FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Application of Machine Learning techniques on the Discovery and annotation of Transposons in genomes

Tiago David Soares da Cruz Loureiro

Master in Informatics and Computing Engineering

Supervisor: Jorge Vieira (PhD)

Supervisor: Nuno Fonseca (PhD)

Supervisor: Rui Camacho (PhD)

7 February 2012

Abstract

Transposable elements or transposons are sequences of DNA that move and transpose within a genome. Known as mutation agents, these elements are broadly studied due to their importance in disease research, genome alteration and because of their importance on species evolution.

Several methods were developed to discover and annotate transposable elements and they are classified in four main categories: *De novo*, Structure-based, Comparative Genomic and Homology-based.

There are different tools based on these methodologies that detect transposable elements, although, there isn't any single tool which has good results in detecting all the different types of transposable elements.

Taking this into account, this dissertation will have three distinct phases. At first there will be generated datasets of curated DNA sequences with transposable elements inserted in known positions. These datasets will be as diversified as possible so they can cover all the different scenarios found in real genomes.

Following, using these datasets, transposon detection tools are evaluated and their accuracy is measured.

Using the results of the transposon detection tools' evaluation, the last step of this dissertation is to use Machine Learning techniques to identify and characterize the context where these tools fail short in transposon detection. After this, the aim is to create a meta-tool using models generated by Machine Learning that combines the best of different transposable elements detection tools in order to improve the overall detection accuracy.

Acknowledgements

I would like to acknowledge Rui Camacho, Nuno Fonseca and Jorge Vieira for their support, help and suggestions in this report. With their precious help and expertise, the context and objectives of this dissertation became much clear to me.

Tiago David Soares da Cruz Loureiro

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations and symbols

DNA	Deoxyribonucleic acid
TE	Transposable Element
mtDNA	Mitochondrial DNA
cDNA	DNA copy
BLAST	Basic Local Alignment and Search Tool
Transposase	An enzyme that is responsible for the catalysis of transposition
LTRs	Long Terminal Repeats
MITE	Miniature Inverted Repeat Transposable Element
SINE	Short Interspersed Nuclear Element
LTR	Long Terminal Repeat
TIR	Terminal Inverted Repeat
A	Adenine
G	Guanine
C	Cytosine
T	Thymine
bp	Base Pairs
FASTA	Text-based format for representing nucleotide sequences, each one represented by a letter
Indel	A mutation class that includes both insertions and deletions.

Chapter 1

Introduction

This Chapter introduces the work of this thesis by presenting its context. It follows by presenting the motivation and the goals of this work. Lastly, it presents the document's structure.

1.1 Context

Transposable elements (TEs) or simply transposons are a large class of repetitive DNA sequences that have the ability to move within a given genome. Is is estimated that 40% or more of the human genome is composed of transposon-derived sequences and although many are remnants of active elements, the mobility of transposons has had an important role in the structure and evolution of genes and genomes.

Having the ability to replicate, transposons can occupy large fractions of genome sequences, especially higher in eukaryotes.

The identification of TEs has become increasingly necessary for both genome annotation and evolutionary studies. The contribution of TEs to genome structure and evolution as well as their impact on genome sequencing, assembly, annotation and alignment has generated increasing interest in developing new methods for their computational analysis.

Many approaches to detect TEs were developed since transposons were found by Barbara McClintock [McC50]. Based on different proprieties of this TEs, these approaches look for structural similarities, compare sequences with known transposons or try to identify repetitive patterns.

Given the existence of different transposon detection methodologies, there are several software tools that implement these different approaches. Each tool has its own strengths and weaknesses in the detection of particular TE category. Hence there are specific tools to achieve better detection rates on specific transposon categories. Adding to this, different tools can disagree on the detection of a TE, whether not agreeing on its length or boundaries (beginning position and/or the ending position).

In this perspective, transposon detection tools may have the potential to benefit from comparison between each other and on the integration of multiple tools in a computational dynamic system.

1.2 Motivation and Goals

Transposable elements are very important entities to be studied as they have preponderant roles in genome structure, size, rearrangement and have great contribution to host gene and regulatory evolution. Also they are very useful in plant molecular biology since they are mutation agents and as such they can introduce desirable characteristics where they are inserted.

Due to their importance, various approaches were made in order to identify and annotate transposon elements such as *de novo*, homology-based, structured-based and comparative genomic methods.

De novo methods compare several sub-sequences of a given genome and if they are repeated several times within that genome then they can potencially be TEs. The key step on this approach is to distinguish TEs from all other repeat classes. On Homology based methods sets of known TEs are compared with DNA sequences of a genome to find similarities. Structure based Methods use prior knowledge about the common transposon structural features, such as long terminal repeats to identify potential transposon elements on a given genome. Finally comparative genomics methods use multiple alignments of closely related genomes to detect large changes between genomes. Although different, all these approaches pursue the detection of TEs and each has its own strengths and weaknesses.

Several studies have been published describing and comparing transposon detection tools [BQ07] [CD03]. In these studies several transposon detection tools are described and their implementations compared. However the results of transposon detection vary from tool to tool. Transposon detection agreement between different tools is one of the main problems as tools can identify or not a given transposon and even if identified they can disagree on its length and start and end positions within the genome.

On another perspective, and according to [BQ07] and [CD03], integration of multiple approaches will further advance the computational analysis of this dynamic component. This means that integrating the best of each relevant tool can improve the overall detection accuracy and provide researchers better result in transposon detection.

The aim of this dissertation is to improve transposon detection by analyzing existing transposon detection tools and evaluate their performance regarding the detection and annotation of TEs. Based on their performance, it is expected that a set of tools is integrated based on machine learning models so that the efficiency of transposon detection is increased and the combination of the tools will take advantage of the strengths of each one on the different contexts.

With this in mind, the main idea is to use datasets of DNA sequences that have transposon sequences present in *a priori* positions. Using these datasets, several discovery methodologies are tested and evaluated accordingly to the expected results of the datasets. Using the results of the

previous step as a data source, machine learning techniques will be used in order to combine different transposon detection methodologies and create a model to increase the accuracy of transposon element detection and annotation.

1.3 Document Structure

The next Chapter on this document, Chapter 2, provides a background of transposon related biological information and gives an insight on the Machine Learning subject. First in Section 2.1 it is given a description of what are transposons, what types do exist, how they are classified among other useful definitions of biological components related to the subject. After this, in Section 2.2 it is described the concept of Machine Learning and the most important techniques relevant for this dissertation.

In chapter 3, state of the art on transposon is presented and some discovery approaches will be described and some considerations will be made about the strength of them in finding particular types of transposons elements. Transposon detection tools will also be analyzed later in this chapter.

Chapter 4 summarizes the technologies that will be used on this project and describes the experimental setup, validation and evaluation of the results of this work.

Chapter 5 describes the planned work for the following semester and Chapter 6 concludes this report.

Chapter 2

Background

In this Chapter several concepts related to transposable elements (TEs) and genomes are introduced. The concept of TEs is presented and their importance in the genome is emphasized. The main different TEs kinds are presented and characterized.

Finally in 2.3 there are taken some conclusions regarding the transposable elements and their discovery methodologies.

2.1 Introduction to Transposable Elements

Initialy discovered during the middle part of the twentieth century by Barbara McClintock, transposable elements (TEs), "jumping genes" or simply transposons are DNA sequences that move from one location on the genome to another. According to the The American Heritage Science Dictionary [Ame05] the definition of transposon is the following:

> A segment of DNA that is capable of independently replicating itself and inserting the copy into a new position within the same or another chromosome or plasmid.

McClintock's work [McC50] was revolutionary in that it suggested that an organism's genome is not a stationary entity, but rather it is subject to alteration and rearrangement. Eventually, she was awarded the Nobel Prize in 1983 for her work.

2.1.1 DNA sequences

Firstly published in 1953 by Watson and Crick [WC53], DNA stands for deoxyribonucleic acid. It is the genetic material which makes up of all living cells and many viruses. It consists of two long strands of nucleotides linked together in a helicloidal structure. In eukaryotic cells, the DNA is contained in the nucleus, where it is called nuclear DNA, and in mitochondria where it is called mitochondrial DNA or mtDNA.

DNA has asymmetric ends called the 5' (five prime) and 3' (three prime) ends, with the 5' end having a terminal phosphate group and the 3' end a terminal hydroxyl group.

The information in DNA is stored as a code made up of four chemical bases: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). The order, or sequence, of these bases determines the information available for building and maintaining an organism. DNA bases pair up, adenine with thymine and cytosine with guanine, to form units called base pairs (bp), as shown in Figure 2.1. Each base has a five-carbon sugar,deoxyribose, and one phosphate group attached. This group forms a nucleotide which is present in the form of a spiral called double helix [Ref12].

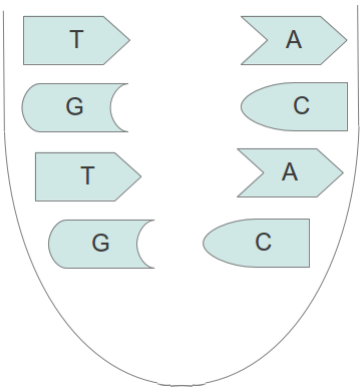

Figure 2.1: DNA structure

The DNA segments carrying its genetic information are called genes.

2.1.2 Gene

Like transposons, genes are genomic sequences that represent the basic physical and functional unit of heredity. Genes, which are made up of DNA, act as instructions to make molecules called proteins.

Although the classic view of genes defines them as compact, information-laden gems hidden among billions of bases of junk DNA, they are neither compact nor uniquely important. According to [Pen07] and [GP08] genes can be sprawling, with far-flung protein-coding and regulatory regions that overlap with other genes.

2.1.3 Mutations

Mutations are sudden and spontaneous changes genome sequences caused by radiation, viruses, transposons and mutagenic chemicals, as well as errors that occur during meiosis or DNA replication. TEs, as integral elements of genomes are possible targets of these mutations.

Mutations in small scale can be classified as:

- Point mutations: exchange a single nucleotide for another, these changes are classified as transitions or transversions. Most common is the transition that exchanges a A to G or G to A or a T to C or C to T. A transition can be caused by nitrous acid, base mis-pairing,

or mutagenic base analogs. Less common is a transversion, which exchanges a C/T for A/G. A point mutation can be reversed by another point mutation, in which the nucleotide is changed back to its original state or by second-site reversion (a complementary mutation elsewhere that results in regained gene functionality). Point mutations that occur within the protein coding region of a gene may be classified into three kinds, depending upon what it codes for:

Silent mutations: which code for the same amino acid.

Missense mutations: which code for a different amino acid.

Nonsense mutations: which code for a stop and can truncate the protein.

- Indel mutations [KR04]:

  Insertions are additions of one or more extra nucleotides into a DNA sequence. It may be caused by TEs or by errors during replication of transposons. The consequences of an insertion in a coding region of a gene may be altering the splicing of the mRNA or causing a shift in the reading frame. These insertions can be reverted by excision of the TEs element.

  Deletions are remotions of one or more nucleotides from a DNA sequence and, as a consequence, they can alter the reading frame of the gene in which they happen.

2.1.4 Importance of Transposable Elements

The fact that more than 40% of human genome is made up of transposon raises an important question regarding their importance in the genome [nat01].

TEs can be the source of genome construction and at the same time destruction [BHD08]. TEs can damage the genome of their host cell in different ways, namely by:

- Insertions: TEs can insert themselves into functional genes and disable them. This process can cause numerous diseases depending on the TE. Among these diseases are hemophilia A and B, colon Cancer or even Cystic fibrosis. An example can be seen in Figure 2.2.

- Deletions: if a transposon leaves a gene, the resulting gap may not be repaired correctly.

- TE multiplication: multiple copies of the same sequence can hinder precise chromosomal pairing during mitosis and meiosis, resulting in unequal crossovers, one of the main reasons for chromosome duplication.

Figure 2.2: Example of an insertion of a TE on a DNA segment in four steps

On the other hand, insertion of transposons is accompanied by the duplication of a short flanking sequence of a few base pairs. Transposons excise imprecisely, generally leaving part of the duplication at the former insertion site and the consequences of the insertion and excision depend on the location within the coding sequence. This commonly results in either an altered gene product or a frame-shift mutation. Ultimately there can be generated an exon shuffling. Exon (coding section of an RNA transcript, or the DNA encoding it, which are translated into a protein) shuffling results in the juxtaposition of two previously unrelated exons, usually by transposition, thereby potentially creating novel gene products [MDK99].Transposons also can promote illegitimate recombination of large DNA segments movement of large segments of DNA either by transposition or by illegitimate recombination [FRI86].

TEs are not all active. In fact, most TE are silent and do not actively move around the genome in which they are inserted. Some silenced TEs are inactive because they have mutations that affect their ability to move from one chromosomal location to another while others are perfectly intact and capable of moving but are kept inactive by defense mechanisms such as DNA methylation, chromatin remodeling, and miRNAs [Pra08].

Transposons have, therefore, the ability to increase genetic diversity. Adding to this the fact that most TE are inhibited by the genome, results in a balance that makes TEs an important part of evolution and gene regulation in all organisms that carry these sequences.

2.1.5 Transposable Elements Classification

In the past, TE classification was based in a wide variety of factors. Although useful, this approach was limited due to the fact that many of the newly identified transposons do not contain the signature structural elements that are found in the earlier classes of transposon. In an era of large-scale genome sequencing, in which new elements are being described from diverse organisms at an unprecedented rate, a better way of categorizing TEs is by how they transpose [CD03].

In an attempt to introduce a universal classification scheme, Wicker *et al* [WSHV+07] defined a classification scheme based on the transposition process and [KJ08] implemented in Repbase, which is is a large database of eukaryotic repetitive and TEs.

Using this criteria, transposons are first assigned in one of two classes. If their mechanism of transposition is of "copy and paste" they are classified as Retrotransposons. If the mechanism of transposition is of "cut and paste" they are classified as DNA transposons. The way they move within a genome is dictated by their transposase proteins which can be: DDE-transposases, rolling-circle (RC) or Y2-transposases, tyrosine (Y)-transposases, serine (S)-transposases and a combination of reverse transcriptase and endonuclease (RT/En).

Depending on the transposase used in the transposition process, they belong to one of five major classes: long terminal repeat (LTR) retrotransposons, non-LTR retrotransposons, cut-and-paste DNA transposons, rolling-circle DNA transposons (Helitrons) and self-synthesizing DNA transposons (Polintons). Each of these classes of TE are composed by a small number of superfamilies or clades and each superfamily is composed by many families of TE [KJ08].

2.1.5.1 Retrotransposons

Retrotransposons, Class I transposons or simply "copy and paste" generate a copy of their DNA (cDNA) by reverse transcription of their RNA genome. The cDNA is then inserted into the genome in a new position. Reverse transcription is catalyzed by a reverse transcriptase, which is often coded by the TE itself .

These elements can be divided in three classes according to the enzymatic activities they encode in addition to reverse transcriptase [CCGL02].

Long Terminal Repeats (LTR) and retroviruses have opted DDE-transposases to integrate the cDNA. The transcription process described in the next figure.

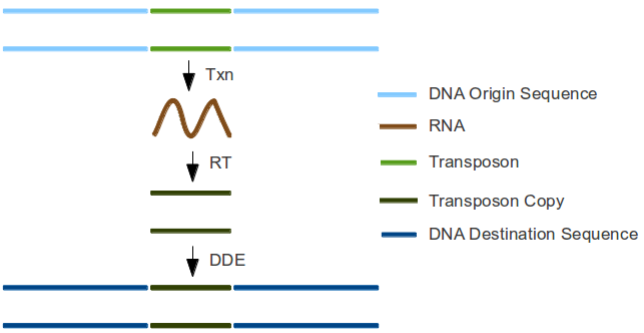

Figure 2.3: Retrotransposons copy-out themselves by reverse-transcribing a full-length copy of their RNA, generated by transcription (Txn). They make a cDNA from their RNA and integrate this into a target using a DDE-transposase.

DIRS1 use Y-transposases and do not have LTRs. The insertion process is described in the following picture.

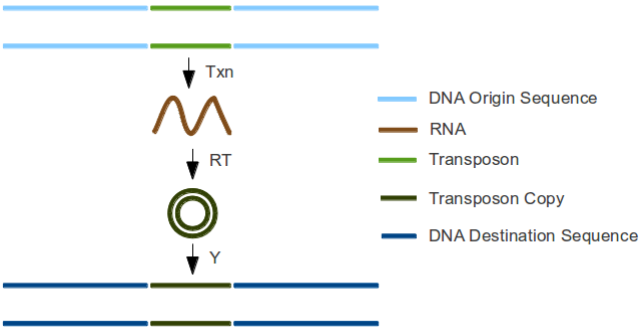

Figure 2.4: Y-retrotransposons copy-out themselves by reverse-transcribing a full-length copy of their RNA, generated by transcription (Txn). They generate a circular cDNA intermediate by reverse transcription and a Y-transposase integrates the element into the target.

The third class of is non-LTR-retrotransposons, also called TP-retrotransposons that use a combination of reverse transcriptase and endonuclease (RT/En) to transpose. These transposons lack terminal repeats but often have A-rich sequence at their 3' end. The process of their transposition is described in the following picture.

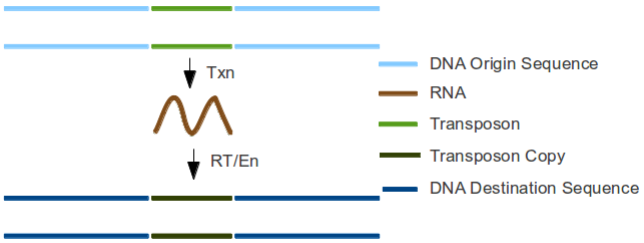

Figure 2.5: TP-retrotransposons use reverse transcriptase to copy their RNA directly into a target that has been cut by a transposon-encoded nuclease.

This category consists of two sub-types [Ala02]:

- LINEs: Long Interspersed Elements(LINEs) encode reverse transcriptase and lack LTRs. They are transcribed to an RNA using an RNA polymerase II promoter that resides inside the LINE.
- SINEs: Short Interspersed Elements (SINEs) are short DNA sequences that don't encode a functional reverse transcriptase protein that are transcribed by RNA polymerase III.

2.1.5.2 DNA transposons

DNA transposons, Class II transposons or simple "cut and paste" do not involve an RNA intermediate in the transposition process. The transposase makes a staggered cut at the target site producing sticky ends, cuts out the DNA transposon and ligates it into the target site. A DNA polymerase fills in the resulting gaps from the sticky ends and DNA ligase closes the sugar-phosphate backbone.

This results in target site duplication and the insertion sites of DNA transposons may be identified by short direct repeats followed by inverted repeats [CD03].

Due to the nature of the transposition process, these transposons have unique characteristics: they create target site duplications upon insertion, they have an ORF containing the catalyst domain for transposase and they have Terminal Inverted Repeats (TIRs) in the extremities.

They can be classified as either "autonomous" or "non-autonomous". The autonomous ones have an intact gene that encodes an active transposase enzyme; the TE does not need another source of transposase for its transposition. In contrast, non-autonomous elements encode defective polypeptides and accordingly require transposase from another source.

Miniature Inverted-repeat Transposable Elements (MITEs) are non-autonomous DNA transposons that are relatively small and share the TIR sequence motifs with other DNA transposons. They are abundant in the non-coding regions of the genes of many plant and animal species [HW10].

DNA transposons may be duplicated if transposition takes place during S phase of the cell cycle when the donor site has already been replicated, but the "target" site has not.

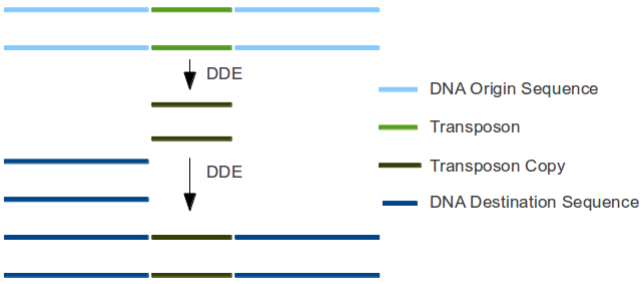

Figure 2.6: In general, DNA transposons excise from the flanking DNA to generate an excised linear transposon, which is the substrate for integration into a destination sequence.

2.2 Machine Learning

In this section it is described the concept of Machine Learning and its applicability in Data Mining problems. The algorithms more relevant for this dissertation are characterized and described.

2.2.1 Introduction

Solving complex computing problems does require intelligence. The learning process is crucial in building intelligence as it is a source of knowledge and it is in this context that the Machine learning concept becomes relevant.According to the Tom M. Mitchell [Mit97] definition, a Machine Learning program can be described as follows:

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine learning is a type of artificial intelligence that provides computers with the capability of acquiring and integrating the knowledge automatically.

In the acquiring process, it is expected that it is used a set of examples (training data) from which it can extract regularities.

2.2.2 Datasets as Machine Learning inputs

A Machine Learning method takes as an input a training dataset in which it will work on. In this dataset it is needed to clarify four different concepts [WF05]:

- Instance: an instance is a single example in a data set. For instance, a row in 2.1 table.
- Attribute: an aspect of an instance, also called feature. For instance Outlook, Temperature, Humidity in 2.1. Attributes can take categorical or numeric values.
- Value: category that an attribute can take. For instance Sunny, Rainy for the attribute Outlook in 2.1 table.
- Concept: the aim, the label to be learned. In 2.1 table, the answer to Play Sport (Yes/No).

Outlook	Temperature	Humidity	Play Sport?
Sunny	Warm	Low	Yes
Sunny	Cold	Medium	No
Rainy	Cold	Low	No
Sunny	Hot	Medium	Yes
Rainy	Hot	High	Yes
Rainy	Cold	High	No

Table 2.1: Sample table

2.2.3 Algorithm Types

According to [RN03], the type of feedback available for learning is usually the most important factor in determining the nature of the learning problem that the agent faces. The field of machine learning usually distinguishes three cases: supervised, unsupervised, and reinforcement learning.

- Supervised learning: is a learning process from which machine learning can infer a function from supervised (labeled) training data set. Each example in this training data set is formed by an input object with one or more attribute values and a desired output value. These examples are analyzed and it is produced an inferred function, which is called a classifier or a regression function depending on the output being discrete or continuous. The result function should predict the correct output given any valid input and is evaluated regarding it's accuracy of predicting correct outputs.

- Unsupervised learning: is a learning process which tries to learn relations between data components or learning patterns with no specific output values (no labeled data). Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution and therefore this learning process cannot learn what to do due to the fact that there is no information on what is or isn't a desirable output.

- Reinforcement learning: is a learning process which is based on learning from reinforcement. It is based on the notion of maximizing the decisions value by some notion of cumulative reward. Reinforcement learning typically includes the subproblem of learning how the environment works.

2.2.4 Machine Learning Techniques

Following several popular approaches of Machine Learning are presented based on different types of machine learning types described in 2.2.3.

2.2.4.1 Decision Tree learning

Decision tree learning is a method commonly used in data mining, statistics and machine learning. The goal of this learning method is to create a model that predicts the value of a target variable based on several input variables. There are two main types of decisions trees:

- Classification trees: when the predicted outcome is the class to which the data belongs.

- Regression trees: when the predicted outcome can be considered a real number.

To build a decision trees there are several appropriate algorithms. Among them, the most recognized are:

- ID3 Algorithm

- C4.5 Algorithm

Decision tree learning can handle numeric and categorical data and create easy to understand and interpret models. On the other hand, decisions tree can fail to generalize data and may require pruning.

2.2.4.2 Association Rule learning

Association rule learning is a popular method for discovering interesting relations, frequent patterns, associations, correlations, or causal structures among sets of items in large databases.

According to [BJ93] the problem of association rule mining is defined as the follow:

Let I be a set of n binary attributes called items and D be a set of n transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in

I. A rule is defined as an implication of the form X $\rightarrow$ Y where X and Y belong to I. X is called antecedent and Y is called the consequent of the rule.

To select interesting rules from the set of all possible rules, various measures of significance and interest can be used to evaluate their pertinence and the their real value. The most used are:

- Support: denotes the frequency of the rule within transactions. A high value means that the rule involves a great part of database.

$$Support(A \rightarrow B) = p(A \cup B) \tag{2.1}$$

- Confidence: denotes the percentage of transactions containing A which also contain B. It is an estimation of conditioned probability.

$$Confidence(A \rightarrow B) = p(B|A) \tag{2.2}$$

- Lift: denotes the ratio of the observed support to that expected if A and B were independent.

$$Lift(A \rightarrow B) = \frac{p(B|A)}{p(B)} \tag{2.3}$$

2.2.4.3 Artificial neural networks

Artificial neural networks are learning algorithms that are inspired by the structure and functional aspects of biological neural networks. These are composed by groups of structures (neurons) which are interconnected and have a relation between ones outputs and other inputs. These models are non-linear and can find patterns in the dataset where they are applied. The main advantages of using neural networks are their ability to be used as an arbitrary function approximation mechanism that is continuously learning from observed data. Although for tunning the learning algorithms to achieve better results can take significant amount of experimentation and if the model chosen is overly complex, neural networks tend to have problems in learning.

2.2.4.4 Inductive logic programming

Inductive logic programming (ILP) is an approach to rule learning using logic programming as a uniform representation for examples, background knowledge, and hypotheses. Based on known background knowledge and a set of examples represented as a logical database of facts, an ILP system will find hypotheses of a certain format that can predict the class labels of target tuples. Although many ILP approaches achieve good classification accuracy, most are not highly scalable due to the computational expense of repeated joins.

2.2.4.5 Support vector machines

Support vector machines are a set of related supervised learning methods used for classification and regression. Given training data, each marked as belonging to one of two categories, an SVM

training algorithm builds a model that predicts whether a new example falls into one category or the other. They use a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane, which is a boundary that separates the tuples of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors and margins.

2.2.4.6 Bayesian networks

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. The nodes of the graph represent random variables which may be observable quantities, latent variables, unknown parameters or hypotheses and edges represent conditional dependencies. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. There are efficient algorithms that can perform inference and learning in Bayesian networks.

2.2.5 Evaluating Learning Performance

One subject of great importance in the Machine Learning process is the evaluation of the performance. Evaluation is the key to determine which is the best solution to a particular problem.

There are several measures to evaluate the fitness of models, namely:

- Accuracy: the proportion of correct predictions, both true positives and true negatives in the set of examples considered. It is defined as:

$$accuracy = \frac{truepositives + truenegatives}{Totalnumberofexamples} \tag{2.4}$$

- Precision: the proportion of outcomes predicted correctly in all the outcomes predicted. It is defined as:

$$precision = \frac{CorrectclassifiedexamplesasX}{TotalexamplesclassifiedasX} \tag{2.5}$$

- Recall: the proportion of outcomes that are correct and were, in fact, retrieved. It is defined as:

$$recall = \frac{CorrectclassifiedexamplesasX}{Totalexamplesclassified} \tag{2.6}$$

- F-measure: a weighted harmonic mean of precision and recall. It is defined as:

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{2.7}$$

While high recall means that an algorithm returned most of the relevant results, high precision means that an algorithm returned more relevant results than irrelevant.

It is often need to compare two different learning methods on the same problem to see which is the better one to use. The model should be able to generalize, deal with new data and do not overfit.

In order to evaluate this, an approach partitions the data into a training and test sets. The model is trained using one partition (training set). As the model is created, it should be tested with the training and the test sets and evaluation metrics referenced in **??** should be calculated. The results of the evaluation using the training set and the test set are then compared to measure the model's ability to generalize.

Since this process requires a training and a test sets, one possible approach is using Cross-validation. Cross-validation is an approach to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available. It optimizes the model parameters to make it fit the training data as well as possible so the model can fit well in the test data, avoiding the overfitting problem. Overfitting is when a model works best for the training data than for the test data and therefore the result model is not able to generalize well. This problem often happens when the size of the training set is small or when there are many parameters in the model.

The first step in the cross-validation involves partitioning the input data in two subsets: the training and the test sets. In k-fold cross validation, this process is then repeated k times (the folds). The input data set is randomly partitioned into k subsets and of the k subsets, one is used as the validation data for testing the model, and the remaining k-1 are used as training data, with each of the k subsamples being used only once as the validation data. The k results from the folds are then compared. The most common fold number is 10 [WF05].

2.3 Conclusions

This Chapter presented some background information regarding the TEs, mainly support concepts for the aim of this project. It followed by giving a general overview of what are Machine Learning techniques and their importance in the scope of this work.

The next Chapter, chapter 3, uses the definitions and the background introduced in these sections to introduce transposon detection methodologies and tools.

Chapter 3

State of the Art

This Chapter describes several studies and approaches that were made regarding the discovery and annotation of TEs in genome sequences. It reviews the principles of transposon discovery and the different methods available, based on the different studies and approaches made through the years. Adding to this, the technologies and tools which will be used further in this work are described and their choice is justified.

The section 3.2 describes the main TEs detection methodologies, referring how they discover and what do they need to do this and in section 3.2 transposon detection tools are described.

3.1 Transposon Detection Methodologies

In this section all the transposon detection methodologies are described. A recent work of Saha et al. [SBMP08] classifies methodologies based upon the algorithms they use to identify TEs in three categories:

- Library-based Techniques: identify known repetitive sequences by comparing input datasets against a set of TEs;
- Signature-based Techniques: identify TEs based on amino acids sequence and spacial arrangements based on *a priori* knowledge of particular transposon types;
- Ab initio Repeat Techniques: identify repetitive elements without using reference sequences or known transposons in the process. It is a two step process where first the algorithms identify repetitive sequences and then they identify the boundaries of the repeats and extraction of the consensus sequence for each family;

Although this is a valid classification, in this work the classification of transposon detection methodologies is according the view of Bergman and Quesneville [BQ07] since this article is more cited than Saha's article. Bergman and Quesneville classify the methods in four categories:

- *De novo*: this approach looks for similar sequences found at multiple positions within a sequence;
- Structure-based: this approach finds TEs using knowledge from their structure;
- Comparative Genomics: compares genomes to find insertion regions which can be TEs or caused by TEs;
- Homology-based: uses known TEs as a mean to discover TE in genome sequences.

On Table 3.1 it is shown different available transposon detection tools.

Name	Type	Operating System	URL
CENSOR[JKDP96]	Homology - based	Unix and online	http://www.girinst.org/censor/download.php
LTR_Finder[XW07]	Structure - based	Unix	http://tlife.fudan.edu.cn/ltr_finder/
MGEScan[RT09]	*De novo*	Unix	http://darwin.informatics.indiana.edu/cgi-bin/evolution/nonltr/nonltr.pl
MITE-Hunter[HW10]	Structure-based	Unix	http://target.iplantcollaborative.org/mite_hunter.html
PILER[EM05]	*De novo*	Unix	http://www.drive5.com/piler/
RECON[BE02]	*De novo*	Unix	http://selab.janelia.org/recon.html
REPEATFinder[VHS01]	*De novo*	Unix	http://cbcb.umd.edu/software/RepeatFinder/
RepeatMasker[Smi]	Homology - based	Unix and online	http://www.repeatmasker.org/
TEseeker[KUC+11]	Homology - based	Unix	http://repository.library.nd.edu/view/16/index.html
VariationHunter[HHD+10]	Structure - based	Unix	http://compbio.cs.sfu.ca/strvar.htm

Table 3.1: Transposable Elements Detection Tools

Methods are now described in detail.

3.1.1 *De novo*

De novo methods are based on the assumption that similar sequences found at multiple positions within a genome have a strong possibility of being TEs. As such, the main challenges for *de novo* methods are to distinguish TEs from all other repeat classes. This intrinsic repetition of transposons in genome sequences makes these methods very effective in finding TEs with high prevalence and not very effective in identifying degraded TEs or transposons with low copy number inside a genome (one or two copies) [KUC+11].

Initially these methods start by using computational strategies such as suffix trees or pairwise similarity to detect repeat regions. With these they try to detect pairs of similar sequences at

different locations in a given genome. Typically *de novo* repeat discovery methods use assembled sequence data, and therefore are critically dependent on both sequencing and assembly strategies.

The second step in *de novo* approach is to cluster pairs of aligned DNA segments into repeat families and filter out the ones that are not TEs. To filter out the false-positives clusters which correspondes to repetitive elements that are not transposons, there are two different approaches. The first approach considers the multiple alignments of all repeat copies of a cluster. Multiple alignments are split into 'sub-clusters' when several sequences end at the same, or similar, position in the alignment, as expected for interspersed repeats that arise by the process of transposition. Sequences are split according to these boundaries and the underlying alignment pairs are re-clustered. In this way, nested repeats can be detected and separated from one another. Moreover,they will be splitted in 'sub-clusters' according to the presence of long non-conserved regions (i.e. mismatching regions) between instances to deal with interfamily similarity between closely related TE families. The second approach tries to find complete copies of the repeat among all instances of a family.It is searched for the longest sequences in a cluster and filtered according to their occurrence and if it is an active transposon family, there will be at least a few copies found, normally more than three [BQ07].

3.1.2 Homology-based

Homology based methods use known TEs (of other species) to detect through homology (similarity) new TEs in a given genome. Obviously, homology-based methods are biased towards the detection of previously identified TE families and to TEs active recently enough to retain substantial protein homology.

These methods have strong advantages in finding known transposons and degraded transposons although they fail to recognize transposons unrelated to the known ones. They are also not applicable to certain classes of TEs that are composed entirely of non-coding sequences, such as miniature inverted repeat TEs (MITEs) and short interspersed nuclear elements (SINEs) [KUC+11] [BQ07].

One common approach is to align a genome using fast alignment algorithms with known TEs and analyze the results of the hits. Another known approach uses hidden Markov models to detect common TEs. This second approach, despite being effective for closely related genomes, fails when used in distantly related species.

3.1.3 Structure based

Structure based Methods use prior knowledge about the common structural features shared by different TEs that are necessary for the process of transposition, such as long terminal repeats. These methods rely on detecting specific models of TE architecture, rather than just the expected results of the transposition process, making them less sensitive to similarity between the sequences and the known transposons. Like homology-based methods, these can also detect low copy number families of transposons. Purely structure-based methods are limited by the fact that specific models

must be designed and implemented for each type of transposon in analysis and that some classes of transposons have stronger structure characteristics than others and, therefore, are more easily detected using these kinds of methods [BQ07].

Several structure-based methods have been developed recently to detect LTR retrotransposons, by searching for the common structural signals in this subclass of TE—LTRs, target site duplications (TSDs), primer-binding sites (PBSs), polypurine tracts (PPTs) and ORFs for the gag, pol (containing the RT domain) and/or env genes [XW07] [HW10].

3.1.4 Comparative Genomics

Firstly described by Caspi and Pachter [CP06], comparative genomics methods use multiple alignments of closely related genomes to detect large changes between genomes. The idea behind this method is that insertion regions can be TEs or caused by TEs.

These methods are based on the fact that transposition creates large insertions that can be detected in multiple sequence alignments [BQ07]. They start by searching for insertions regions in whole-genome multiple alignments where there are disrupted sequences by large insertions, normally more than 200 bp. After applying filtering and concatenating the insertion regions are locally aligned with all other insertion regions to identify repeat insertion regions.

This approach can identify new transposon families and instances although it is dependent on the quality of whole genome alignments, which can be compounded by the multiple alignment of draft genomes.

3.2 Transposon Detection Tools

In this section the software tools that will be used on this work are described and grouped accordingly to Bergman and Quesneville [BQ07], who classify the methods in four categories: *de novo*, structured-based, comparative genomic and homology-based.

In order for a tool to be suitable for this project, it must meet certain demands:

- Must be an open source tool;
- Must be publicly available;
- Must be runnable from command-line;
- It must be supported by scientific studies.

As all this requirements are meet, that tool can be used in the aim of this work.

3.2.0.1 *De novo* Tools

In the 3.2 table there are presented the tools that will be used in this work that implement *De novo* methodologies for detecting TEs.

Name	Description
MGEScan[RT09]	A computational tool for the identification of non-LTR retrotransposons in genomic sequences, following a computational approach inspired by a generalized hidden Markov model.
PILER[EM05]	A package of efficient search algorithms for identifying characteristic patterns of local alignments induced by certain classes of repeats.
RECON[BE02]	A computational tool that uses *De novo* methodologies based on extensions to the usual approach of single linkage clustering of local pairwise alignments between genomic sequences.
REPEATFinder[VHS01]	A computacional tool for clustering and analysis of the repeat data captured in suffix trees. Finds repeats in individual genome sequences or sets of sequences and accurately creates repeat databases from small and large genomes.

Table 3.2: *De novo* Tools

3.2.0.2 Homology-Based Tools

In the 3.3 table there are presented the tools that will be used in this work that implement Homology-Based methodologies for detecting TEs.

Name	Description
TEseeker[KUC+11]	An automated homology-based approach for identifying TEs.
RepeatMasker[Smi]	Is a program that Screens DNA sequences for interspersed repeats and low complexity DNA sequences. The output of the program is a detailed annotation of the repeats that are present in the query sequence as well as a modified version of the query sequence in which all the annotated repeats have been masked.
CENSOR[JKDP96]	Is a software tool designed to identify and eliminate fragments of DNA sequences homologous to any chosen reference sequences, in particular to repetitivte elements.

Table 3.3: Homology-Based Tools

3.2.0.3 Structure-Based Tools

In the 3.4 table there are presented the tools that will be used in this work that implement Structure Based methodologies for detecting TEs.

Name	Description
LTR_Finder[XW07]	Given DNA sequences, it predicts locations and structure of full-length LTR retrotransposons accurately by considering common structural features.
MITE-Hunter[HW10]	A program pipeline that can identify MITEs as well as other small DNA transposons from genomic DNA data sets.
VariationHunter[HHD+10]	A tool that uses combinatorial algorithms to detect transposon insertions.

Table 3.4: Structure-Based Tools

Starting with *de novo* tools, MGEScan is a tool for the identification of non-LTR retrotransposons in genomic sequences, following a computational approach inspired by a generalized hidden Markov model. Three different states represent two different protein domains and inter-domain linker regions encoded in the non-LTR retrotransposons, and their scores are evaluated by using profile hidden Markov models (for protein domains) and Gaussian Bayes classifiers (for linker regions), respectively. It is implemented in three modules: the first prepares the input for evaluation process by locating the signals of domains in a given genomic sequence. The second module finds the optimal path of states, which corresponds to the annotation of the protein domains and linker regions in the clades. The third module post-processes the results and reports the coordinates and sequences of non-LTR retrotransposons classified into specific clades. Since this method focuses on the identification of elements which retain the protein domains, the results include elements containing all the domains in ORFs, and those containing one of the domains but with partially truncated 5'- or 3'- ends [RT09].

PILER adopts a novel heuristic-based approach to *de novo* repeat annotation that exploits characteristic patterns of local alignments induced by certain classes of repeats. The PILER algorithm is designed to analyze assembled genomic regions and find only repeat families whose structure is characteristic of known subclasses of repetitive sequences. PILER works on the premise that the entire DNA sequence is assembled with a reasonably low number of errors because the algorithm is completely dependent on the position of repeats in the genome for all classification. The output of the clustering step is recorded in terms of start and end coordinates. Similar elements are then clustered into piles, which are sets of overlapping hits [EM05].

RECON starts to do the initial all-versus-all comparison of the genome and using the datafile containing pairwise alignments. It uses multiple alignment information to define the boundaries of individual copies of the repeats and to distinguish homologous but distinct repeat element families. It is not useful for processing short-period tandem repeats since because it discard short elements generated during splitting [BE02].

REPEATFinder relies on suffix tree data structure to detect exact repeats and merges neighboring and overlapping exact repeats to detect non-exact repeats [VHS01].

Regarding the homology-based tools, TEseeker begins with BLAST searches against the genome using representative TEs for the chosen family. The resulting BLAST hits are then combined if they overlap or are very close together and are then extracted from the genome. It assembles with CAP3 and runs another BLAST search against the genome and process the hits in the same manner. This second run, when extracting the sequences from the genome, it adds flanking regions. The length of the flanking region is dependent on the type of transposon and is utilized to enable the capture of the entire transposon. The results are then aligned and a consensus is generated. This consensus is used to perform a final BLAST search, again combining, extracting, and assembling the sequences. CAP3 then produces the high-quality, full- length consensus TEs [KUC+11].

RepeatMasker discovers repeats and removes them so as to prevent complications in sequence assembly and gene characterization. This tool includes a set of statistically optimal scoring matrices permitting estimation of the divergence of query sequences compared to a curated repeat library. A search engine such as BLAST, WU-BLAST, or Crossmatch is utilized in the comparison process. The degree of similarity required for a query sequence to be paired with a reference sequence can be specified by the user. Identification of repeats by RepeatMasker is based entirely upon shared similarity between library repeat sequences and query sequences. The output of the program is a detailed annotation of the repeats that are present in the query sequence as well as a modified version of the query sequence in which all the annotated repeats have been masked [Smi] .

CENSOR is a program designed to identify and eliminate fragments of DNA sequences homologous to any chosen reference sequences. It uses BLAST to identify matches between input sequences and a reference library of known repetitive sequences. The length and number of gaps in both the query and library sequences are considered along with the length of the alignment in generating similarity scores. This tool reports the positions of the matching regions of the query sequence along with their classification [SBMP08] [JKDP96].

In the structure-based tools we have LTR_Finder. Given DNA sequences, it predicts locations and structure of full-length LTR retrotransposons accurately by considering common structural features. LTR_FINDER identifies full-length LTR element models in genomic sequence in four main steps. It first selects possible LTR pairs by searching for all exactly matched string pairs in the input sequence by a linear time suffix-array algorithm. Then it selects pairs of which distances and overall sizes satisfy given restrictions. It is calculated the distances and then it is recorded as an LTR candidate for further analysis. After that, Smith–Waterman algorithm is used to adjust the near-end regions of LTR candidates to get alignment boundaries. At the end of this step, a set of regions in the input sequence is marked as possible loci for further verification. Secondly, LTR_FINDER tries to find signals in near LTR regions inside these loci. The program detects PBS by aligning these regions to the 30tail of tRNAs and PPT by counting purines in a 15 bp sliding window along these regions. This step produces reliable candidates. In the end, this program reports possible LTR retrotransposon models at different confidence levels [XW07].

MITE-Hunter, a program pipeline that can identify MITEs as well as other small Class 2 non- autonomous TEs from genomic DNA data sets. It starts by identifying transposon elements candidates and then filters the false-positives based on the PSA. Then it selects TEs examples from each group and filters the false-positives based on the MSA, predicting TSDs and generating consensus sequences. Finally it selects new exemplars and groups TEs into families [HW10].

VariationHunter is a structural variation discovery algorithm that aims to improve both the sensitivity and specificity of structural variation detection. It uses combinatorial algorithms to detect transposon insertions [HHD+10].

3.3 Conclusion

This Chapter presented the main methodologies used in the process of discovering TEs, mainly their approach to this problem, their main advantages and disadvantages. It also summarized the transposon detection tools that will be used in this work. The following Chapter 4 will describe the experimental setup that will be used in this dissertation.

Chapter 4

Experimental Setup

This chapter presents the experimental setup that will be used in the Master thesis. This includes: the generation of curated data sets; evaluation of existing tools on those data sets; and the application of Machine Learning algorithms to help in the improvement of TEs detection.

In **??** the software tools used are enumerated and the criteria of their selection is clarified. Also it is described how the datasets are generated. The Section 4.2 it is established how the different transposon detection tools are going to be evaluated and the expected outcomes of this evaluation. In Section 4.3 it is described how machine learning methodologies will be used in the aim of this work and how their results will be evaluated. Finally in Section 4.4 we present the chapter's conclusions.

4.1 Experimental Setup

4.1.1 Datasets Generation

In this work it is essential to have curated datasets of genome sequences so that transposon detection tools can can process them and detect and annotate the TEs they find.

The datasets will be composed by genome sequences in FASTA format7 with the annotation of the elements in the sequence (transposon, genes, repeats, ...). The datasets will be as diversified as possible, having distinct genome sequences so the outcomes of transposon detection tools evaluation are as comprehensive as possible.

To this end we will propose and implement a Dataset Simulator as described in 4.1.2. It allows the generation of different genome sequences, varying both in length and composition. These sequences will have transposons in different proportions and variety. In addition, these sequences will have genes and other repetitive elements that are not transposons. Finally these sequences can have mutations, either point mutations or indel mutations. This last condition makes the dataset more in agreement with what real genomes are like.

4.1.2 Dataset Simulator

To build the datasets needed in this dissertation, it is needed a simulator. This simulator will produce sequences that have transposon elements in known positions. In order generate a new sequence, there should be defined these input parameters:

- Sequence Length: length of the output sequence.
- Gene Percentage: % of genes that should be included in the sequence in relation to it's total length.
- Transposon Percentage: % of transposons that should be included in the sequence in relation to it's total length.
- Repetitive Elements Percentage: % of repetitive elements (no transposons included) that should be included in the sequence in relation to it's total length.
- Set of Genes: a set of genes in FASTA format that are used in the creation of this sequence.
- Set of Transposable elements: a set of TEs in FASTA format that are used in the creation of this sequence.
- Set of Repetitive elements: a set of repetitive elements (no transposons included) in FASTA format that are used in the creation of this sequence.
- Rate of Insertions: number of inserted nucleotides per 1000 bp.
- Rate of Deletions: number of deleted nucleotides per 1000 bp.
- Rate of Replacements: number of substituted nucleotides per 1000 bp.

The output of the simulator will be set of sequences, written in FASTA format, and an annotation file containing all TEs and repetitive elements locations inside each sequence.

The result genome sequence consists of genes, transposons and other repetitive elements filled with random nucleotides in the gaps between them. The quantity of TEs, genes and repetitive elements are defined by the Transposon, Gene and Repetitive Elements Percentages respectively. These elements are from the Set of Genes, TEs and Repetitive Elements given in the input parameters.

There is also the possibility for the generated sequence to have point mutations using the Rate of Replacements parameter, replacing a given number of bases per 1000 bp. In the same way, indel mutations [KR04] can also be applied to the result sequence by changing the Rate of Insertions and Deletions parameters.

The sequences generated are personalized accordingly to the situation to be tested. For instance if the aim of the sequence to generate is to evaluate the capacity of the transposon detection tools regarding LTR transposon detection the set of TEs given should consist of elements of this class.

4.1.3 Raw Data

The simulator that produces the artificial dataset requires data related to TEs, genes and repetitive elements that is in FASTA format.

In this dissertation we use a set of genes from FlyBase [Fly] which are real genes annotated from Drosophila Melanogaster, a specie of fly.

The sources for TEs were: Repbase [Rep] which is a database of repetitive DNA elements and from Gydb [Gyd] which is also a cooperative repository of TEs organized and classified by type.

4.2 Evaluation of Transposon Detection Tools

Evaluating the different transposon detection tools is a key step in the scope of this dissertation. Using a dataset generated with the simulator described in 4.1.2, each transposon detection tool will process the dataset and predict the TEs.

Each tool analyzes all the dataset sequences and produces as a result the annotations of TEs and repetitive elements. These annotations are then compared with the annotations generated by the simulator and it is measured the general accuracy of that tool in detecting TEs. Furthermore it enables a more in-depth analysis regarding the tool's capacity in detecting specific sequences generated by the simulator.

his information will be quite valuable for the combination of the tools.

4.3 Model generation and Evaluation of the Results

This work involves machine learning and data mining over the results of the transposon detection. To learn the models it will be used Weka [oW], a collection of machine learning algorithms for data mining tasks. This software contains a set of tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Each example in this dataset will have a set of features that are used to describe it. The sequence length, the types and number of TEs present, the mutations present in the sequence and the results of the tools in discovering TEs in it are some of the most relevant.

Using the sequences features it is possible to extrapolate association rules and identify relevant patterns regarding transposon detection. Also, each transposons detection tools will be run with different input parameters and the results are compared so one can identify what are the values for those parameters that maximize the transposon detection accuracy.

The results gathered by evaluating all the transposon detection tools with the curated dataset will be used as a data source in this stage. Each example from this dataset will have information regarding the transposon features, tools' parameters and identification results.

Since the aim of this dissertation is to integrate transposon detection tools, several machine learning algorithms will analyze these datasets. Each algorithm will be subject to a cross-validation process.

After this, all the models are compared and the one with the best overall performance is chosen as a potential solution of integration of different transposon detection methodologies.

4.4 Conclusion

It is now clear how the genome sequence dataset will be generated and what is its aim. The dataset simulator was described and so was the raw data that will be used in this process.

The process of transposon detection tools' evaluation was described and the expected outputs were identified. This chapter also concluded how machine learning methodologies will be used in the scope of the dissertation to achieve better overall results in TEs detection.

Chapter 5

Workplan

This chapter describes the main tasks of this thesis in 5.1 and then the work plan throught the ahead semester in 5.2.

5.1 Main Tasks

This work is divided in eight different tasks from writing state of the art to writing the final thesis.

The first task is to write te state of the art of transposable element discovery and annotation, planning the work and defining objectives and evaluation measures for them. After this, it is important to create a simulator to generate artificial data in order to create datasets where we have genome sequences with transposable elements inserted in known positions. As a result, the next major step is to use this simulator to generate the datasets of genome sequences. This is done by giving this simulator transposable elements, genome sequences and input parameters related to the data that is needed to be generated. Having the datasets ready, the next big step is evaluating the different transposable elements discovery tools and annotate the results of this step. After this, the results are analyzed accordingly to the aims of this work. Once the analysis is done, it is time to use machine learning techniques to combine and improve transposon detection. Finally, the solution is evaluated and the writing of the thesis is finallized.

5.2 Scheduling

In figure 5.1 it is presented a gantt chart with the scheduling of all the major tasks of the work assigned throught the weeks from January 2012 to July 2012.

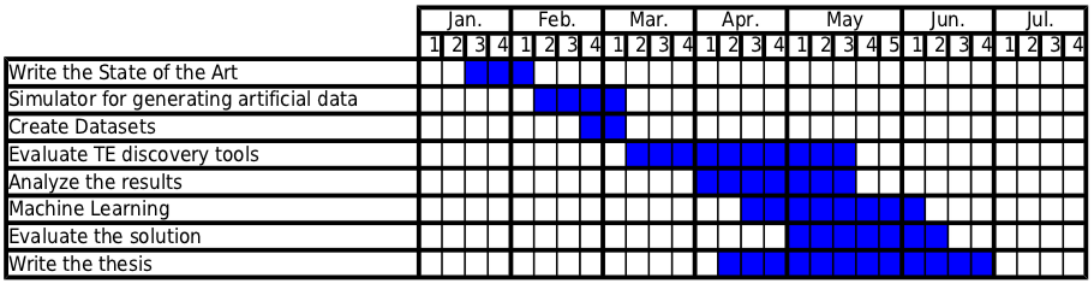

Figure 5.1: Gantt Chart for the semester

Chapter 6

Conclusions and Future Work

This chapter concludes this report emphasizing the importance of the problem and the potential solution.

6.1 Conclusion

As we have seen transposable elements are very important entities to be studied as they have preponderant roles in the genome evolution. Because of their importance, various approaches were made in order to identify and annotate them, although none is fully capable of detecting all the different types of transposons.

Considering the need for a more complete tool that has a better overall accuracy in TE detection, this dissertation main aim is to combine different transposon detection tools and provide a tool that is capable of detecting TEs with a better accuracy.

This dissertation will focus in three main steps: first it will be used a dataset simulator that will generate curated DNA sequences with TEs within them. The next step will be using the generated dataset to evaluate the transposon detection tools accuracy. Using the results of these evaluations, the final step is to use Machine Learning techniques to integrate different tools and combine their strengths to improve TE detection.

References

[Ala02] Weiner Alan M. SINEs and LINEs: the art of biting the hand that feeds you. *Current Opinion in Cell Biology*, 14(3):343–350, June 2002.

[Ame05] *The American Heritage Science Dictionary*. Houghton Mifflin Harcourt, 2005.

[BE02] Zhirong Bao and Sean R Eddy. Automated De Novo Identification of Repeat Sequence Families in Sequenced Genomes. *Genome Research*, 12(8):1269–1276, 2002.

[BHD08] Victoria P Belancio, Dale J Hedges, and Prescott Deininger. Mammalian non-LTR retrotransposons: For better or worse, in sickness and in health. *Genome Research*, 18(3):343–358, 2008.

[BJ93] Peter Buneman and Sushil Jajodia, editors. *Mining Association Rules between Sets of Items in Large Databases*, Washington, D.C., 1993.

[BQ07] Casey M Bergman and Hadi Quesneville. Discovering and detecting transposable elements in genome sequences. *Briefings in Bioinformatics*, 8(6):382–392, November 2007.

[CCGL02] N L Craig, R Craigie, M Gellert, and A M Lambowitz. *{M}obile {D}{N}{A} II*. ASM Press, 2002.

[CD03] M Joan Curcio and Keith M Derbyshire. The outs and ins of transposition: from Mu to Kangaroo. *Nat Rev Mol Cell Biol*, 4(11):865–877, November 2003.

[CP06] A Caspi and L Pachter. Identification of transposable elements using multiple alignments of related genomes. *Genome research*, 16(2):260–270, February 2006.

[EM05] Robert C Edgar and Eugene W Myers. PILER: identification and classification of genomic repeats. *Bioinformatics*, 21(suppl 1):i152–i158, 2005.

[Fly] FlyBase.

[FRI86] TREVOR J. FRIED, MIKE and WILLIAMS. Inverted Duplication-Transposition Event in Mammalian Cells at an Illegitimate Recombination Join. *MOLECULAR AND CELLULAR BIOLOGY*, 6(6):2179–2184, 1986.

[GP08] Graziano and Pesole. What is a gene? An updated operational definition. *Gene*, 417(1–2):1–4, 2008.

[Gyd] Gydb.

[HHD+10] F Hormozdiari, I Hajirasouliha, P Dao, F Hach, D Yorukoglu, C Alkan, E E Eichler, and S C Sahinalp. Next-generation VariationHunter: Combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, 26(12):i350–i357, 2010.

[HW10] Yujun Han and Susan R Wessler. MITE-Hunter: a program for discovering miniature inverted-repeat transposable elements from genomic sequences. *Nucleic Acids Research*, 2010.

[Inf07] National Center for Biotechnology Information. Blast Help: FASTA, 2007.

[Ins] European Bioinformatics Institute. About Nucleotide And Protein Sequence Formats.

[JKDP96] Jerzy Jurka, Paul Klonowski, Vadim Dagman, and Paul Pelton. Censor—a program for identification and elimination of repetitive elements from DNA sequences. *Computers & Chemistry*, 20(1):119–121, 1996.

[KJ08] Vladimir V Kapitonov and Jerzy Jurka. A universal classification of eukaryotic transposable elements implemented in Repbase. *Nat Rev Genet*, 9(5):411–412, May 2008.

[KR04] Alexey S Kondrashov and Igor B Rogozin. Context of deletions and insertions in human coding sequences. *Human Mutation*, 23(2):177–185, 2004.

[KUC+11] Ryan Kennedy, Maria Unger, Scott Christley, Frank Collins, and Gregory Madey. An automated homology-based approach for identifying transposable elements. *BMC Bioinformatics*, 12(1):130, 2011.

[Lab04] Göttingen Genomics Lab. FASTA format description, 2004.

[McC50] B McCLINTOCK. The origin and behavior of mutable loci in maize. *Proceedings of the National Academy of Sciences of the United States of America*, 36(6):344–355, June 1950.

[MDK99] J V Moran, R J DeBerardinis, and H H Kazazian. Exon shuffling by L1 retrotransposition. *Science*, 283(5407):1530–1534, 1999.

[Mit97] Tom Michael Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. WCB/McGraw-Hill, Boston, MA, 1997.

[nat01] Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, February 2001.

[oW] University of Waikato. Weka.

[Pen07] E Pennisi. Genomics. DNA study forces rethink of what it means to be a gene. *Science (New York, N.Y.)*, 316(5831):1556–1557, June 2007.

[Pra08] Leslie A. Pray. Transposons: The Jumping Genes. *Nature Education*, 2008.

[Ref12] Genetics Home Reference. *Handbook: Help Me Understand Genetics*. Genetics Home Reference, 2012.

[Rep] Repbase.

REFERENCES

[RN03] Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[RT09] Mina Rho and Haixu Tang. MGEScan-non-LTR: computational identification and classification of autonomous non-LTR retrotransposons in eukaryotic genomes. *Nucleic Acids Research*, 37(21):e143, 2009.

[SBMP08] Surya Saha, Susan Bridges, Zenaida Magbanua, and Daniel Peterson. Computational Approaches and Tools Used in Identification of Dispersed Repetitive DNA Sequences. *Tropical Plant Biology*, 1(1):85–96, March 2008.

[Smi] Green P Smit AFA, Hubley R. RepeatMasker Open-3.0.

[VHS01] N Volfovsky, B J Haas, and S L Salzberg. A clustering method for repeat analysis in DNA sequences. *Genome Biol*, 2(8):RESEARCH0027+, 2001.

[WC53] J D Watson and F H C Crick. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[WF05] Ian H Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, San Francisco, CA, 2nd edition, 2005.

[WSHV+07] Thomas Wicker, Francois Sabot, Aurelie Hua-Van, Jeffrey L Bennetzen, Pierre Capy, Boulos Chalhoub, Andrew Flavell, Philippe Leroy, Michele Morgante, Olivier Panaud, Etienne Paux, Phillip SanMiguel, and Alan H Schulman. A unified classification system for eukaryotic transposable elements. *Nat Rev Genet*, 8(12):973–982, December 2007.

[XW07] Zhao Xu and Hao Wang. LTR_FINDER: an efficient tool for the prediction of full-length LTR retrotransposons. *Nucleic Acids Research*, 35(suppl 2):W265–W268, 2007.

REFERENCES

Chapter 7

FASTA format

According to [Lab04] [Ins] [Inf07] this format each sequence consists of a single header line started with ">" providing the sequence name and, optionally, a description followed by lines of sequence data. There should be no space between the ">" and the first letter of the identifier and usually each line is formated to 60 characters long and should be no longer than 80 characters. Lines started with a semicolon are ignored as they are treated as comments. Bellow is an example of a DNA sequence in FASTA format.

```
>Name and description
GTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCACAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCCAGCAATCCGCGCGCCGGGACAGAATG
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

In FASTA files containing multiple sequences, one sequence ends when a ">" appears to mark the start of another one. In the scope of this work, the sequences are filled with combinations of four letters representing the four nucleotids present in the DNA sequences:

- A - Adenine
- C - Guanine
- G - Cytosine
- T - Thymine