REPCLASS: CLUSTER AND GRID ENABLED AUTOMATIC CLASSIFICATION OF TRANSPOSABLE ELEMENTS IDENTIFIED DE NOVO IN GENOME SEQUENCES

by

NIRMAL RANGANATHAN

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

ACKNOWLEDGEMENTS

I would like to thank my supervising professor, Mr. David Levine, Department of Computer Science and Engineering for his constant encouragement and support, providing me guidelines in my research work.

I would like to thank Dr. Cedric Feschotte, Department of Biology for having provided constant encouragement and direction in my thesis work and for having a great deal of patience in explaining the intricacies of the Biology involved.

I would like to thank Dr. Nikola Stojanovic, Department of Computer Science and Engineering for serving on my committee.

I would like to thank Patrick Mcguigan for helping out in troubleshooting issues on the cluster and providing all the extra resources I needed.

I would like to thank my friends who provided constant support and helped me with my other obligations.

Finally, my sincerest gratitude to Mum and Dad who have always been there for me, providing constant encouragement and support.

November 11, 2005

ABSTRACT

REPCLASS: CLUSTER AND GRID ENABLED AUTOMATIC CLASSIFICATION OF TRANSPOSABLE ELEMENTS IDENTIFIED DE NOVO IN GENOME SEQUENCES

Publication No. _____

Nirmal Ranganathan, M.S.

The University of Texas at Arlington, 2005

Supervising Professor: David Levine

In the last few years many computer and laboratory improvements in the production and analysis of DNA sequences have made possible the complete sequencing of whole genomes. This provides us with a wealth of raw genomes that needs to be processed and annotated. 5% to 80% of eukaryotic genomes contain repetitive DNA consisting of transposable elements and tandem repeats which needs to be identified, classified and annotated in order to sequence and annotate the entire genome accurately.

Existing tools allow us to identify and annotate transposable elements (TE) but no tool exists for their classification. This thesis work introduces REPCLASS an

automated tool for the classification of transposable elements that are identified de novo in new genomes. REPCLASS consists of a workflow consisting of several methods to provide a tentative classification of TE consensus sequences. REPCLASS is also a distributed application utilizing high performance cluster computing for performing the computationally intensive task of classification.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

Over the past few years many genomes have been sequenced and with the advancement of technology new genomes are being sequenced rapidly. This leaves a void in the study of new genomes and we are not able to keep up with the detailed study of each of the sequenced genomes. Automated tools have been developed for every aspect of bioinformatics, improving the productivity of researchers, but there are some fields that, though important, until recently were less touched upon topics. One such field is the analysis of repetitive DNA, which so far was mainly considered junk DNA without any specific purpose.

1.1 Repetitive DNA

Some portion of every genome contains repetitive DNA and the genome size does not correlate well with the organism complexity. The variation in genome size also does not depend on the gene numbers, as most of the variation is due to the variation in the amount of repetitive DNA. Repetitive DNA is of two types, tandem repeats and transposable elements (TE). These elements are not junk DNA as they were thought to be. They are now implicated in a broad variety of processes directly related to the host

cell. They can affect host replication, gene silencing, gene regulation and the evolution of genes and proteins.

1.2 Detection and analysis of repetitive DNA

Detection of repetitive elements is a basic step in many biologically important analyses, including, but not limited to, sequence assembly during genome sequencing, genome annotation, similarity searches, gene and coding sequence prediction. Annotation or masking the repetitive elements has been mainly done based on the similarity to previously known repeats. Tools such as Repeat Masker [1], Masker Aid [2] and Censor [3] are widely used to mask repeat sequences based on similarity, usually by replacing detected repeats by special characters. Many tools such as Tandem repeat finder [4] and EMBOSS (etandem and equicktandem) [5] have been developed to detect tandem repeats which are simple repeating sequences. The classification of repeats has been predominantly done manually. A database of these manually curated repeats was first published in 1992 as Repbase [6]. Repbase became a regularly updated general database of eukaryotic repetitive elements and it was later renamed as Repbase Update (RU). All programs which perform the similarity based search depend on the accuracy and content of RU.

The process of repeat identification is complex and a de novo approach is followed which is based on the repetitive nature and the characteristic properties of the elements. The identification and classification are individual problems and they need to be tackled differently. The problem of identifying repetitive elements de novo is a

complex algorithmic task, which has been tackled by tools like, RECON, RepeatScout, ReAS and Piler. The classification of these repeats is another separate complex problem which is being done manually. Moreover, the elements exhibit varied difference in their properties so it is almost impossible to detect them based on any single method.

1.3 Computational Biology

Computational biology is the use of algorithmic techniques from applied mathematics, statistics and computer science applied to biology. Major research efforts include the development of tools for sequence alignment, gene finding, genome assembly, predicting protein structures and interactions, predicting gene expression and a host of other applications. One other interesting aspect of computational biology is the use of computing power to process tasks faster and in a distributed environment. Recent advances in high performance computing have led to the development of powerful, cost effective systems called clusters. Clusters consist of individual machines grouped together by means of software and network hardware to act as a single unified entity. Another new paradigm in computing has evolved where several of these clusters which are geographically distributed are grouped together to form a computing grid. There are many problems in biology which can be tackled with the use of clusters and grids. Sequence analysis is one such domain and the identification and classification of transposable elements can also fall under this category. De novo identification and classification of transposable elements in new genomes is a computationally intensive task because repetitive DNA is often the single largest component of the genome (for

instance half of the human genome consists of repeated sequences). Moreover any biological analysis rarely consists of a single step; rather multiple steps are involved to complete a single process. These steps can often be modeled as workflows which can then be deployed on clusters or grids, where the whole process can be completed faster than on traditional systems.

1.4 Goal of the Thesis

This thesis focuses on the computational and biological aspects of identifying and classifying transposable elements through genomic sequence analysis. We introduce REPCLASS, an automated high throughput workflow model, leveraging various programs to identify and classify transposable elements in new genomes. Recent advances in the field of gene annotation indicate that high-quality gene models can be produced by combining multiple independent outputs of computational programs [7]-[10]. The idea of a combined evidence annotation of transposable elements [11] has already been introduced by Quensneville et al. REPCLASS follows a similar model based on workflows and utilizing clusters and grids for performing the tasks.

REPCLASS attempts to classify newly identified TEs taking into account various characteristics which characterize a repetitive element. REPCLASS was predominantly designed for the classification of repeats identified in complete genome sequences and grouped into families by de novo identification programs like RECON or RepeatScout. Even though REPCLASS will accept any sequence as input, it is essential that the input is a consensus of the intact and ancestral sequence of an active repeat

element, containing all structural and sequence information necessary for the classification. In an ideal case de novo identification is performed on a new genome and the complete repeat library consisting of the consensus sequences of the repeat elements is provided as input to REPCLASS.

REPCLASS screens each repetitive element consensus sequence generated by RepeatScout for DNA and structural motifs typical of known repeat types, TE classes and superfamilies. These include conserved protein-coding domains and motifs, terminal, direct and inverted repeats, simple sequence repeats (SSR), palindromic motifs, RNA secondary structure prediction and terminal nucleotide signature sequences. Another important piece of information is the structure of the alteration of genomic DNA associated with the insertion of the elements. This is generally the duplication of short target genomic sequence, target site duplications (TSD). TSDs are dependent on the enzymes involved in TE integration (such as transposases, endonucleases or integrases) and therefore the length and sometimes the sequence of the TSD is generally conserved within TE superfamilies or classes. REPCLASS then assigns a tentative biological classification of the repeats with a confidence score based on the results of each of the above mentioned methods.

This document is structured to provide the reader with the basic knowledge about genome biology and the computing tools used in bioinformatics before proceeding with the focus of this thesis. Chapter 2 gives a background on genome biology, characteristics, properties and classification of transposable elements, some bioinformatics programs used for repeat identification and sequence comparison and

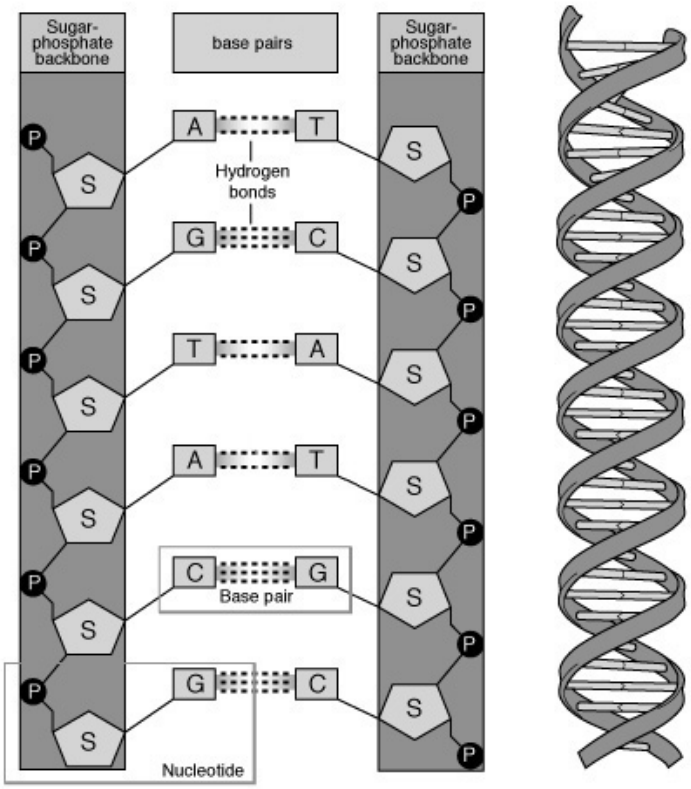
finally an introduction to clusters and grids and their use in bioinformatics. Chapter 3 provides the complete architecture of REPCLASS along with the methods and algorithms involved, followed by the description of the benefits of using a cluster for the processing of repeats. Chapter 4 provides the experiments and results, separated into (1) biological results which provide significant information about the classification and (2) computational results which show the increase in speed and other benefits of using a cluster. Chapter 5 concludes the thesis with the information and perspectives on future development of the related software.

CHAPTER 2

BACKGROUND

2.1 Genome Biology

All life on earth can be categorized into three phylum, namely archaea, bacteria and eukaryotes. Archaea and bacteria are single celled organisms also referred to as prokaryotes. Prokaryotes do not have a nucleus and a nuclear membrane whereas the eukaryotes have them. Taking humans as an example, we are composed of many different organs, and each of these organs serve a specific function. All organs contain cells which are the fundamental working units of any living organism. Cells consist of a nucleus which is their headquarters, regulating all cellular activity. The nucleus is surrounded by cytoplasm which primarily consists of water. Within the nucleus is the DNA (Deoxyribonucleic Acid) responsible for providing the cell with its unique genetic information and characteristics. DNA comprises of structural elements called nucleotides. Each nucleotide is characterized by one of four bases (A – Adenine, C – Cytosine, T – Thymine, G – Guanine) as well as a molecule of sugar and a molecule of phosphoric acid, Figure 2.1.

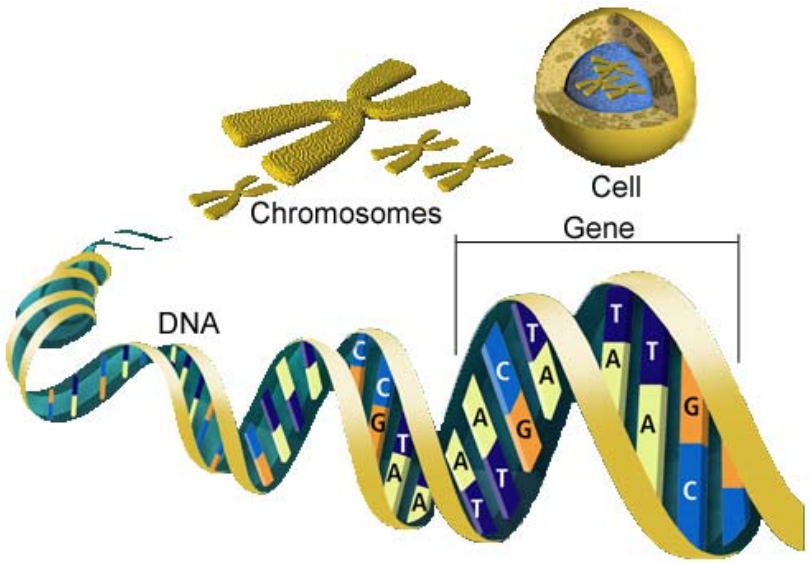


(Image Credit: U.S. Department of Energy Human Genome Program, http://www.ornl.gov/hgmis)

Figure 2.1: Structure and composition of nucleic acid

Nucleotides are linked (ex. ATTGACGT) to form sequences. Nucleotides may form ordered sequences called genes which act as functional subunits for hereditary information. The DNA is the same in every cell of the body, but depending on the specific cell type, some genes may be turned on or off, when they are responsible for the different functions of the cells. Genes encode information on how to make proteins which in turn perform most life functions and are also responsible for cellular structures. The nucleotides are paired together in a double helix coil, A binding with T and C binding with G. These pairs are called base pairs. Figure 2.2 shows the cell, chromosomes and the DNA structure. DNA is organized in tight coils known as

chromosomes. A typical organism has between 2 to 24 chromosomes of varying length. Chromosomes are typically 400 thousand to 400 million base pairs long.



(Image Credit: U.S. Department of Energy Human Genome Program, http://www.ornl.gov/hgmis)
Figure 2.2: Cell, chromosomes and DNA structure

The nucleus also contains other molecules of proteins and RNA (Ribonucleic Acid). The function of some RNA is to copy genetic information from DNA and export this information into the cytoplasm where it is translated into proteins. RNA is single stranded and DNA double stranded, and RNA contains uracil instead of thymine.

Two strands which form the double helix coil are referred to as a forward strand and a reverse complement strand going from left to right and right to left in a positive direction. In the positive orientation the start of the strand is called 5' (5 prime) and the end is 3' (3 prime). There are two general types of genes: non-coding genes and protein coding genes. Non-coding genes encode various functional RNA molecules. Coding genes serve as a template for encoding proteins through a two step process where the gene is first transcribed into RNA and then translated into amino acid chains. The

boundaries of a protein-encoding gene are defined as the points at which transcription begins and ends. The core of a protein coding gene is the coding region, which contains the nucleotide sequence that is eventually translated into an amino acid (and eventually a protein) sequence.

The genes are composed of two primary alternating structural components called exons and introns (Figure 2.3). The exons carry the information required for protein synthesis and they are translated into the corresponding proteins. The non-coding parts of the gene are called introns. The complete set of DNA of any organism is collectively known as the genome. Eukaryotic cells also have a mitochondrial genome derived from a prokaryotic organism and some eukaryotes (plants and algae) also have a chloroplast genome. The present study focuses on the nuclear genomes of eukaryotes. The genome size varies for each organism. Table 1.1 shows the genome sizes of some eukaryotic organisms along with the estimated number of genes [12] and percentage content of repetitive DNA. The table illustrates that the variations in genome size cannot be explained by the variation in gene numbers, but that it is relatively well correlated to the amount of repetitive DNA. Larger eukaryotic genomes, such as those of some plants or those of mammals contain larger amount of repetitive DNA than the smaller genomes of nematode or yeast, for example.

Table 1.1 Genome sizes and number of genes [12, 14, 15]

Organism	Genome Size (bases)	Estimated Genes	% repetitive DNA
Human (Homo sapiens)	~3 billion	~30,000	44.4
Lab mouse (M. musculus)	~2.6 billion	~30,000	39
Fruit Fly (D. melanogaster)	~137 million	~13,000	22
Roundworm (C. elegans)	~97 million	~19,000	6

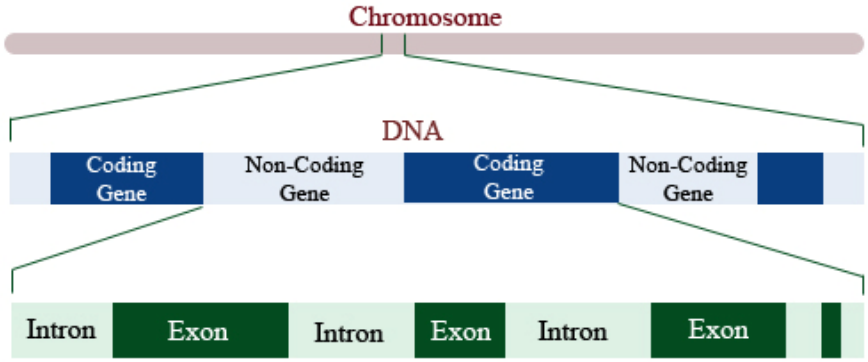


Figure 2.3: Gene, exon and introns representation

2.2 Repetitive DNA and Transposable Elements

Repetitive DNA can be divided into (1) tandem repeats and (2) interspersed repeats. Tandem repeats consist of multiple direct repetitions of the same sequence motifs in a head to tail orientation. Interspersed repeats are copies of repetitive elements dispersed at multiple locations throughout the genome. Almost all interspersed repeats are generated by transposition and therefore are referred to as TEs. Transposition is a reaction which mediates the movement of discrete DNA segments between many chromosomal sites. Almost all interspersed repeats result from the propagation and amplification of mobile genetic elements known as insertion sequences, transposons, repetitive elements or transposable elements (TE) throughout the genome. Mobile DNA was first discovered by Barbara McClintock [13] when she described controlling

elements in maize in the 1950s. She discovered the genetic elements which could move from one place to another within the genomes as a result of chromosomal breaks.

In most eukaryotic genomes the largest fraction of repetitive DNA is made of interspersed repeats. Tandem repeats may account for a substantial fraction of the genome, but they are typically excluded from genome sequence projects because of inherent difficulty in cloning tandem repeats and chromosomal regions rich in tandem repeats (centromeres and telomeres). However, tandem repeats are relatively easy to identify and classify based on their particular arrangement. They are classified on the basis of the pattern or sequence, unit length of the sequence and the number of units. Microsatellites are typically shorter than 1000 bp and composed of tandemly repeated units 2bp to 10bp long. Minisatellites are composed of tandemly repeated units longer than microsatellites, from a few bp to 100bp. Minisatellites span from 1 kbp to 100 kbp. Satellites are generally made of larger units (>100bp) and occur in arrays of 10^2-10^3 tandemly repeated units and are predominantly located in the well defined chromosomal regions, such as centromeres.

In contrast, interspersed repeats are derived from a very diverse range of TEs for which a much more complex classification has been proposed. They have different structural properties and different methods of transposition, by which they replicate. More details on their classification and characteristics are provided in the following sections.

Transposable elements are DNA sequences that will either copy-paste (Class I) or cut-paste (Class II) itself in another part of the genome. In the cut-paste mechanism

a TE is excised from one site (a position in the DNA sequence) and inserted at another site either on the same or a different chromosome. In the copy-paste mechanism the transposable element creates an RNA copy of itself, which is reverse transcribed into a DNA molecule that is subsequently reinserted in the genome. The details are explained in the next section. TEs comprise a significant fraction, ~22% [14] of the fruit fly to ~50% of the human genome. The movement of TEs can generate mutations and chromosomal rearrangements and thus affect the expression of other genes. TEs are generally 100 to 20,000 base pairs long. The human genome contains 44.4% [15] of interspersed repeats, 3% of coding regions (excluding those of TEs) and 2% of satellites and micro-satellites. The remaining are non-coding regions of unknown origin. The chart in Figure 2.4 shows the composition of the human genome.

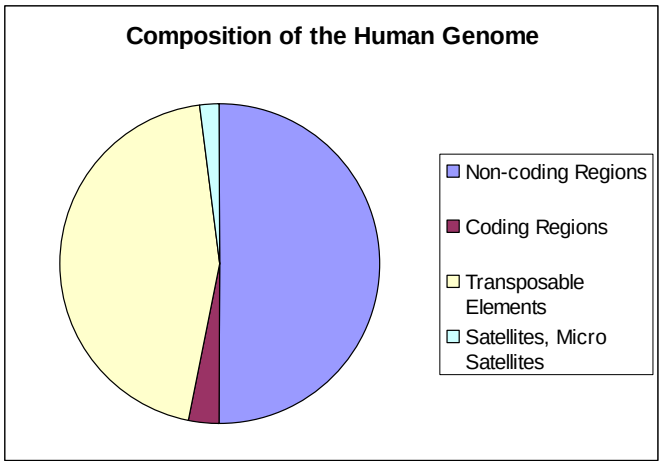


Figure 2.4: Composition of the Human Genome

2.3 Classification of Transposable Elements

There are two main classes of transposable elements based on their method of transposition: (1) transposition with an RNA intermediate and (2) direct DNA transposition.

2.3.1 Transposition with an RNA intermediate

This class of transposable elements (CLASS I) follows a mechanism of transposition which is similar to retrovirus replication. These elements code for an enzyme reverse transcriptase which catalyzes reverse transcription, where a DNA sequence is synthesized from the RNA transcript of the TE. This newly created sequence inserts itself into target DNA sites through the copy-paste mechanism. Some of these elements have long direct repeats on either ends of the transposon known as long terminal repeats (LTR). This result in two subclasses: (i) LTR retrotransposons and (ii) Non-LTR retrotransposons.

2.3.2 Direct DNA Transposition

Class II elements transpose directly from the DNA and do not form intermediate sequences. This class is subdivided into two major subclasses, DNA transposons and Helitrons. DNA transposons are elements bounded by terminal inverted repeats (TIR) on either end. The DNA transposition [16] requires three sites, one at each end of the transposon and another at the target site. The transposition starts by the excision of DNA at the ends of the terminal inverted repeats at the donor site. This is followed by the merging of the ends of the transposon with the target DNA at positions such that the newly inserted transposon is flanked by short gaps. The host site repairs these gaps, creating target site duplications (TSD) (Figure 2.5), characteristic of most transposons. This method where the transposon is completely excised from the donor DNA and inserted into the target DNA is often called a “cut-and-paste” mechanism. The other subclass of elements, Helitrons, transposes in a similar way but they do not form target

site duplications. These elements insert between A on the 5’ end and T on the 3’ end. They also have different structural properties when compared to DNA transposons.

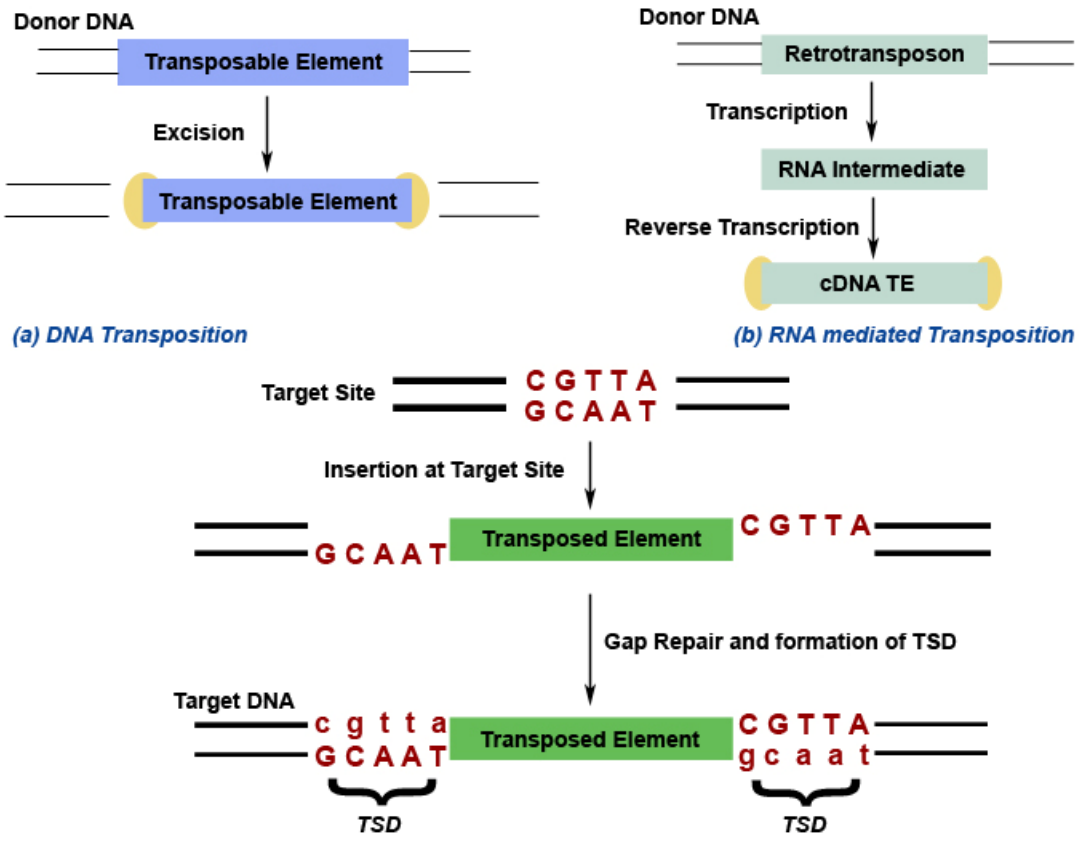


Figure 2.5: Transposition of Transposable Elements (TE)

The TE classification system is constantly evolving in response to the continuous discovery of new groups of elements and the ever-growing number of subgroups and families with particular structural and genetic variations described within existing groups. TEs are classified in a hierarchical family structure: class, subclass, superfamily/group and clade/lineage. There have been various proposed standards for the classification of TEs where the class and subclass structure have been universally accepted. There are some differences in the use of the subdivisions such as superfamily,

family, subfamily, clade and group. Here we have provided a classification that provides distinct differences between the various levels of classification. The classification chart [17] (Figure 2.6) provides detailed information about the classification of TEs for eukaryotic genomes, along with the classification criteria.

The levels of classification are based on certain properties of the transposable elements which distinguish them from one another. The class distinctions are created based on the transposition intermediate and method of transposition. There are two classes that arise based on the presence or absence of reverse transcriptase and the transposition intermediate, through RNA intermediate and without an RNA intermediate, forming Class 1 and Class II respectively. The next level, subclass is distinguished based on structural properties, integration mechanism and the coding capacity. The structural properties are terminal inverted repeats, long terminal repeats, tRNA structures and simple sequence repeats. The integration mechanism represents the target site duplications which the TE creates at the target site. The TSD is not the main distinguishing factor for the subclass and is used only for the Helitron subclass. The coding capacity deals with the types of enzymes each element encodes. There are some enzymes that are distinct in certain subclasses, whereas others are common to several subclasses. These enzymes are integrase, endonuclease, protease, helicase, peptidase and reverse transcriptase. The next level of classification is superfamilies which are distinguished by phylogenetic analysis and integration mechanism.

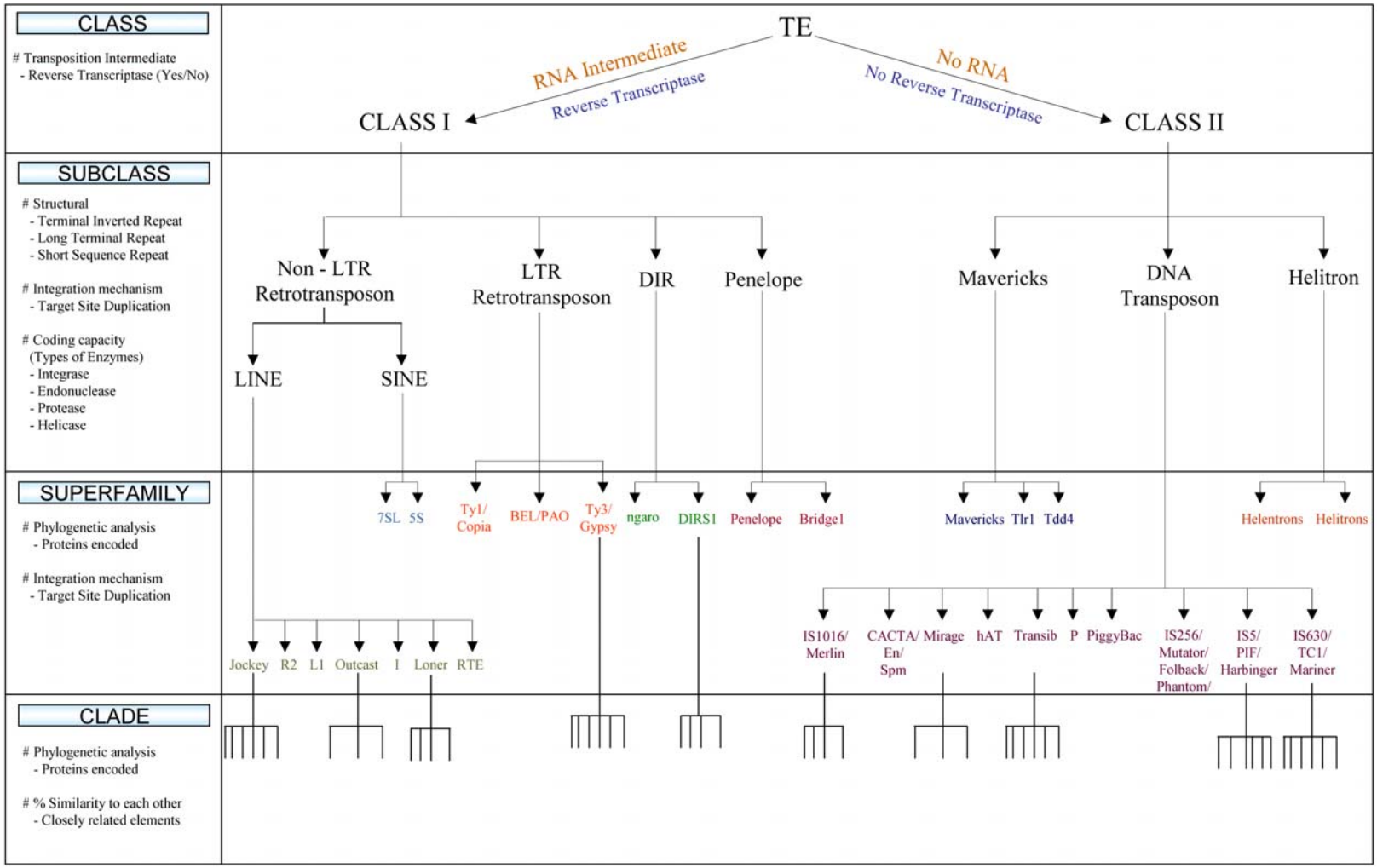


Figure 2.6: TE Classification Chart

By phylogenetic analysis we mean organizing the elements into phylum or group based on similar protein motifs or signatures. A motif is a short sequence of a few conserved bases. The last level is the clade which is a grouping of elements within a superfamily with sequence and protein similarities. A classification using XML is shown in Figure 2.7.

```
<?xml version="1.0" encoding="UTF-8"?>
<TE>
    <CLASS name="I" keywords="RNA Intermediate">
        <SUBCLASS name="Non-LTR Retrotransposons - LINE's" keywords="TPRT">
            <SUPERFAMILY name="R2" keywords="" rm="" tsd="" enzymes="">
                <CLADE name="CRE" keywords="" rm="" tsd="4-30" enzymes="RT,EN" />
                <CLADE name="NeSL" keywords="" rm="" tsd="" enzymes="" />
            </SUPERFAMILY>
            <SUPERFAMILY name="L1" keywords="" rm="" tsd="" enzymes="" />
        </SUBCLASS>
        <SUBCLASS name="Non-LTR Retrotransposons - SINE's" keywords="TPRT">
            <SUPERFAMILY name="tRNA derived" keywords="" rm="" tsd="" enzymes="" />
        </SUBCLASS>
        <SUBCLASS name="LTR Retrotransposons" keywords="VLP-mediated">
            <SUPERFAMILY name="Ty1/copia" keywords="Pseudoviridae" rm="" tsd="" enzymes="">
                <CLADE name="Pseudoviridae" keywords="" rm="" tsd="" enzymes="" />
            </SUPERFAMILY>
            <SUPERFAMILY name="Ty3/gypsy" keywords="Metaviridae" rm="" tsd="" enzymes="" />
            <SUPERFAMILY name="BEL/PAO" keywords="" rm="" tsd="" enzymes="" />
        </SUBCLASS>
    </CLASS>
```

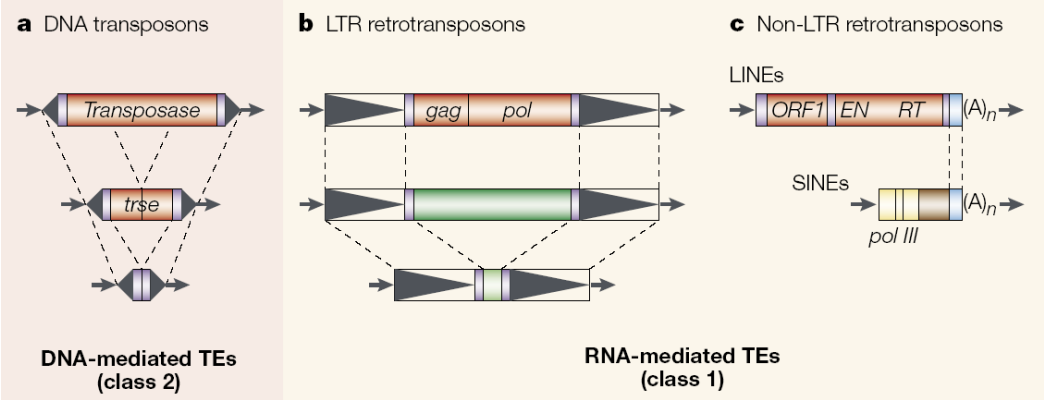
Figure 2.7: XML file of TE classification

2.4 Characteristics of TEs

All TEs exhibit some characteristics common to all their copies. Some of these characteristics are:

- Similar proteins exhibiting homology
- Target Site Duplications (TSD)
- Terminal Inverted Repeats (TIR)
- Long Terminal Repeats (LTR)
- Simple Sequence Repeats (SSR)
- tRNA Structure

Most of these families of TEs occur in large copy numbers within a genome. These copy numbers vary depending on the element. The Alu family and all its subfamilies are the most abundant in the human genome, with at least a million copies. Figure 2.8 shows the different classes and some subclasses with their structure. The DNA transposons are flanked by terminal inverted repeats, the LTR retrotransposons by direct long terminal repeats. The non-LTR retrotransposons do not have structural motifs, but instead feature simple sequence repeats towards the 3' end (see definition below). Each of these elements code for different enzymes, which helps us distinguish them. Each group of TEs further consists of autonomous and non-autonomous elements. Autonomous elements encode their own protein-coding domains (transpose, gag, pol, ORF1, EN, RT), as shown in the figure below and non-autonomous elements do not have protein-coding domains. Non-autonomous elements may nevertheless still propagate by using the enzymes encoded by the autonomous elements.



ORF – Open Reading Frame
trse – transpose
EN – Endonuclease
RT – Reverse Transcriptase

Figure 2.8: Structure of some subclasses [17]

2.4.1 Target Site Duplications (TSD)

TSDs are formed during the insertion of a TE into a target site. One of the strands breaks, thus forming a gap. This gap is then repaired by the host genome to form the target site duplications. It has been found that these TSDs are sometimes conserved in length and in certain cases a clear sequence preference for the TSD may be observed. Not all TEs create TSDs upon insertion, as it can be seen in the case of Helitrons. The formation of a TSD is shown in Figure 2.5. Some examples of TSDs are:

- 4-6bp for most LTR retrotransposons
- TA for the Tc1/Mariner Superfamily
- 8bp for the hAT Superfamily

2.4.2 Terminal Inverted Repeats (TIR)

An inverted repeat is one where two different segments of the double helix read the same but in the opposite directions. Terminal inverted repeats are when the inverted repeat occurs at the ends of a transposable element. This structure is common in all DNA transposons. The length of the TIRs vary from 10bp-500bp.

Figure 2.9: Terminal Inverted Repeat (TIR)

2.4.3 Long Terminal Repeats (LTR)

Long terminal repeats are long repeating sequences of DNA that occur on either end of the TE. LTRs are a structural feature of LTR retrotransposons. LTRs vary in

length from 100bp to several kbp. For illustration, only a short stretch of an LTR is shown, in direct orientation, in Figure 2.10.

Figure 2.10: Long Terminal Repeats (LTR)

2.4.4 Simple Sequence Repeats (SSR)

SSRs are small sequences of 2bp-10bp length which repeat in constant intervals. They are mainly characteristic of satellites, SINEs (Short Interspersed Nucleotide Elements) and LINEs (Long Interspersed Nucleotide Elements). They occur in the flanking regions or within the element, but exist either on the 3' or 5' end, but not on both.

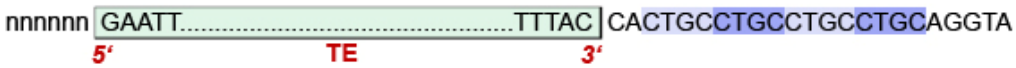


Figure 2.11: Simple Sequence Repeats (SSR)

2.4.5 tRNA Structure

Transfer RNA (tRNA) is a clover leaf structure which facilitates protein synthesis. The tRNA binds specifically to mRNA (messenger RNA) codons via anticodons and interacts with the ribosome and rRNA (ribosomal RNA). During the transcription SINEs form tRNA structures such as the one shown in Figure 2.12.

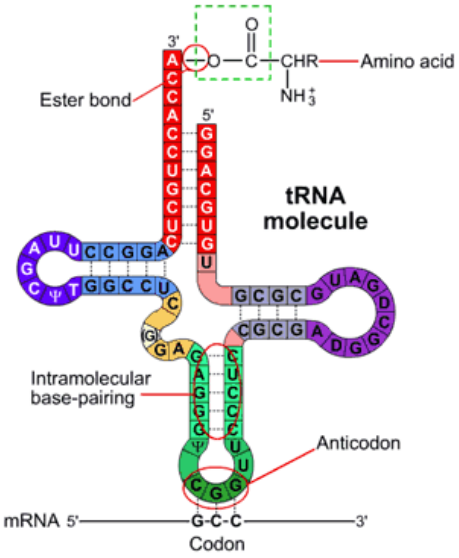


Figure 2.12: tRNA molecule

2.5 Identification of Transposable Elements

Transposable elements are generally classified based on their structure. However, several subdivisions within the main classes are distinguished based on their degree of identity and similarity in terms of their DNA and protein sequences. Since there are various forms of TEs it requires substantial work to identify or classify them. Various identification methods are:

1) Analysis of natural or artificial unstable mutations.
2) Polymerase Chain Reaction (PCR) [18] amplification using conserved regions.
3) Genome sequence analysis.

The first two methods are laboratory based methods and they will not be covered in this thesis. This thesis will focus on identifying and classifying transposable elements by analyzing the genome sequence.

2.6 Genome Sequence Analysis

Sequence analysis is performed by a suite of Blast [19] programs and other software specific to TE identification and classification. There are three main steps in the analysis of TEs. They are: (1) Identification, (2) Classification and (3) Annotation or masking.

Identification of TEs is a computationally complex task and no single program has been able to come up with a good model to identify them. Many algorithms are being designed to tackle the problem of TE identification, but none of them are able to define the ends very well. Some of the programs are: RECON [20], RepeatScout [21], Piler [22] and ReAS [23]. All these programs are described in the next section. Though they produce acceptable results, none provide a classification for the discovered elements.

The next step is the classification of the identified repeats, which is the main focus of this thesis and the REPCLASS tool. Repeats have been classified manually for more than a decade now. With the rate of new genomes being sequenced, the time taken to classify the repeats in these genomes manually would take several years. The architecture and features of REPCLASS are explained in the next chapter.

The final step is the annotation or masking of repeat sequences. Annotation is the process of labeling and describing sequences. The most used and relied upon tool for TE masking and annotation is RepeatMasker, which recent studies indicate may be “neither the most efficient nor the most sensitive approach” [24] for TE annotation.

Some other tools used are Censor and MaskerAid, which is a faster version of RepeatMasker.

2.7 Identification and annotation tools

2.7.1 RECON

RECON is good for a first pass of automatic identification and grouping of repeats into families, but it does not provide a biological classification. RECON groups sequences into families identified on the basis of their repetitive nature through genome-vs-genome pairwise alignments. The underlying algorithm of RECON is based on extensions to the single linkage clustering of local pairwise alignments between genomic sequences. It uses multiple alignment information to define the boundaries of individual repeat families and distinguishes homologous but distinct repeat element families. The success of this procedure depends on the presence of elements reiterated in multiple copies and on the efficient detection by similarity search algorithms like wu-blast.

2.7.2 Piler

Piler is a repeat identification tool that exploits characteristic patterns of local alignments induced by certain classes of repeats. It mainly focus on identifying a subset of local alignment hits which form a pattern characteristic of a given type of repeat. Though this method achieves high specificity of those particular repeats it has a low sensitivity in finding new elements. The piler suite of algorithms finds tandem arrays,

dispersed families, pseudosatellites, and elements with terminal repeats, such as families belonging to the LTR retrotransposon subclass.

2.7.3 ReAS

ReAS is an algorithm to recover ancestral sequences of TEs from the unassembled whole genome shotgun (WGS) reads. The main assumption of the algorithm is that these TEs must exist in large number of copies across the genome and must not be so old that they are no longer recognizable when compared to their ancestral sequences. The unique feature of this algorithm is in that all other algorithms work on assembled genomes whereas ReAS uses unassembled reads of a whole genome shotgun.

2.7.4 RepeatScout

RepeatScout is an algorithm which identifies repeat families through the extension of consensus seeds. Most importantly, it enables a rigorous identification of repeat boundaries, which is a major issue in de novo identification of repeats. RepeatScout is more sensitive and faster than RECON in identifying repeats in newly sequenced genomes. RepeatScout produces consensus sequences of the families, though it does not report the number of copies in a family. It creates the consensus sequence boundaries using pairwise alignment scores.

2.7.5 Repeat Masker

The common method for automated annotation of repetitive sequences relies on a single program, RepeatMasker. It does an extensive search of the significant matches between the genome to be annotated and the libraries of known repetitive elements from

Repbase. It is mainly useful for masking known repetitive elements and it is not useful for identifying or classifying new repetitive elements. The drawbacks of this approach are: 1) the annotation is good as long as the libraries are complete and correct. 2) Creation of these libraries is a daunting task which requires substantial manual work.

2.8 Cluster and Grid computing

Increasing demands for high performance and computational power by scientists, academicians and companies have led to the development of low cost, high performance and reliable computing systems, known as clusters. A cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers working together as a single, integrated computing resource. Clusters generally consist of large storage networks, high processing power and large memory, interconnected with high speed, low latency networks like Myrinet, Infiniband or Gigabit Ethernet.

The Grid is the next generation in distributed computing. The Grid is a conglomeration of hardware and software resources pooled together to provide huge computing power and storage capabilities. Grid computing as stated by Ian Foster, is "coordinated resource sharing and problem solving environment in dynamic, multi-institutional virtual organizations" [25].

The key distinction between clusters and grids is mainly in the way resources are managed. In the case of clusters, the resource allocation is performed by a centralized resource manager and all nodes cooperatively work together as a single

unified resource. In the case of Grids, each node has its own resource manager and does not aim at providing a single system view.

2.9 Clusters and Bioinformatics

There has been recent increase in the use of clusters and grids in bioinformatics research with the emergence of Bio-clusters and Bio-Grids. Apart from computational needs, bioinformatics requires the grid and clusters for data storage, workflow interactions and sharing of semantic data. myGrid [26] is one of the pioneers of biogrids, building high level services for integrating applications and data resources, concentrating on dynamic resource discovery, workflow specification and distributed query processing. Bioinformatics work often requires many databases to be linked together for annotation and querying purposes.

Our use of clusters for REPCLASS reflects the fact that it is a computational workflow which requires considerable database interactions and independent programs that can be run as batch jobs. REPCLASS utilizes the DPCC[1] (Distributed and Parallel Computing Center) at UTA for enacting the workflow.

CHAPTER 3

REPCLASS

In the chapter we introduce REPCLASS, the first automated tool for the classification of repeats. REPCLASS is a multi-program workflow that performs the classification of repeat sequences based on the various characteristics exhibited by transposable elements. Moreover, REPCLASS is a high throughput computing tool which utilizes the power of cluster computing to quickly classify repeats in entire genomes.

REPCLASS generates the final classification based on outputs of various stages, which may result in combined evidence for the classification. Along with the classification, it reports other useful information such as the TSD and structural properties and coding sequences. The input to REPCLASS is a repeat library of consensus sequences which are run through parallel steps of homology, TSD and structural based searches. A consensus sequence is a representative sequence of several closely related sequences. Such consensus is generated for every possible repeat family. The Repbase Update database hosts all the consensus sequences of known repeats. It is manually curated and it has evolved over a decade to contain several hundred TE consensus sequences of different genomes. Our final result is the compilation of all

discovered elements and their matching against the existing classification database in order to verify the classification. The complete workflow is shown in Figure 3.1.

3.1 De novo identification of repeat families

Identifying families of repetitive elements is a large and complex algorithmic problem which is beyond the scope of this thesis. There are only a few algorithms and programs performing de novo identification of repeat families in whole genome sequences. In order to generate the input to REPCLASS we have used RepeatScout, which is one of the methods to identify repeat families de novo. Other programs can be used (instead of RepeatScout) to generate an initial list of all TE sequence consensuses for a genome. We initially experimented with two programs, RECON and RepeatScout. RECON generated more fragments then RepeatScout and it did not define the boundaries of the repeats very well. We compared both the tools on several genomes and consistently experienced the above problems with RECON. Hence the better choice for REPCLASS was RepeatScout.

We start with providing the complete genome sequence to RepeatScout, which outputs a Fasta [27] format file of the consensuses of the identified repetitive elements. RepeatScout reports all kinds of repetitive elements, including tandem repeats, satellites, micro-satellites and transposons. All tandem repeats, satellites and micro-satellites need to be filtered out, as they do not pertain to TE classification. We use Tandem Repeat Finder [4] and Nseg [28] programs to filter out these elements and RepeatMasker to filter simple sequence repeats and low complexity repeats.

RepeatScout creates many fragments of complete TEs, so fragments less than 100bp are removed. The reason for the cutoff being100bp is because all TEs described in the literature are >100bp. After filtering all these elements, the remaining elements are classified based on the three methods described below.

3.2 Automated REPCLASS workflow

Most problems in bioinformatics are tackled by multiple steps to arrive at a solution. It applies to TE classification, as no single method is likely to be sufficient to arrive at a proper classification with good confidence. Our workflow consists of four main parts:

1. Homology based classification
2. TSD based classification
3. Structural properties based classification
4. Validation and grouping of the results

The input to REPCLASS is the consensus sequences of the identified repeats. They are fed into the workflow where the homology, TSD and structural based classifications can be processed individually. Each of these steps involves multiple sub-processes which will be discussed in detail below. The final step involves the validation of the results of each of the previous steps, and grouping them together in order to arrive at a tentative classification. The classification based on multiple sources of information is more accurate than any single method.

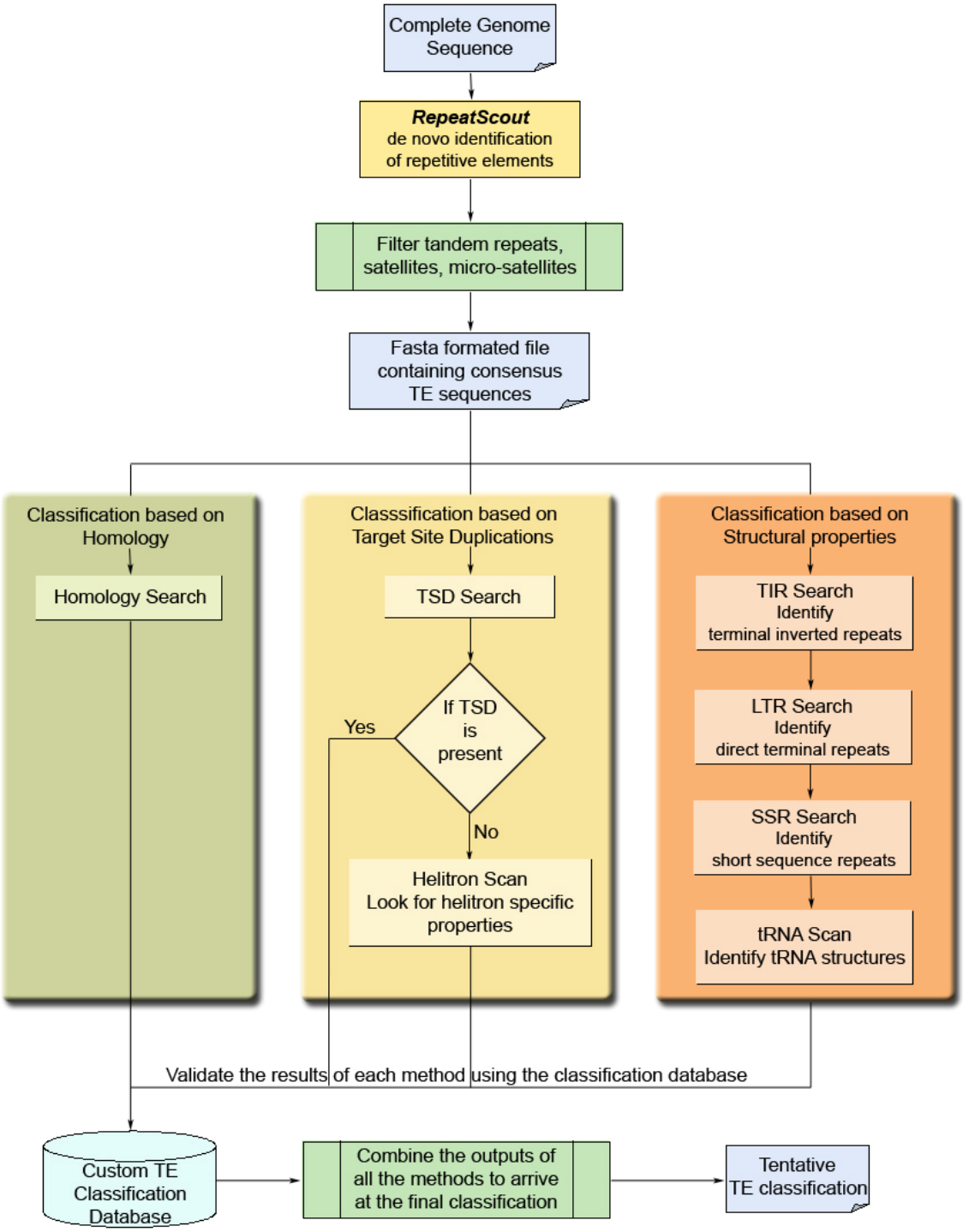


Figure 3.1: REPCLASS workflow

3.3 Homology based classification

This method contains only one sub process, the Homology search, discussed in detail below.

3.3.1 Homology Search

All elements have some sort of conservation among different protein motifs. Elements belonging to the same families will generally exhibit similar motifs. They can be identified by performing a tblastx search of the query sequence against already known transposable elements. Repbase, a manually curated database of known and classified transposable elements will serve as the database for the tblastx search. This method relies heavily on the accuracy and availability of TEs in Repbase.

The procedure for homology based search adopted in REPCLASS is outlined below. First a keyword index of the Repbase database is created. This step involves the extraction of keywords and descriptions from the Repbase database in EMBL [30] format. The indexing searches for specific keywords such as subclass, superfamilies, families and groups. The index consists of the Repbase assigned ID for the TE along with information on subclass (SC), superfamily (SF), family (FM), group (GP), subgroup (SG) and keywords (KW). Figure 3.2 shows a snapshot of the index file.

```
ID  CR1-2_AG
SC  Non-LTR Retrotransposon
SF  CR1
GP
FM
SG
KW  DNA/RNA-binding;PHD finger;endonuclease;reverse transcriptase;

ID  HARBINGERN1_AG
SC  DNA Transposon
SF  Harbinger
GP
FM
SG
KW
```

Figure 3.2: Repbase Keyword Index file snapshot

The next step involves a tblastx search of the TE sequences against a Fasta based Repbase database. The BLAST algorithm detects regions of local similarity, which helps us compare novel sequences with previously annotated repeats. The BLAST package consists of a suite of programs: blastn, tblastx, tblastn, blastx and blastp. The tblastx program converts the nucleotide sequences into protein sequences of six frame translations. It then performs a heuristic search for local alignments of the protein sequences which detects regions of similarity between the query and the sequences in its database. Blastn is a nucleotide against nucleotide comparison that returns the most similar DNA sequences. All blast programs used in RECLASS are based on WU-Blast [29] and not NCBI Blast [19]. The blast output files are parsed and the first *x (default of 10)* hits with greater and 85% similarity and an e-value lesser than e^{-5} is chosen. The classification for these *x* elements are retrieved from the Repbase keyword index. Two measurements for each keyword are calculated, one based on the e-values (p_e) and the other based on the keyword counts (p_k).

Some of the problems are from non-coding sequences of individual repeat families which are poorly conserved not only across species, but also within species and among families that belong to the same superfamily. Therefore, most detectable similarities are expected to occur within protein coding sequences like reverse transcriptase, integrases, transposases, endonulease and helicase encoded by the transposable elements. We calculate the P_e measure as:

$P_e = |ln\ (e\text{-}value)| / 100$, for e-values < e^{-100} are set to e^{-100}.

A weighted average of p_e is taken based on the number of hits. p_k is calculated as the weighted average of the occurrence of a particular keyword to the total number of hits.

$P_k = keyword\ count/no.\ of\ hits$.

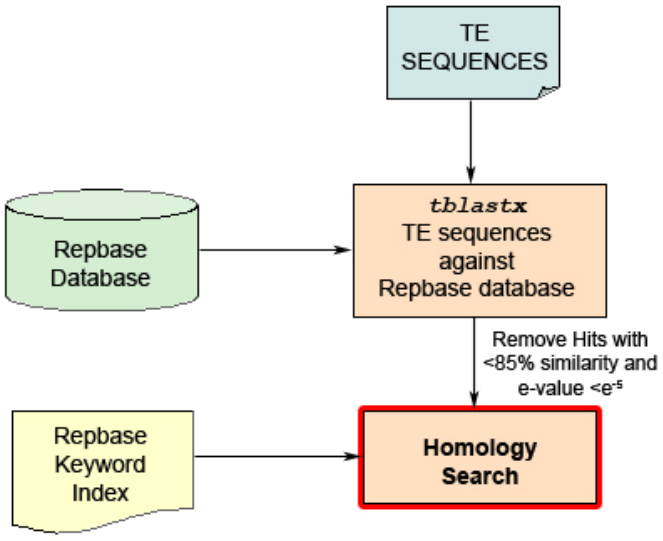


Figure 3.3: Homology Based Classification

The tblastx parameters for the homology search are, E=0.0001 and hspmax=5000. Keywords were extracted for hits with >85% similarity. The maximum number of hits considered is 10.

3.4 TSD based classification

Target site duplications occur in many elements, but there are also elements which do not produce a TSD, especially the Helitron subclass. We first search for TSDs, then for elements in which a TSD was not found, we perform an additional step, searching for properties specific to the Helitron subclass of DNA transposons.

3.4.1 TSD Search

This method of classification identifies target site duplications caused by the insertions of transposable elements. The steps involved in the TSD Search are shown in Figure 3.4 and described below:

1) First a blastn search is performed with the query consisting of the TE sequences against the Fasta database of the target genome.
2) Blastn returns a set of similar sequences, from which we select only sequences with conserved ends and look at a 50 bp flanking them at both sides. We retrieve only elements with both ends because only those elements can possibly have the TSDs. This is because some copies may have mutated over time, loosing the TSDs in the process.
3) The flanking sequences are searched for possible target site duplications. A moving window is used to check for direct repeats on the flanking ends near the 5’ and 3’

ends of the sequence. We allow a mismatch of 1bp in a match >5bp and 2bp mismatches are allowed for a match >10bp.

4) The maximum number of elements having the same length TSD is retrieved and a consensus of those TSD sequences is formed.
5) Based on the most likely TSD length, and the conserved consensus, we can predict the class, subclass and superfamily of that particular sequence. Most subclasses and superfamilies have specific length TSDs, while others feature varying lengths within the family. Certain superfamilies have a consensus sequence for the TSDs, such as TA or TTAA. For most of the repeats in Repbase these TSDs have been removed from the TE sequence itself. However in the TE consensus sequences provided by RepeatScout, these TSDs will occur as part of the repeat. Thus the need to search the ends of the input TE consensus sequence in order to identify such TSDs.
6) A probability score is determined for the TSD scan (p_{tsd}). It is based on the occurrence of a particular length as compared to the total number of complete copies found.

The 50bp flanking sequences are joined together and used in a blastn search against the genome in order to find empty sites, those without the repeat insertion. Such empty sites are commonly found because repeats are frequently nested within other repeats. By comparing the repeat containing the empty site, we can infer the likely effect of the insertion on the flanking genomic DNA and therefore confirm the TSD.

The TSD search performs better in families with higher copy numbers and conserved ends.

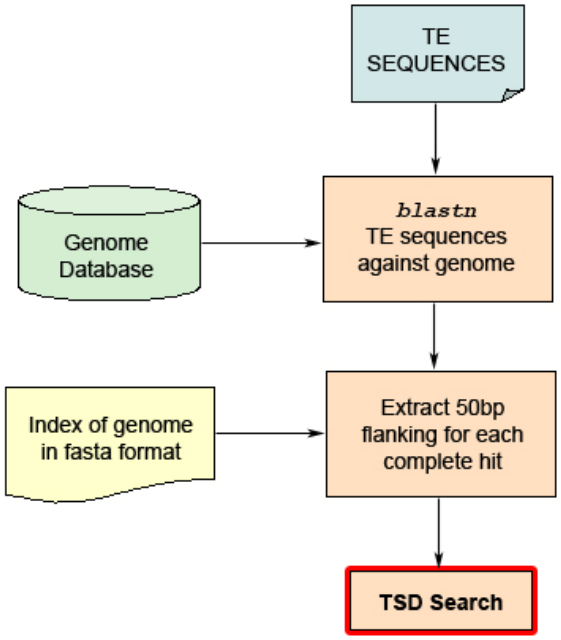


Figure 3.4: Target Site Duplication (TSD) Search

The blastn parameters used in our search are, hspmax=0, gspmax=0, filter=none, Q=2 and R=1.

3.4.2 Helitron Scan

The Helitron [31]-[32] subclass exhibits some unique structural properties that distinguish it from other repeated elements. All elements belonging to this class have conserved 5'-TC and CTRR-3' (R = A or G) ends which do not have terminal inverted repeats. They contain 16 - 20bp long palindromes separated by 10-12bp from the 3' end and transpose precisely between the 5'-A and T-3', with no modifications at the AT target sites and the palindrome is rich in GC content.

In most cases no TSDs will be found flanking helitrons. A special search of the flankings for 5'-A and T-3' is performed. In the case of repeats identified by RepeatScout the 5'-A and the T-3' will be part of the TE sequence. So the 5' end of the sequence is searched for ATC or TC accepting a match from the end till 5bp within the sequence. Similarly the 3' end is searched for the presence of CTRRT or CTRR where R is either A or G. A combined match (H_{53}) of 5'-TC and CTRR-3' accounts for 50% of the final score. A tentative similarity score H_{53} is calculated based on the average of the number of hits (Σi) to the total number of copies (T_C).

$$H_{53} = (\Sigma i / T_C)*0.5$$

The palindrome program from the EMBOSS suite is used to find possible palindromes in the transposon sequence. Its output is then parsed to search for the characteristics of Helitrons. An accepted palindrome will have one end 5-12bp from the 3' end and account for a mismatch of 2-5bp pinhead within the palindrome as indicated in Figure 3.5. The occurrence of this palindrome accounts for 50% of the final score. The palindrome score (H_P) is calculated based on the GC content (C_{GC}) of the palindrome.

$$H_P = (C_{GC} / (\text{length of palindrome}*2))*0.5$$

The final score is a sum of the above scores:

$$H_T = H_{53}+H_P.$$

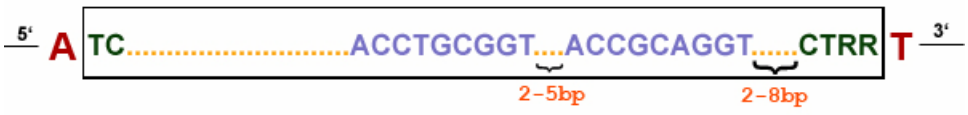


Figure 3.5: Helitron structure

A similarity of above 75% will confirm that the element belongs to the Helitron subclass. The parameters for palindrome are minpallen=5, maxpallen=70, gaplimit=70, nummismatches=0 and nooverlap.

3.5 Structural properties based classification

Majority of the transposable elements exhibit some sort of structural properties, like direct repeats, inverted repeats, simple sequence repeats and tRNA structures. The output of these searches are combined together to classify the subclass level of the particular TE element. Each search resembles the property of a particular subclass. The following are the structural scans:

- LTR (Long Terminal Repeat) search
- TIR (Terminal Inverted Repeat) search
- tRNA scan
- SSR (Simple Sequence Repeat) search

3.5.1 LTR (Long Terminal Repeat) Search

Long Terminal Repeats (LTR) or direct repeats are a set of sequences that repeat in the same direction at both the 5’ and 3’ ends. LTRs vary in size from 100bp to several kbp. The LTR retrotransposons which is a subclass of class I exhibit this structural property.

LTR search scans for these directly repeated sequences. Given below are the steps involved in identifying a LTR:

1) Start with a window size of 50bp in length or the shortest LTR length as specified by the user.
2) Set one window to the start of the sequence.
3) Search for a similar sequence of the specified window size towards the other end of the sequence.
4) If there is no sequence for that window size, move the first window by 1 and similarly continue step 3. This shift is done till 20bp, taking into account that the ends of the TE sequence are not determined correctly.
5) If a matching window is found, increase both the window sizes by 1 and compare the newly added base. Repeat step 5 as long as there is a match. Also a mismatch of 1bp for every 10bp is allowed.
6) If the final match occurs within 20bp towards the ends of the sequence and is >100bp in length, then the current window is considered as the direct repeat sequence or LTR.
7) Report that the TE element is a LTR retrotransposon.

3.5.2 TIR (Terminal Inverted Repeat) Search

Terminal inverted repeats (TIRs) are characteristic of DNA transposons. They are inverted repeats ranging for 10bp to 500bp, on either ends of the transposable element sequence. The TIR search follows a similar approach to that of the LTR search. Here an external program called einverted, which is part of the Emboss suite of packages is used to find inverted repeats within the TE sequence. This program returns

a number of inverted repeats, which need not be terminal inverted repeats. The output of einverted is parsed in order to search for inverted repeats that occur towards the ends of the TE sequence. The ends of the TIR are allowed to be 30bp from the ends on either side, to facilitate for ends that are not well defined by RepeatScout.

The presence of a terminal inverted repeat confirms that the corresponding TE belongs to the DNA transposon subclass of class II. The parameters for einverted are gap=12, threshold=50, match=3, mismatch=4 and maxrepeat=10000.

3.5.3 tRNA Search

tRNA (transport RNA) is a special motif that is present in most SINEs which belong to the subclass of non-LTR retrotransposons. These tRNA structures are specific protein structures and can be identified. For REPCLASS the tRNA-SE [33] program is used for detecting tRNAs. tRNA-SE was run with the default parameters.

3.5.4 SSR (Simple Sequence Repeat) Search

Simple Sequence Repeats (SSRs) are ubiquitous in both prokaryotes and eukaryotes. They are formed by multiple repetitions of the same basic sequence motif. SSRs are characterized by their base sequence, the length of the sequence and the number of units. All these parameters vary a lot, with SSRs from 1bp to 10 bp long. Repeats with more than 10bp sequence are considered as tandem repeats, satellites or micro-satellites. In our case we have the following characterization for SSRs:

Table 3.1 SSR reference table for SSR Search

Sequence Length – Composition	Number of Units
1 – N	10
2 – NN	7
3 – NNN	5
4 –NNNN	4
5 – NNNNN	3

We are interested in SSRs which occur only at any one end of the sequence along with the flanking. They are not part of the whole sequence as are satellites and micro-satellites. SSRs are mainly characteristic of non-LTR retrotransposons, SINEs and LINEs. SSR Search is a two step approach:

1. Search for Poly A tails.
2. Search for simple sequences.

3.5.4.1 Poly A Tail Search

Poly A tails are a stretch of base “A” occurring towards the 3’ end of the sequence. They occur in lengths of more than 10bp.

Figure 3.6: Poly A Tail

The SSR search scans for Poly A tails at the 3’ end of the sequence allowing for a lapse of 10bp from the end and also searching the 3’ end flanking. This is done considering that RepeatScout might not have defined the ends properly.

3.5.4.2 Simple Sequence Search

This search is based on a sliding window algorithm, where the window size is set from 1 to 5 as described in Table 3.1. This search is performed on only hundred base pairs towards the end along with 50bp of flanking. Given below is the pseudo code:

```
for window_size = 1 to 5
    for i in sequence
        unit_sequence = sequence[i] to sequence[i+window_size]
        adj_sequence = sequence[i+window_size] to sequence[i+(window_size*2)]

        while (unit_sequence equals adj_sequence)
            i = i + window_size
            unit_count[unit_sequence]++;
            unit_sequence = sequence[i] to sequence[i+window_size]
            adj_sequence = sequence[i+window_size] to sequence[i+(window_size*2)]
    foreach unit_count[unit_sequence]
        if (unit_count[unit_sequence] > specified number of units for  unit_length)
            indicate presence of SSR
```

Figure 3.7: Algorithm for Simple Sequence Search

The presence of a SSR at one terminus of the consensus sequence indicates that the target sequence is a non-LTR retrotransposon. This search does not distinguish between a LINE and SINE.

3.6 Validation and grouping of results

The final process in the workflow is the grouping of the results obtained by the three methods of classification. We created a custom classification database that mirrors the classification method we dealt with earlier having a list of all known classes, subclasses, superfamilies and clades. This information is used to validate the results

produced by previous steps. It is further used to augment the information received in the previous steps. For example the homology based classification may report the superfamily, but will not have the details of the subclass and class. This is because the Repbase keyword index which is extracted from the Repbase Update is not accurate for old entries. Some entries in Repbase are not updated and do not have accurate information leading to the above mentioned problem. For these elements the complete classification is updated from the classification database. Also there are cases where the subclass might be reported erroneously, even for such entries the classification database is used to correct those errors and provide a more meaningful classification.

Next the classification results from all the methods are combined together to provide a tentative classification. During the combination some elements may be classified by more than one method. These elements are further checked to verify that they provide the same classification in all the methods. Some of the ambiguities are cleared based on the accuracy of the information gathered by the respective method. In cases where this cannot be resoled, the result of the individual method is shown to the user who can decide on the classification. The final classification for each element contains complete details about the element, such as the target site duplication length and consensus, TIRs, LTRs and SSRs and also the name of the coding sequences detected by the homology search.

3.7 How does the cluster augment the search process?

The amount of computing power required by REPCLASS is dependent on the specific genome and the number of TE consensus sequences. Our results illustrate that results can be obtained 30-50 times faster using a cluster than running on a high end standalone machine. Further time analysis will be dealt with in the next section.

The automatic workflow defined earlier has been designed to make use of either a cluster or a single computer. Some of the main advantages of using a cluster are faster turn around time, high scalability and better fault tolerance. The idea of executing workflows on a cluster deals with two aspects, that some of the steps can be executed in parallel and most of the bioinformatics workflows generally involve multiple data sets, which can be executed in parallel. There are some issues that need to be considered when designing applications to work on clusters. Some of the issues we have considered and addressed are: turn around time, scalability, resource utilization, load balancing and fault tolerance. We discuss below how we have incorporated the above issues into REPCLASS.

3.7.1 Turnaround time

Turnaround time is the total time taken for the job, including the queuing delays of the batch processing system. During periods of high loads on the cluster, turnaround time can fluctuate a lot. The tasks are divided based on the current status of the cluster. Moreover the mechanism we use allows us to get partial results, which the user can view at regular inntevals. This provides the user with near immediate results, and the

user need not wait till all the TE sequences are classified. This is also useful when you want to run the same data set with different sets of parameters.

3.7.2 Scalability

Scalability is the ability to handle varying amounts of load and computing power with equal efficiency. REPCLASS is highly scalable in nature. Here scalability depends on one main factor, the number of TE sequences to classify. REPCLASS is scalable up to the point where each sequence can be run on a single node. At the start of the program, the current state of available nodes on the cluster is determined and it is directly reflected in the number of individual tasks.

3.7.3 Resource utilization and load balancing

We have a mechanism that utilizes the maximum available CPUs. Work is distributed among all the CPUs and during periods of heavy load, we make sure that we fully utilize the available CPUs. This is handled by having a common data pool that all the individual jobs access to get their input data. All the jobs terminate when the data pool is exhausted. This is very useful in situations when some of jobs are running while others are in the batch queue. The common data pool facilitates efficient load balancing across the cluster.

3.7.4 Fault tolerance

Fault tolerance deals with handling errors occurred during processing. Errors are detected at runtime and the data sets are put back into the pool to be processed again. A deadlock condition occurs when the pool is exhausted and all the jobs have terminated.

In such cases a post processing step which aggregates all the results, looks for these left out jobs and reschedules them.

CHAPTER 4

EXPERIMENTS AND RESULTS

The initial run of REPCLASS on various genomes and already annotated TEs gives an insight into the efficiency and accuracy of the classification. REPCLASS was run on the DPCC cluster which consists of 81 dual processors, 2.667 GHZ and 2.44 GHZ Xeon compute nodes with 2GB memory each. The software was run on varying number of processors on the cluster and on different genomes. The results are divided into two sections, the biological significance of the results and the computational aspect of the results. The biological part of the results has four portions:

- Classification of annotated TE elements in Repbase for Caenorhabditis elegans (worm) and Drosophila melanogaster (fruit fly).
- Classification of TEs identified de novo for the Caenorhabditis elegans genome.
- Classification of TEs identified de novo for previously unclassified genomes, Ciona intestinalis (sea squirt) and Strongylocentrotus purpuratus (sea urchin).
- Classification of TEs identified de novo for the Human X Chromosome.

4.1 Classification of annotated TE repeats in Repbase

The classification of previously annotated TEs in Repbase acts as a control for the efficiency and accuracy of REPCLASS. Here we have considered two genomes; C. elegans and D. melanogaster. Before running REPCLASS, we performed the following steps to generate a dataset of consensus sequences for the manually classified TEs of these genomes.

- All the simple repeats, tandem repeats, satellites and micro-satellites were removed from Repbase for the two genomes. Being a manually curated database Repbase also has a list of these repeats. Since these repeats do not always fall under TE classification they are removed.
- In each case, the corresponding TE element information was removed from the Repbase index, to demonstrate better validation for the results. This was done since the homology based search would identify these repeats based off the element, itself.

Considering the case for the previously described genomes, the rational for using these specific genomes for validation purposes are two fold:

- C. elegans and D. melanogaster are the most extensively studied species for repeats. The staff at Repbase have compiled a comprehensive list of repeats and carefully annotated them. The annotations of Repbase are the results of over a decade of work by their staff.
- The two species have a assortment of TEs. This provides an understanding of the performance of REPCLASS on a wide variety of TE elements.

Below are the details of the classification:

Caenorhabditis elegans:

- Number of repeats in Repbase: 171
- Number of repeats after removing unclassified repeats: 124
- Number of repeats correctly classified by REPCLASS: 111
- Accuracy of classification: 92%
- Percentage classified: 89.52%

REPCLASS was not able to classify 13 repeats that were classified in Repbase. The accuracy of classification is defined as matching the level of classification provided by REPCLASS to the level of detail as that of Repbase. Accuracy was affected when there were two possible classifications identified for a single element and one of them matched with that of Repbase. REPCLASS failed to classify 3 solo LTR retrotransposons which are part of Repbase and 2 Helitron elements which did not have a similarity >75% to that of a Helitron, 5 DNA transposons and 1 SINE element. The diagram below shows the distribution of the number of repeats classified by each method and also the overlap.

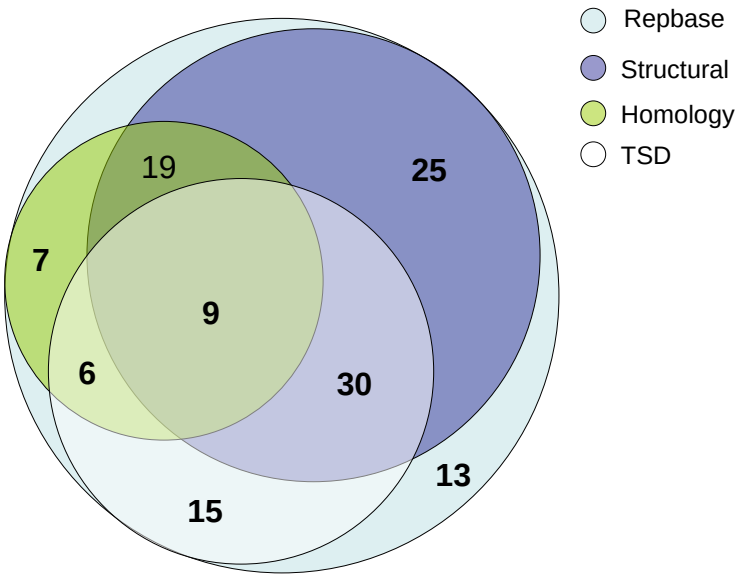


Figure 4.1: Classification distribution for C. elegans repeats in Repbase

Overlapping repeats, in this case 9 repeats were classified by all three methods and the results also matched that of Repbase. This shows that several TEs have all characteristic properties but others exhibit only a single property. Here we have 64 repeats (57.65%) classified by more than one method and 47 repeats (42.35%) classified by only a single method. We note that even those repeats that were classified by only a single method also provided a classification that matched with that of Repbase. Hence, it is not a limiting factor when an element is classified by only one method. The following table shows the number of elements and the percentage of those repeats classified by various methods. A majority of those repeats; 24.19% were classified by the structural and TSD search and match with entries in Repbase.

Table 4.1 Split of classification by different methods for C. elegans

Classification based on	No. of Repeats	% of Total
Repbase + Homology + TSD + Structural	9	7.25
Repbase + Homology + TSD	6	4.82
Repbase + TSD + Structural	30	24.19
Repbase + Homology + Structural	19	15.34
Repbase + Homology	7	5.65
Repbase + TSD	15	12.1
Repbase + Structural	25	20.17
Repbase	13	10.48

Drosophila Melanogaster:

- Number of repeats in Repbase: 235
- Number of repeats after removing unclassified repeats: 156
- Number of repeats correctly classified by REPCLASS: 141
- Accuracy of classification: 95%
- Percentage classified: 90.38%

In the case of the D. melanogaster genome, REPCLASS was not able to classify 15 repeats that were classified by Repbase. The repeats that were not classified are 7 solo LTR retrotransposons, 3 non-LTR retrotransposons and 5 DNA transposons. The following diagram shows the number of repeats classified by each of the methods and the overlap.

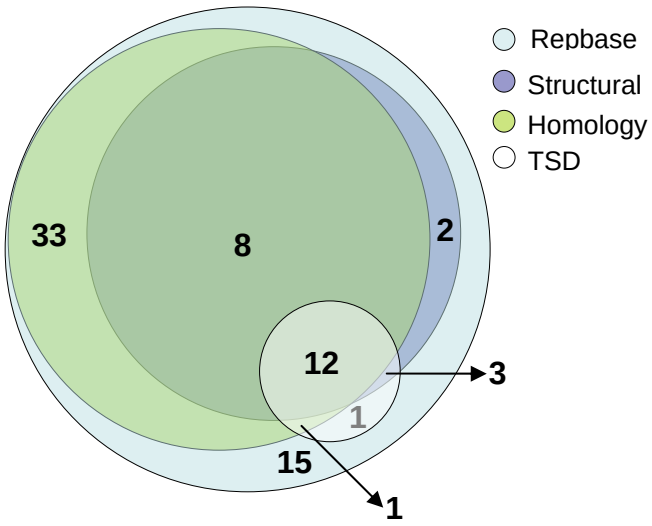


Figure 4.2: Classification distribution for Drosophila melanogaster in Repbase

There is significant overlap in multiple methods in the D. melanogaster genome. 12 repeats were classified by the methods and 89 repeats were classified by the homology and structural searches. Here, only 17 repeats had a TSD based classification. 105 repeats (74.47%) were classified by more than one method and 36 repeats (25.53%) were classified by only one method. This table shows the number of repeats and the percentages classified by the combination of the various methods.

Table 4.2 Split of classification by different methods for D. melanogaster

Classification based on	No. of Repeats	% of Total
Repbase + Homology + TSD + Structural	12	7.7
Repbase + Homology + TSD	1	0.64
Repbase + TSD + Structural	3	1.92
Repbase + Homology + Structural	89	57.05
Repbase + Homology	33	21.15
Repbase + TSD	1	0.64
Repbase + Structural	2	1.28
Repbase	15	9.62

There is a large amount of variation in the percentage of repeats classified by one method and multiple methods as compared to those of C. elegans. This also demonstrates that one particular method will not be sufficient for classification and that the combination of several methods is required.

4.2 Classification of C. Elegans repeats identified de novo with RepeatScout

The next step was to compare the repeats identified de novo with RepeatScout and those existing in Repbase. This provides an estimation of the performance of RepeatScout and REPCLASS combined. 35.8% (145 repeats) of the RepeatScout output was classified by REPCLASS.

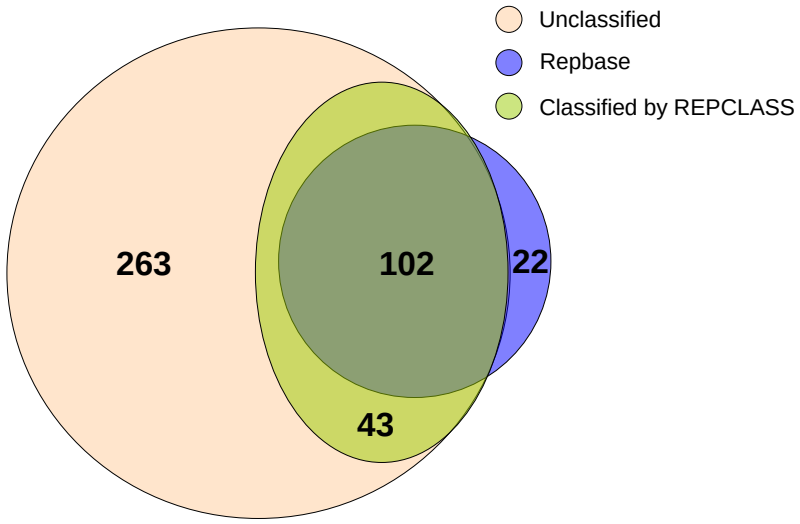


Figure 4.3: Classification distribution for C. elegans repeats identified de novo using RepeatScout

408 repeat consensuses were identified and 102 repeats match those of Repbase. RepeatScout identified 38 repeats with proper ends matching with repeats in Repbase. 22 repeats that are part of Repbase were not identified by the de novo method. 43 of the

identified repeats were classified by REPCLASS but do not show any significant similarity with repeats in Repbase. These 43 repeats potentially represent new families or subfamilies. This shows that REPCLASS combined with RepeatScout can be used as a powerful tool for the discovery of new repeat families. The remaining 263 repeats may include new repeat families but could not be classified by REPCLASS. There may be a fraction of these repeats that are fragments and further analysis is required to find their nature.

4.3 Classification of repeats in Ciona intestinalis and Strongylocentrotus purpuratus

The next step was to use the combination of REPCLASS and RepeatScout to identify and classify TEs in new genomes that do not have a comprehensive library of TE consensus sequences. The two genomes chosen, C. intestinalis and S. purpuratus represent a different taxonomy than C. elegans and D. melanogaster. C. intestinalis is a primitive species with a relatively small genome of 200 Mbp (Mega basepairs) long. The S. purpuratus is a much bigger genome with 800 Mbp. These two genomes do not exist as complete chromosomes and they consist of scaffolds and contigs. These genomes are currently being sequenced to provide better coverage and a more complete genome.

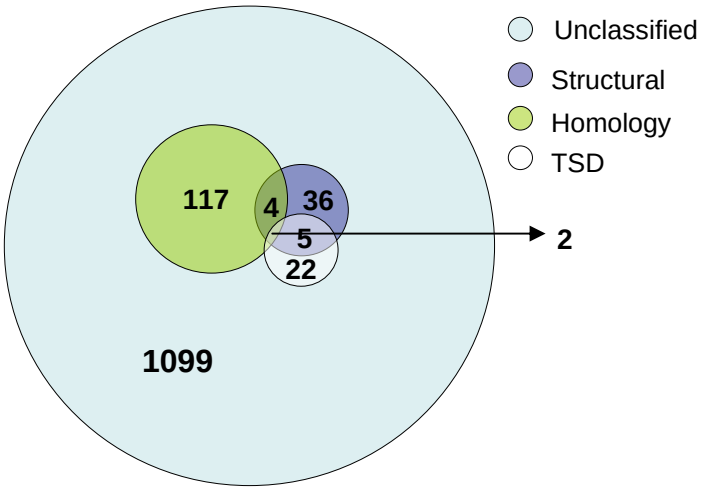


Figure 4.4: Classification distribution for C. intestinalis repeats identified de novo using RepeatScout

In the C. intestinalis genome 1285 repeats were identified of which only 186 TEs (14.47%) were classified and 1099 repeats were not classified by REPCLASS. 123 repeats were classified by the homology based search, 29 repeats by the TSD search and 47 repeats by the structural search. Only 11 repeats were classified by more than one method and 175 elements were classified by one method.

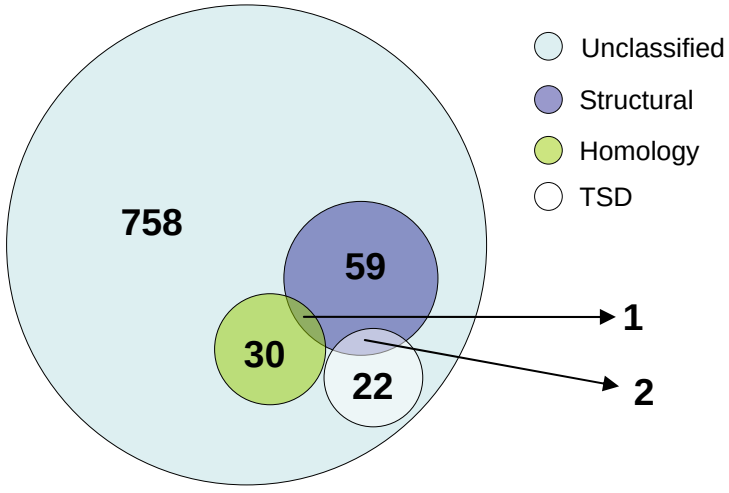


Figure 4.5: Classification distribution for S. purpuratus repeats identified de novo using RepeatScout

In the S. purpuratus genome 872 repeats were identified, of which 114 TEs (13.07%) were classified by REPCLASS and 758 repeats were not classified. An interesting observation here is that only 3 repeats were classified by more than one method and no repeats were classified by all three methods. 111 repeats were classified by only one method. The highest fraction of repeats (54.38%) was classified by the structural search. 31 repeats were classified by the homology search, 24 repeats by the TSD search and 62 repeats by the structural search.

Some of the difficulties faced in the classification of these repeats are:

- The de novo identified repeat consensus does not have well defined ends.
- There are many repeats that occur as fragments.
- Since the genomes are not complete and exist in contigs, the repeats were not correctly identified by RepeatScout.

4.4 Classification of repeats in the Human X Chromosome

The human genome was used because it is a much larger and more completely sequenced genome. The X chromosome was analyzed because of two reasons: (1) as a representative of one of the chromosomes, being neither the smallest nor the largest and (2) it is computationally more intensive to process the entire human genome and due to time constraints we chose only a single chromosome. Thus it is worthwhile to classify the human genome chromosome by chromosome and combine all of the repeats and remove the duplicates.

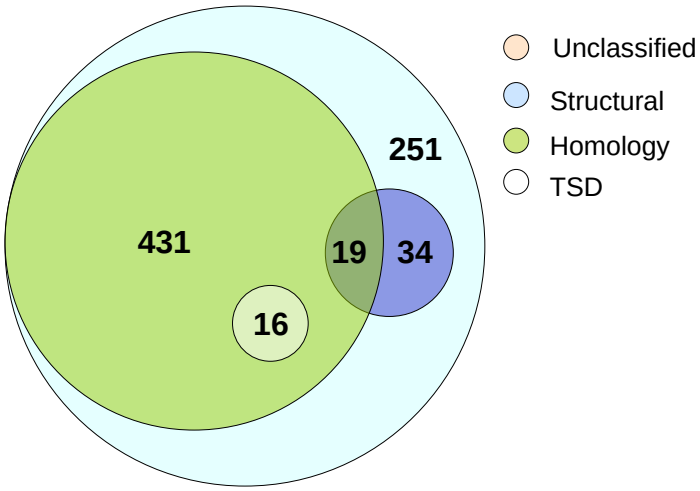


Figure 4.6: Classification distribution for Human X chromosome repeats identified de novo using RepeatScout

In the human genome the coding sequences are more conserved which returns more frequent hits from the homology based search. REPCLASS classified 500 repeats (66.6%) of the 751 repeat consensuses that were identified de novo. The repeats classified by the homology search was 466 (93.2% of the total classified). 16 repeats were classified by the TSD search and 53 elements were classified by the structural search. The low percentage of repeats identified by the other two methods is because of the nature of the input library sequences which do not have well conserved ends. Only 35 repeats (7%) were classified by more than one method and 465 repeats (93%) were classified by one method.

4.5 Cluster Performance

We measure the performance of REPCLASS using a computational cluster, based on two performance criteria: (1) Scalability and (2) Load balancing. All of the

experiments were performed on the DPCC cluster UTA at various load levels. REPCLASS exploits the data parallelism in the classification of repeats. A single program executes on different data sets, in this case the library of TE consensus sequences.

4.5.1 Scalability

We measure scalability in terms of the relationship of the increase in the number of processors leading to the decrease in total processing time for the classification. The TEs of each genome was run at differing system load, utilizing a fixed number of processors. The graphs depict the total time for classification as a function of increase in the number of processors. The scalability graphs were plotted for the classification of C. elegans repeats in Repbase, C. elegans repeats identified de novo by RepeatScout and D. melanogaster repeats in Repbase.

In an ideal case one might expect to see a near linear increase in speedup as the number of processors increase. Real biology data tends to have abnormalities and does not result in linear scalability, as shown in the following graphs. There is a saturation point for every data set, which is the sequential portion of the code. Upon reaching that point, regardless of the increase in the availability of resources, a constant time is required for processing that particular data. This saturation point will depend on each data set. In our case, the saturation points for C. elegans and D. melanogaster repeats in Repbase are 30 and 40 processors respectively. In the case of C. elegans repeats identified de novo using RepeatScout, the saturation point is much higher, because of the nearly 10 times more repeats found. Another notable point is, there is a tremendous

decrease in running time until we reach 10 processors after which the run-time decrease is more gradual. We see a linear increase in speedup from 2 to 5 processors.

Genomes with fewer repeats can be processed in several hours, even with a single processor, but there are organisms with a higher number of repeats which will take a longer time to process. Ciona intestinalis repeats identified de novo took approximately 5 days to complete on a single machine. With the use of the cluster we have classified all of the repeats in under 2 hours.

The variation in the processing scalability for different genomes is due to the size of the genome, the number of copies of repeats present in the genome, the TE size and the total number of TEs in the input library of TE consensus sequences. The saturation point is dependent on the size and complexity of the repeat. For a repeat with large number of copies the time taken for the TSD search will be longer because it has to retrieve all copies from the genome and moreover if the genome size is large, the search time increases.

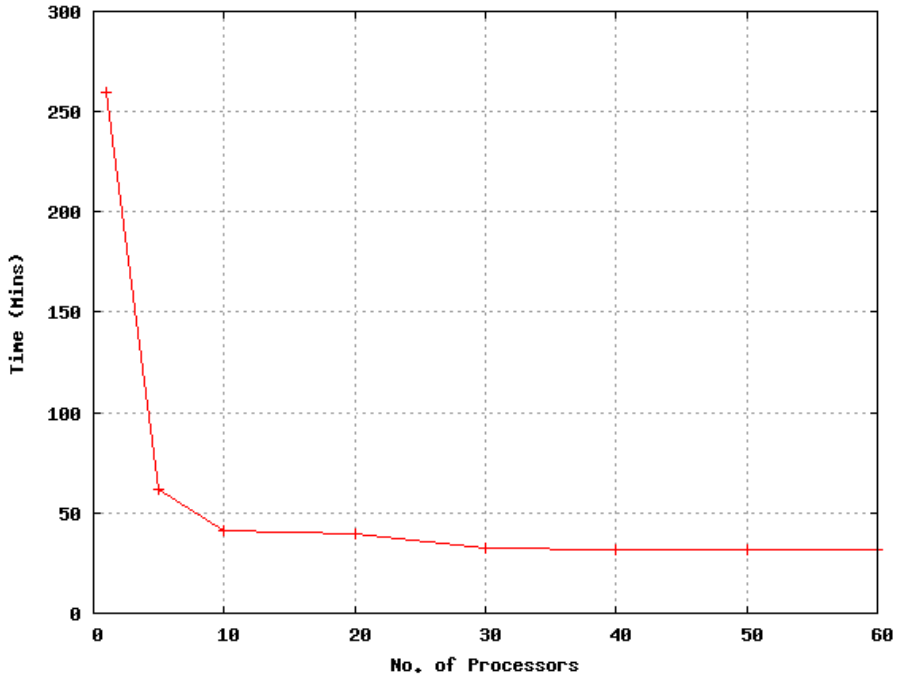


Figure 4.7: Scalability for Caenorhabditis elegans classification of repeats in Repbase

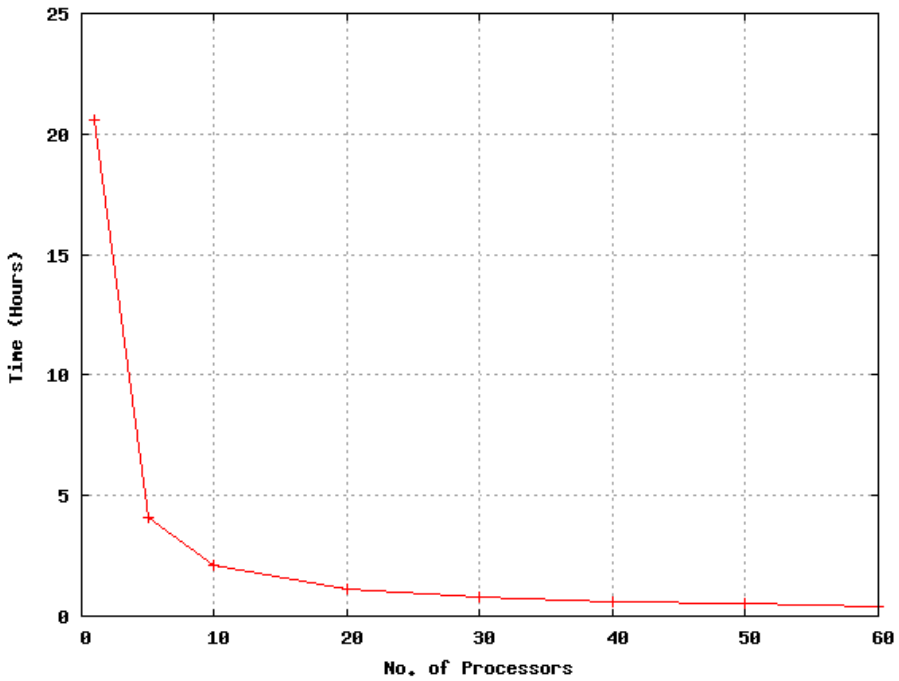


Figure 4.8: Scalability for Caenorhabditis elegans classification of repeats identified de novo using Repeat Scout

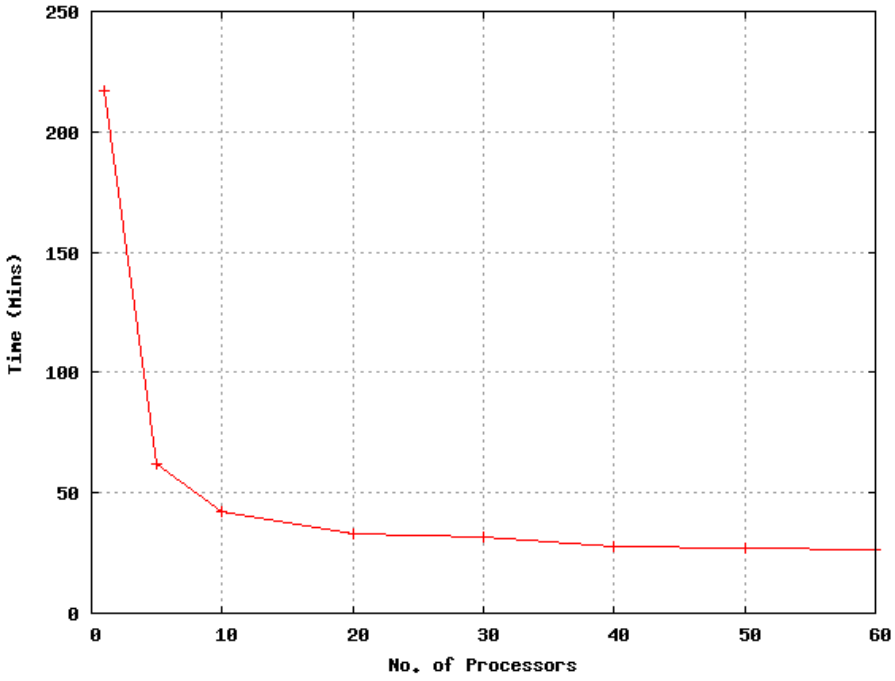


Figure 4.9: Scalability for Drosophila melanogaster classification of repeats in Repbase

4.5.2 Load Balancing

Load balancing is an important factor when using clusters. Load balancing is the distribution of work to each computational unit. REPCLASS achieves a good level of load balancing by dynamic spooling of data units rather than allocating the data set for each node at the beginning of processing. The following sets of graphs plot the cumulative work done by a fixed number of processors sorted according to time. The same data set was run first using 5 processors and then varying from 10 until 60 processors.

In C. elegans and D. melanogaster repeats available at Repbase, there are a few elements which require more computation than the average, and hence tend to take longer to finish. This is shown in the case with C. elegans in Figure 4.10 where for

processor 1, when the load is distributed across 20 to 60 processors, the time taken is significantly more than the remaining processors. This is due to some elements which take a longer time to complete, increasing the overall time. The same phenomenon is observed with D. melanogaster in Figure 4.12.

For C. elegans repeats identified de novo shown in Figure 4.11 there is a more even distribution of load across all of the processors, this is the closest to the ideal case that we obtained. In the other species, we show that apart from the first 10 processors the load across the remaining processors is distributed evenly.

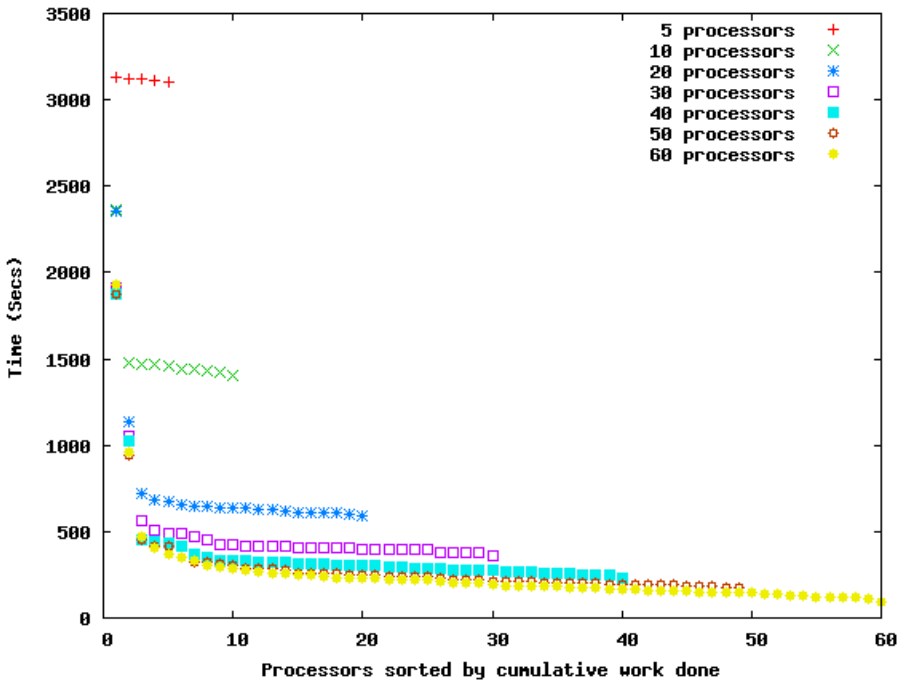


Figure 4.10: Load balancing for Caenorhabditis elegans classification of repeats in Repbase

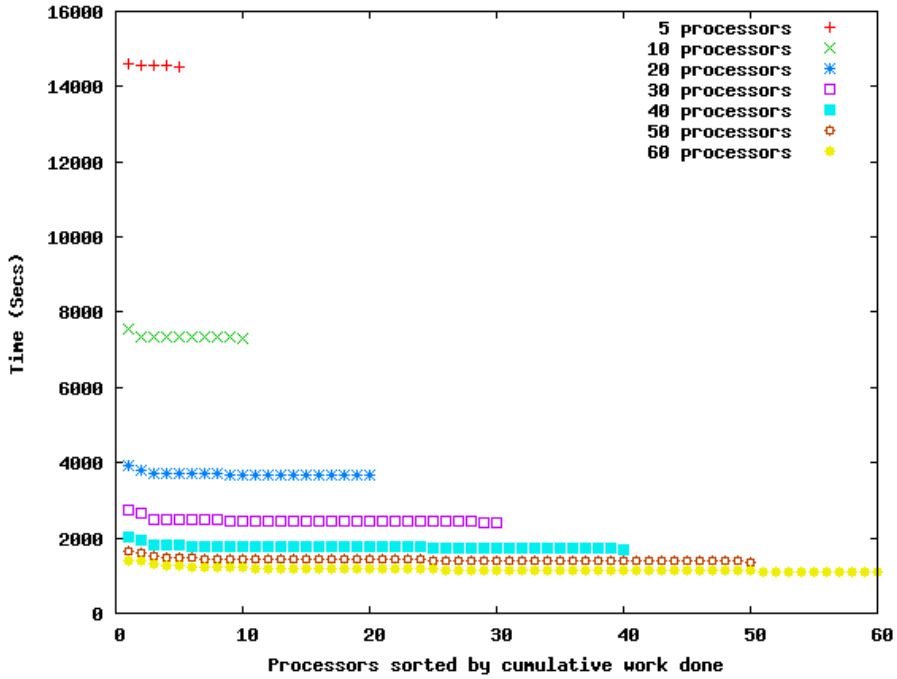


Figure 4.11: Load balancing for Caenorhabditis elegans classification of repeats identified de novo using RepeatScout

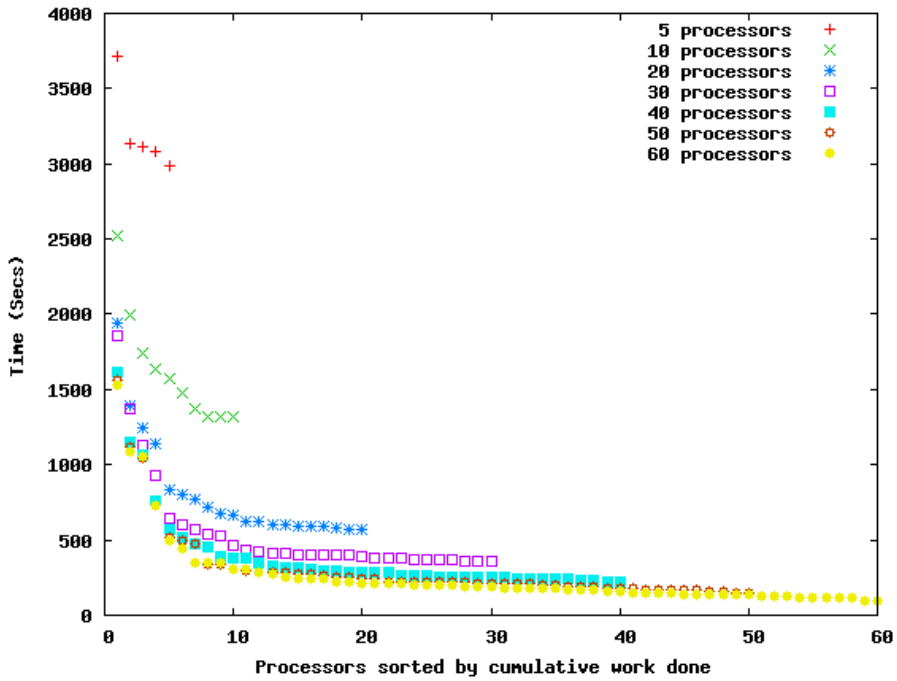


Figure 4.12: Load balancing for Drosophila melanogaster classification of repeats in Repbase

4.5.3 Turnaround time

The C. elegans and D. melanogaster genomes are small and have a small numbers of repeats. The largest genome is the human genome and has a potential of having a large number of repeats. Due to the size of the human genome, only the X chromosome was considered as a test case. Figure 4.13 shows the turnaround time for the classification of the human X chromosome repeats identified de novo.

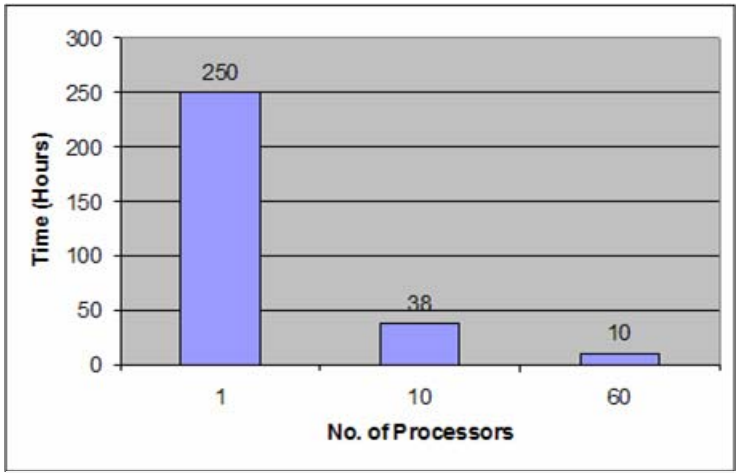


Figure 4.13: Turnaround time for classification of Human X Chromosome repeats identified de novo using RepeatScout

It took 10 days and 10 hours to complete the classification of the human X chromosome one a single node. There was drastic decrease in the turn around time with an increase in the number processors. We were able to complete the classification in 10 hours using 60 processors, a speedup of 25. The X chromosome represents 4.8% of the entire human genome and it might take nearly 7 months to complete the classification for the entire human genome on a single machine. With the use of the cluster that task

can be completed within 8 days using 60 processors and even faster with more processors. This variation is not constant across all the genomes and depends largely on the time taken to classify unusual repeats.

The use of clusters may not be justified for small genomes, with a small number of repeats that can be processed in a short time frame. But there is so much variation in the size of the genomes that it becomes more efficient to use a cluster, considering the case of the human genome and other large genomes. Moreover, even for small genomes, tuning different parameters to get fine grained results will benefit with the use of clusters, by achieving a faster turn around time for multiple parameter sets of the same genome.

CHAPTER 5

CONCLUSION

5.1 Conclusion

We have developed a tool that will automatically perform the classification of transposable elements. The tool has been tested for various genomes, each having diversity in their transposable element content.

5.1.1 Validation of REPCLASS design and functionality

The control for the efficiency of REPCLASS was the repeats available in Repbase for C. elegans and D. melanogaster. We were able to classify 89.52% and 90.38% of those repeats. Moreover the accuracy of the classification was 92% and 95% respectively. These two genomes have varied TEs and REPCLASS performed well under both conditions. C. elegans has more Class II elements and D. melanogaster has more Class I elements. REPCLASS is extremely efficient at retrieving the correct annotation of known repeats. These experiments validate the design and structure of REPCLASS.

5.1.2 Assessing the efficiency of the combination of REPCLASS and RepeatScout to classify repeats de novo

Repeats in Repbase have been manually curated and are more accurate. Repbase is not able to keep up with the number of new genomes that are being sequenced. Hence

the process of identification of repeats needs to be performed de novo. We compare the classification of C. elegans repeats identified de novo with those existing in Repbase. It revealed that there are more repeats that are not annotated by Repbase. At first this seems surprising because C. elegans has one of the smallest genomes among multicellular eukaryotes and one might expect this genome to be exhaustively characterized in terms of repeat content. The fact that RepeatScout identified 306 repeat families that do not significantly match any repeats catalogued in Repbase suggests that RepeatScout generates a large number of false positives and artifacts and/or that many repeats are still unknown and remain to be characterized in this species. Among those 306 potentially novel repeats, REPCLASS was able to classify 43 unambiguously as members of known TE subclasses and/or superfamilies. These clearly represent new TE families and shows that the combination of REPCLASS and RepeatScout can be used as an efficient tool for TE discovery. Nevertheless, 263 repeats identified by RepeatScout remain unclassified by REPCLASS. There are several explanations for this result. First, it is possible that a large fraction of these repeats do not represent TEs but other types of repetitive DNA, including segmental duplications and gene families. It may be possible to identify gene families by searching for non-TE proteins using blastx against the Genbank protein databases. Second, it is likely that a fraction of the unknown repeats represent entirely new types of mobile elements that have not yet been described and possess unique characteristics that cannot be recognized by the current version of REPCLASS. This emphasizes one of the obvious limitations of REPCLASS; the fact that it can only classify elements harboring sequences and features typical of

known repeats. As new repeats are described, it will be necessary to update REPCLASS to incorporate the unique characteristics and signatures of the novel elements. In this regard, one way to see REPCLASS is as a useful tool to filter known repeats and isolate repeats potentially representing entirely new types of mobile elements. We plan to further analyze the unclassified elements to differentiate between these non-mutually exclusive hypotheses.

5.1.3 Exploration of repeats identified de novo in new genome sequences

We next assessed the ability of REPCLASS to assist in the classification and annotation of repeats identified de novo in yet unexplored or poorly characterized genomes. As a first step, we decided to focus on the genome sequences of two very different model organisms, the tunicate C. intestinalis (sea squirt) and the echinoderm S. purpuratus (sea urchin) for which little information was available regarding their repeats and TEs. RepeatScout generated a very large number of consensus repeat sequences for C. intestinalis (1285), despite its very compact genome. For S. purpuratus, which has a genome size 4 times larger than C. intestinalis, RepeatScout yielded a smaller number of consensus sequences (872), although still a significantly higher amount than for C. elegans (408). This suggests either that both C. intestinalis and S. purpuratus harbor an extremely rich diversity of repetitive sequences and/or that RepeatScout tend to produce a high rate of repeat fragmentation with these input sequences, leading to an increased number of consensus sequences in the output. In this regard, it is important to note that both species have been sequenced using a shotgun approach with a moderate genomic coverage. By definition, such sequencing methods

are expected to give rise to a large amount of repeat fragmentation during the assembly of contigs, especially if the repeats are large in size. In contrast, the genome sequence of C. elegans was obtained by a clone-by-clone approach and the overall quality of the sequence is very high. Most gaps in the sequence have been filled and the genome is assembled into long chromosome-sized pseudomolecules with essentially no interruption. We believe that this is one of the major reasons why REPCLASS is unable to classify a larger fraction of the repeats identified de novo by RepeatScout. This is because the two ends of the repeats are often required for RECLASS to find characteristic features of TEs (TSD and structural searches).

In order to test this hypothesis, we used RepeatScout and REPCLASS to identify and classify de novo repeats on the human X chromosome. We selected this species because it contains a high diversity and a very large amount of repeats (~45% of the genome and >600 families). Because of time and computation constraint, we decided to focus on a single chromosome and selected the X chromosome because of its large size (152 Mbp) and relatively high-quality assembly. RepeatScout identified and compiled 751 consensus repeats on this chromosome and 66.6% of them could be classified by REPCLASS. It should be noted that 93.2% were classified based on the homology search module of REPCLASS, reflecting the fact that most human repeats contain coding sequences. Therefore, this experiment does not allow us to assess the quality and integrity of the repeat consensus sequences generated by RepeatScout. It is likely that this remains a major limiting factor in the accuracy of REPCLASS. Nonetheless, this is a very encouraging result considering the complexity and the

abundance of repeats in humans. This result indicates that the combination of RepeatScout and RECLASS provide a powerful tool for repeat finding and annotation of complex genome sequences, provided that the input sequence is of relatively good quality.

5.1.4 Assessing computational performance of REPCLASS

Classification of TE is computationally intensive and can take from 5 hours for the C. elegans genome to 7 months in the case of the human genome. Using the cluster provides faster turn around and high scalability for the performance of REPCLASS. The speedup with the increase in processors varied from 5 for the C. elegans genome up to 25 for the human X chromosome. The speedup varies significantly due to the size of the genome, the complexity of the repeats and the number of repeats in the input TE library of consensus sequences. The distribution of load by REPCLASS ensures proper load balancing across all nodes. The load balancing also varies in cases where certain TE sequences take more time to process. The speed up can further be increased with the use of a computational grid. Finally the performance of REPCLASS greatly depends on the quality, size and complexity of the genomes and the TE sequences.

5.2 Future Work

Very few repeats are known to populate the mitochondrial and chloroplast genomes, although a few elements similar to those found in nuclear genomes have also been identified in some organelle genomes. REPCLASS could also be used to search and classify TEs in organelle genomes. The largest fraction of repeats in prokaryotic

genomes is composed of insertion sequences, which share structural and coding similarity to those of class II transposons in eukaryotic genomes. REPCLASS will be able to recognize and classify the vast majority of prokaryotic repeats. We plan to add specific feature searches for prokaryotic mobile elements (phages, integrons, replicons) in order to comprehensively analyze the repeat content of prokaryotic genomes.

A major limitation in the classification of repeats is the quality of the input library of repeats. RepeatScout or other de novo repeat identification tools needs to define the boundaries of repeats more accurately and reduce the amount of repeat fragmentation. Some parts of REPCLASS function based on existing knowledge of TEs and their properties. This knowledge needs to be transferred for new repeats that are discovered by creating new modules and updating the information for the TSD classification. Furthermore we need to make REPCLASS open source under GPL license and create a web interface to access REPCLASS. The repeats that are annotated by REPCLASS can be used to update the Repbase database and assist in the masking of repeats using programs such as RepeatMasker.

REFERENCES

[1] Smit AF. Repeat Masker, http://www.repeatmasker.org.

[2] Bedell J, Korf I, Gish W. (2000) Masker Aid: a performance enhancement to RepeatMasker. Bioinformatics, 16: 1040–1041.

[3] Jurka J, Klonowski P, Dagman V, Pelton P. (1996) CENSOR - a program for identification and elimination of repetitive elements from DNA sequences. Computers and Chemistry, 20: 119-122.

[4] Benson G, (1999). Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Research, 27: 573-580.

[5] Emboss (The European Molecular Biology Open Software Suite), http://emboss.sourceforge.net/

[6] Jurka J, Kapitonov VV, Pavlicek A, Klonowski P, Kohany O. (2005) Repbase Update, a database of eukaryotic repetitive elements. Cytogenetic Genome Research, 110: 462-467.

[7] Misra S, Crosby MA, Mungall CJ, Matthews BB, Cambpell KS, et al. (2002) Annotation of the Drosophila melanogaster euchromatic genome: A systematic review. Genome Biology, 3(12): research0083.1 – 0083.22.

[8] Mungall CJ, Misra S, Berman BP, Carlson J, Frise E, et al. (2002) An integrated computational pipeline and database to support whole-genome sequence annotation. Genome Biology 3(12): research0081.1 – 0081.11.

[9] Allen JE, Pertea M, Salzberg SL. (2004) Computational gene prediction using multiple sources of evidence. Genome Research, 14: 142–148.

[10] Ding L, Sabo A, Berkowicz N, Meyer RR, Shotland Y, et al. (2004) EAnnot: A genome annotation tool using experimental evidence. Genome Research 14: 2503 – 2509.

[11] Quesneville H, Bergman CM, Andrieu O, Autard D, NouaudD, et al. (2005) Combined evidence annotation of transposable elements in genome sequences. PLoS Comp. Biol., 1(2): e22.

[12] DOEgenomes.org genomics primer, http://www.ornl.gov/sci/techresources/Human_Genome/publicat/primer2001/1.shtml

[13] McClintock B. (1950) The origin and behavior of mutable loci in maize. Proc. Natl. Acad. Sci., 36: 344-349.

[14] Kapitonov VV, Jurka J. (2003) Molecular paleontology of transposable elements in the Drosophila melanogaster genome. Proc. Natl. Acad. Sci., 100: 6569-6574.

[15] Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, et al. (2001) Initial sequencing and analysis of the human genome. Nature 409: 860-921.

[16] Harshey RM and Bukhari AI (1981) A mechanism of DNA transposition. Proc. Natl. Acad. Sci., 78: 1090-1094.

[17] Feschotte C (private communication), 2005.

[18] Mullis KB. The Polymerase Chain Reaction (Nobel Lecture). Angewandte Chemie Internationl Edition in English, 33: 1209-1213.

[19] Alrschul SF, Gish W, Myers EW, and Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology, 215: 403-410.

[20] Bao Z and Eddy SR. (2002) Automated de novo Identification of Repeat Sequence Families in Sequenced Genomes. Genome Research, 12: 1269-1276.

[21] Price AL, Jones NC and Pevzner PA. (2005) De novo identification of repeat families in large genomes. Bioinformatics, 21: i351-i358.

[22] Edgar RC and Myers EW. (2005) PILER: identification and classification of genomic repeats. Bioinformatics 21(Suppl 1): i152-i158.

[23] Li R, Ye J, Li S, Wang J, Han Y, et al. (2005) ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. PLoS Computational Biology 1(4): E43.

[24] Juretic N, Bureau TE, Bruskiewich RM. (2004) Transposable element annotation of the rice genome. Bioinformatics, 20: 155–160.

[25] Foster I, Kesselman C and Tuecke S, (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. Supercomputer Applications, 15(3), 2001.

[26] Stevens RD, Robinson AJ, Goble CA, (2003). ^{my}Grid: personalised bioinformatics on the information grid. Bioinformatics, 19: i302-i304.

[27] Lipman DJ and Pearson WR. (1985) Rapid and Sensitive Protein Similarity Searches. Science, 227: 1435-1441.

[28] Wooton JC and Federhen S. (1993) Statistics of local complexity in amino acid sequences and sequence databases. Computers and Chemistry, 17: 149–163.

[29] Gish WR, Washington University BLAST. http://blast.wustl.edu/.

[30] Baker W, Broek A, Camon E, Hingamp P, Sterk P, Stoesser G and Tuli MA. (2000) The EMBL Nucleotide Sequence Database. Nucleic Acids Research, 28: 19-23.

[31] Kapitonov VV, Jurka Jerzy. (2001) Rolling-circle transposons in eukaryotes. Proc. Natl. Acad. Sci., 98: 8714-8719.

[32] Feschotte C, Wessler SR. (2001) Treasures in the attic: Rolling circle transposons discovered in eukaryotic genomes. Proc. Natl. Acad. Sci., 98: 8923-8924.

[33] tRNAscan-SE, tRNA detection in large-scale genome sequence. http://selab.wustl.edu/cgi-bin/selab.pl?mode=software#trnasca

BIOGRAPHICAL INFORMATION

Nirmal Ranganathan joined the University of Texas at Arlington in the fall of 2003. He received his Bachelor's degree in Computer Science and Engineering from Kongu Engineering College, Perundurai, India, affiliated to the Bharathiar University, Coimbatore, India. His research interests are grid computing, high performance computing and bioinformatics work involving transposable elements. He received his M.S in Computer Science and Engineering in 2005.