schema for representing genome-associated biological information," Bioinformatics, vol. 23, no. 13, pp. i337–346, July 2007. [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/btm189>
- [24] L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis, "The generic genome browser: a building block for a model organism system database." Genome research, vol. 12, no. 10, pp. 1599–1610, October 2002. [Online]. Available: <http://dx.doi.org/10.1101/gr.403602>
- [25] M. E. Skinner, A. V. Uzilov, L. D. Stein, C. J. Mungall, and I. H. Holmes, "Jbrowse: A next-generation genome browser," Genome Research, vol. 19, no. 9, pp. 1630–1638, September 2009. [Online]. Available: <http://dx.doi.org/10.1101/gr.094607.109>
- [26] S. E. Lewis, S. M. J. Searle, N. Harris, M. Gibson, V. Iyer, J. Richter, C. Wiel, L. Bayraktaroglu, E. Birney, M. A. Crosby, J. S. Kaminker, B. B. Matthews, S. E. Prochnik, C. D. Smith, J. L. Tupy, G. M. Rubin, S. Misra, C. J. Mungall, and M. E. Clamp, "Apollo: a sequence annotation editor," Genome Biology, vol. 3, no. 12, 2002. [Online]. Available: <http://dx.doi.org/10.1186/gb-2002-3-12-research0082>
- [27] "Generic model organism database (gmod)." [Online]. Available: [www.gmod.org](http://www.gmod.org)
- [28] L. A. Mueller, T. H. Solow, N. Taylor, B. Skwarecki, R. Buels, J. Binns, C. Lin, M. H. Wright, R. Ahrens, Y. Wang, E. V. Herbst, E. R. Keyder, N. Menda, D. Zamir, and S. D. Tanksley, "The sol genomics network. a comparative resource for solanaceae biology and beyond," Plant Physiol., vol. 138, no. 3, pp. 1310–1317, July 2005. [Online]. Available: <http://dx.doi.org/10.1104/pp.105.060707>
- [29] L. G. P. Almeida, R. Paixao, R. C. Souza, G. C. da Costa, F. J. A. Barrientos, M. T. dos Santos, D. F. de Almeida, and A. T. Vasconcelos, "A system for automated bacterial (genome) integrated annotation—sabia," Bioinformatics, vol. 20, no. 16, pp. 2832–2833, November 2004. [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/bth273>

## BIBLIOGRAPHY

---

- [30] T. M. Lowe and S. R. Eddy, "trnscan-se: a program for improved detection of transfer rna genes in genomic sequence." Nucleic acids research, vol. 25, no. 5, pp. 955–964, March 1997. [Online]. Available: <http://dx.doi.org/10.1093/nar/25.5.955>
- [31] S. Hunter, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, U. Das, L. Daugherty, L. Duquenne, R. D. Finn, J. Gough, D. Haft, N. Hulo, D. Kahn, E. Kelly, A. Laugraud, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, J. Mistry, A. Mitchell, N. Mulder, D. Natale, C. Orengo, A. F. Quinn, J. D. Selengut, C. J. A. Sigrist, M. Thimma, P. D. Thomas, F. Valentin, D. Wilson, C. H. Wu, and C. Yeats, "Interpro: the integrative protein signature database," Nucl. Acids Res., vol. 37, no. suppl.1, pp. D211–215, January 2009. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkn785>
- [32] M. Kanehisa and S. Goto, "Kegg: Kyoto encyclopedia of genes and genomes," Nucl. Acids Res., vol. 28, no. 1, pp. 27–30, January 2000. [Online]. Available: <http://dx.doi.org/10.1093/nar/28.1.27>
- [33] R. L. Tatusov, N. D. Fedorova, J. D. Jackson, A. R. Jacobs, B. Kiryutin, E. V. Koonin, D. M. Krylov, R. Mazumder, S. L. Mekhedov, A. N. Nikolskaya, B. S. Rao, S. Smirnov, A. V. Sverdlov, S. Vasudevan, Y. I. Wolf, J. J. Yin, and D. A. Natale, "The cog database: an updated version includes eukaryotes." BMC bioinformatics, vol. 4, no. 1, pp. 41+, September 2003. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-4-41>
- [34] "Laboratory of phil green." [Online]. Available: <http://www.phrap.org/>
- [35] "Image: The fingerprint image analysis system." [Online]. Available: <http://www.sanger.ac.uk/resources/software/image/index.html>
- [36] C. Soderlund, I. Longden, and R. Mott, "Fpc: a system for building contigs from restriction fingerprinted clones," Comput. Appl. Biosci., vol. 13, no. 5, pp. 523–535, October 1997. [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/13.5.523>
- [37] (2010) Richfaces. [Online]. Available: <http://www.jboss.org/richfaces>
- [38] (2010) Primefaces. [Online]. Available: <http://www.primefaces.org/>
- [39] "Java." [Online]. Available: <http://www.oracle.com/technetwork/java/index.html>
- [40] R. C. G. Holland, T. Down, M. Pocock, A. Prlic, D. Huen, K. James, S. Foisy, A. Drager, A. Yates, M. Heuer, and M. J. Schreiber, "Biojava: an open-source framework for bioinformatics," Bioinformatics, vol. 24, no. 18, pp. btn397–2097, August 2008. [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/btn397>

- [41] T. J. Hubbard, B. L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, G. Coates, S. Fairley, S. Fitzgerald, J. Fernandez-Banet, L. Gordon, S. Graf, S. Haider, M. Hammond, R. Holland, K. Howe, A. Jenkinson, N. Johnson, A. Kahari, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, D. Rios, M. Schuster, G. Slater, D. Smedley, W. Spooner, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, S. Wilder, A. Zadissa, E. Birney, F. Cunningham, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, J. Herrero, A. Kasprzyk, G. Proctor, J. Smith, S. Searle, and P. Flicek, "Ensembl 2009." Nucleic acids research, vol. 37, no. Database issue, pp. D690–697, January 2009. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkn828>
- [42] R. Drysdale and FlyBase Consortium, "Flybase : a database for the drosophila research community." Methods in molecular biology (Clifton, N.J.), vol. 420, pp. 45–59, 2008. [Online]. Available: [http://dx.doi.org/10.1007/978-1-59745-583-1\\_3](http://dx.doi.org/10.1007/978-1-59745-583-1_3)
- [43] D. Lawson, P. Arensburger, P. Atkinson, N. J. Besansky, R. V. Bruggner, R. Butler, K. S. Campbell, G. K. Christophides, S. Christley, E. Dialynas, M. Hammond, C. A. Hill, N. Konopinski, N. F. Lobo, R. M. Maccallum, G. Madey, K. Megy, J. Meyer, S. Redmond, D. W. Severson, E. O. Stinson, P. Topalis, E. Birney, W. M. Gelbart, F. C. Kafatos, C. Louis, and F. H. Collins, "Vectorbase: a data resource for invertebrate vector genomics," Nucl. Acids Res., vol. 37, no. suppl\_1, pp. D583–587, January 2009. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkn857>
- [44] S. E. J. C. A. K. Z. A. V. P. Bowes JB, Snyder KA, "Xenbase: gene expression and improved integration." Nucleic Acids Res., vol. 38, pp. 27–30, 2009.
- [45] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. Gilbert, I. Korf, H. Lapp, H. Lehv slaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney, "The bioperl toolkit: Perl modules for the life sciences." Genome research, vol. 12, no. 10, pp. 1611–1618, October 2002. [Online]. Available: <http://dx.doi.org/10.1101/gr.361602>
- [46] (2010) Softberry. [Online]. Available: <http://linux1.softberry.com/berry.phtml>
- [47] O. M. Group, "Omg unified modeling language (omg uml), superstructure, v2.1.2," Tech. Rep., November 2007. [Online]. Available: <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF>
- [48] L. Cui, N. Veeraraghavan, A. Richter, K. Wall, R. K. Jansen, J. Leebens-Mack, I. Makalowska, and C. W. dePamphilis, "Chloroplastdb: the chloroplast genome database." Nucleic Acids Res, vol. 34, no. Database issue, January 2006. [Online]. Available: <http://view.ncbi.nlm.nih.gov/pubmed/16381961>

## BIBLIOGRAPHY

---

- [49] U. Vothknecht and J. Soll, "Chloroplast membrane transport: interplay of prokaryotic and eukaryotic traits."
- [50] M. G. Bausher, N. D. Singh, S.-B. Lee, R. K. Jansen, and H. Daniell, "The complete chloroplast genome sequence of citrus sinensis (l.) osbeck var 'ridge pineapple': organization and phylogenetic relationships to other angiosperms," *BMC Plant Biology*, vol. 6, pp. 21+, September 2006. [Online]. Available: <http://dx.doi.org/10.1186/1471-2229-6-21>
- [51] K. Knapp and Y.-P. P. P. Chen, "An evaluation of contemporary hidden markov model genefinders with a predicted exon taxonomy." *Nucleic acids research*, vol. 35, no. 1, pp. 317–324, January 2007. [Online]. Available: <http://dx.doi.org/10.1093/nar/gkl1026>
- [52] P. E. V. S. S. D. S.-C. H. B. S. F. P. G. D. S. X. A. J. K. D. W. R. F. A. B. H. Paiva, J.A.P. and J. Grima-Pettenati, "Advancing eucalyptus genomics: identification and sequencing of lignin biosynthesis genes from deep-coverage bac libraries," *Unpublished*, 2010.
- [53] A. Coghlan, T. Fiedler, S. McKay, P. Flicek, T. Harris, D. Blasiar, the nGASP Consortium, and L. Stein, "ngasp - the nematode genome annotation assessment project," *BMC Bioinformatics*, vol. 9, no. 1, pp. 549+, December 2008. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-9-549>