FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Information and Data Analysis System for Gene Expression

Diogo André Rocha Teixeira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Camacho

Second Supervisor: Nuno Fonseca (EMBL-EBI, Cambridge, UK)

June 9, 2014

Information and Data Analysis System for Gene Expression

Diogo André Rocha Teixeira

Mestrado Integrado em Engenharia Informática e Computação

June 9, 2014

Abstract

The advent of next generation sequencing methods has revolutionized the field of molecular biology in the past few years. Following this theme, we will discuss the usage of RNA Sequencing methods in order to understand how the final portion of genetic code in a gene's exon affects the transcription speed of that same exon. We will describe the several components of the information system that will be developed to address this problem. A literature review will be presented, for both the areas of transcriptome assembly and data mining, addressing the most relevant concepts regarding our particular problem. Lastly, we will provide an estimated work plan for the project development and describe the methods that will be used for validating the project's results.

Resumo

O advento das técnicas de sequenciação de nova geração revolucionou o campo da biologia molecular nos últimos anos. Seguindo este tema, vamos discutir o uso de métodos de *RNA Sequencing* para perceber de que forma a porção final do código genético do exão de um gene afeta a velocidade a velocidade de transcrição desse mesmo exão. Vamos descrever os vários componentes do sistema de informação que vai ser desenvolvido para endereçar este problema. Será apresentada a revisão da literatura, nas áreas de assemblagem de *RNA* e *data mining*, endereçando os conceitos mais relevantes relativamente ao nosso problema particular. Por fim, vamos apresentar uma estimativa para o plano de trabalho para o desenvolvimento do projeto e descrever os métodos que serão usados para a validação dos resultados do projeto.

Acknowledgements

<TODO>
I'd like to thank the academy...
</TODO>

Diogo André Rocha Teixeira

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

API	Application Programming Interface
AUC	Area Under the Curve
BLAST	Basic Local Alignment Search Tool
cDNA	Complementary DNA
CSS	Cascading Style Sheets
DBMS	Database Management System
DNA	Deoxyribonucleic Acid
GUI	Graphical User Interface
HTML	HyperText Markup Language
IBMC	Institute for Molecular and Cell Biology *(Instituto de Biologia Molecular e Celular)*
ILP	Inductive Logic Programming
K-NN	K-Nearest-Neighbors
mRNA	Messenger RNA
NCBI	National Center for Biotechnology Information
NGS	Next Generation Sequencing
RBP	RNA Binding Protein
RNA	Ribonucleic Acid
RNA-Seq	RNA Sequencing
ROC	Receiver Operating Characteristic
SAM	Sequence Alignment/Map
SVM	Support Vector Machine
rRNA	Ribosomal RNA
tRNA	Transfer RNA
WTSS	Whole Transcriptome Shotgun Sequencing

Chapter 1

Introduction

NOTES

- Introduction seems fine, check later if emphasis on sequencing should be toned down.

Molecular biology is a branch of biology that studies biological activities of living beings, at a molecular level. The grounds for this field of study were set in the early 1930s, although it only emerged in its modern form in the 1960s, with the discovery of the structure of DNA. Among the processes studied by this branch of biology is gene expression. Gene expression (further explained in Chapter 2) is the process by which DNA molecules are transformed into useful genetic products, typically proteins, which are essential for living organisms. This knowledge is not only important in fields like evolutionary or molecular biology, but has crucial applications in fields such as medicine. One example of such an application is the usage of gene expression analysis in the diagnosis and treatment of cancer patients [PASH03].

With the advent of NGS (*Next Generation Sequencing*) techniques, researchers have at their disposal huge amounts of sequencing data, that is not only cheaper and faster to produce, but also more commonly available. This data can then be used to obtain relevant information about organisms' gene expression. But, as the cost of sequencing genomes was reduced, the cost of processing such information was increased. NGS techniques tend to produce much smaller reads[1] than previously used techniques, presenting a more complicated problem, from a computational standpoint [Wol13].

1.1 Domain Problem

NOTES

- Genome assembly is not part of the work, read alignment is.

[1] A *read* is a single fragment of a genome/transcriptome, obtained through sequencing techniques.

- Work objectives float around differential expression analysis, gene enrichment and protein binding site discovery.

Despite its great advancements in the past decades, molecular biology is still a relatively new subject and, as such, there are still some unknowns and partial knowledge in this area. In respect to gene expression, some mechanisms of this intricate process are yet to be fully understood. One such mechanism is the one that regulates the transcription speed of RNA. One of the objectives of the thesis is to understand how the final sequences of a gene's exons are responsible for the speed at which the exons themselves are transcribed. The other objective is to understand how RNA binding protein (RBP) manipulation can be used to better understand an organism's gene expression. This are, however, complex tasks that can be further decomposed in the three main problems that will be addressed in the thesis, namely:

- Sequencing read alignment against a reference genome and differential expression analysis between samples of different individuals (of the same species). This is effectively one of the most complex problems addressed in the thesis. We will use data obtained through a sequencing method called RNA Sequencing[2]. Further insight about this method will be given in Chapter 2, with particular emphasis for tools used to align and analyze this data (Section 2.2.1).

- Gene enrichment and RBP analysis. This part of the work aims to collect as much relevant information as possible about the particular genes being studied at the time, to help biologists better understand their function. RBP knowledge is particularly important for gene manipulation and a great tool for better understanding gene expression, as will be further described in Chapter 2.

- Further analysis of the produced data, using machine learning techniques applied to data mining, specifically to clustering analysis. These techniques will be employed in an effort to try to give biologists more relevant information about gene expression, uncovering possible relationships in the retrieved information. This topic will be developed in Section 2.3.

Solving these problems requires the use of computational tools. As such, the development of a computer system (or multiple systems) to tackle these problems emerges as a secondary objective of the thesis. The details of the idealization of this system will be presented in Chapter 3, while its concrete implementation will be discussed in Chapter 4.

1.2 Motivation and Objectives

NOTES

- Check previous notes, genome assembly is not needed.

[2]RNA Sequencing is also referred to as *Whole Transcriptome Shotgun Sequencing*, or WTSS.

- Don't forget to mention the gene enrichment/protein binding site analysis. - Talk about how simplicity is needed because many times biologists don't understand the tools already available.

As mentioned above, the assembly of a transcriptome is a very complex problem, even more so when RNA Sequencing is used. Aligning such large amounts of data in the form of small reads to a complete (and therefore extensive) genome is a complicated task. It is, however, an essential problem that needs to be solved, as without an assembled transcriptome there is no analyzable data. In turn, this analysis of the transcriptome is essential to further scientific development in fields like molecular biology and medicine, as stated above. With this project we aim at developing an automated computer system, capable of solving these problems. More than applying exclusively to the particular problem at hand, we want to develop an useful and intuitive solution, that might easily be used by researchers in this field of study to solve similar problems, extending its benefits to a broader scientific community.

1.3 Project

NOTES

- Change project description to fit the alignment/differential expression pipeline and the PBS tool. - Refer both tools as independent, but mention that they're supposed to be integrated with each other.

The project itself will revolve around the development of a prototype computer system. The first objective of this prototype is to solve the aforementioned problems, namely the transcriptome's assembly and analysis. Beyond this objective, the prototype should become an useful and intuitive tool for any researcher investigating this or similar problems. To fulfill these objectives, we will need to develop a complex system, composed by several smaller systems. Therefore, the envisioned system architecture is divided into three major components, to wit:

Information system is responsible for storing and managing genetic data, coordinating interaction between the other components of the system and providing a web interface for user interaction. This component will be based mainly on typical web technologies, that is, relational databases for data storage (SQL DBMSs), web frameworks for business logic implementation (Ruby on Rails, Padrino, NodeJS[3]) and web markup and styling languages for interface implementation (HTML, CSS).

Assembly pipeline will use genetic data stored in the information system in order to produce assembled transcriptomes. This pipeline will be composed by several tools, corresponding to each phase of the assembly process, possibly intercalated with data format conversion programs. The tools to be used in this component will be further discussed in Section 2.2.1.

Transcriptome analysis will be responsible for the data mining analysis of the assembled transcriptomes, in the context of the problem of the thesis. It is expected that this component

[3] Ruby on Rails and Padrino are Ruby based web frameworks, while NodeJS is a Javascript based web framework.

integrates with the rest of the system. Further information about the tools that will be used in this component is given in Section 2.3.3.

From here, this document will not dwell in the details of the implementation of such a system, focusing instead the specificities of the problem's solution, from the molecular biology and data mining perspectives. This is due to the fact that the development of the system itself is not the focus of the thesis, but rather a natural consequence of the project's work process.

1.4 Structure of the Report

NOTES

- Update this last.

Besides the introduction chapter, this document is composed by three additional chapters. Chapter 2 introduces some basic biology and RNA-Seq concepts, that are essential to understand the problems with which this document deals. Furthermore, we describe the main techniques used for genome/transcriptome sequencing and assembly, their differences, applications and the tools and data formats typically used in those areas. Lastly, we give some insight about data mining algorithms and how they will be applied in the context of the project. Chapter **??** outlines the main steps in the development of the project (and the respective software prototype) and attempts to provide a feasible schedule for the work's execution. It also presents the datasets that will be studied and used in this work, their origins and features, as well as the validation methods that will be used to ascertain the quality of our results. Chapter **??** sums up the what has been defined in the report, emphasizing the problem that the thesis addresses and the work that will be executed towards solving that problem. It will also give a brief idea of what are the expected results at the end of the project.

Chapter 2

State-of-the-Art

NOTES
- Change transcriptome assembly to read alignment.

In this chapter we will begin by making a more in depth presentation of the process of gene expression. This will be followed by a literature and state of the art review in the fields of transcriptome assembly and data mining. Lastly, we will present some of the tools used in each of those areas and common result evaluation techniques, as well as some relevant data representation formats for genetic information.

2.1 Biological Base Concepts

NOTES
- Base concepts look good.
- Include part about RNA binding proteins.

Before dwelling in the details of the state of the art that are on the foundation of the thesis, it is important to explain some concepts of the domain of molecular biology. As explained in Chapter 1, gene expression is the mechanism by which an organism's DNA can be expressed into functional genetic products, like proteins. This process starts with the genetic code, or nucleotide sequence, of each gene. Different genes in an organism's DNA are responsible for the creation of different genetic products. The process of gene expression itself is composed by two main stages, transcription and translation [GEN].

Transcription is the stage at which genetic data in the form of DNA is used to synthesize RNA, being this the process that concerns the thesis' main question. Several different types of RNA are produced by this process, including mRNA (which specifies the sequences of amino acids that form a protein), rRNA and tRNA, both later used in the translation stage. Simplifying a gene's

structure, it can be seen as composed by two types of sequences, introns and exons, as seen in Figure 2.1.

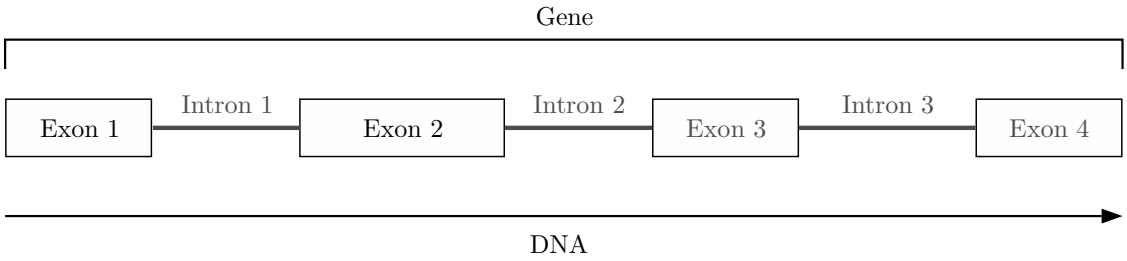

Figure 2.1: Overall structure of a gene, with its different areas (simplified).

Only the exons are useful in the gene expression process, being also known as coding regions. Introns, on the other hand, are not used in the process. They are present in an early stage mRNA molecule, the precursor mRNA, but are later removed (or spliced) in the final molecule before the translation stage [GEN]. Figure 2.2 illustrates the removal of introns from the mRNA molecule, during the splicing process. As stated before, the main goal of this thesis is to explain how the final nucleotide sequence of each exon affects the transcription speed of the exon itself.

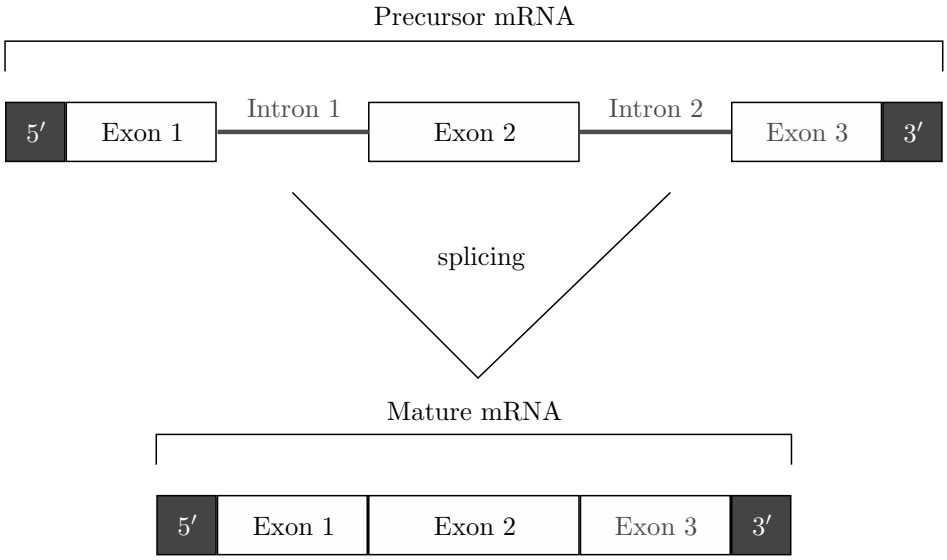

Figure 2.2: The removal (splicing) of introns from the precursor mRNA, during the transcription process.

After the conclusion of the transcription process comes the translation process. In this process, the synthesized mRNA is used to specify the sequence of amino acids that constitute the particular protein being produced. The other types of RNA molecules (rRNA and tRNA) are also used in this stage of the gene expression process.

Obtaining this genetic information is done experimentally, by employing a sequencing technique. For quite some time this process was carried out using the Sanger's and other similar

sequencing methods methods [RF09]. Though effective, such methods were notably slow and costly, with large projects like the Human Genome Project (HGP) consuming roughly thirteen years and US$ 3 billion. These limitations were so severe that, other than the realm of human genetics, this kind of study was restricted to model organisms, such as the fruit fly and mouse genomes [Wol13]. The past few years have seen the appearance and rise in popularity of the NGS techniques. These techniques differ from the more classical ones by producing larger amounts of information, at less cost. They are also typically more cost effective than previous techniques and can be easily employed by single laboratories, which has greatly contributed to their popularity. As a disadvantage, NGS techniques produce shorter reads than their older counterparts, being that "*(...) transcriptome assembly from billions of RNA-Seq reads (...) poses a significant informatics challenge*" [MW11, p. 671]. Although the thesis will not deal with the problems of sequencing techniques, it is important to indicate that the read dataset that will be used is the result of NGS techniques. As such, we will use assembly techniques more suited to situations where short reads are available.

2.2 RNA Sequencing and Transcriptome Assembly

NOTES

- Again, careful with genome assembly references.
- The part about genome assembly seems nice to have here, but explain that the focus is read alignment.

Transcriptome assembly is the process by which experimentally obtained genetic data reads can be organized and merged together in a partial or complete genome expression profile. As stated above, the advent of next generation sequencing techniques, with their reduced costs, greatly increased the availability of genome sequencing data.

For years, microarrays were the standard tool available for examining features of the transcriptome and global patterns of gene expression [Wol13]. However, microarrays are typically more oriented towards assembly against existing reference data, hence limiting its application to species with well known reference genomes. This is impractical, as NGS techniques allow to cheaply obtain genetic information of previously non-studied species. This is one of the reasons that led to the inception of RNA-Seq. Contrary to microarrays, RNA-Seq techniques are able to wield results that are suitable for both reference guided assembly and *de novo* assembly approaches [WL09]. *De novo* or exploratory assembly has captured the interest of researchers in the past few years, leading to the appearance of multiple RNA-Seq tools that are capable of making this type of assembly without a reference genome [FEB+11]. Despite this amazing capability, we will restrict this project to reference genome guided problems, which are simpler and more suitable for a Masters level thesis.

2.2.1 RNA Sequencing Tools

NOTES

- Add section about differential expression analysis tools.
- Refer the whole Tuxedo Suite, but remind that some are not used.
- Refer iRAP. - Refer any other used tools.

Below we will present some bioinformatic tools, used to support the multiple steps of the RNA Sequencing process. Although there are several tools capable of executing all steps of this process, it has been decided we will create our own alignment/assembly pipeline, using specialized tools for every step. We will also present some of the most popular file formats used in this context, along with some tools used to manipulate those formats. We will focus in open source, Unix command line based tools.

Tuxedo Suite

The Tuxedo suite is a free, open-source collection of applications that has been widely adopted as analysis toolset for fast alignment of short reads. It is composed by four separate tools, Bowtie, TopHat, Cufflinks and CummRbund, briefly reviewed below. These tools are extensively used for RNA Sequencing analysis. Although the applications are made for command line execution, there are several workflow managers, like Galaxy[1], that easily integrates with the suite, providing a web interface for its use.

Bowtie. Bowtie[2] is an ultrafast, memory-efficient short read aligner. Bowtie is typically used to build a reference index for the genome of the organism being studied, for posterior use by other tools, like TopHat. It can also output alignments in the standard SAM format, allowing Bowtie to interoperate with tools like SAM Tools. However, it should not be used as a general purpose alignment tool, as it was created and is more effective when aligning short read sequences against large reference genomes.

TopHat. TopHat[3] is a fast splice junction mapper for RNA Sequencing reads. It uses Bowtie as the underlying alignment tool, using its results and a FASTA formated reference genome to identify splice junctions between exons.

Cufflinks. Cufflinks[4] assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA Sequencing samples. It uses the SAM or BAM formatted files as input, typically the ones produced by TopHat, outputting GTF files as a result.

[1] http://galaxyproject.org/
[2] http://bowtie-bio.sourceforge.net/index.shtml
[3] http://tophat.cbcb.umd.edu/
[4] http://cufflinks.cbcb.umd.edu/

CummeRbund. Lastly, CummeRbund[5] is an R package (see Section 2.3.3) designed to help the visualization and analysis of Cufflinks' RNA Sequencing output. As such, it is not directly involved in the trascriptome alignment process. It takes the various output files from Cufflinks and uses them to build a SQLite database describing appropriate relationships between genes, transcripts, etc. It implements several plotting functions, as well as commonly used data visualizations.

SAM Tools

SAM Tools[6] is a library package designed for parsing and manipulating alignment files in the SAM/BAM format [LHW+09] (see Section 2.2.2). SAM Tools has two separate implementations, one in C and the other in Java, with slightly different functionality. Beyond manipulation of SAM and BAM files, this package is able to convert between other read alignment formats, sort and merge alignments and show them in a text-based viewer.

BLAST

BLAST[7] is a tool, implemented in C++, that is used to find regions of local similarity between biological sequences. It uses FASTA sequences (see Section 2.2.2) as search input and outputs the results reports in XML, HTML or plain text. There are several different BLAST programs available at the moment, that can be used depending on our objective and type of data. BLAST is particularly useful to search biologic sequence databases, but can be used for other purposes, like identifying an unknown species or comparing common genes in two related species.

2.2.2 Relevant Standard File Formats

NOTES

- Still relevant, not much to change.

As expected, the great diversity of RNA Sequencing tools brings with it a wealth of file formats. Some of these formats are developed from the ground up to satisfy a specific need, while other are mere contextual adaptations or specializations of already established formats. Below we will present a few of the most popular and widely spread file formats, talking about their basic structure, the types of data they represent and their applications.

FASTA

FASTA is the standard line and character sequence format used by NCBI [NCB], using this last organization's character code conventions. It is a simple format, that can be used to easily store

[5] http://compbio.mit.edu/cummeRbund/
[6] http://samtools.sourceforge.net/
[7] http://blast.ncbi.nlm.nih.gov/Blast.cgi

data represented by character sequences, like nucleotide (DNA, RNA) or amino acid (protein) sequences. This file format is widely use to store sequencing reads, DNA/RNA sequences and other character sequences in database systems. Its simplicity makes it extremely easy to manipulate and parse, presenting also an attractive solution for data transfer between different tools.

FASTQ

FASTQ is used to store store character sequences, typically nucleotide sequences [CFG+10]. It is quite similar to the standard FASTA format, in respect to the manner in which character sequences are represented. However, for every sequence, there is a second sequence of equal length, representing the quality scores of the original sequence. These quality scores are also represented as single characters, taking values between and including ASCII-33 to ASCII-126. It is typically used in the same situations as the FASTA format, when quality scores are available/relevant.

SAM and BAM

The SAM format is a text format for storing sequence alignment data [Lab]. It is widely used to store mapping information between sequencing reads and a given reference genome. This sort of information is typically the product of sequencing alignment tools, that consume sequencing reads from FASTQ files and align them with a reference genome.

The BAM format contains exactly the same information as the SAM format and the same rules apply for both formats. The difference between both formats lies in their encoding. While SAM is a text based format, BAM is a binary format. This means that BAM sacrifices human readability for increased machine processing performance, as it is more efficient to work with compressed and indexed binary data.

VCF

VCF is a text file format used to store gene sequence variants [Smi13]. In the past few years, as larger and larger genome sequencing projects became more common (like the 1000 Genomes Project[8]), storing such large amounts of information became a serious concern. To address these concerns the VCF format was created. Instead of storing the complete genome, VCF stores only the variations (and their respective positions) of newly sequenced genomes relatively to a known reference genome, typically in a compressed text file. As such, it is a format often used when building genome databases.

GFF and GTF

GFF is a text based file format to store gene features [San11]. Many genome assembly tools execute this process in two separate steps: feature detection for identification of specific regions

[8]The 1000 Genomes Project, started back in 2008, is an international effort to establish the most comprehensive catalogue to date of human genetic variations.

(exons, introns, etc.) and genome assembly, using those features as reference. However, often times it is beneficial to decouple these two steps, using different and more efficient tools for each. As such, the GFF format emerged as a protocol for feature information transfer between tools.

The GTF format is similar to the GFF format, in which it is based. It is also used in similar situations. However, GTF builds on top of GFF, defining additional conventions, specific to the domain of genetic information. Despite their initial relation, both formats continue to be developed individually.

2.3 Data Mining

NOTES

- Still relevant, don't change.
- Important part is now clustering, not classification.

Data mining is the process of "*extracting or "mining" knowledge from large amounts of data*" [HKP06, p. 5]. As such, it consists of a set of techniques that can be used to find interesting patterns in large data sets, that translate in newfound knowledge. Data mining borrows techniques from multiple fields, such as artificial intelligence, machine learning, statistics, and database systems [CEF+12]. Its ultimate goal is to combine all those techniques and transform large and (apparently) meaningless sets of data into understandable and useful information. Thus, data mining was motivated by the perspective of harnessing the abundance of data, that characterizes today's information systems, to produce meaningful knowledge.

Because of their large quantities of input data, data mining tasks are usually totally, or at least partially, automated. As such, there are several algorithms for these tasks and tools that implements such algorithms, as presented in Section 2.3.1 and Section 2.3.3, respectively.

We can divide data mining into main types: descriptive data mining and predictive data mining [FPSS96]. Descriptive data mining is focused on finding the underlying structure of a given set of data. Instead of predicting future values, it concerns the intrinsic structure, relations and interconnectedness of the data being analyzed, presenting its interesting characteristics without having any predefined target. On the other hand, predictive data mining is used to predict explicit values, based on patterns determined from the dataset. With predictive data mining we try to build models using known data and use those models as a base to predict future behavior.

As we're seeing, data mining does not represent a single type problem. In fact there are several different types of problems that can be addressed by data mining techniques. Each of these problems may require a different data mining method. A brief review of the most common methods is given below.

Classification is a method that tries to generalize the already known structure of a dataset, so that it applies to new datasets. In other words, with classification we try to learn a function that is capable of mapping our data into predefined classes.

Regression tries to learn a function that models relationships between variables in the dataset. That function can latter be used to find real value predictions of future behavior of the same or similar datasets.

Clustering consists in identifying a finite set of categories or clusters of similar values, to describe the dataset. As such, it is used without prior knowledge about data structure.

Summarization provides a more compact representation of a subset of data, in a way that the summarized data retains the central points of the original data. This can be accomplished in several different ways, like using report generation or multivariate visualization techniques.

Dependency modeling finds a model which describes relationships between variables, revealing their dependencies.

Change and deviation detection tries to discover the most significant changes in the data, when compared with previously measured data. This method is useful to find interesting data variations or data errors.

2.3.1 Data Mining Algorithms

NOTES

- Focus is now on clustering algorithms. - See Wikipedia's page on "clustering" for a good reference on algorithm types.

In this project we will be concerned with the classification side of data mining. Below, we will review some algorithms that can be used in classification problems.

Inductive Logic Programming

NOTES

- Still seems relevant, later refer that it wasn't possible to conduct ILP clustering due to lack of time/problems with tools.

ILP is a subfield of machine learning that uses first order logic to represent both data and models [LD98]. ILP induces hypotheses (models) from examples and background knowledge. Examples are of two types: instances of the concept to be "learned" and non-instances of the concept. Background knowledge is a set of predicates encoding all information that the experts find useful to construct the models. ILP might be used to tackle several machine learning and data mining problems, like classification, regression and clustering.

The first and most important motivation for ILP systems is that they overcome the representation limitations of attribute-value learning systems, such as the previously mentioned data mining algorithms. Attribute-value systems base their representations of data in table based representations. Although effective in many situations, these representation is not very expressive and might

not even be feasible for certain problems [BM95]. The second motivation for ILP is that by using a logical representation, the hypotheses are understandable and interpretable by humans, being therefore useful to explain the phenomenons that produce the data. This representation also means that background knowledge can be represented and employed in the induction process, in contrast to attribute-value models, where this information is difficult to represent.

Despite these advantages, ILP cannot be applied indiscriminately to any classification or regression problem. ILP systems are typically very heavy when it comes to computational resource consumption and run for long periods of time [FCSC03].

2.3.2 Model Evaluation Procedures and Measures

NOTES

- No longer relevant.
- Explain internal and external evaluation.
- Explain that due to the novelty of our analysis we will merely focus on internal analysis.
- Describe some internal analysis measures (don't forget silhouettes).

2.3.3 Data Mining Tools

Except in rare cases of very specific problems, it typically makes no sense for someone to implement any data mining algorithm that they might need. In fact, today we have lots of data mining tools (many of which are free), that already implement many of those algorithms. These tools are usually customizable, making it easy to adapt them to most problems. Below we'll briefly review some of the most popular data mining tools, that apply to the specific needs of this thesis.

RapidMiner

NOTES

- Still relevant.
- Refer that RapidMiner and Weka were only used for testing purposes, and are note part of the final solutions.
- Refer other relevant R packages.

RapidMiner[9] is a complete solution for data mining problems. it is available as a standalone GUI based application, as seen in Figure 2.3. It is a commercial application, although its core and earlier versions are distributed under an open source license and it offers a free version, beyond its multiple paid versions. Being one of the most popular data mining tools used today, its applications span several domains, including education, training, industrial and personal applications,

[9] http://www.rapidminer.com/

among others. Its functionality can also be easily extended through the use of plugins[10], reflecting in an increased value for this tool. One such example in the area of bioinformatics is the integration plugin between RapidMiner and the Taverna[11] open source workflow management system [JEF11].

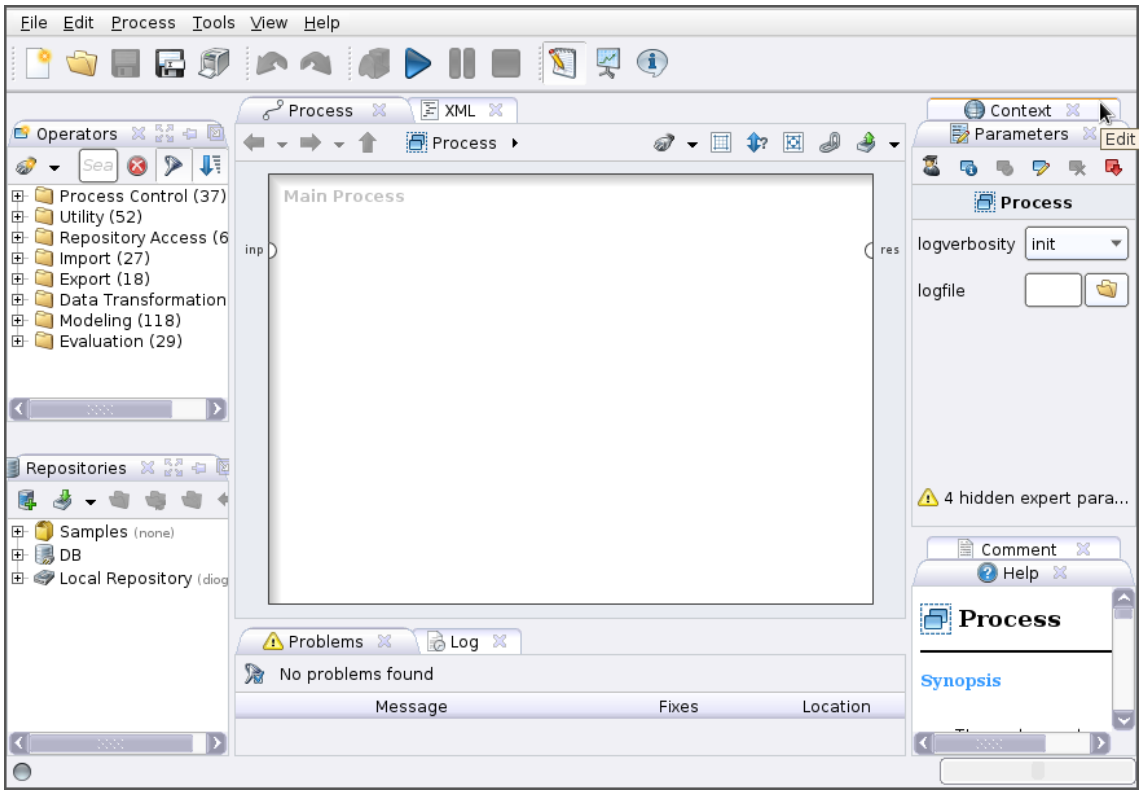

Figure 2.3: RapidMiner user interface.

Weka

Weka[12] is an open source tool that collects several machine learning algorithms and allows its user to easily apply those algorithms to data mining tasks [HNF+09]. Created at the University of Waikato, New Zeland in 1997 (the current version was completely rewritten in 1997, despite the first iteration of the tool being developed as early as 1993), it is still in active development to date. Weka supports several common data mining tasks, like data preprocessing, classification, clustering, regression and data visualization. it is core libraries are written in Java and allow for an easy integration of its data mining algorithms in pre existing code and applications. Other than that, Weka can be used directly through a command line/terminal or through one of its multiple

[10] Plugin is a software module that adds new functionality to an existing software application. Plugins are typically dependent on the platform they extend and can't be used as standalone tools.

[11] http://www.taverna.org.uk/

[12] http://www.cs.waikato.ac.nz/ml/weka/

GUIs (Figure 2.4). Its simple API and well structure architecture allow it to be easily extended by users, should they need new functionalities.

Figure 2.4: Weka interface selection.

R Language

R[13] is a free programming language and software environment for statistical computing and graphics generation. Originally developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand in 1993 [Iha98], it is still under active development. R is typically used by statisticians and data miners, either for direct data analysis or for developing new statistical software [FA05].

R is an implementation of the S programming language[14], borrowing some characteristics from the Scheme programming language. it is core is written in a combination of C, Fortran and R itself. It is possible directly manipulate R objects in languages like C, C++ and Java. R can be used directly through the command line or through several third party graphical user interfaces like Deducer[15]. There are also R wrappers for several scripting languages.

R provides several different statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, among others. It can also be used to produce publication-quality static graphics. Tools like Sweave [Lei02] allow users to embed R code in LaTeX documents, for complete data analysis.

Bioconductor Package. Bioconductor is a free and open source set of tools for genomic data analysis, in the context of molecular biology [Lei02]. It is primarily based on R. It is under active development, with two stable releases each year. Counting with more than seven hundred different packages, it is the most comprehensive set of genomic data analysis tools available for the R programming language. It also provides a set of tools to read and manipulate several of

[13] http://www.r-project.org/
[14] S is an object oriented statistical programming language, appearing in 1976 at Bell Laboratories.
[15] http://www.deducer.org/pmwiki/index.php

the most common file formats used in molecular biology oriented applications, including FASTA, FASTQ, BAM and GFF.

2.4 Chapter Conclusions

NOTES

- Remove the uncertainty part, talk about what was done.

In this chapter we gave a brief introduction of the molecular biology concepts that serve as base of the thesis. We also reviewed the concepts on RNA-Seq and data mining and presented short analyses of concrete tools that will likely be used during the project.

At this moment we do not possess all the necessary information about the dataset that will be used in the data mining phase of the project. We are unable, at the present, to determine the nature of our data, which means that we cannot predict whether it could be modeled by classification algorithms. As such, we presented ILP as an alternative approach for this situation. In case ILP techniques become in fact necessary, further work will include a more profound and complete revision.

Chapter 3

Solution Description

NOTES

- Talk about iRAP and how the web interface for analysis.
- Talk about PBS Finder, how it is standalone and it's web interface.
- Talk about integration of both tools, and the idealized complete system (a diagram would be nice :)).

Chapter 4

Implementation

NOTES

- Talk about iRAP configuration, the result combination tool and more.

- Talk about PBS Finder configuration, requisites and analysis flow (show that analysis flow diagram).

- Talk about platform extensibility, deployment alternatives and more.

Chapter 5

Conclusions

NOTES

- Check PDIS conclusion.
- Talk about finishing iRAP's web integration and further exploration of its results.
- Talk about full automated integration between both tools.

Conclusions

References

[BM95] Ivan Bratko and Stephen Muggleton. Applications of Inductive Logic Programming. *Commun. ACM*, 38(11):65–70, November 1995.

[CEF+12] Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, and Wei Wang. Data mining curriculum: a proposal, version 1.0. Available at www.kdd.org/curriculum/CURMay06.pdf, last access on February 2014, April 2012.

[CFG+10] Peter J a Cock, Christopher J Fields, Naohisa Goto, Michael L Heuer, and Peter M Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic acids research*, 38(6):1767–71, April 2010.

[FA05] John Fox and Robert Andersen. Using the R statistical computing environment to teach social statistics courses. *Department of Sociology, McMaster University*, (January), 2005.

[FCSC03] Nuno Fonseca, VS Vitor Santos Costa, Fernando Silva, and Rui Camacho. On the implementation of an ILP system with Prolog. Technical report, Technical report, DCC-FC & LIACC, UP, 2003.

[FEB+11] Nuno Fonseca, Dent Earl, Keith Bradnam, et al. Assemblathon 1: A competitive assessment of *de novo* short read assembly methods. *Genome Research*, 21(12):2224–2241, December 2011.

[FPSS96] UM Fayyad, G Piatetsky-Shapiro, and P Smyth. Knowledge Discovery and Data Mining: Towards a Unifying Framework. *KDD*, 1996.

[GEN] GENIE. Gene expression and regulation. Available at http://www2.le.ac.uk/departments/genetics/vgec/schoolscolleges/topics/geneexpression-regulation, last access on February 2014.

[HKP06] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan kaufmann, 2006.

[HNF+09] Mark Hall, Hazeltine National, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA Data Mining Software : An Update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[Iha98] Ross Ihaka. R: Past and future history. *COMPUTING SCIENCE AND STATISTICS*, 1998.

REFERENCES

[JEF11] Simon Jupp, James Eales, and Simon Fischer. Combining RapidMiner operators with bioinformatics services–a powerful combination. In *RapidMiner Community Meeting and Conference, (RCOMM)*, 2011.

[Lab] Abecasis Lab. Sam - genome analysis wiki. Available at http://genome.sph.umich.edu/wiki/SAM, last access on February 2014.

[LD98] N Lavrac and S Dzeroski. *Inductive logic programming*, volume 1446 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/Heidelberg, 1998.

[Lei02] Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

[LHW+09] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16):2078–9, August 2009.

[Lyo98] Robert Lyons. A molecular biology glossary. Available at http://seqcore.brcf.med.umich.edu/doc/educ/dnapr/mbglossary/mbgloss.html, last access on February 2014, July 1998.

[MW11] Jeffrey Martin and Zhong Wang. Next-generation transcriptome assembly. *Nature reviews. Genetics*, 12(10):671–82, October 2011.

[NCB] NCBI. Web blast page options. Available at https://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml, last access on February 2014.

[PASH03] Lajos Pusztai, Mark Ayers, James Stec, and Gabriel N Hortobágyi. Clinical Application of cDNA Microarrays in Oncology. *The Oncologist*, 8(3):252–258, January 2003.

[RF09] Jorge S. Reis-Filho. Next-generation sequencing. *Breast cancer research : BCR*, 11 Suppl 3:S12, January 2009.

[San11] Sanger Institute. Gff (general feature format). Available at http://www.sanger.ac.uk/resources/software/gff/spec.html, last access on February 2014, 2011.

[Smi13] Smith, Steven and Browning, Brian. Introduction to variant call format. Available at http://faculty.washington.edu/browning/beagle/intro-to-vcf.html, last access on February 2014, 2013.

[WL09] Brian T Wilhelm and Josette-Renée Landry. RNA-Seq-quantitative measurement of expression through massively parallel RNA-sequencing. *Methods (San Diego, Calif.)*, 48(3):249–57, July 2009.

[Wol13] Jochen B W Wolf. Principles of transcriptome analysis and gene expression quantification: an RNA-seq tutorial. *Molecular ecology resources*, 13(4):559–72, July 2013.

Appendix A

Glossary

This brief glossary was based on a similar work by Robert Lyons [Lyo98].

cDNA	DNA which has been reverse transcribed using RNA as a template.
Exon	The portions of a genomic DNA sequence which will be represented in the final, mature mRNA. Exons may include coding sequences, the *5'* untranslated region or the *3'* untranslated region.
Expression	To "express" a gene is to cause it to function. A gene which encodes a protein will, when expressed, be transcribed and translated to produce that protein. A gene which encodes an RNA rather than a protein (for example, a rRNA gene) will produce that RNA when expressed.
Gene	A unit of DNA which performs one function. Usually, this is equated with the production of one RNA or one protein. A gene contains coding regions, introns, untranslated regions and control regions.
Genome	The total DNA contained in each cell of an organism. There are somewhere in the order of a hundred thousand genes, including coding regions, *5'* and *3'* untranslated regions, introns, *5'* and *3'* flanking DNA.
Intron	Introns are portions of genomic DNA which are transcribed (and thus present in the primary transcript) but which are later spliced out. Thus, they are not present in the mature mRNA.
mRNA	"Messenger RNA" contains sequences coding for a protein. The term mRNA is used only for a mature transcript (with all introns removed), rather than the primary transcript in the nucleus.
rRNA	"Ribosomal RNA" describes any of several RNAs which become part of the ribosome, and thus are involved in translating mRNA and synthesizing proteins.

Shotgun cloning	The process of randomly shearing an organism's genomic DNA and cloning it into a suitable vector, resulting in a genomic library.
Shotgun sequencing	Sequencing the DNA library created by shotgun cloning.
Transcription	The process of copying DNA to produce an RNA transcript. This is the first step in the expression of any gene. The resulting RNA will produce the desired protein molecule by the process of translation. [2]
Translation	The process of decoding a strand of mRNA, thereby producing a protein based on the code.
tRNA	"Transfer RNA" represents one of a class of rather small RNAs used by the cell to carry amino acids to the enzyme complex (the ribosome) which builds proteins, using an mRNA as a guide.

Appendix B

iRAP Example Configuration