# Interruptional Activity and Simulation of Transposable Elements

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Lingling Jin

# Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

# Abstract

Transposable elements (TEs) are interspersed DNA sequences that can move or copy to new positions within a genome. The active TEs along with the remnants of many transposition events over millions of years constitute 46.69% of the human genome. TEs are believed to promote speciation and their activities play a significant role in human disease. The 22 *AluY* and 6 *AluS* TE subfamilies have been the most active TEs in recent human history, whose transposition has been implicated in several inherited human diseases and in various forms of cancer by integrating into genes. Therefore, understanding the transposition activities is very important.

Recently, there has been some work done to quantify the activity levels of active *Alu* transposable elements based on variation in the sequence. Here, given this activity data, an analysis of TE activity based on the position of mutations is conducted. Two different methods/simulations are created to computationally predict so-called harmful mutation regions in the consensus sequence of a TE; that is, mutations that occur in these regions decrease the transposition activities dramatically. The methods are applied to *AluY*, the youngest and most active *Alu* subfamily, to identify the harmful regions laying in its consensus, and verifications are presented using the activity of *AluY* elements and the secondary structure of the *AluYa5* RNA, providing evidence that the method is successfully identifying harmful mutation regions. A supplementary simulation also shows that the identified harmful regions covering the *AluYa5* RNA functional regions are not occurring by chance. Therefore, mutations within the harmful regions alter the mobile activity levels of active *AluY* elements. One of the methods is then applied to two additional TE families: the *Alu* family and *L1* family, in detecting the harmful regions in these elements computationally.

Understanding and predicting the evolution of these TEs is of interest in understanding their powerful evolutionary force in shaping their host genomes. In this thesis, a formal model of TE fragments and their interruptions is devised that provides definitions that are compatible with biological nomenclature, while still providing a suitable formal foundation for computational analysis. Essentially, this model is used for fixing terminology that was misleading in the literature, and it helps to describe further TE problems in a precise way. Indeed, later chapters include two other models built on top of this model: the sequential interruption model and the recursive interruption model, both used to analyze their activity throughout evolution.

The sequential interruption model is defined between TEs that occur in a genomic sequence to estimate how often TEs interrupt other TEs, which has been shown to be useful in predicting their ages and their activity throughout evolution. Here, this prediction from the sequential interruptions is shown to be closely related to a classic matrix optimization problem: the Linear Ordering Problem (LOP). By applying a well-studied method of solving the LOP, Tabu search, to the sequential interruption model, a relative age order of all

TEs in the human genome is predicted from a single genome. A comparison of the TE ordering between Tabu search and the method used in [47] shows that Tabu search solves the TE problem exceedingly more efficiently, while it still achieves a more accurate result. As a result of the improved efficiency, a prediction on all human TEs is constructed, whereas it was previously only predicted for a minority fraction of the set of the human TEs.

When many insertions occurred throughout the evolution of a genomic sequence, the interruptions nest in a recursive pattern. The nested TEs are very helpful in revealing the age of the TEs, but cannot be fully represented by the sequential interruption model. In the recursive interruption model, a specific context-free grammar is defined, describing a general and simple way to capture the recursive nature in which TEs nest themselves into other TEs. Then, each production of the context-free grammar is associated with a probability to convert the context-free grammar into a stochastic context-free grammar that maximizes the applications of the productions corresponding to TE interruptions. A modified version of an algorithm to parse context-free grammars, the CYK algorithm, that takes into account these probabilities is then used to find the most likely parse tree(s) predicting the TE nesting in an efficient fashion.

The recursive interruption model produces small parse trees representing local TE interruptions in a genome. These parse trees are a natural way of grouping TE fragments in a genomic sequence together to form interruptions. Next, some tree adjustment operations are given to simplify these parse trees and obtain more standard evolutionary trees. Then an overall TE-interaction network is created by merging these standard evolutionary trees into a weighted directed graph. This TE-interaction network is a rich representation of the predicted interactions between all TEs throughout evolution and is a powerful tool to predict the insertion evolution of these TEs. It is applied to the human genome, but can be easily applied to other genomes. Furthermore, it can also be applied to multiple related genomes where common TEs exist in order to study the interactions between TEs and the genomes.

Lastly, a simulation of TE transpositions throughout evolution is developed. This is especially helpful in understanding the dynamics of how TEs evolve and impact their host genomes. Also, it is used as a verification technique for the previous theoretical models in the thesis. By feeding the simulated TE remnants and activity data into the theoretical models, a relative age order is predicted using the sequential interruption model, and a quantified correlation between this predicted order and the input age order in the simulation can be calculated. Then, a TE-interaction network is constructed using the recursive interruption model on the simulated data, which can also be converted into a linear age order by feeding the adjacency matrix of the network to Tabu search. Another correlation is calculated between the predicted age order from the recursive interruption model and the input age order. An average correlation of ten simulations is calculated for each model, which suggests that in general, the recursive interruption model performs better than the sequential interruption model in predicting a correct relative age order of TEs. Indeed, the recursive interruption model achieves an average correlation value of $\rho = 0.939$ with the correct simulated answer.

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to Dr. Ian McQuillan, my Master's and Ph.D. supervisor, for his incredible amount of guidance and support in overcoming numerous difficulties that I have faced through my research.

Special thanks to Dr. Longhai Li, my cognate committee member, who has provided me with lots of ideas and guidance on statistical aspects of my thesis.

I am also grateful to the other members of my advisory committee: Dr. Tony Kusalik, Dr. Mark Keil, and Dr. Michael Domaratzki for their invaluable suggestions and comments.

I want to acknowledge my colleagues in the Bioinformatics lab and my friends for their friendship and helpful discussions.

I am grateful to my husband, Shi, and my children, April and Avery, for being the love and happiness of my life. Very special thanks goes to my husband for helping me with the implementation and graphic design in the thesis.

Last but not the least, I would like to acknowledge my parents, Xin and Zhimei, and my parents-in-law, Zhendong and Shuping. Thank you for all your love and support.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| TE | Transposable Elements |
| bp | base pairs |
| Kbp | Kilobase pairs |
| Mbp | Megabase pairs |
| RU | Repbase Update |
| RM | RepeatMasker |
| Myr | Million years |
| MYA | Million Years Ago |
| LINE | Long Interspersed Nuclear Elements |
| SINE | Short Interspersed Nuclear Elements |
| LTR | Long Terminal Repeat |
| TIR | Terminal Inverted Repeat |
| TSD | Target Site Duplication |
| ORF | Open Reading Frame |
| IM | Interruption Matrix |
| LOP | Linear Ordering Problem |
| CFG | Context-free Grammar |
| SCFG | Stochastic Context-free Grammar |
| AM | Adjacency Matrix |

# Chapter 1

# Introduction and objectives

Transposable elements were first discovered by Barbara McClintock in the 1950s during her studies of maize, work for which McClintock received the Nobel Prize in Physiology or Medicine in 1983. The patterns of colour in maize kernels changed in different breeding crosses, which was interpreted in her study as a result of the regulation of gene activity by some mobile genetic elements. These elements can move from place to place within/between the chromosomes. As the elements move, they mutate genes in some of the cells and change the colour of maize kernels due to their effects on pigmentation genes [14]. These mobile genetic elements are named *transposable elements* (TEs), or *transposons*.

Transposable elements were dismissed at one point as being useless, but they are emerging to be thought of as major players in evolution. Indeed, the impact of TEs on genome evolution appears to be extensive and they are even believed to promote speciation [42] and can therefore be seen as a driver of evolution. The evolutionary history of a TE family in a species may represent a plentiful source of information about genome evolution. Additionally, more and more evidence is emerging that active TEs play a significant role in human biology and disease as they create genetic diversity in human populations and can integrate into genes, potentially causing disease. However, little analysis is currently being undertaken in determining what factors influence their activity, what occurred throughout evolution, and in understanding how TEs change over time.

## 1.1 Motivations

In this section, the motivations of the thesis will be discussed from the impacts of TEs on genomes as well as some cases of human diseases caused by TE retrotranspositions.

### 1.1.1 The impact of TEs on genomes

Though the number of genes in a genome grows from bacteria to higher organisms, it is the repeats, especially TEs, that account for the major differences in genome size within species, and even between closely related species [14]; that is, genome size is not correlated with the complexity of the organism [143]. Retrotransposons are major players in promoting the increase of genome size. It has been shown using genomic studies of ancient human remains that the human genome is continuing to expand at a rate between 1 and 10 million base pairs per million years, and this expansion is heavily influenced by *retrotransposition*, the transposition of retrotransposons [71]. For example, there are ∼2,000 *L1* and ∼7,000 *Alu* copies accumulated over the past ∼6 Myr of human evolution [127]. Not only in humans, TEs influence plant genome sizes significantly as well. The sizes of plant genomes span across many orders of magnitude ranging from about 63 Mbp of the *Genlisea* genome [51] to more than 110,000 Mbp of the lily *Fritillaria assyriaca* [4], which is primarily the consequence of polyploidization and TE proliferation [27]. Moreover, studies of maize [122] and the rice *Oryza australiensis* [116] show that LTR retrotransposition doubled the genome size in the two species independently.

TEs also impact host genomes by generating genomic instability due to their continuous activity over years. The major way that a retrotransposon alters genome function is by inserting itself into protein-coding or regulatory regions. There are a number of examples of genetic disorders that are caused by the expansion of microsatellites [65, 26, 28], and non-LTR retrotransposons are the source of microsatellites. For example, it is known that about 20% of all microsatellites shared by the human and chimpanzee genomes lie within *Alu* elements [72].

Retrotransposons can generate genomic rearrangements as well, such as deletions, duplications, inversions, or translocations. It was observed in cultured human cells that ∼20% of *L*1 insertions were related with structural rearrangements [30], including insertion-mediated deletions; that is, concurrent deletions at the insertion site. It has been estimated that during primate evolution, about 45,000 insertion-mediated deletions might have removed over 30 Mbp of genomic sequence [29].

Moreover, transposons can mediate the formation of a new gene family [106] or impact gene expression. TEs may influence the expression of surrounding genes often due to a regulatory promoter[1] and terminator[2] sequence occuring in LTRs [101]. Thus, TEs can be used as tools in genetic research to either open or terminate the expression of nearby genes.

Last but not least, TEs are of interest for their own sake as they have been a powerful evolutionary force in shaping genomes. The ubiquity of TEs raises a number of questions about the relationships between TEs

[1] A *promoter* is a region of DNA that facilitates the transcription of a particular gene. Promoters are located near the genes they regulate, on the same strand and typically upstream (towards the 5' region of the sense strand).

[2] A *terminator* is a section of genetic sequence that marks the end of gene or operon on genomic DNA for transcription.

and their host genomes and the significance of these elements on the evolution of their hosts. Knowledge of the location of TEs can also be helpful in the determination of the evolutionary history of a locus of a genome.

### 1.1.2 Transposable elements causing human diseases

TEs have co-existed with their host organisms for an exceptionally long period, during which their transposition activities have contributed to their host genomes in both positive and negative ways. Transposition of currently active elements as well as recombination involving repetitive sequences can be responsible for genetic diseases, and there is a growing understanding of the specific negative impacts of TEs in human disease. Generally, the TE-associated insertional mutagenesis and recombination may cause DNA damage and contribute to human diseases. In this subsection, some examples of human diseases caused by TE activities will be described.

Retrotransposition events occur in both the germline and somatic tissues [30]. The transposition of TEs has been implicated in processes ranging from cancer to brain development. The brain has one of the highest average frequencies of transposable element activity of all tissues in the human body [86]. In very recent research on Alzheimer's disease, a molecular mechanism of the Alzheimer's process was proposed to be caused by *Alu* elements losing their normal controls as a person ages, wreaking havoc on the machinery that supplies energy to brain cells and leading to a loss of neurons and dementia [86]. The authors hypothesize that through human-specific neurologic pathways, *Alu* insertions in mitochondrial genes can cause progressive neurological disfunction, which may underlie the origin of higher cognitive function [86]. Therefore, retrotransposons have played an important role in primate evolution.

Human TEs have been reported to cause several types of cancer, such as breast cancer, colon cancer, retinoblastoma, neurofibromatosis, hepatoma, etc., through insertional mutagenesis of genes that are important to malignant transformation [10]. For example, the most up-to-date research in [126] has shown that a hot *L1* (defined as showing at least one-third of the activity of $L1_{RP}$ in [17]) source TE on Chromosome 17 of a patient's genome interrupted somatic repression in normal colon tissues, and initiated colorectal cancer by disrupting the APC tumor suppressor gene [103]. The fact that *L1* transpositions occur in human tumors suggests the possibility that somatic *L1* insertions may play a role as driver mutations during the stages of initiation, progression, and metastasis of tumors [126].

Non-LTR retrotransposons are deemed as the major source of TE-related mutagenesis in the human genome. There have been a number of cases shown to cause heritable diseases, as a result of human genetic disorders caused by *de novo L1*, *Alu* and *SVA* insertions, such as haemophilia, cystic fibrosis, Apert syndrome, neurofibromatosis, Duchenne muscular dystrophy, $\beta$-thalassaemia, and hypercholesterolaemia [30]. Table 1.1,

| TE insertion | Chromosome | Disease caused by TE insertion | Reported papers |
|---|---|---|---|
| *Alu* insertions | X | Hemophilia B | [34, 25] |
| | X | Hemophilia A | [25] |
| | X | Dent's disease | [25] |
| | X | X-linked agammaglobulinemia | [25] |
| | X | X-linked severe combined immunodeficiency | [34] |
| | X | Glycerol kinase deficiency | [34] |
| | X | Adrenoleukodystrophy | [25] |
| | X | Menkes disease | [52] |
| | X | Hyper-immunoglobulin M syndrome | [5] |
| | 1 | Retinal blinding | [25] |
| | 1 | Type 1 antithrombin deficiency | [25] |
| | 2 | Muckle-Wells syndrome | [25] |
| | 2 | Hereditary non-polyposis colorectal cancer | [25] |
| | 3 | Hypocalciuric hypercalcemia and hyperparathyroidism | [34] |
| | 3 | Cholinesterase deficiency | [34] |
| | 3 | Aplasia anterior pituitary | [25] |
| | 5 | Associated with leukemia | [34] |
| | 5 | Hereditary desmoid disease | [34] |
| | 7 | Chronic hemolytic anemia | [95] |
| | 7 | Cystic fibrosis | [24] |
| | 8 | Branchio-oto-renal syndrome | [25] |
| | 8 | Lipoprotein lipase deficiency | [110] |
| | 8 | CHARGE syndrome | [142] |
| | 9 | Walker Warburg syndrome | [15] |
| | 10 | Autoimmune lymphoproliferative syndrome | [25] |
| | 10 | Apert syndrome | [34] |
| | 11 | Complement deficiency | [34] |
| | 11 | Acute intermittent porphyria | [25] |
| | 12 | Human-specific evolutionary change | [15] |
| | 12 | Mucolipidosis type II | |
| | 13 and 17 | Breast cancer | [34] |
| | 17 | Neurofibromatosis | [34] |
| *L1* insertions | X | Choroideremia | [25] |
| | X | Chronic granulomatous disease | [25] |
| | X | X-linked Duchenne muscular dystrophy | [25, 111] |
| | X | Hemophilia A | [25] |
| | X | Hemophilia B | [25] |
| | X | X-linked retinitus pigmentosa | [25] |
| | X | Coffin-Lowry syndrome | [25] |
| | 5 | Colon cancer | [25] |
| | 9 | Fukuyama-type congenital muscular dystrophy | [25] |
| | 11 | Beta-thalassemia | [25] |
| | 11 | Pyruvate dehydrogenase complex deficiency | [15] |

**Table 1.1:** Human diseases caused by TE insertions.

adapted from [9], gives more details about some diseases caused by the insertions of *L1* and *Alu*, the active elements in humans.

### 1.1.3 Perspective

It is important to understand the patterns of the activities of TEs and the factors that may change their activities, because such TE activities can powerfully influence the structure of the genome, including the capacity of chromosomes to rearrange and to regulate transcription. And, as discussed above, they are often important in understanding human disease. Further, identifying repetitive DNA sequences in eukaryotic DNA is essential in genome analysis, because these repeats offer an opportunity to study molecular evolution as "molecular fossils" in evolutionary studies based on comparative analysis of genomes from different species. It is amazing how much regarding evolution can be inferred from a single genome sequence, as most evolutionary analysis require sequences from multiple genomes. This might be useful in situations where TEs change quickly and multiple genomes are not available. The dynamic of what causes TE families as a whole to evolve is also of interest.

## 1.2 Objectives and layout of the thesis

The research of this thesis will be conducted only on data regarding human TEs. We intend to contribute to the understanding of how the TE propagation through evolution shapes the genome, how the age and lifespan of TEs can be predicted from a single genome, and a determination of certain factors that affect the activities of active TEs that may cause human disease. The major work of the thesis will be composed of four major goals with their corresponding chapters:

**Goal 1** Create a model that describes TEs and remnants of TEs formally (Chapter 4).

After an extensive literature survey on transposable elements, we found that there did not exist a standard model that describes/defines the topic in a clear and consistent way. For example, the use of many terms are frequently ambiguous, such as a transposable element, a subfamily of a transposon, a clade of transposons, a group of transposable element fragments, etc. Moreover, some computational approaches regarding TEs were described in a prose-like language, without any formal algorithms, which brought in different ambiguities when reproducing the method. Therefore, our first goal is to create a formal model that consists of an initial definition of TEs, TE fragments, and interruptions between TEs, etc. This model does not attempt to capture the molecular operations of TE movement, but only describes the order and distance between TE fragments in genomic sequences by grouping homologous TEs together. It is a high-level abstraction defining TEs and their positional relationships, which serves as a baseline in describing our other formal models.

**Goal 2** Understand the factors that affect the activities of active TEs, and understand how activity is affected (Chapter 3).

The factors that change transpositional activities is largely a mystery, and it might be the result of a number of factors or combinations of them. Our goal is to understand what factors affect a transposon's activity level, and determine how they affect them. We will study "harmful mutation regions" in active TEs, where mutations within these regions will decrease the activities of the host element.

**Goal 3** Predict the age, lifespan, and activity of TEs in the human genome from the remnants of these elements from a genome. In order to make this prediction, two formal models are created in Chapters 5 and 6 that describe and can be used to analyze and predict the ages of TEs.

By analyzing TE remnants in genomic sequences, the knowledge of TE activities can be inferred. Then understanding how TEs interrupted within each other will reveal information regarding the age and lifespan of a transposon; that is, when TEs activated and deactivated through evolution. Then the dynamics of TE transpositions through evolution can be predicted. This goal can be achieved by two sub-goals as follows.

- Understand interruption activities between TEs.

  The interruptions are classified into two different types based on the fashion in which they nested. The formal model in **Goal 1** is applied to these two models. The first application was the Sequential Interruption model, which captures the interruptions between pairs of TEs, and structures the abundance of these interruptions into a so-called interruption matrix. The second application was the Recursive Interruption model that describes the nested nature of the recursive interruptions with a context-free grammar. The parse trees of the grammar can illustrate the relationships of TE nesting using the structure of the trees.

- Predict an overall TE-interaction network.

  Several of the small interruption trees can be combined together to form a weighted directed graph to illustrate all TE activity, which can also help in the understanding of TE evolution as a whole. Such a graph is created for all TEs in order and is used to predict the age order and lifespans of these TEs.

**Goal 4** Understand the dynamics of TEs transpositions through evolution (Chapter 7).

Another goal of the thesis is to create a simulation that imitates the evolutionary history of the propagation of TEs in the human genome. This is done in such a way that the remnants of TEs with their relative positions in the genome are comparable to the actual TE remnants in the human genome. Then by analyzing the generated TE remnants and predicting the TE evolution using the tools generated for **Goal 3**, this work can be used as a type of verification of the models and algorithms in **Goal 3**. A simulation also allows for specific hypotheses regarding TEs to be tested, which is of interest to the community. Moreover, some aspects of **Goal 2** inform the simulation itself. The two sub-goals are listed as follows.

- Simulate TE propagation through evolution.
- Use the simulated interruptions to verify the prediction models.

Hence, the various goals of this thesis are quite interconnected:

- **Goal 1** is used as a basic frame work for **Goal 3** and **Goal 4**;
- **Goal 2** and **Goal 3** inform the process of simulation of **Goal 4**;
- **Goal 4** is used as a verification for **Goal 3**.

These goals all help in the study of transposable elements, their influence on genomes, and their ability to be used for evolutionary prediction.

# Chapter 2

# Background

Transposable elements are one type of repetitive DNA sequences in eukaryotic genomes. Typically, repetitive DNA sequences are broadly classified into two large families within eukaryotic genomes, tandem repeats and dispersed repeats, and each of these two families can be further divided into several subfamilies as shown in Figure 2.1.

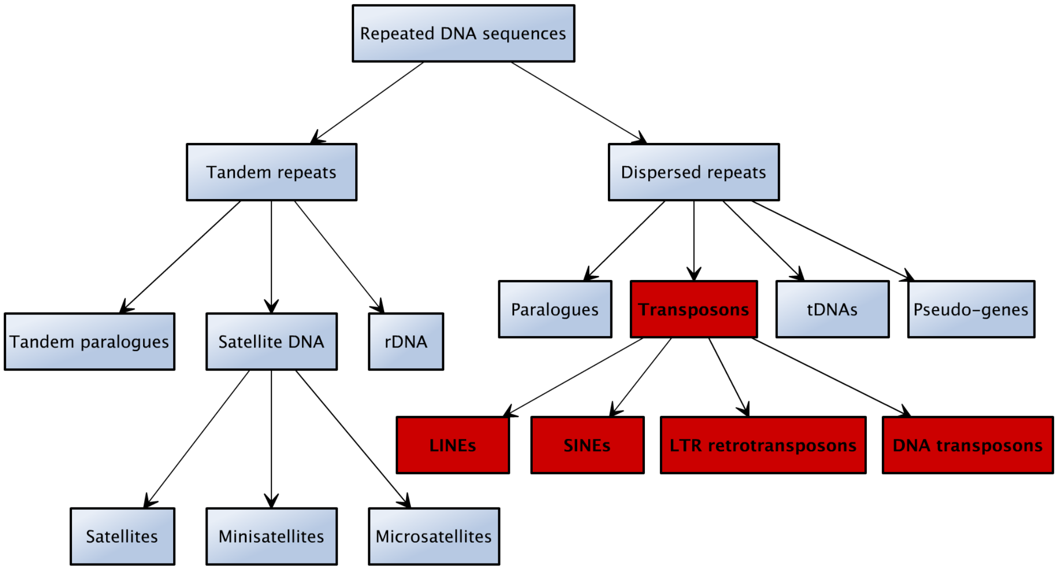


**Figure 2.1:** Repeated DNA sequences in eukaryotic genomes, adapted from [119]. Transposons are marked in red.

TEs are found in both eukaryotic and prokaryotic organisms, including plants, animals, bacteria, and archaea. As shown in Figure 2.2 (summarized from [14, 125, 148, 136]), the proportion of TEs in a genome differs broadly depending on the organism, ranging from a few percent (0.3%) in the bacteria *Escherichia coli* to almost the entire genome (>80%) in maize *Zea mays*. In humans, 66–69% of the genome is repetitive or repeat-derived [32], whereas coding sequences comprise less than 5% of the genome. The majority of repeats in human are transposable elements, making up about 45% of the genome [88].

Some TEs have an evolutionary history dating back hundreds of millions of years during which they adaptively

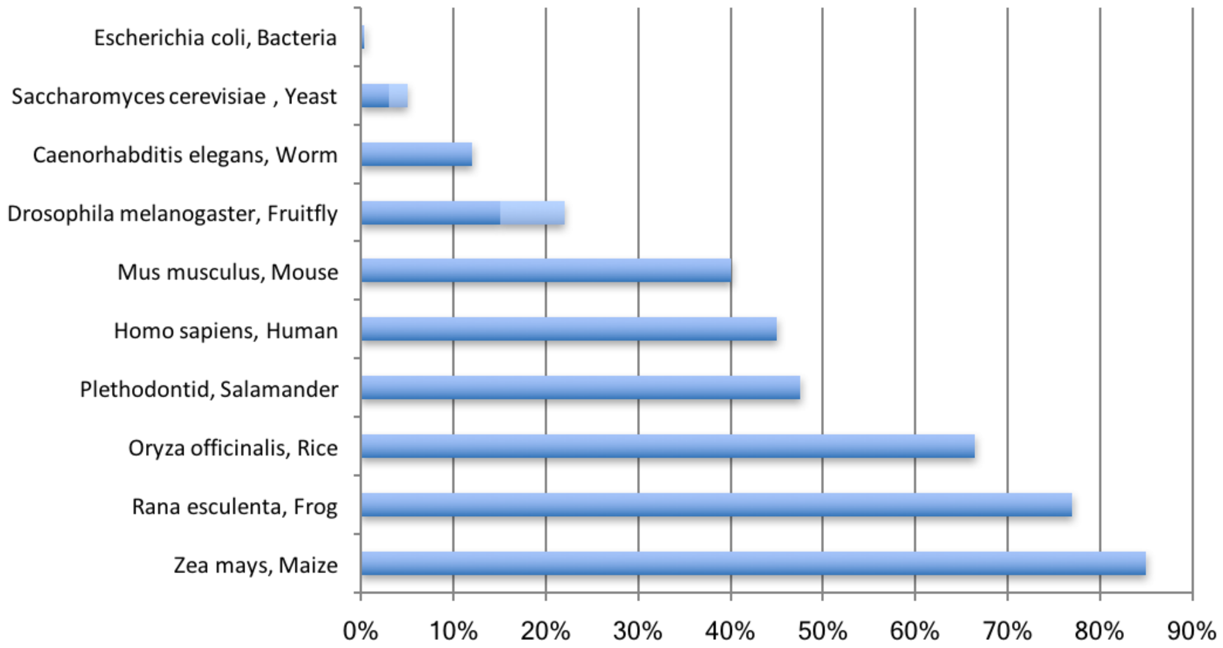


**Figure 2.2:** The proportions of TEs in several genomes. As shown in the stacked bars, the TE proportions in yeast and fruitfly are shown as value ranges (3% to 5% in yeast, and 15% to 22% in fruitfly). The reason that there exists ranges is because the transposons in these species are transient components of those genomes, which means the repeat fraction of these genomes evolves very rapidly within species [100].

diversified into forms that share very little sequence homology. Over time, inactivated copies of these elements have accumulated and now comprise a significant proportion of many genomes, serving as an important opportunity to study molecular evolution. This is because every element in the genome represents a "fossil record" that throughout evolution accumulates mutations randomly and independently, meaning that they can be used to study genomic changes both between and within species. "The mammalian genome could be compared, somewhat poetically, with a coral reef, in which the transposable elements are the coral, the reef is built of the fossils of their ancestors, and the genes are the inhabiting fish, anemones, sea stars, and so on." [130]

So far in this chapter, a brief introduction about TEs including their proportions in genomes and some key terminologies has been provided, which gives a general idea about what TEs are. In the next sections, some necessary knowledge regarding the topic of the thesis will be provided from the perspectives of biology, bioinformatics, and previous research on TEs respectively.

## 2.1 Biological background

### 2.1.1 Classification of transposable elements

*Transposition* is defined as the movement of genetic material from one genomic location, the *donor site*, to another, the *target site*, within the same genome [50]. Transposable elements are traditionally classified into two broad classes on the basis of their transposition mechanism and sequence organization [43]:

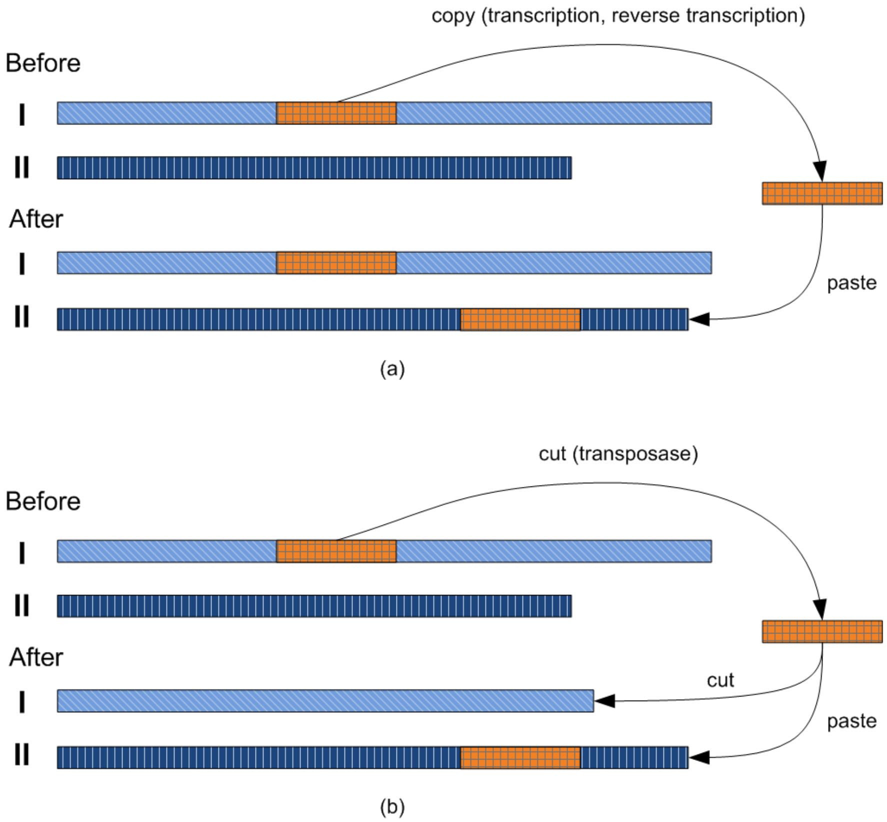


**Figure 2.3:** Conceptual diagrams representing the transposition mechanisms. (a) Retrotransposons ("copy-and-paste" mechanism) copy themselves in two stages: first from DNA to RNA by transcription, then from RNA back to DNA by reverse transcription. The DNA copy is then inserted into the genome in a new position. (b) DNA transposons ("cut-and-paste" mechanism) do not involve an RNA intermediate. The transpositions in these classes are catalyzed by various types of transposase enzymes.

- Class I elements ("copy-and-paste" mechanism as the conceptual diagrams shown in Figure 2.3 (a)) are those that transpose via reverse transcription of an RNA intermediate, referred to as *retrotransposons*. The RNA intermediate is first transcribed from a genomic copy of DNA, then reverse-transcribed into DNA that is identical to the original DNA by a reverse transcriptase[1] encoded in the TE sequence,

[1] Reverse transcriptase is an RNA-independent DNA polymerase that catalyzes the synthesis of DNA from RNA.

and each complete replication cycle produces one new copy into the host DNA [145]. Consequently, retrotransposons can increase the copy numbers of elements and thereby increase genome size; indeed, they are major players in promoting the increase of genome size.

- Class II elements ("cut-and-paste" mechanism as the conceptual diagrams shown in Figure 2.3 (b)) move primarily through a DNA-mediated mechanism of excision and insertion, and are often called *DNA transposons*.

The class I and class II elements coexist in an extensive range of eukaryotes, which suggests their ancient evolutionary origins; however, there exist many variations in the activity, copy number, and diversity of TEs in the genomes of different species [49].

TEs can also be divided into several types on the basis of the structural features of their sequences: LTR retrotransposons, LINEs, SINEs, and DNA transposons (summarized as in Figure 2.4). Note that this classification is according to structural features of TEs as in Section 2.1.2, which is equivalent to the simplified classification shown in the red subtree in Figure 2.1. Among the four types of TEs, non-LTR retrotransposons (LINEs and SINEs) have been major factors of genome evolution by providing diversity and plasticity to the genome [71].

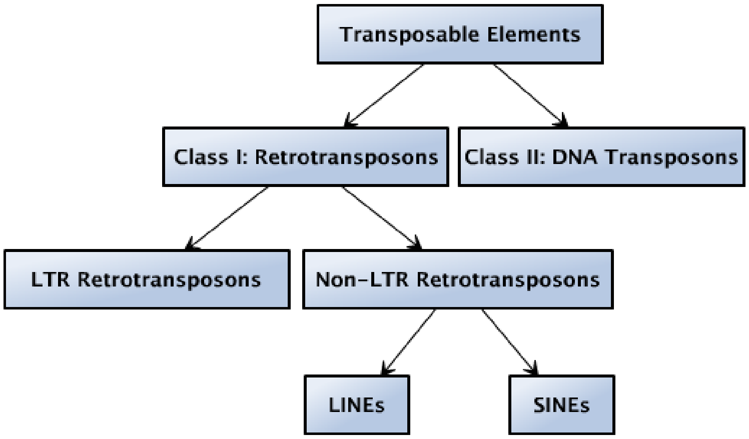


**Figure 2.4:** The classification of transposable elements can be represented as a tree.

TEs are also described as being *autonomous* or *non-autonomous* based on whether or not they encode their own genes for transposition. Those transposable elements that possess a complete set of transposition protein domains are called *autonomous*. Transposable elements that lack an intact set of mobility-associated genes are called *non-autonomous* TEs. The transposition of non-autonomous TEs requires involvement of protein(s) from either autonomous element(s) or from the genome in which they reside. For example, the autonomous *Ac* elements in maize can transpose themselves regardless of the other TEs present in the genome ; in contrast, the non-autonomous *Ds* elements cannot transpose without the aids of one or more copies of *Ac* elements in the genome [50].

Nevertheless, the term autonomous does not indicate that a TE is active or functional. A TE is defined as *active* if it can transpose either autonomously or non-autonomously. Typically, the lifespan of one transposable element starts from an activation of the transposon, followed by a burst of transpositions, while accumulating mutations, followed by the slowing of mobile activity after additional mutations. The transposon then ebbs further until it becomes inactive. The inactive elements, referred to as *fossil transposable elements*, become relics and can get interrupted by the transpositions of other active elements [47]. Active elements comprise only a tiny proportion of the TE content of the genomes of most organisms. The genomes of eukaryotes are filled with thousands of copies of the remnants of inactive TEs. For example, there are roughly 50,000 autonomous and 200,000 non-autonomous fossil DNA transposons in the human genome, and none of them are active any more [50].

The human genome consists of a large amount of TEs and their remnants. Table 2.1 lists the percentage of TE contents in each chromosome calculated from the *hg38* assembly[2] of the human genome. It should be noted that the total percentage of TEs we calculated on *hg38* is 46.69% in the thesis, which is slightly higher than the 45% estimated in the year of 2009 [88] from a previous version of the human genome.

Within each type, TEs can be even further subdivided into families then subfamilies, based on their details of the transposition mechanism, and sequence similarity. For example, $L1$, $L2$ are families under LINEs, while *Alu*, *SVA* are families under SINEs; furthermore, there are subfamilies $AluY$, $AluJ$, $AluS$ under the family of *Alu*. Table 2.2 (information from [84]) is a detailed summary of each type of TE in the human genome.

A gust of transposition of *L1* and *Alu* elements in the primate lineage occurred about 40 million years ago (MYA), followed by a slowing of transpositional activity since then [73]. Recent evidence indicates that there are 35 to 40 subfamilies of *Alu*, *SVA*, and *L1* elements staying actively mobile in the human genome [104, 71], and all of the active transposable elements only comprise less than 0.05% of the nucleotides in the human genome. It has been estimated that active human transposons generate about one insertion for every 10 to 100 live births [69, 89, 31]. The rate of *L1* retrotransposition is estimated as 1/140 live births per generation [38], and one new *Alu* insertion is generated for every 20 live human births [31]. The active TEs along with their copy numbers are listed in Table 2.2 as well.

### 2.1.2 Structural features of TE sequences

In mammals, almost all transposable elements fall into one of the four types listed in Figure 2.4, of which three transpose through RNA intermediates and one transposes directly as DNA. Transposable elements use different strategies for their evolutionary survival (summarized from [84]):

[2] *hg*38 is the December 2013 assembly of the human genome.

| | LINE | SINE | LTR | DNA | Total TE percentage |
|---|---|---|---|---|---|
| Chromosome 1 | 20.24% | 14.39% | 8.23% | 3.27% | 46.19% |
| Chromosome 2 | 22.63% | 11.90% | 9.16% | 3.83% | 47.58% |
| Chromosome 3 | 23.52% | 12.10% | 9.80% | 3.99% | 49.46% |
| Chromosome 4 | 24.57% | 10.27% | 11.51% | 3.70% | 50.11% |
| Chromosome 5 | 23.80% | 11.26% | 9.92% | 3.79% | 48.83% |
| Chromosome 6 | 23.40% | 11.62% | 9.67% | 3.83% | 48.58% |
| Chromosome 7 | 21.72% | 13.81% | 8.87% | 3.51% | 47.98% |
| Chromosome 8 | 23.15% | 12.03% | 10.04% | 3.61% | 48.90% |
| Chromosome 9 | 19.75% | 12.28% | 7.78% | 3.09% | 42.96% |
| Chromosome 10 | 21.00% | 13.94% | 8.47% | 3.64% | 47.11% |
| Chromosome 11 | 22.99% | 13.44% | 8.96% | 3.38% | 48.81% |
| Chromosome 12 | 21.75% | 14.94% | 9.43% | 3.72% | 49.90% |
| Chromosome 13 | 19.41% | 8.58% | 8.90% | 3.07% | 40.01% |
| Chromosome 14 | 18.61% | 11.42% | 8.18% | 3.01% | 41.28% |
| Chromosome 15 | 17.80% | 12.56% | 6.24% | 3.00% | 39.65% |
| Chromosome 16 | 14.70% | 18.05% | 7.26% | 3.08% | 43.12% |
| Chromosome 17 | 15.17% | 21.21% | 6.27% | 3.12% | 45.81% |
| Chromosome 18 | 20.88% | 10.39% | 8.84% | 3.48% | 43.64% |
| Chromosome 19 | 13.55% | 27.14% | 8.53% | 2.11% | 51.35% |
| Chromosome 20 | 19.14% | 15.68% | 8.55% | 4.06% | 47.47% |
| Chromosome 21 | 16.17% | 9.05% | 9.82% | 2.64% | 37.73% |
| Chromosome 22 | 12.13% | 15.81% | 5.00% | 2.05% | 35.01% |
| Chromosome X | 33.63% | 10.36% | 11.24% | 3.22% | 58.50% |
| Chromosome Y | 11.61% | 4.63% | 7.81% | 0.81% | 24.87% |
| Total percentage in *hg38* | 21.42% | 12.82% | 9.00% | 3.40% | 46.69% |

**Table 2.1:** A summary of the percentage of TE content in the human genome (calculated on *hg38*). It also lists the percentage of TEs of four different types and summarized by chromosomes.

| **TE Class** | Class I Elements | | | | Class II Elements | |
|---|---|---|---|---|---|---|
| **Class Name** | Retrotransposons | | | | DNA Transposons | |
| **Type** | LTR Retrotransposons | | Non-LTR Retrotransposons | | | |
| | | | LINE | SINE | | |
| **Mode of Transposition** | Autonomous | Non-autonomous | Autonomous | Non-autonomous | Autonomous | Non-autonomous |
| **Length** | 6-11 Kbp | 1.5-3 Kbp | 6-8 Kbp | 100-300 bp | 2-3 Kbp | 80-3000 bp |
| **Copy Number** | 450,000 | | 850,000 | 1,500,000 | 300,000 | |
| **Major Families** | ERV, ERVK, ERVL,<br>Tf series, Ty serie,<br>MaLR, copia, Toms, 17.6 | | *L1* (16.9% of the human genome), L2, L3 | *Alu* (10.6% of the human genome)<br>MIR, Ther2/MIR3 | hAT, Tc1/mariner, piggyBac,<br>Sleeping Beauty, Tn series, Mu, Mos1,<br>Tol2, hobo, transits, MITEs | |
| **Copy Number and age** | | | *L1* (>500,000 copies, ∼150 Myr)<br>less than 100 copies are functional | *Alu* (∼1.1 million copies, ∼65 Myr)<br>*SVA* (∼3,000 copies) | | |

**Table 2.2:** Overview of different classes of transposable elements (reflecting the classification in Figure 2.4) in the human genome.

- LINEs and SINEs depend on vertical transmission, meaning that the new "offspring" TEs are produced from their "parent" TEs within the host genome.
- DNA transposons depend on horizontal transfer, meaning that the transfer between members of the same species are not in a parent-child relationship.
- LTR retrotransposons use both strategies.

It should be noted that it is beneficial to discuss the biological details in order to gain an understanding of the variation, since different parts of this thesis deal with different specific TEs and TE properties. Also, different existing TE-related tools take into account specific features, which are worth discussing. The biological insights of each type of TE is summarized as follows (information from [41], [84], [121], and [71]):

### DNA transposons

DNA transposons are prevalent in bacteria, but are also found in the genomes of many metazoa, including insects, worms, and humans [70]. These elements are generally excised from one genomic site and integrated into another by a "cut-and-paste" mechanism (Figure 2.3).

As shown in Figure 2.5, DNA transposons are usually composed of terminal inverted repeat sequences (TIRs; in Figure 2.5, big blue arrows are in opposite directions) at their front and rear ends. Between TIRs is one ORF[3] sequence (red boxes in Figure 2.5, 2.6, and 2.7) that encodes a transposase protein that recognizes

[3] An *open reading frame (ORF)* is a DNA sequence that does not contain a stop codon (a nucleotide triplet within messenger

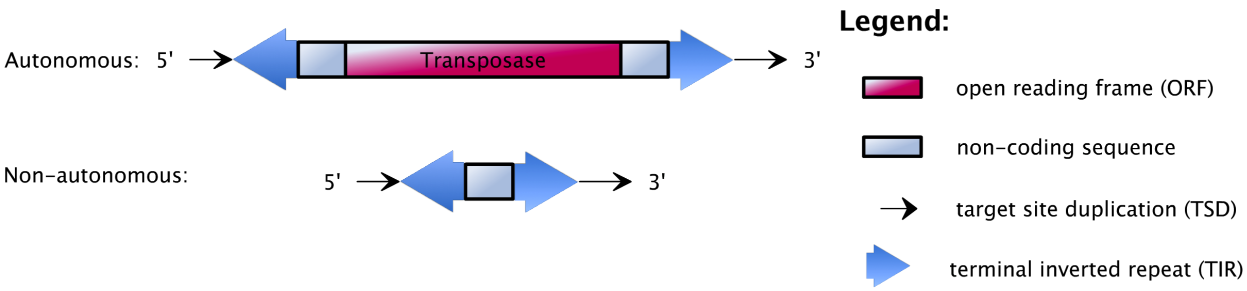


**Figure 2.5:** Structural features of DNA transposons, which may contain both autonomous and non-autonomous elements.

the TIRs and cuts the transposon out of its genomic site. The two ends of the transposon are then held together by the transposase while it finds another site in the DNA to cut and insert into. Thus, the process uses a so-called "cut-and-paste" mechanism. All DNA transposons, both autonomous and non-autonomous, are surrounded by short duplications of the genomic sequence at their insertion sites, called *target site duplications* (TSDs)[4]. This occurs because the double-stranded target site is cut in a staggered manner, the single-stranded flanks are then repaired, and two repeats in the same orientation (called *direct repeats*, opposite to inverted repeats), are created on both sides of the integrated TE [50]. The TSDs can either be of fixed or variable lengths, depending on the type of elements. In fact, the integration of almost all TEs results in the target-site duplications as shown in black thin arrows flanking the element in Figure 2.5, 2.6, and 2.7. In fact, non-autonomous DNA transposons are usually derived from an autonomous transposon by an internal deletion.

There are nearly no known active DNA transposons in mammals (except bats[5]) [71]. It has been previously believed that DNA transposons have not been active in the mammalian lineage for at least 40 million years (Mys). There are only 15 superfamilies to which currently known eukaryotic DNA transposons belong (despite their enormous diversity and abundance): *hapaev, En/Spm (CACTA), hAT, Harbinger (Pif), ISL2EU (IS4EU), Kolobok, Mariner, Merlin, Mirage, MuDR(MULE), Novosib, P, PiggyBac, Rehavkus,* and *Transib* [68].

The human genome contains the remnants of at least five major families of DNA transposons, which can be subdivided into many transposons with independent origins. Table 2.3, derived from [112], is a summary of currently recognizable DNA transposons in the human genome with copy number greater than 100.

DNA transposons generally transpose to genomic sites less than 100 Kbp from their original site, called

RNA that signals a termination of translation) in a given reading frame.

[4] When a transposon inserts itself into host DNA, a short (7-20bp) segment of host DNA is replicated at the site of insertion, which is called *'target site duplication' or TSD*.

[5] Eight different families of DNA transposons, including hAT family members and piggyBac-like elements, were found active in the genome of the little brown bat, *Myotis lucifigus*.

| Family | DNA Transposons | Number of Transposons | Copy Number |
|---|---|---|---|
| hAT | *Autonomous:* | | |
| | Blackjack, Charlie, Cheshire, Zaphod | 19 | 46,133 |
| | *Nonautonomous:* | | |
| | Arthur1, FordPrefect, MER102, MER106, MER107, MER112, MER113, MER115, MER117, MER119, MER1, MER20, MER3, MER30, MER33, MER45, MER58, MER5, MER63, MER69, MER81, MER91, MER94, MER96, MER99, ORSL | 52 | 218,059 |
| | *Total* | 71 | 264,192 |
| MuDr | *Nonautonomous:* | | |
| | Ricksha | 3 | 985 |
| | *Total* | 3 | 985 |
| piggyBac | *Autonomous:* | | |
| | Looper | 1 | 521 |
| | *Nonautonomous:* | | |
| | MER75, MER85 | 3 | 1,569 |
| | *Total* | 4 | 2,090 |
| Tc1/mariner | *Autonomous:* | | |
| | HSMAR, Tigger, Kanga | 22 | 53,320 |
| | *Nonautonomous:* | | |
| | MADE, MARNA, MER104, MER2, MER44, MER46, MER53, MER6, MER8, MER82, MER97 | 23 | 54,718 |
| | *Total* | 45 | 108,038 |
| Unknown | MER103, MER105 | 2 | 7,567 |
| | *Total* | 2 | 7,567 |
| | **Grand Total** | **125** | **382,872** |

**Table 2.3:** A summary of currently recognizable DNA transposons in the human genome with copy number greater than 100.

"local hopping" [70] (e.g., the *Drosophila* P element), and some are able to make distant "hops" (e.g., the fish Tc1/mariner element) as well. Moreover, DNA transposons are inclined to have short lifespans within a species compared to LINEs, and why DNA transposons have lost their ability to move for millions of years of mammalian evolution requires further studies.

### Retrotransposons

Retrotransposons are very different from DNA transposons. They replicate and mobilize through an RNA intermediate via a "copy-and-paste" mechanism involving the enzyme reverse transcriptase and an endonuclease.

Retrotransposons typically can be divided into long terminal repeat (LTR) retrotransposons (Figure 2.6) and non-LTR retrotransposons (Figure 2.7), and non-LTR retrotransposons are subdivided into long interspersed nuclear elements (LINEs) and short interspersed nuclear elements (SINEs). Most of retrotransposons are no longer able to retrotranspose. Retrotransposons have taken over large portions of the genomes of most plants and animals. In plants, most are LTR-retrotransposons , while in mammals non-LTR retrotransposons predominate.

Some retrotransposons are site-specific (only insert at specific sites in the genome). For instance, the non-LTR retrotransposons of the *R1* and *R2* families insert themselves only at specific sequences within the ribosomal RNA genes of insects [71]. In contrast, there are some non-LTR retrotransposons of the *L1* family that insert at many different sites that are AT-rich (e.g., 5'-TTTT/AA-3', where "/" signifies the cut site) [71].

**LTR Retrotransposons.** The LTR retrotransposons have many characteristics akin to retroviruses. They are called LTR retrotransposons because they have long terminal repeat (LTR) sequences of 300 to 1000 nucleotides at their two ends in direct orientation. These direct repeats have the same sequence in the same order, e.g., ABCD. In contrast, inverted repeats in DNA transposons are ABCD at one end and DCBA at the other (example from [70]). The LTRs contain promoters that stimulate transcription of the RNA of the element. Figure 2.6 shows the structural features of both the autonomous and non-autonomous LTR retrotransposons. Autonomous LTR retrotransposons have the products required for transposition encoded in open reading frames (ORFs), while non-autonomous LTR retrotransposons lack most or all coding sequence for transposition, and their internal region can be variable in length and unrelated to the autonomous elements.

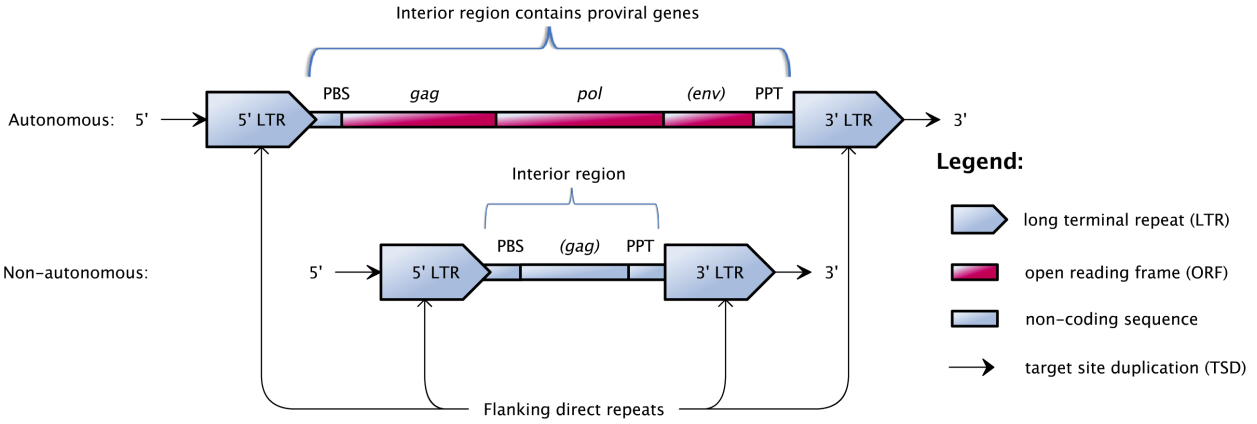


**Figure 2.6:** Structural features of the autonomous and non-autonomous LTR retrotransposons (the genes in parentheses are optional).

According to [68], there are 6 superfamilies in LTR retrotransposons: *Copia, Gypsy, BEL, ERV1, ERV2*, and *ERV3*. Among a variety of LTR retrotransposons, only the vertebrate-specifc endogenous retroviruses (*ERVs*) appear to have been active in the mammalian genome. Most (85%) of the LTR retrotransposon-derived TE remnants consist only of an isolated LTR, where the internal sequence has been lost by homologous recombination [102].

**Non-LTR Retrotransposons.** Non-LTR retrotransposons are quite different from LTR-retrotransposons in their mode of regulation, replication, and structure. They are divided into autonomous elements (LINEs) and non-autonomous elements (SINEs). As shown in Figure 2.7, non-LTR retrotransposons have an internal promoter at their 5' end (*pol II* for LINEs and *pol III* for SINEs) that is important for starting expression or transcription of the element RNA. Non-LTR Retrotransposons end by a simple sequence repeat at their 3' end, usually a *poly(A)* tail (a region containing many A nucleotides in a row). All LINEs described so far usually encode two proteins necessary for their retrotransposition. The 3' tail of some SINEs and the 3' tail of LINEs present in the same genome are related to each other (share homology) [41], which indicates that SINEs must be aided by the transposition machinery of partner LINEs in the process of transposition [35].

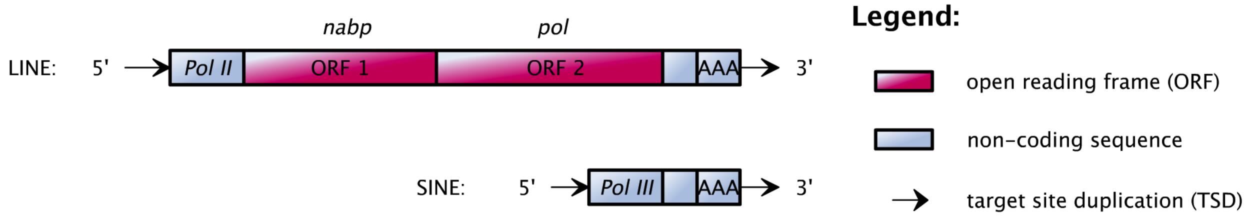


**Figure 2.7:** Structural features of the non-LTR retrotransposons, that contain autonomous LINEs and non-autonomous SINEs.

In humans, LINEs are about 6 $Kbp$ long, harbouring an internal *polymerase II* promoter and encoding two open reading frames (ORFs[3]). ORF 1 encodes a nucleic acid binding protein (*nabp*) with chaperone[6] and esterase[7] activities, and ORF 2 encodes a *pol* protein with reverse transcriptase[1] and endonuclease[8] activities. It is believed that the LINE machinery is responsible for most reverse transcription in the genome, that also includes the transposition of the non-autonomous SINEs. Three distantly related LINE families are found in the human genome: *L1*, *L2*, and *L3*, among which, only *L1* is still active.

The non-autonomous *Alu*s are thought to use the transposition machinery of LINEs. They are about $100 - 300$ $bp$ with no terminal repeats, harbour an internal *polymerase III* promoter and encode no proteins. The human genome contains a few families of SINEs: the active *Alu*, *SVA*, and the inactive *MIR* and

[6] In molecular biology, *molecular chaperones* are proteins that assist the non-covalent folding or unfolding and the assembly or disassembly of other macromolecular structures, but do not occur in these structures when the structures are performing their normal biological functions, having completed the processes of folding and/or assembly.

[7] An *esterase* is a hydrolase enzyme that splits esters into an acid and an alcohol in a chemical reaction with water called hydrolysis.

[8] *Endonucleases* are enzymes that cleave the phosphodiester bond within a polynucleotide chain.

*Ther2/MIR3*.

One of the key facts about human retrotransposons is that humans have active TEs, called *L1*s (the only active family in LINEs in humans), and these active TEs make an endonuclease and a reverse transcriptase that drives the retrotransposition of themselves and of other TEs, called *Alu* and $SVA$ (the active families in SINEs in humans).

## 2.2 Bioinformatics background

### 2.2.1 Sequence alignment

Many biological structures can be naturally represented by strings/sequences, such as DNA, RNA, and proteins. Sequence alignment is a method for biological sequence comparison, which can reveal similarity between different sequences. There exist a number of sequence alignment methods/tools, and some of them are based on dynamic programming alignment algorithms, such as the Needleman-Wunsch algorithm [109] or the Smith-Waterman algorithm [133], which compute the optimal alignments between two sequences.

A multiple sequence alignment (MSA), a natural extension of two-sequence comparisons, is a sequence alignment of three or more sequences. MSAs are a powerful way to study biological sequences. In a MSA, similar characters among a set of sequences are aligned together in columns. Often, the goal with aligning sequences is to reveal *homology*, which indicates similar position, structure, function, or characteristics due to evolutionary relatedness [57]. Sequences can be aligned to visualize the effect of evolution across the whole family. Ideally, a column of aligned characters all diverge from a common ancestor. The resulting MSA can infer sequence homology and guide phylogenetic analysis to assess the sequences' shared evolutionary origins. From this, a *consensus* sequence can be calculated from the result of a MSA. The consensus sequence of a multiple alignment is, informally, a "best" single sequence to represent the alignment. For example, a consensus for three DNA sequences

$$\begin{array}{c} \mathrm{A\ C\ A\ G\ T\ A\ G} \\ \mathrm{A\ C} - - \mathrm{T\ C\ G} \\ \mathrm{A\ G} - - \mathrm{G\ C\ G} \end{array}$$

is $ACAGTCG$. Notice that it is possible to have more than one consensus.

It is computationally expensive to calculate the optimal alignment between multiple sequences, therefore, most MSA tools use heuristic methods rather than global optimization.

### 2.2.2 Repbase Update — the database of repetitive sequences

Most eukaryotic repetitive sequences have been reconstructed into a database called Repbase Update (RU) [63], a database of the consensus sequences of repetitive elements (not only TEs, but also other repeats), that are present in diverse eukaryotic organisms. RU is the major reference database of repetitive sequences used in DNA annotation and analysis. Each sequence in the database is accompanied by a short description and references to the original contributors. It has been developed by Dr. Jerzy Jurka since 1990. It continues to grow through its community-driven annotation and submission tools and now is widely used in genome sequencing projects worldwide as a reference collection for masking and annotation of repetitive DNA. Consequently, the repeat classification based on RU is used in many other databases (such as UCSC genome database [120], Ensembl annotation [2]) and in secondary databases of repetitive elements. Some TE discovery and annotation tools also use RU as their reference library, such as the ones discussed in Section 2.2.3.

### 2.2.3 TE discovery tools

According to [13], there are usually two major goals in identifying TEs in genomic sequences:

- mask them as a preprocessing step in some bioinformatic tasks, such as gene finding;
- study them directly to make inferences about the biology or evolution of TEs.

These aims are incorporated into the most common systems used to detect individual instances of TEs in genome sequences.

The detection of TEs can be conducted in different ways, depending on the level of knowledge about the repeats that are taken into account when detecting them in a genome sequence. As suggested in [13], [121], [88], and [91], the approaches can be classified into four categories as follows.

- Library-based approaches search the repetitive sequences by comparing input data to a set of reference sequences (known TEs) contained in a library.
- Signature-based approaches search TEs using knowledge from their known structures.
- Comparative-genomics approaches use the fact that transposition creates large insertions that can be detected in multiple sequence alignments and rely on neither library nor structural features.
- *De novo* approaches look for similar subsequences found at multiple positions within a sequence.

In the next subsections, we will elaborate on each of these approaches in more detail.

### Library-based techniques

Library-based methods identify repetitive sequences by comparing input datasets against a set of reference repeat sequences (known TEs) [121]. The library can either be user-defined, or it can be a general library, such as the commonly used Repbase Update [63]. The advantages of library-based techniques are that this method is usually efficient and effective at finding repeats in the library, whereas the disadvantages are that it heavily depends on how much we already know, and fails to detect the repeats that do not exist in the library.

The most widely used library-based program is RepeatMasker [131], which is the major library-based tool used in repeat identification. It has been identified as one of the most accurate tools in detecting TEs and it has become a standard tool for finding repeats in genomes [91]. Using precompiled repeat libraries, RepeatMasker finds copies of known repeat families represented in Repbase Update. As the name implies, it was designed to discover repeats and mask them. The program performs a similarity search based on local alignments, then outputs masked genomic DNA and a tabular summary of TE content.

Table 2.4 is an example of the tabular summary output by RepeatMasker, which shows eight repeat fragments identified in a query sequence named *HSU08988*, and each row represents one fragment. The columns are information about each fragment. For example, the first fragment in the list is from position 6563 to position 6781 in the query sequence. It is a fragment from position 103 to position 336 of the complementary sequence of element *MER7A* (belongs to a DNA transposon *MER2* family), with 15.6% percent divergence, 6.2% deletion, and 0% insertion compared to the *MER7A* consensus in the Repbase Update.

| | | | | | position in query | | | | | | position in repeat | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| score | % | % | % | query | | | | C | matching | repeat | left | end | begin |
| | div. | del. | ins. | sequence | begin | end | left | + | repeat | class/family | begin | end | left |
| 1306 | 15.6 | 6.2 | 0 | HSU08988 | 6563 | 6781 | -22462 | C | MER7A | DNA/MER2 | 0 | 336 | 103 |
| 12204 | 10 | 2.4 | 1.8 | HSU08988 | 6782 | 7714 | -21529 | C | TIGGER1 | DNA/MER2 | 0 | 2418 | 1493 |
| 279 | 3 | 0 | 0 | HSU08988 | 7719 | 7751 | -21492 | + | (TTTTA)n | Simple_repeat | 1 | 33 | 0 |
| 1765 | 13.4 | 6.5 | 1.8 | HSU08988 | 7752 | 8022 | -21221 | C | AluSx | SINE/Alu | -23 | 289 | 1 |
| 12204 | 10 | 2.4 | 1.8 | HSU08988 | 8023 | 8694 | -20549 | C | TIGGER1 | DNA/MER2 | -925 | 1493 | 827 |
| 1984 | 11.1 | 0.3 | 0.7 | HSU08988 | 8695 | 9000 | -20243 | C | AluSg | SINE/Alu | -5 | 305 | 1 |
| 12204 | 10 | 2.4 | 1.8 | HSU08988 | 9001 | 9695 | -19548 | C | TIGGER1 | DNA/MER2 | -1591 | 827 | 2 |
| 711 | 21.2 | 1.4 | 0 | HSU08988 | 9696 | 9816 | -19427 | C | MER7A | DNA/MER2 | -224 | 122 | 2 |

**Table 2.4:** An example of a tabular output of RepeatMasker. In the tabular summary, there are eight repeat fragments listed, which were identified in the query sequence named *HSU08988*. Each row represents one fragment, and the columns are the detailed information about the fragments.

RepeatMasker uses multi-processor systems and a simple database search approach (e.g., BLAST [78]), making it one of the fastest and most effective repeat finders [90]. Because of the importance of this tool in the field of TE identification, in the thesis, TE fragments are defined with the aid of RepeatMasker in Chapter 4.

In addition to the tabular output as in Table 2.4, there are several library-based repeat detection tools that use visualization to display the repeat fragments, such as CENSOR [64]. It applies the same kind of approach used in RepeatMasker, then graphically maps detected repeats with colour-coding of different types of repeats.

### Signature-based techniques

Unlike library-based tools, all signature-based tools employ prior knowledge about the common structural features shared by different TEs in the same class as introduced in Section 2.1.2, and it is less biased by similarity to the set of known elements. Given a particular repeat group, signature-based repeat detection tools search query sequences for motifs[9] and spatial arrangements characteristic of that group [121]. This approach can be used to find new elements of known groups, but not new groups of elements. The limitation of such approaches is that it depends entirely on how much is known about the structure of elements belonging to particular groups, and also on the existence of characteristic structures.

LTR retrotransposons are bordered by long terminal repeats (LTRs), with a more detailed structure shown in Figure 2.7. Based on the structure, there have been several tools developed to detect solely LTR retrotransposons, by searching for the common structural signals existing in LTR retrotransposons as listed below. Some of the features correspond to parameters that can be adjusted by the users of these tools, such as:

- a range of lengths of the LTR sequences;
- the distance between the two LTRs of an element;
- the presence of TSDs[4] at each end;
- the presence of critical regions for replication, such as the primer binding site (PBSs)[10] and the polypurine tract (PPTs);
- the percent identity between the two LTRs;
- the existence of some conserved motifs corresponding to the genes they encode.

[9] A sequence *motif* means a sequence pattern of nucleotides in a DNA sequence or amino acids in a protein.

[10] A *primer binding site* is a region of a nucleotide sequence where an RNA or DNA single-stranded primer binds to start replication. The primer binding site is on one of the two complementary strands of a double-stranded nucleotide polymer, in the strand which is to be copied, or is within a single-stranded nucleotide polymer sequence.

For example, the method LTR_STRUC [98] searches a query sequence for pairs of similar LTRs separated by a distance expected for this group of retrotransposons. The pairwise alignment of LTRs is then used to calculate the boundaries of the LTRs on the original segment, which should span a full-length element from the start of the 5' LTR to the end of the 3' LTR. LTR_par [66] is a similar structure-based method to detect LTR retrotransposons, which improves on some of the weaknesses in LTR_STRUC. Both LTR-detection tools have significant advantages over the library-based methods in the case of LTR retrotransposon families; thus, they are discovery tools that complement the library-based methods.

There are also some programs designed to detect non-LTR retrotransposons (SINEs, LINEs). For example, as illustrated in Figure 2.7, LINES and SINEs are flanked by target site duplications (TSDs), so the tools such as TSDfinder [137] are designed to precisely identify transposon boundaries and refine the coordinates of *L1* insertions that are detected by RepeatMasker [131], using the structural feature shared by the *L1* family. The SINEDR [141] identification tool can detect known SINEs that are flanked by TSDs. The RTAnalyzer [92] is designed to detect sequences of retrotransposed origin. It is used to detect the common signatures of *L1* transposition to find out if the sequences have been transposed by an *L1*, by calculating a transposition score on the basis of the common signatures, such as the presence of a *poly(A)* tail, TSDs, and an endonuclease cleavage site in the 5' end of the sequence.

#### Comparative genomic method

An innovative method in [19] uses the fact that transposition creates large insertions to detect new TE families and instances, depending on neither library nor structural features. These large insertions can be detected in multiple sequence alignments. This method has two major advantages over other repeat-finding methods: first, in contrast to determining the location of repeats using sequence similarity, this method utilizes the genomic artifacts of the transposition mechanism itself in the context of multiple alignments; second, it can derive the lifespan of each repeat instance with the aid of phylogenetic trees.

This method looks for insertion regions (IRs) in multiple alignments of orthologous genome sequences that are interrupted by a large insertion in one or more species. The large insertions detected are then filtered and concatenated as IRs; then IRs are locally aligned with all other IRs to identify repeat IRs. This method is useful in detecting new TE families and instances, especially in placing TEs to branches of a phylogenetic tree.

In particular, this approach has the following three advantages: first, it depends less on the sequence similarity than do the commonly used library-based methods, thus it provides a complementary approach to TE identification; second, this method allows placing each insertion event to branches of a phylogeny, which is valuable information for understanding the transposition mechanism and the evolution of the host genomes; finally, the method is flexible in choosing between stringent criteria and low-quality cutoffs on repeat content and structure, which allows mining deep into the past of TEs of the genomes. However, there exists some potential drawbacks caused by the huge difficulty in computing whole genome alignments.

### *De novo* **repeat discovery**

*De novo* repeat discovery algorithms identify repetitive elements without using reference sequences or known structures in the repeat identification process. As the number of sequenced genomes increases, there is little or no information about their repeat content, thus *de novo* repeat discovery approaches are proposed to discover new repeats using different kinds of methodology, with different final goals. These methods usually use assembled sequence data, so both sequencing and assembly strategies are critical to the results [13]. The major challenges are to characterize TEs from other TE classes and to distinguish new TE families.

All *de novo* discovery of repeat families starts with identifying relatively short sequences that are found multiple times in a sequence or sequence set using classical computational strategies. The following list briefly describes some general approaches that have been utilized in identification and clustering of repeats.

- Self-comparison approaches

  The self-comparison approach compares DNA sequence with itself to identify groups of similar sequences. This approach is used by the Repeat Pattern Toolkit (RPT) [1], the first attempt to detect repeats using the self-comparison approach. The RPT is based on a sequence similarity scoring system, and uses BLAST [3] to perform the self-comparison. The grouping of repeats is then formed by clustering. RECON [6], one of the most commonly used programs, also uses the BLAST program to perform the self-comparison, followed by a clustering method to form repeat families. PILER [37], repeat identification tool for assembled genomic regions, uses another procedure to perform the self-alignments called Pairwise Alignment of Long Sequences.

- $k$-mer approaches

  A substring of length $k$ occurring more than once with perfect matching in a sequence is identified as a repeat in the $k$-mer or “word counting” approaches. One of the first programs to apply the $k$-mer approach is called REPuter [82]. Based on a suffix-tree data structure, it can determine all the exact repetitive substrings in a complete genome. RepeatScout [117] first builds a library of high frequency $k$-mers with fixed length, then uses these as seeds for a greedy search.

- Spaced seed approaches

  Spaced seed approaches are an extension of the $k$-mer approach, which allows some differences in the sequence of the seed, such as the length and/or the percentage identity. PatternHunter [93], as the first spaced seed tool, allows mismatches in fixed positions and at the same time requires identical matches in others. RAP [18] uses a complex indexing strategy allowing space efficient counting of words of a specific size.

- Other approaches

  There exists some other approaches which detect and classify TEs, but do not fall into any of the above categories. One example is the RepeatGluer [115] program, which represents the mosaic structure of sub-repeats using A-Bruijn graph representation. The A-Bruijn graph created by RepeatGluer derives the mosaic repeat structure from a set of pairwise similarities; furthermore, by traversing the graph, it can also illustrate evolutionary history of repeats.

## 2.3 Previous studies on TE phylogeny and evolution

### 2.3.1 TE phylogeny based on sequence divergence

Some TEs have an evolutionary history dating back hundreds of millions of years in which they adaptively diversified into different forms. In the meantime, inactive copies of these elements have accumulated and now comprise a significant proportion of many genomes. It is possible to perform a phylogenetic analyses via a multiple sequence alignment of transposons, under the assumption that transposon evolution behaves like a molecular clock after activation (a discussion of this assumption will follow below). Indeed, the molecular clock hypothesis is that for every given gene (or protein), the rate of molecular evolution is approximately constant. The age of transposons can be estimated by comparing the percentage of sequence divergence with some known age of divergence of species such as humans and Old World Monkeys (adapted from [114]) as follows:

- most transposable elements in the human genome are ancient ($\sim$ 100 MYA), which get removed from the genome very slowly;
- non-LTR retrotransposons, LINEs and SINEs, have very long lineages, some dating back 150 MYA;
- DNA transposon are extinct, as there is no evidence for their activity in the human genome in the past 50 million years.

Specifically, each copy of a TE in a genome is derived from an active TE sequence that accumulates mutations randomly and independently from other copies of this TE. Consensus sequences of the original active copies (these consensus records are in Repbase Update [63]) are derived from multiple sequence alignments of the present-day copies. The approximate age of these elements can be calculated from the average sequence divergence of the present-day copies from the consensus sequence.

The divergence-based method has been applied to both *Alu* [67, 7] and *L1* [132] elements to assign approximate ages to TEs. For example, to estimate the age of *Alu* subfamilies in [67], pairwise alignments were performed using the Smith-Waterman algorithm [133], and the divergence ($d$) between the two sequences was calculated by

$$d = \frac{\text{Number of mismatches}}{\text{Number of Matches + mismatches}}. \tag{2.3.1}$$

Note that a deletion or insertion of any length was deemed as a single mismatch. Then using Kimura's distance measure[11] [74], the average age of all major *Alu* subfamilies were calculated based on the assumption that the rate of change of these sequences is 0.16% per site per million years [67].

However, these divergence-based calculations are limited by the assumption of the constant mutation rate (molecular clock) both over time and between the different classes of transposable elements [118, 16]. Furthermore, the difference in percent divergence of a TE family is dependent on not only the length but also the age of the element. In addition, these methods do not help predict the periods of activity and inactivity of the TEs. Thus, a more thorough method taking into account of more factors other than a mutation rate is needed to predict TE evolution more accurately, and one innovative method called interruption analysis can produce an estimation of relative ages of TEs using the frequency of interruptions between TEs that will be elaborated on in the next subsection.

### 2.3.2 TE phylogeny based on interruptional analysis

Each transposable element has a distinct period of transposition activity when it is active, in which it spreads through the genome, followed by inactivation and accumulation of mutations. Though transposable elements make up about half of the human genome, most of them are inactive relics. DNA transposons have become completely inactive and LTR retrotransposons may have done so as well [84]. Because of the ubiquity of inactive TEs in many genomes, throughout evolution newer TEs end up nesting recursively, often multiple levels deep, inside existing inactive TEs. The result of the transposition activities can be described as (summarized from [47] and [81]):

1. older TEs are heavily interrupted by younger TEs, but have not inserted into younger elements;
2. younger TEs, with a relatively recent period of activity, have inserted into older elements that were present in the genome, but are not interrupted by older elements;
3. TEs of intermediate age have both inserted into older elements and have themselves been fragmented by younger elements;

[11] In 1980, Kimura introduced a model to estimate the level of nucleotide substitutions $K$: $K = -0.5 \cdot ln\{(1-2P-Q)\sqrt{1-2Q}\}$, where $P$ and $Q$ are the proportions of transitional and transversional differences, respectively, between two homologous sequences.

4. younger TEs interrupt not only older TEs, but also the fragmented TEs that had been previously interrupted. That is, interruptions are nested recursively.

A method in [47] called interruptional analysis estimates relative TE ages based on the frequencies with which every TE has inserted itself into every other TE in a genome. The resultant ordering that was obtained from a positional distribution agreed reasonably with published chronologies. This is in contrast to the more common divergence-based methods (Section 2.3.1) to estimate TE ages, which has been unreliable for older more diverged elements. The approach only relies on data from a single genome.

The interruptional analysis is strategized in two major steps (summarized from [47]):

**Step one:** generate an interruption matrix based on the identified transposon clusters.

> Many TEs have split other TEs into two noncontiguous TE fragments by inserting into the sequence of those TEs already present in the genome. The occurrence of TEs that are inserted into other TEs are named "transposon clusters" in [47]. These transposon clusters in the human genome were identified by defragmentation of TEs, and the number of times every TE inserted itself into every other TE were counted and grouped into an $n \times n$ matrix, called an interruption matrix, where $n$ is the number of TEs (a formal definition and examples of interruption matrix will be given in Chapter 5).

**Step two:** generate a TE chronological order using a repositioning method.

> A computational method called interruptional analysis then performs a repositioning of all elements on the axes of the interruption matrix, in order to search for an ordering of all elements that minimizes a penalty score defined as the summation of nonzero entries in the upper triangle matrix. Theoretically, the best ordering of TEs corresponds to the order that achieves the minimum penalty score of the matrix.

Figure 2.8 (adapted from [47]) shows the resultant relative age order of 360 TEs calculated by the interruptional analysis method.

Essentially, this interruptional analysis method uses exhaustive search with the computational complexity of $O(n!)$ over all orders. Though the authors in [47] tried different strategies to decrease the complexity, it is still not practically feasible when the number of TEs in consideration is large. Therefore, the authors only try seven to ten rounds of repositions to reach a local optimum, then do the same 100,000 times to compute a distribution of positions of each TE over all orderings. They are only able to do so on 360 of the over 1000 TEs in the human genome. Besides complexity, there are some other limitations listed as follows:

- The interruption matrix can only record the interruptions between every two TEs, but fails to take into account the recursively nested interruptions which can be informative (more details on this limitation in Chapter 6).

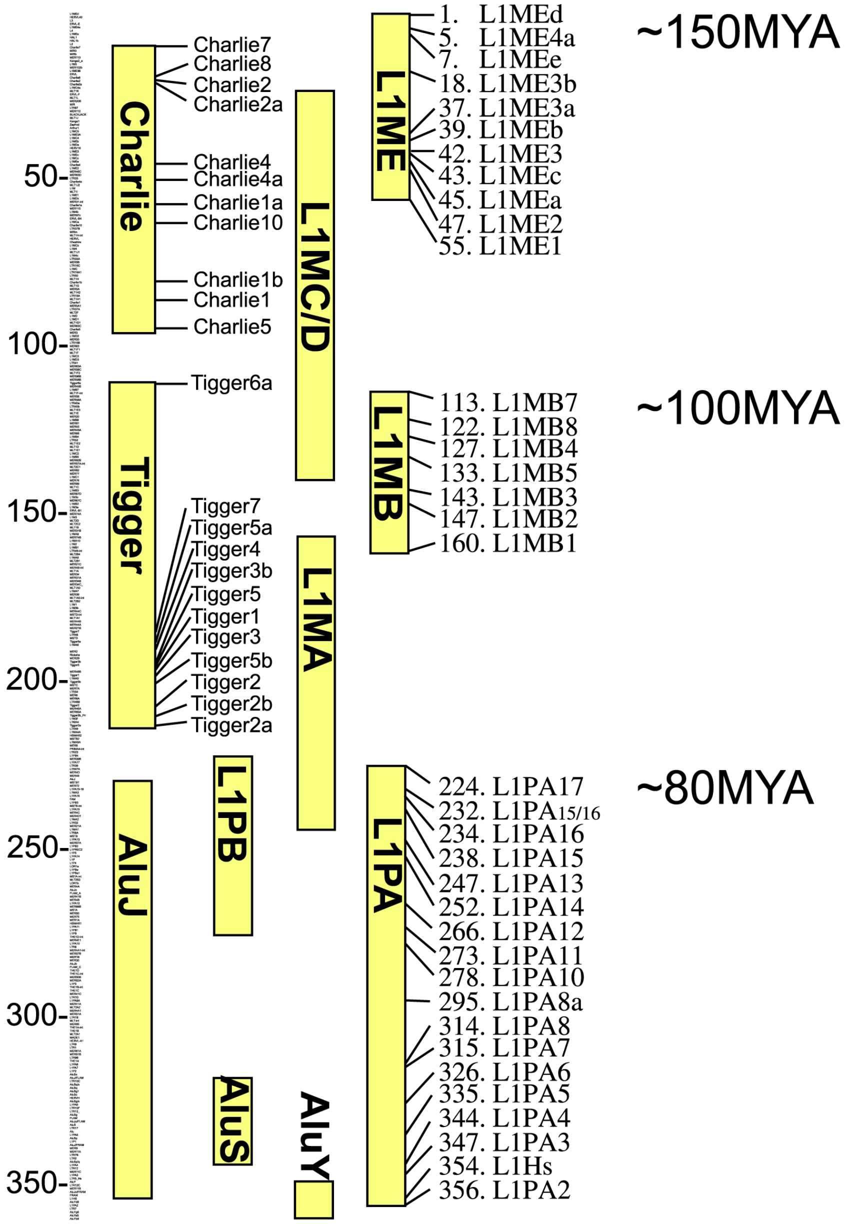


**Figure 2.8:** The resultant relative age order of 360 TEs calculated in [47], where the numbers on the left axis represent positions in the age order.

- A chronological order of TEs was calculated, which tells a relative age order of TEs, but without knowing how much older one TE is than another.
- Since each TE has a distinct period of activity, the age order (positional distribution) derived from this method cannot tell the exact timespan of transposition activity of each individual TE.
- Moreover, the relative age of the TEs that have not interrupted or been interrupted by other TEs can not be estimated by this method.

Overall, the interruptional analysis provides a novel analysis of the evolutionary history of some of the most abundant and ancient repetitive DNA elements in mammalian genomes by analyzing a single genome, which is important for understanding the dynamic forces that shape the genomes during evolution.

### 2.3.3 Perspective

The sequence divergence analysis is a traditional and common method for TE phylogeny, and it has been extensively used in estimating the age of younger TE families; however, other methods are worthy of investigation especially for older and more diverged elements. In contrast, the estimation of TE evolution based on interruptional analysis brought in a new idea of considering the interruptional activities between TEs, which can be used to predict the age and evolutionary activity of these TEs.

# Chapter 3

# Computational identification of harmful mutation regions that influence transpositions of active transposable elements in the human genome [1]

## 3.1 Abstract

As the most abundant transposable elements, *Alu* elements have 1.1 million copies interspersed throughout the human genome, and about 11% of the human genome consists of *Alu* sequences [50]. Recent evidence indicates that the 22 *AluY* and 6 *AluS* TE families have been the most active TEs in recent human history [104], whose transposition has been implicated in several inherited human diseases and in various forms of cancer by integrating into genes; therefore, understanding the transpositional activity and factors that change the activity levels of these TEs is very important. There has been some work done to quantify and analyze the transposition of active *Alu* transposable elements in mobile assays. Based on this activity data, a method/simulation was created in this chapter to computationally identify the regions on a TE consensus sequence that may change the transpositional activity. This method was applied to *AluY*, the youngest and most active *Alu* subfamily, to identify the harmful mutation regions laying in its consensus. Mutations occurring within these regions have crucial effects in decreasing the elements' transposition. The identified regions were then verified by the secondary structure of the *AluY* RNA, where the harmful regions overlapped with the *AluY* RNA major SRP9/14 contact sites. An additional simulation also showed that the identified harmful regions covering the *AluY* RNA functional regions are not occurring by chance. Therefore, we conclude that mutations within the harmful regions identified alter the mobile activity levels of active *AluY* elements. The method was then applied to the *Alu* family and *L1* family in detecting the harmful regions in these elements.

[1] Part of the work in this chapter has been published in [60], and some is submitted in [61].

## 3.2 Introduction and motivation

Active TE elements only constitute a tiny fraction of the TE complement of the genomes of most organisms; for example, there are only about 100 active *L1* copies out of a total of over 500,000 *L1*s in the human genome [41]. Recent evidence indicates that 35 to 40 subfamilies of *Alu*, *L1*, and *SVA* elements remain actively mobile in the human genome [104, 71]. Active human transposons have been estimated to generate about one new insertion per 10 to 100 live births [69, 89, 31]. Specifically for the *Alu* family, it is estimated that one new *Alu* insertion occur for every 20 live human births [31], and there is one *Alu* insertion for every 3000 bp in the human genome on average [84]. *Alu* transposition events have a major impact on human biology and diseases [104] because the active TEs can create genetic diversity in human populations and integrate into genes that cause diseases. Indeed, forty-three disease-causing *Alu* insertions have been identified [9]. As mentioned in Chapter 1, in recent research in [86], the authors hypothesize that *Alu* insertions in mitochondrial genes can lead to progressive neurological disfunction. Therefore, it is of importance to understand how the activity level of *Alu*s can change based on possible mutations and mutation loci.

*Alu* elements are approximately 300 base pairs long, and do not contain any protein-coding sequences for transposition. They rely upon *L1*-encoded proteins for their own mobilization [35]. It is believed that *Alu* elements are derived from the 7SL RNA, and modern *Alu* elements emerged from a head-to-tail fusion of two distinct fossil antique monomers, hence its dimeric structure of two similar, but distinct monomers (left and right arms) joined by an A-rich linker and terminated by a poly(A) tail [55]. The left arm contains functional, but weak, A and B boxes of the RNA polymerase III internal promoter [104], as shown in Figure 3.1 and also Figure 3.8.

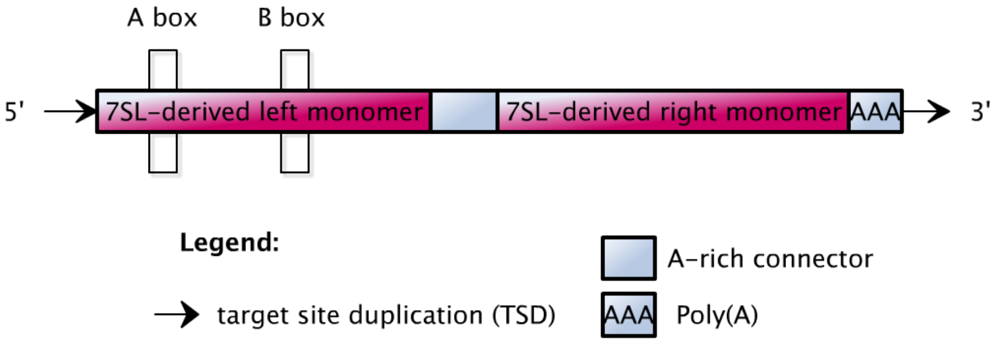


**Figure 3.1:** Structure of *Alu* elements.

Different periods of evolutionary history have given rise to different families and subfamilies of *Alu* elements, each containing a small number of active *Alu* elements that serve as the source of subsequent families [50]. According to Repbase Update, there are three *Alu* subfamilies. *AluJ* is the most ancient (about 65 million years old), and is thought to be functionally extinct [8, 104, 9, 12]; the second oldest is the *AluS* subfamily, which became active approximately 30 million years ago, and only some intact elements were found to be active in humans [12, 105]; *AluY* is the youngest subfamily, and most elements of this subfamily are currently active [77]. Because there is no specific mechanism for removal of *Alu* insertions, *Alu* evolution is dominated by the accumulation of new *Alu* inserts [50]. These new copies of *Alu* accumulate mutations independently over time.

In order to identify active *Alu* copies that exist in the human genome and analyze their transpositional activities, Bennett et al. [12] designed an *in vivo* experiment to systematically examine the mobilization capacity of *Alu* copies across the human genome, in particular the transposition capacity of the 280 bp central "core" regions of *Alu* copies using a plasmid-based mobilization assay. The experimental procedure is described in more detail in [35]. Briefly, the *Alu* retrotransposition was detected on induction by LINE expression vectors. Human HeLa cells were co-transfected with a marked *Alu* and an expression vector for the human *L1* under the control of the CMV promotor. Cells were amplified and retrotransposition events were detected. This method allows for comparing the relative mobilization efficiencies of diverse core elements by keeping all other factors constant and eliminating possible variation due to flanking sequences.

An annotated database of 850,044 full-length human *Alu* copies was first developed in [12], then some representative elements were carefully selected from the database, as well as several synthetic older consensus elements that are no longer present in the modern human genome, totalling 89 elements, with 9 *AluJ*, 28 *AluS*, and 52 *AluY*. These elements were then cloned and tested in a mobile assay. From the functional analysis of the *Alu* elements, it was shown that the elements with fewer changes relative to the consensus sequences generally had the highest levels of activity. No elements with more than 10% mutations, which would occur with at least 28 bp changes, were active [12]. This indicated that the amount of sequence variation is an effective factor in altering the transpositional activity. However, the fact that polymorphic *AluY* copies[2] generally had robust levels of mobilization in contrast to the randomly chosen *AluY* copies with sequence variation, indicated that some sequence changes are more effective than others in altering activity.

In light of the experiments and the functional analysis done in [12], a more detailed analysis of more than just sequence similarity with the consensus is desired to understand more precisely what influences TE activity. In this chapter, a computational method is developed to further analyze how the sequence of an element influences its transpositional activity; specifically, this method identifies the critical regions within the *AluY* consensus that have crucial effects in deactivating the elements' transposition, called "*harmful mutation regions*". This analysis can be applied to any TE family or other organism, but it requires further experiments akin to those in [12], where a quantified transposition fraction is available for each TE.

[2] Some elements from the young *Alu* subfamilies, known as *Y, Yc1, Yc2, Ya5, Ya5a2, Ya8, Yb8*, and *Yb9*, have inserted into the human genome so recently that they are *polymorphic* with respect to the presence or absence of insertion in different human genomes.

## 3.3 Materials and notations

### 3.3.1 Materials

Because *AluY* is the youngest *Alu* subfamily that harbours the largest number of active elements, in this section, the *AluY* sequences from the experiment in [12] will be analyzed.

First, pairwise sequence alignments of the *AluY* elements against the *AluY* consensus sequence from Repbase Update were calculated, giving pairwise scores for every *AluY* element sequence with the *AluY* consensus. Pairwise scores are simply the number of identities between the two sequences, divided by the length of the alignment, giving the *percent identity.*

In the experiment in [12], *AluYa5* elements were used as a standard for comparing the retrotransposition activity. An element is considered more active than *AluYa5* when the cell culture of this element showed greater fluorescence intensity than the cell culture of *AluYa5*, and vice versa. The average *activity fraction* of a TE is defined as a percentage of the fluorescence intensity of the cell culture of this TE over that of *AluYa5* elements. The *Alu* elements can then be categorized by their average activity fraction (ranges from 0% to 118% of *AluYa5* activity — it can be over 100% if the activity is higher than *AluYa5*). Starting from these activity fractions, all *Alu* elements were organized into four activity level groups as in Definition 1.

**Definition 1.** *A set of elements is defined as in the same* activity group *if their activity fractions are in the same range:*

- *the* inactive group *consists of elements with activity fractions that range from* 0% *to* < 5%*,*
- *the* low activity group *consists of elements with activity fractions that range from* 5% *to* < 40%*,*
- *the* moderate activity group *consists of elements with activity fraction that range from* 40% *to* < 66.6%*,*
- *the* high activity group *consists of elements with activity fraction greater than* 66.6%*.*

The activity fractions of all the *AluY* elements were plotted against their percent identity in Figure 3.2, where each data point represents one *AluY* element; the x-axis is the percent identity of the elements to the *AluY* consensus sequence; the y-axis is the activity fraction of these elements.

The elements in Figure 3.2 are also grouped into similar activity levels (e.g., high activity group, moderate activity group, low activity group, and inactive group, as marked in the figure). It can be seen that, roughly, the elements with higher percent identity tend to have higher activity level. However, a linear relationship is not clear. For example, there exist some elements with high activity level (in the high activity group) but a low percent identity, while some elements have a high percent identity but a low activity level (in the low

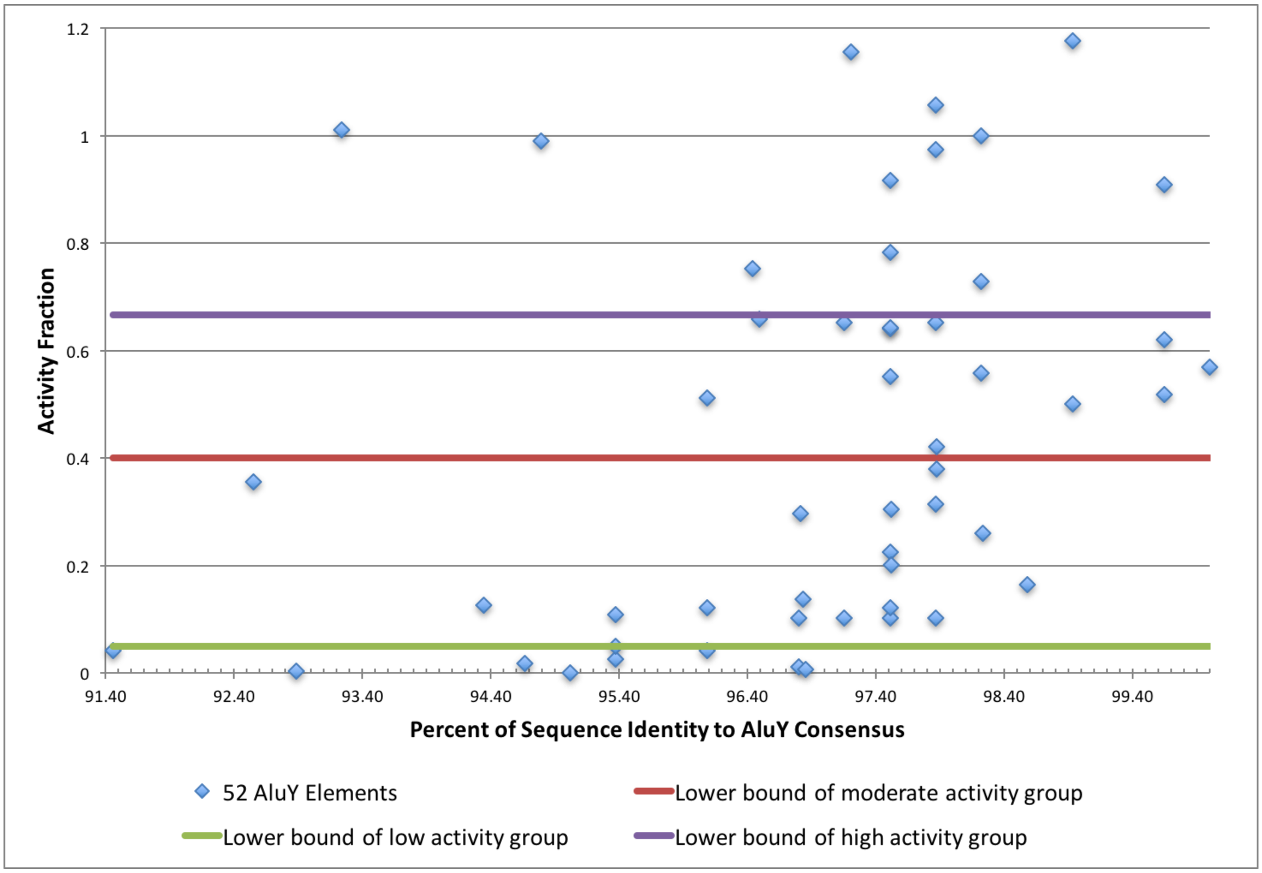


**Figure 3.2:** The plot of the 52 *AluY* elements from [12], where the x-axis is the percent identity and the y-axis is the activity fractions. The elements are partitioned into different groups of activity levels. Some elements are also grouped into vertical bins for further analysis in Section 3.4.3.

activity or inactive groups). The lack of a clear linear trend leads to the hypothesis that some mutation sites are less effective and some are in contrast more effective in altering the elements' transpositional activities. A computational method is proposed in the next sections to identify these affective mutation sites.

### 3.3.2 Notations

In order to formulate the description of the problem and the computational method that will be proposed, some notations need to be defined first.

**Definition 2.** *The* total number of elements *to be considered in the TE family is denoted by $N$, and the* length of the consensus sequence *of this TE family is denoted by $L$.*

*For example, considering the 52 AluY elements in [12] ($N = 52$), the length of the AluY consensus in RU is $L$ = 282.*

*A* window *is a region within the consensus sequence, and is defined by a window size, denoted by wsize, and* *a start position of the window.*

*For example, a window denoted by $w_i$ is the region from the ith position to the jth position, where $j = i + wsize - 1$, in the consensus sequence.*

*Given the length of consensus and a window size, the* number of windows*, denoted by nw, can be calculated as $nw = L - wsize + 1$.*

**Definition 3.** Mutations in the window $w_i$ *of one TE element is defined as the total number of mutations (versus the consensus) of this element lying within the window, denoted by $m_i$.*

For example, for an element with mutated positions at $2, 3, 7, 15, 80, 224$ in the consensus, given $wsize = 10$, then $m_1 = 3$ (number of mutations in the window from position 1 to 10), and $m_{10} = 1$ (number of mutations in the window from position 10 to 19).

**Definition 4.** *For every element, the window is "slid" from the beginning to the end of the consensus, to generate a vector of mutations in all windows for this element. Mutations in all windows in all elements can then be represented as a* mutation matrix*, denoted as $M(N \times nw)$.*

$$M(N \times nw) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & \dots & m_{1nw} \\ m_{21} & m_{22} & m_{23} & \dots & m_{2nw} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{N1} & m_{N2} & m_{N3} & \dots & m_{Nnw} \end{bmatrix} \tag{3.3.1}$$

Taking the example of the $AluY$ elements in [12], there are $N = 52$ rows and $nw = 273$ (where $L = 282, wsize = 10$) columns in the matrix. The mutation matrix of the $AluY$ elements, representing the mutations of every element in each window, is shown in the heat map in Figure 3.3. The windows on the $AluY$ consensus are shown in the x-axis; the $AluY$ TE elements sorted by their activity fractions in descending order are shown in the y-axis. The activity groups are also marked with black lines in the figure.

From Figure 3.3, it is easy to see that certain windows are apparently darker than others in most elements, which indicates that certain regions in the sequence tend to have more mutations; however, the heat map does not show whether the positions where mutations occurred are correlated with mobile activities, nor does it show how they are related.

In the next sections, by using correlation analysis (Section 3.4) and group comparison analysis (Section 3.5) respectively, it will be shown that mutations in some windows are indeed harmful to elements' activities.

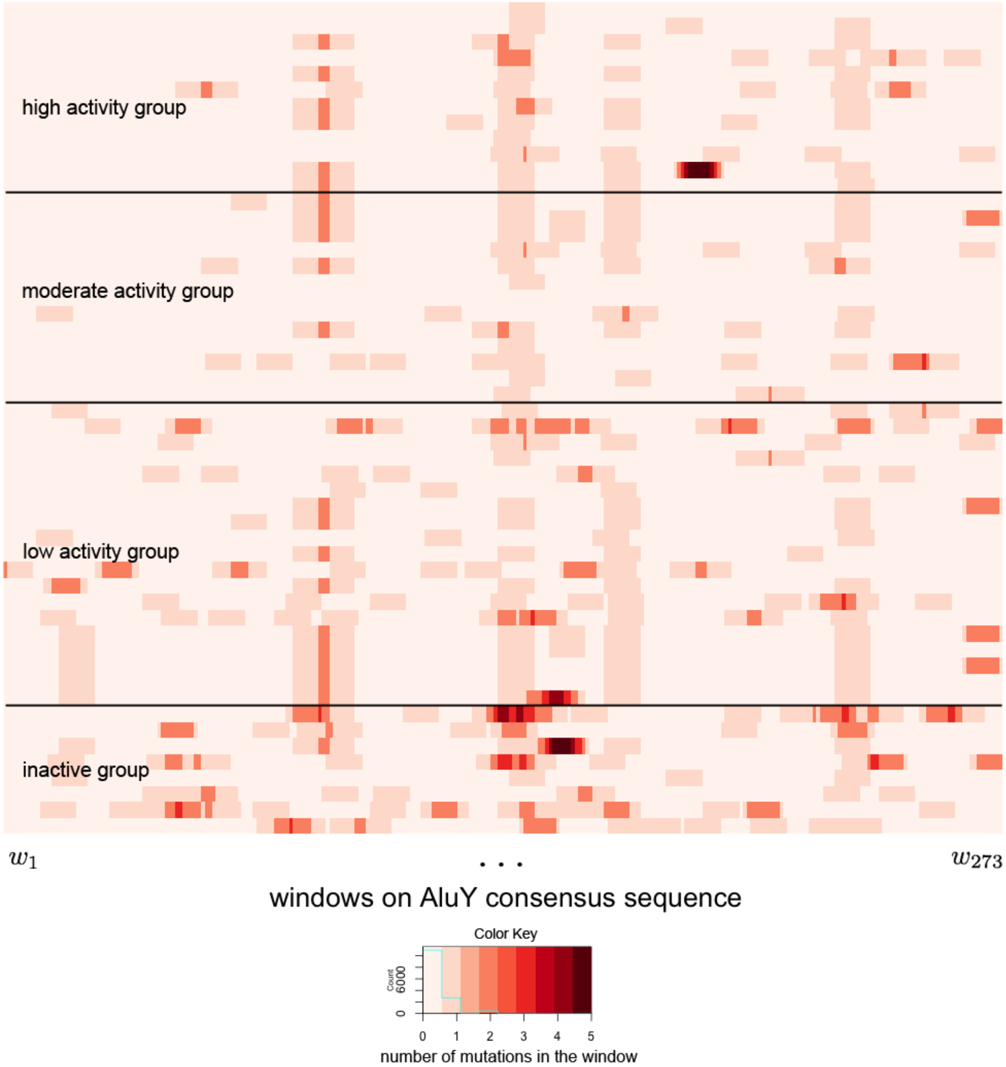


**Figure 3.3:** The number of mutations in each window ($wsize = 10$) of all 52 *AluY* element of Figure 3.2, where the x-axis is all windows on the *AluY* consensus sequence and the y-axis is the *AluY* elements. The elements are sorted by their activity fractions in descending order from the top to the bottom of the chart.

## 3.4 Method I: identification of harmful regions by correlations between mutations and mobile activity

In this section, a computational method is proposed to identify the harmful regions in an active TE family using the Pearson's coefficient of correlation.

### 3.4.1 Pearson's coefficient of correlation and multiple test correction

The *Pearson's coefficient of correlation*, normally denoted by $\rho$, is a measure of the linear correlation between two variables ($X$ and $Y$), whose values range from -1 to 1, with 1 indicating total positive correlation, 0 indicating no correlation, and -1 indicating total negative correlation. It is defined as the covariance of the two variables divided by the product of their standard deviations.

$$\rho = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{3.4.1}$$

where $cov(X,Y)$ is the covariance of $X$ and $Y$, $\sigma_X$ is the standard deviation of $X$, and $\sigma_Y$ is the standard deviation of $Y$.

Though using models for data analysis has many advantages, if the existing data does not fit the model perfectly, results are often misleading. The Pearson's correlation coefficient is a model-free method, and it therefore shows the nature of the data without depending on any existing models.

For each window in the $AluY$ consensus, the variable $X$ is defined as the number of mutations in the window of all $AluY$ elements, and the variable $Y$ is defined as the activity fractions of all $AluY$ elements. The Pearson's coefficient of correlation was calculated by comparing $X$ against Y, using the correlation function `cor` in the `R` Language. The observed correlations from the data in the experiment in [12] are calculated and denoted by

$$\rho_{obs} = (\rho_1, \rho_2, \ldots, \rho_{nw}),$$

as shown in Figure 3.4. It can be seen in the figure that mutations occurring in most of the windows have negative correlation with the mobile activities. The negative correlations indicate that the TE activity decreases as the number of mutations in a window increases; in other words, the mutations in the window are harmful to TE activity. On the other hand, the negative correlations between the activity fractions and the number of mutations may arise by randomness/chance. Therefore, it is necessary to perform a statistical significance test to measure the probability that more negative correlations than what was observed in the data set can be caused solely by chance, which is the $p$-value, a measure of significance in terms of the false positive rate [144].

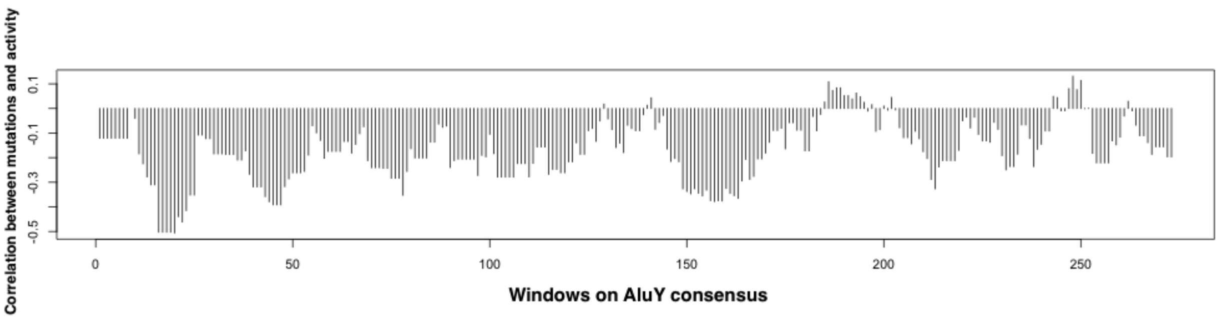


**Figure 3.4:** The Pearson's coefficients of correlation between the number of mutations in each window and the activity fractions of the $AluY$ elements. The x-axis gives the windows in order on the $AluY$ consensus.

In order to correct for multiple comparison bias due to a large number of windows, a $q$-value is also reported. A $q$-value, similar to a $p$-value, is a measurement of the "false discovery rate" (FDR) [11]. The false positive rate and FDR are defined differently — given a rule for calling features significant, the false positive rate is the rate that truly null features are called significant, while the FDR is the rate that significant features are truly null [134]. For example, a false positive rate of 5% in a study means that 5% of the truly null features are called significant on average, while a FDR of 5% means that 5% of all features that are called significant are truly null. In general, the FDR is a sensible measure of the balance between the number of true positives and false positives. Multiple testing correction will be performed using the `qvalue` package [146] under Bioconductor in the `R` Language.

### 3.4.2 Statistical significance tests and results

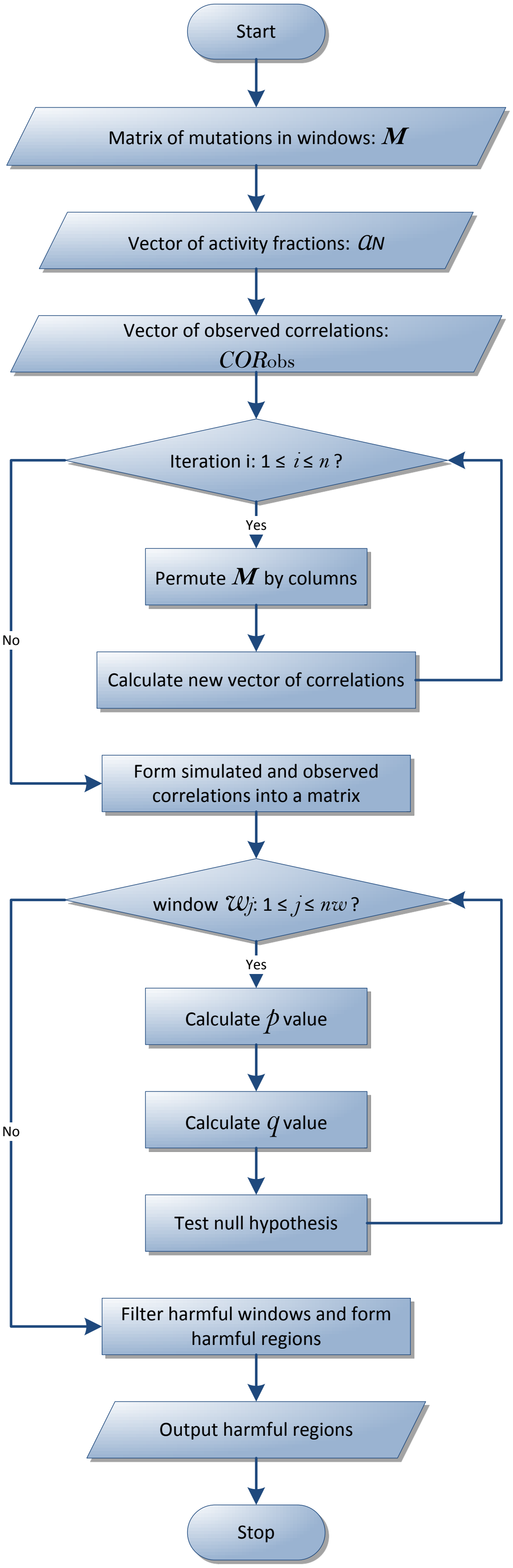
In order to investigate the relationships between the mobile activity and the mutations of a TE, a null hypothesis is proposed as "**mutations in a window are not negatively related (or undifferentiated) to the activity of the TE"**. To test the hypothesis, a statistical simulation is used to generate random data as elaborated in the steps below. The framework of the simulation is a general statistical technique for hypothesis testing.

Given a mutation matrix, $M(N \times nw)$, as in Equation (3.3.1), the activity fractions vector of the $N$ elements, $\alpha_N$, and the observed correlations $\rho_{obs}$, perform the following operations, with the flow chart of the steps in Figure 3.5.

Step 1: generate simulated correlations as follows:

given the number of iterations as $n$ (e.g., $n = 1000$), for each iteration denoted by $i$,

1. permute $M$ by columns as $M^i$;



Start
Matrix of mutations in windows: $\boldsymbol{M}$
Vector of activity fractions: $a_N$
Vector of observed correlations: $COR_{\mathrm{obs}}$
Iteration i: 1 ≤ $i$ ≤ $n$ ?
Yes
No
Permute $\boldsymbol{M}$ by columns
Calculate new vector of correlations
Form simulated and observed correlations into a matrix
window $\mathcal{W}_j$: 1 ≤ $j$ ≤ $nw$ ?
Yes
No
Calculate $p$ value
Calculate $q$ value
Test null hypothesis
Filter harmful windows and form harmful regions
Output harmful regions
Stop

**Figure 3.5:** The flow chart of the algorithm in Section 3.4.2.

2. calculate correlations between $M^i$ and $\alpha_N$. The correlations for each window for this iterations is denoted by $\rho_{(i,1)}, \rho_{(i,2)}, \ldots, \rho_{(i,nw)}$.

Step 2: form simulated and observed correlations into a matrix.

After the $n$ iterations, there are $n$ simulated correlations for each window. The simulated correlations along with the observed correlations are formed into a matrix and summarized in Table 3.1.

| iteration | $w_1$ | $w_2$ | $\ldots$ | $w_j$ | $\ldots$ | $w_{nw}$ |
|---|---|---|---|---|---|---|
| 1 | $\rho_{(1,1)}$ | $\rho_{(1,2)}$ | $\cdots$ | $\rho_{(1,j)}$ | $\cdots$ | $\rho_{(1,nw)}$ |
| 2 | $\rho_{(2,1)}$ | $\rho_{(2,2)}$ | $\cdots$ | $\rho_{(2,j)}$ | $\cdots$ | $\rho_{(2,nw)}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| i | $\rho_{(i,1)}$ | $\rho_{(i,2)}$ | $\cdots$ | $\rho_{(i,j)}$ | $\cdots$ | $\rho_{(i,nw)}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $n$ | $\rho_{(n,1)}$ | $\rho_{(n,2)}$ | $\cdots$ | $\rho_{(n,j)}$ | $\cdots$ | $\rho_{(n,nw)}$ |
| observed correlation ($\rho_{obs}$) | $\rho_1$ | $\rho_2$ | $\cdots$ | $\rho_j$ | $\cdots$ | $\rho_{nw}$ |
| $p$-value | $p_1$ | $p_2$ | $\cdots$ | $p_j$ | $\cdots$ | $p_{nw}$ |
| $q$-value | $q_1$ | $q_2$ | $\cdots$ | $q_j$ | $\cdots$ | $q_{nw}$ |

**Table 3.1:** Simulated and observed correlations between mutations and mobile activities.

Step 3: calculate $p$-values for each window.

For each column $w_j$ $(1 \leq j \leq nw)$ in Table 3.1, calculate a $p$-value of $\rho_j$ in the distribution of $\rho_{(i,j)}(1 \leq i \leq n)$, which is $p_j = P(\rho_{(i,j)} \leq \rho_j)$, where $1 \leq j \leq nw$.

Step 4: calculate $q$-values for each window.

After the $p$-values are calculated for each window, estimate the $q$-values of each window, $q_1, q_2, \ldots, q_{nw}$ using the function `qvalue` in the `R` Language.

Step 5: test the null hypothesis for each window.

For each window $w_j$ $(1 \leq j \leq nw)$, compare its $q$-value, $q_j$, to a confident threshold $\lambda$ (eg. $\lambda = 0.05$). If $q_j < \lambda$, we can reject the null hypothesis that "**mutations in window $w_j$ are not negatively related (or undifferentiated) to the activity of the TE**". If the null hypothesis is rejected, then the window $w_j$ is harmful, and the sites in the window are harmful sites.

Step 6: filter out all windows that are harmful and form overall harmful regions.

Example 1 below shows how to test if a window is a harmful window by comparing the observed correlation and simulated correlations between elements' mobile activities and mutations in a window using the above method.

**Example 1.** *Assume a window size* $wsize = 10$, *the number of iterations* $n = 10,000$, *and consider the window* $w_{20}$ *where the window is between positions 20 and 29. Given* $M$ *as the matrix of mutations in windows, calculate the observed correlation of* $w_{20}$ *by comparing the number of mutations in the 20th window,* $M[,20]$ *(the 20th column of the matrix), and the elements' activity fractions vector,* $\alpha_N$. *The observed correlation is* $\rho_{20} = -0.5059255$. *Then perform the following steps:*

*Step 1: permute* $M[,20]$ $n$ *times and calculate the correlation for every permutation, denoted by* $\rho_{(1,20)}, \rho_{(2,20)}, \ldots, \rho_{(n,20)}$. *The distribution of the simulated correlations* $\rho_{(1,20)}, \rho_{(2,20)}, \ldots, \rho_{(n,20)}$ *is shown in Figure 3.6.*

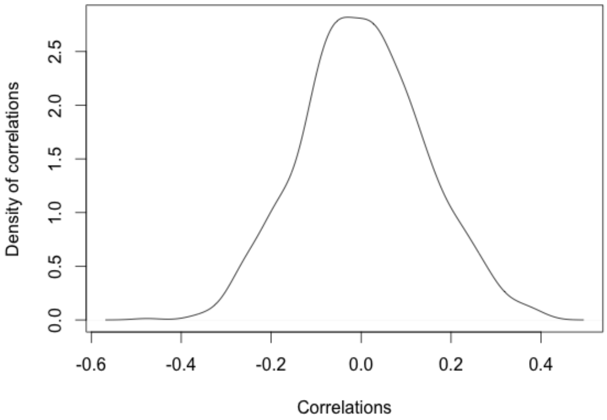


**Figure 3.6:** The distribution of the simulated correlations in the window between position 20 and 29.

*Step 2: calculate the p-value of the observed correlation in the distribution:* $p_{20} = P(\rho_{(i,20)} \leq \rho_{20}) < 0.00001$. *Using the same method, the p-values of all windows can be calculated.*

*Step 3: perform a multiple test correction to calculate the q-values. The q-value of the window in this example is calculated as* $q_{20} < 0.00001$.

*Step 4: given the confident threshold* $\lambda = 0.05$, *the null hypothesis is rejected. Hence, the window* $w_{20}$ *is considered a harmful window, which means that mutations occurring within this window are more affective to the mobile activities of the AluY elements.*

Using this method, the $p$-value and $q$-value are calculated for every window in the *AluY* consensus and the results are shown in Figure 3.7 (a) and (b) respectively.

Given a confidential threshold $\lambda = 0.05$, a window in the *AluY* consensus is identified as a harmful window if and only if its $q$-value $\leq \lambda$. The harmful windows that are overlapped form into harmful regions as listed in Table 3.2.

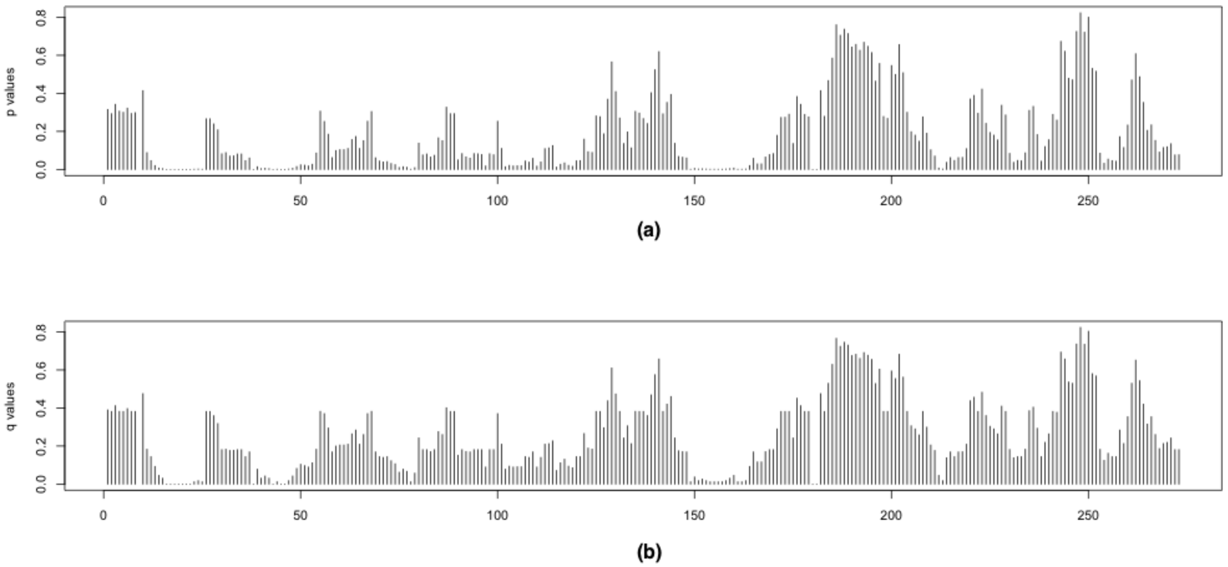


**Figure 3.7:** The $p$-values in (a), and $q$-values in (b) of the *AluY* elements. The x-axis gives the windows in order on the *AluY* consensus.

| region ID | regionStart | regionEnd | average $q$-value |
|---|---|---|---|
| 1 | 14 | 34 | 0.0101 |
| 2 | 38 | 57 | 0.0183 |
| 3 | 78 | 87 | <0.0001 |
| 4 | 149 | 172 | 0.0178 |
| 5 | 180 | 190 | <0.0001 |
| 6 | 212 | 222 | 0.0232 |

**Table 3.2:** The harmful mutation regions in *AluY* elements calculated from correlation analysis ($\lambda = 0.05$).

With $\lambda = 0.05$, the identified harmful regions in Table 3.2 cover 34.5% of the total length of *AluY* consensus sequence. Next, these computationally identified regions will be verified to be harmful to the activity of *AluY* elements.

### 3.4.3 Verifications

In this subsection, the harmful regions in Table 3.2 will be verified in two different ways. First, the *AluY* elements with similar percent identity having various activity is due to whether or not mutations occurred in harmful regions. Second, a possible reason of the harmful regions affecting transpositional activity is because the harmful regions overlap with the functional sites of the *AluYa5* RNA that are important for the transposition of the elements.

### Verification by activity of *AluY* elements

The sequences of the *AluY* elements from [12] are compared to the *AluY* consensus sequence in this section. A relationship between the percent identity of these elements and their levels of mobile activity is plotted in Figure 3.2. It is observed that by grouping some elements with similar percent identity into vertical bins, as marked in the figure, the activity levels of the elements in the same bin vary to a large extent. For example, the elements in bin #1 all have a similar percent identity with the consensus ($\sim$ 97%), but their activities range from 1% to 106% (the activity fraction is in comparison to the activity of *AluYa5*, and if an element is more active than *AluYa5*, the activity fraction could go over 100%).

One possible explanation for this difference is that some mutations occurred in the elements' harmful regions, which decreased their activities dramatically. Thus, all mutations in the high activity group are labelled as "neutral sites", as the elements remain highly active despite the mutations. In other words, these mutations might be "less effective" to their transpositional activities.

| | Activity group | Harmful regions | Neutral sites |
|---|---|---|---|
| bin 1 | low activity | 13 % | 63 % |
| | moderate activity | 13 % | 63 % |
| | high activity | 0 % | 100 % |
| bin 2 | low activity | 0 % | 0 % |
| | moderate activity | 0 % | 5 % |
| | high activity | 0 % | 0 % |
| bin 3 | low activity | 6 % | 0 % |
| | moderate activity | 0 % | 11 % |
| | high activity | 0 % | 8 % |
| bin 4 | low activity | 0 % | 33 % |
| | moderate activity | 40 % | 20 % |
| | high activity | 0 % | 10 % |

**Table 3.3:** The percentage of mutations grouped by bins marked on Figure 3.2 over the total number of mutations in each group.

Table 3.3 lists the percentage of mutations that occurred in harmful region and neutral region respectively for each activity group in the bins in Figure 3.2. It is observed that, in the low activity groups of each bin, there are more mutations in the harmful region compared to other activity groups. Moreover, none of the mutations in the high activity groups falls into the harmful regions. Therefore, the mutations that occurred in the harmful region may cause the low activity levels of these elements.

### Verification by *AluYa5* RNA secondary structure

As introduced in Section 3.2, *Alu* elements are derived from two 7SL RNA forming left and right arms, and the left arm contains A and B boxes of the RNA polymerase III internal promoter. Figure 3.8 shows the secondary structure of the *AluYa5* RNA calculated by Mfold [149] (a program for predicting the secondary structure of RNA) based on previously determined secondary structure in [128, 55]. It is known that SRP9/14 binding is necessary for efficient *Alu* mobilization, and the left *Alu* monomer binding to SRP9/14 is more important for mobilization than the right *Alu* monomer binding [12]. In Figure 3.8, the major and the minor SRP contact sites, as well as the A and B boxes, are marked on the structure in grey; the identified harmful regions from Table 3.2 are marked in yellow. As visually indicated in Figure 3.8, the harmful regions "cover" the two major SRP contact sites and Box B very well, with three other unknown regions that are recognized as harmful. The unknown regions might be caused by the limited amount of *Alu* transposition data, or they may have some interesting unknown functions.

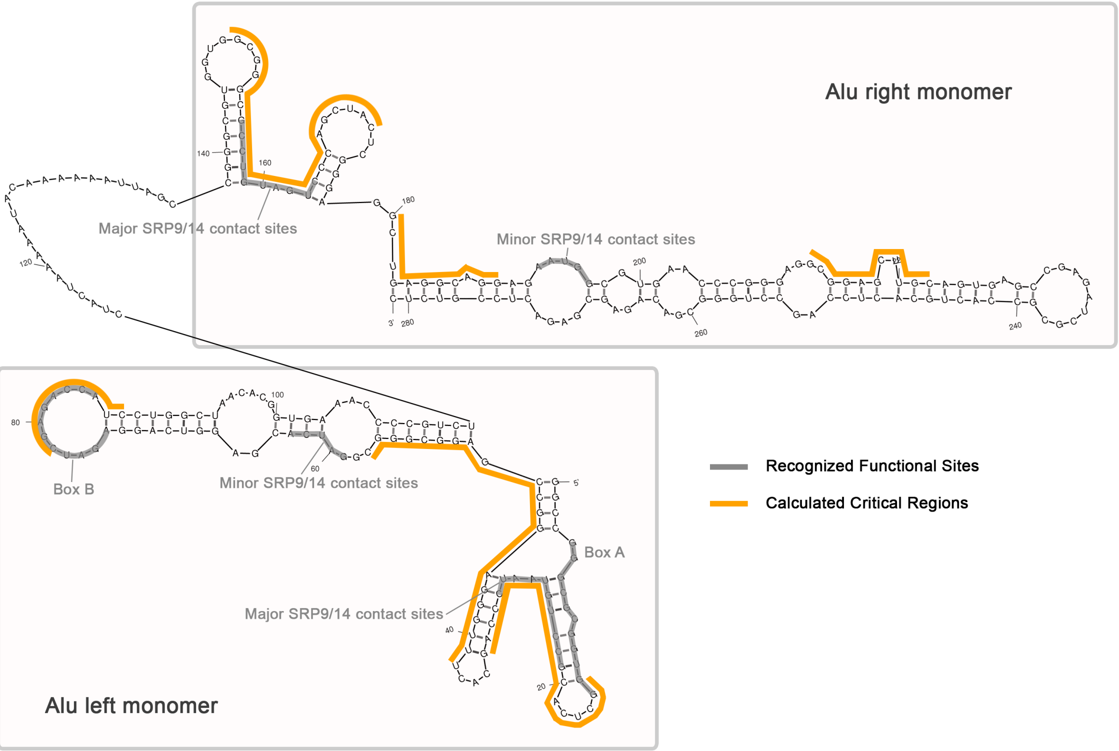

Alu right monomer
Major SRP9/14 contact sites
Minor SRP9/14 contact sites
140
160
180
120
200
260
240
280
3'
Alu left monomer
Box B
Minor SRP9/14 contact sites
Major SRP9/14 contact sites
Box A
80
100
60
40
20
5'
Recognized Functional Sites
Calculated Critical Regions

**Figure 3.8:** The secondary structure of an *AluYa5* RNA. The SRP contact sites and the A and B boxes are marked in grey. The harmful regions identified in Table 3.2 are marked in yellow along the structure as indicated in the legend.

Next, it will be shown that the identified harmful regions covering the functional regions are not picked up totally randomly (by chance). Define the *coverage of harmful regions* as the percentage of the overlapped number of positions between harmful regions (marked in yellow) and functional regions (marked in grey) divided by the total number of positions in the functional regions (marked in grey). Another simulation is developed to compare the coverage of the harmful regions and that of randomly generated regions, which is described as follows: given the lengths and positions of functional regions ($nf$ as number of functional regions), the lengths and positions of harmful regions ($nh$ as number of harmful regions), and the number of trials as $n$,

1. calculate the coverage of harmful regions

$$Cov_{\text{harmR}}.$$

2. For every iteration $i$, where $1 \leq i \leq n$,

   (a) randomly generate $nh$ regions with the same lengths as the harmful regions identified as shown in Table 3.2, and the algorithm makes sure that these regions do not overlap with each other;

   (b) calculate the coverage of randomly generated regions in this iteration, denoted by $Cov_{\text{randR}^i}$.

3. After $n$ iterations, there are $n$ generated coverages, denoted by

$$Cov_{\text{randR}^1}, Cov_{\text{randR}^2}, \ldots, Cov_{\text{randR}^n}.$$

4. Calculate the probability where the coverage of harmful regions is less than the coverage of random regions as

$$P(Cov_{\text{harmR}} < Cov_{\text{randR}}).$$

Using this method and running the simulation on the $AluY$ harmful regions calculated in Table 3.2 for $n = 10,000$ iterations, Figure 3.9 is the distribution of the coverage of random regions, and the blue line on the figure shows the coverage of the harmful regions in Table 3.2.

The probability is calculated as $P(Cov_{\text{harmR}} < Cov_{\text{randR}}) = 22\%$; that is, 78% of randomly generated regions have less coverage than the harmful regions identified by our method. Therefore, we conclude that the harmful regions covering the $AluY$ functional regions and this coverage is probably not by chance.

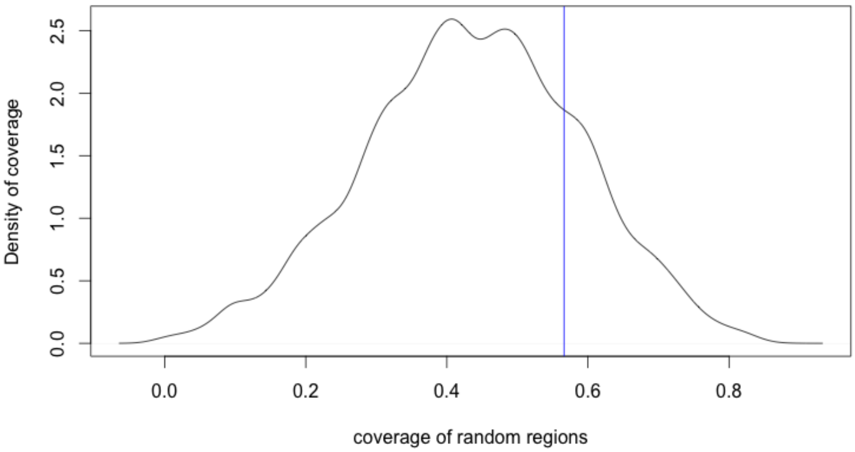


**Figure 3.9:** The distribution of the coverage of random generated regions. The blue vertical line is the coverage of the harmful regions.

## 3.5 Method II: Identification of harmful regions by group comparisons

In this section, the same type of analysis and verifications will be repeated, but with another computational method, called *group comparison analysis*, to identify the harmful regions by randomly introducing mutations into the elements, then comparing them with different activity groups (Definition 1) to evaluate statistical significance tests. This is a statistical technique similar to comparing two population means [99]. Rather than classifying regions as harmful or not, the harmful regions identified by this method are classified into more detailed groups in terms of changing the activity levels in different ways, e.g., mutations in some regions can potentially change a TE from high activity level to moderate activity level, etc.

In describing this approach, some additional terms and labels are used. Based on the activity group defined in Definition 1, *mutations of a set of elements* is defined as follows.

**Definition 5.** *Given a set (group) of related elements, e.g., elements of the same activity group,* the number of mutations *of the group is defined as the total number of the mutations (versus consensus) that occurred in every element in this group.*

Thus mutations can occur more than once at a nucleotide position in the consensus sequence for a group, when more than one element is mutated at the same position.

In the group comparison analysis, it is a set of elements (not a single element) in consideration. Similar to the definition of the mutations in a window of an element (Definition 3), the mutations in a window of a set of elements is defined as below.

**Definition 6.** *Given a window $w_i$, and a set of elements $g$, $m_i^g$ is defined as the number of mutations in the* *elements of the set (group) $g$ in the window $w_i$. The number of elements in the set is denoted as $|g|$.*

It should be noted that there are always two sets of elements from the same consensus, denoted as $g_1$ and $g_2$, in comparison. A ratio is used to compare mutations in elements of two different activity groups, as defined in Definition 7.

**Definition 7.** *Given the mutations in the window $w_i$ in the elements of two different activity groups, $g_1$ and $g_2$, the ratio $R_i$ of the window $w_i$ is defined as*

$$R_i = \frac{m_i^{g_2} + 1}{m_i^{g_1} + 1} \times \frac{|g_1|}{|g_2|} \tag{3.5.1}$$

This ratio describes the relationship between the mutations within a specific window of two groups; more specifically, the ratio indicates how different the number of mutations of the two groups are in a window. Note that there might be some cases where there is no mutated position in the elements of the $g_2$ group in the window, therefore, 1 is added to both the numerator and the denominator to avoid zero denominators.

There are four activity groups defined in Definition 1: the high activity group, the moderate activity group, the low activity group, and the inactive group. Take the example of the *AluY* elements in [12], and compare the elements in a group of higher activity ($g_1$) with elements in a group of lower activity ($g_2$) as listed in Table 3.4.

| | $g_1$: group with higher activity level | $g_2$: group with lower activity level |
|---|---|---|
| Comparison type 1 | the high activity group | the moderate activity group |
| Comparison type 2 | the high activity group | the low activity group |
| Comparison type 3 | the high activity group | the inactive group |
| Comparison type 4 | the moderate activity group | the low activity group |
| Comparison type 5 | the moderate activity group | the inactive group |
| Comparison type 6 | the low activity group | the inactive group |

**Table 3.4:** Comparison types of the group comparison analysis.

The observed ratios were calculated as in Equation 3.5.1 in every window of the *AluY* consensus for each comparison type listed above, and the results were visualized in the heat map in Figure 3.10, where the windows of the *AluY* consensus are shown in the x-axis; the six comparisons in Table 3.4 are shown in the y-axis.

From this figure, it can be seen that certain windows have apparently larger ratios, which indicates that certain windows tend to have a larger difference in the number of mutations in the elements in the two groups comparing to other windows. However, the heat map cannot show whether the differences are correlated with

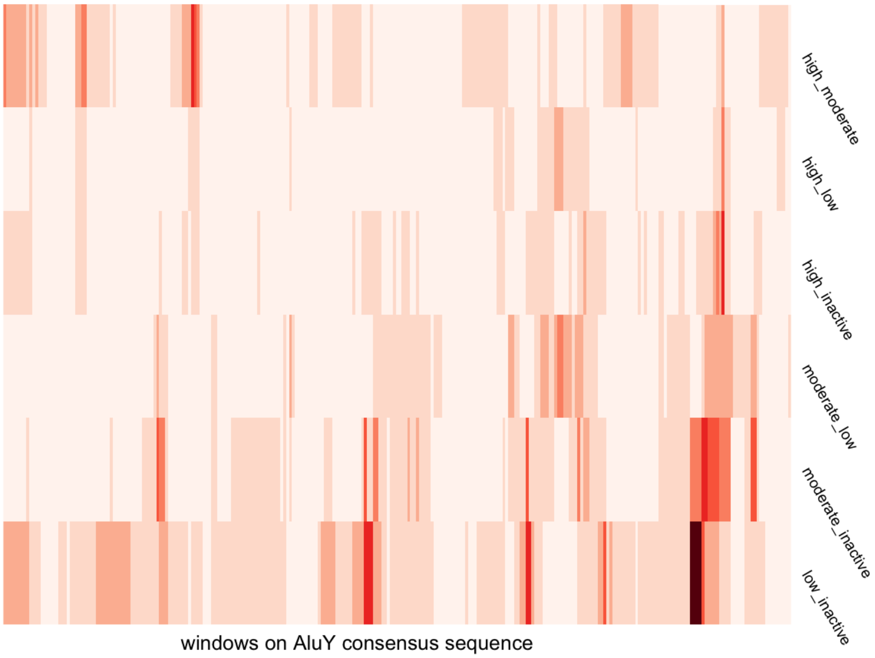


**Figure 3.10:** Observed ratios $R_i$ of the six comparison types in Table 3.4, where the x-axis is all windows on the $AluY$ consensus sequence and the y-axis is the comparison types.

the mobile activities of the elements in the two groups; that is, the difference in mutations may change the elements from a group of higher activity into a group of lower activity.

Next, by using a statistical simulation, the significance of the null hypothesis that "the observed ratio is not greater than expected by chance" will be tested.

### 3.5.1 Statistical significance tests and results

In order to reveal the relationships between the mobile activity and the mutations in a window, the above null hypothesis is rewritten as: "mutations in a window are not negatively related (or undifferentiated) to the activity of the TE", to show a direct relationship between mutations and activity of TEs. A simulation is designed to test the null hypothesis, which is based on the assumption that a mutation can randomly occur at any nucleotide position in a sequence. Note that the evolutionary process of TEs diverging into different subfamilies by accumulating mutations with a mutation rate will not be simulated; instead, given the total number of mutations in the currently existing copies of TEs in an activity group, only the positions where these mutations occurred in the sequence will be simulated. Using the notations defined above, the simulation is described as generally as possible, as follows, so that it can be applied to other TE families. The following steps are performed for each of the comparison types in Table 3.4.

Step 1: calculate observed ratios for all windows, denoted as $R_1, R_2, \ldots, R_{nw}$.

Step 2: generate simulated ratios.

Given the number of iterations as $n$ (e.g., $n = 10,000$), for each iteration denoted by $i$,

1. generate $m^{g_1}$ random positions between 1 and the sequence length $L$ for the $g_1$ group, and generate $m^{g_2}$ random positions between 1 and the sequence length $L$ for the $g_2$ group. The random positions are nucleotide positions in the consensus sequence;

2. calculate simulated ratios in all windows in this iteration, denoted as $r_{i1}, r_{i2}, \ldots, r_{in}$.

Step 3: form observed ratios and simulated ratios into a matrix. After the $n$ iterations, there are $n$ simulated ratios, $r_{i1}, r_{i2}, \ldots, r_{iN}$. The simulated ratios along with the observed ratios are formed into a matrix and summarized in Table 3.5.

| iteration | $w_1$ | $w_1$ | $w_3$ | $\ldots$ | $w_j$ | $\ldots$ | $w_{nw}$ |
|---|---|---|---|---|---|---|---|
| 1 | $r_{11}$ | $r_{12}$ | $r_{13}$ | $\ldots$ | $r_{1j}$ | $\ldots$ | $r_{1(nw)}$ |
| 2 | $r_{21}$ | $r_{22}$ | $r_{23}$ | $\ldots$ | $r_{2j}$ | $\ldots$ | $r_{2(nw)}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $i$ | $r_{i1}$ | $r_{i2}$ | $r_{i3}$ | $\ldots$ | $r_{ij}$ | $\ldots$ | $r_{i(nw)}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $n$ | $r_{n1}$ | $r_{n2}$ | $r_{n3}$ | $\ldots$ | $r_{nj}$ | $\ldots$ | $r_{n(nw)}$ |
| observed ratios | $R_1$ | $R_2$ | $R_3$ | $\ldots$ | $R_j$ | $\ldots$ | $R_{nw}$ |

**Table 3.5:** The matrix of observed and simulated ratios in each window.

Step 4: calculate $p$-values for each window.

For each column $w_j$ ($1 \leq j \leq nw$) in Table 3.5, calculate a $p$-value of $R_j$ in the distribution of $r_{(i,j)}(1 \leq i \leq n)$, which is $p_j = P(r_{(i,j)} \geq R_j)$.

Step 5: calculate $q$-values for each window.

After the $p$-values are calculated for each window, estimate the $q$-values of each window, $q_1, q_2, \ldots, q_{nw}$ using the function `qvalue` in the `R` Language.

Step 6: test the null hypothesis for each window.

For each window $w_j$ ($1 \leq j \leq nw$), compare its $q$-value, $q_j$, to a confidence threshold $\lambda$ (eg. $\lambda = 0.05$). If $q_j < \lambda$, we can reject the null hypothesis that "**mutations in window $w_j$ are not negatively related (or undifferentiated) to the activity of the TE**". If the null hypothesis is rejected, then the window $w_j$ is negatively related to the activity of the TE.

Step 7: filter out all windows that are harmful and form overall harmful regions.

Consider the case comparing the groups of high activity elements ($g1$) and inactive elements ($g2$) to identify potential harmful sites. Mutations at these sites indeed may deactivate a highly active *Alu* element. Example 2 illustrates how to test the hypothesis in a specific window using the *AluY* activity data.

**Example 2.** *Assume a window size is* $wsize = 10$*, the number of iterations is* $n = 1000$*, and consider the case where the window occurs between positions 20 and 29. Here is an example to test if this window is a harmful window by comparing the high activity group with the inactive group. The observed ratio calculated by AluY data using Equation 3.5.1 for this window is* $R_{20} = 5$*.*

*The empirical distribution of the simulated ratios in the experiment is shown in Figure 3.11.*

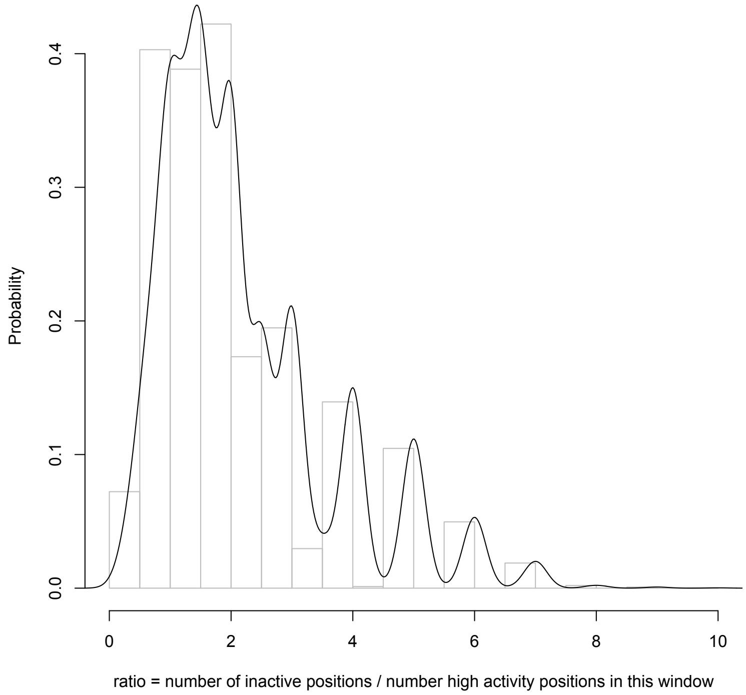


**Figure 3.11:** The empirical distribution of simulated ratios of the window $w_{20}$ (between position 20 and 29). The x-axis is the simulated ratios of the window, and the y-axis is the probability of the ratios.

*The p-value of the observed ratio for this window is* $P(r \geq R_{20}) = 0.367$*, and the q-value is* $q_{20} < 0.0001$*, which is less than the confident threshold* $\lambda = 0.05$*. Therefore, the null hypothesis that "the observed ratio is not greater than expected by chance" can be rejected. The window from position 20 to 29 is considered as harmful.*

Table 3.6 shows all the harmful regions within the *AluY* consensus with $q \leq \lambda$ using the above method by comparing two groups of elements: $g_1$ is the high activity elements group; $g_2$ is the inactive elements group. Mutations occurring in these regions can possibly turn an highly active *AluY* element into an inactive element.

| $w.start$ | $w.end$ | $p$-value | $q$-value |
|---|---|---|---|
| 153 | 162 | $< 0.0000000000$ | $< 0.000000000$ |
| 154 | 163 | $< 0.0000000000$ | $< 0.000000000$ |
| 155 | 164 | 0.0001290323 | 0.007045164 |
| 156 | 165 | 0.0001282051 | 0.007045164 |
| 158 | 167 | 0.0001265823 | 0.007045164 |
| 45 | 54 | 0.0002222222 | 0.007583333 |
| 46 | 55 | 0.0002173913 | 0.007583333 |
| 47 | 56 | 0.0002127660 | 0.007583333 |

**Table 3.6:** Identified harmful windows with $\lambda = 0.05$ comparing two groups of elements: $g_1$ is the high activity elements group; $g_2$ is the inactive elements group. The column of $w.start$ is the start nucleotide position in the $AluY$ consensus sequence; $w.end$ is the end nucleotide position.

Applying the same method to each of the comparison types (pair of activity groups) in Table 3.4, the identified regions of each type are listed in Table 3.7. Note that there are no harmful regions identified in type 1 and type 6 comparisons.

| Type ID | Compared activity groups | Harmful regions |
|---|---|---|
| Type 2 | $g_1$ = high activity group<br>$g_2$ = low activity group | $16 \sim 34$<br>$153 \sim 171$<br>$272 \sim 281$ |
| Type 3 | $g_1$ = high activity group<br>$g_2$ = inactive group | $45 \sim 56$<br>$153 \sim 167$ |
| Type 4 | $g_1$ = moderate activity group<br>$g_2$ = low activity group | $20 \sim 34$<br>$48 \sim 57$<br>$160 \sim 170$ |
| Type 5 | $g_1$ = moderate activity group<br>$g_2$ = inactive group | $45 \sim 58$ |

**Table 3.7:** Identified harmful regions with the start and end positions of each region. The comparison types (from Table 3.4) are coded with different colours, which will be used in the next section.

### 3.5.2 Verifications

Similar to the verification in Section 3.4.3, the identified harmful regions in Table 3.7 will be verified in two different ways as well.

#### Verification by activity of *AluY* elements

Table 3.8 lists the percentage of mutations that occurred in each harmful region for each activity group in the bins in Figure 3.2. It can be seen that, in the low activity groups of each bin, there are more mutations in the harmful regions compared to other activity groups. Therefore, the mutations that occurred in the harmful regions may cause the lower activity levels of these elements. Oppositely, most mutations in the high activity groups are within the neutral regions, and none of them fell into the harmful regions, which indicates that mutations in the neutral regions do not have a large effect on the elements' activity levels. On the other hand, there are more mutations in regions 1 and 2 than in regions 3 and 4 in moderate and low activity groups, which indicates that mutations that occurred in region 1 and 2 have more effects on the mobile activities.

| | activity group | Harmful regions | | | | Neutral sites |
|---|---|---|---|---|---|---|
| | | Type 2 | Type 3 | Type 4 | Type 5 | |
| bin 1 | high | 0% | 0% | 0% | 0% | 100% |
| | moderate | 37.5% | 12.5% | 0% | 0% | 62.5% |
| | low | 37.5% | 0% | 12.5% | 0% | 62.5% |
| bin 2 | high | 0% | 0% | 0% | 0% | 92.3% |
| | moderate | 0% | 0% | 0% | 0% | 65% |
| | low | 20% | 3.3% | 10% | 0% | 20% |
| bin 3 | high | 0% | 0% | 0% | 0% | 58.3% |
| | moderate | 11.1% | 11.1% | 0% | 0% | 66.7% |
| | low | 11.8% | 5.9% | 11.8% | 5.9% | 64.7% |
| bin 4 | high | 0% | 0% | 0% | 0% | 60% |
| | moderate | 40% | 0% | 0% | 0% | 20% |
| | low | 0% | 0% | 0% | 0% | 33.3% |

**Table 3.8:** The percentage of mutations grouped by bins marked in Figure 3.2 over the total number of mutations in each activity group. Because of the overlap between regions, the percentages in the columns in the same row have overlapped parts too.

### Verification by *AluY* RNA secondary structure

The harmful regions in Table 3.7 are marked on the *Alu* RNA secondary structure in Figure 3.12.

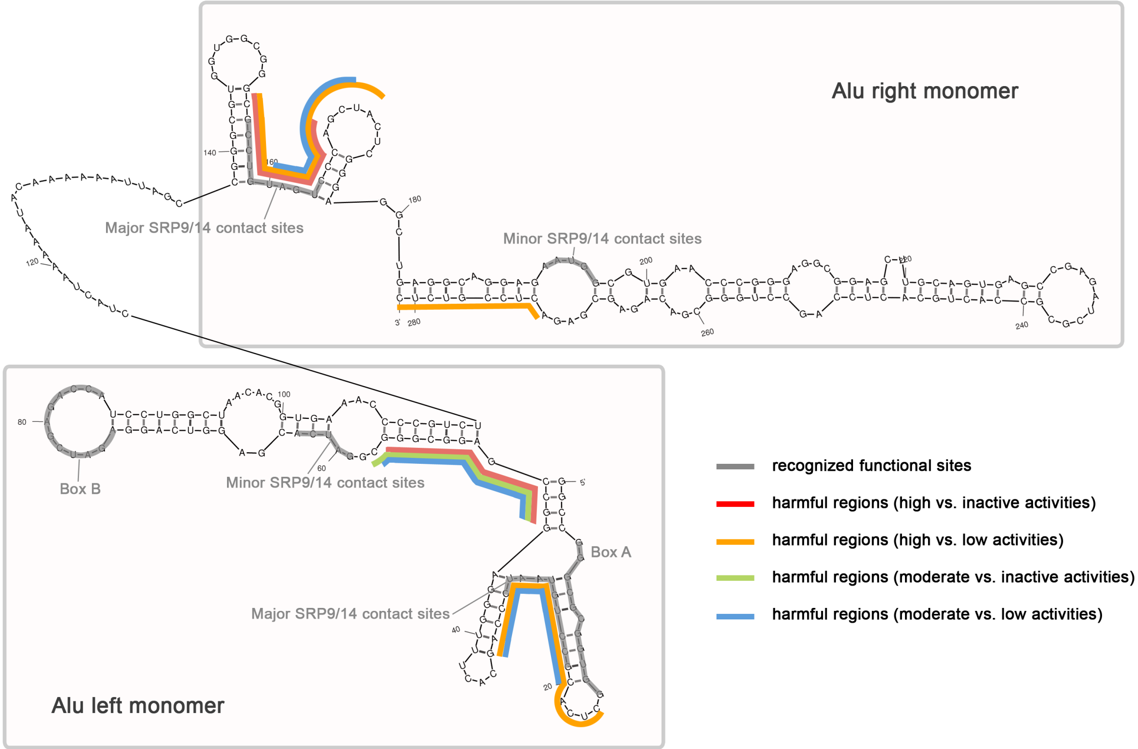

Alu right monomer
Major SRP9/14 contact sites
Minor SRP9/14 contact sites
140
160
180
200
120
260
240
280
3'
Alu left monomer
Box B
Minor SRP9/14 contact sites
Major SRP9/14 contact sites
Box A
80
100
60
40
20
5'
recognized functional sites
harmful regions (high vs. inactive activities)
harmful regions (high vs. low activities)
harmful regions (moderate vs. inactive activities)
harmful regions (moderate vs. low activities)

**Figure 3.12:** The secondary structure of an *AluY* RNA. The SRP contact sites and the A and B boxes are marked in Yellow. The harmful regions identified in Table 3.7 are marked in different colours along the structure as indicated in legend.

As visually indicated in the figure, these regions "cover" the two major SRP contact sites very well, with two other unknown regions that are recognized as harmful.

Another simulation that is the same as in Section 3.4.3 is developed to compare the coverage of the harmful regions and that of randomly generated regions. Running the simulation on the *AluY* harmful regions calculated in Table 3.7 for 10,000 iterations, Figure 3.13 is the distribution of the coverage of random regions, and the blue line on the figure shows the coverage of the harmful regions in Table 3.7.

Similarly, the probability is calculated as $P(Cov_{\text{harmR}} > Cov_{\text{randR}}) = 79.35\%$; that is, 79.35% of randomly generated regions have less coverage than the harmful regions identified by the group comparison method. Therefore, the harmful regions cover the *AluY* functional regions and this coverage is probably not by chance.

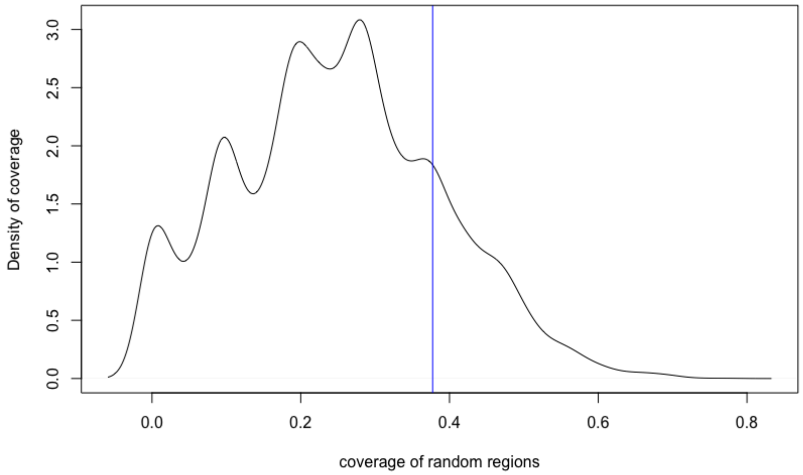


**Figure 3.13:** The distribution of the coverage of random generated regions. The blue vertical line is the coverage of the harmful regions.

## 3.6 Additional case studies

In Section 3.4, the computational method proposed to calculate the harmful mutation regions of TEs was applied to a specific TE family (the *AluY* subfamily) where the transpositional activity fractions of the elements in this family were quantified in [12]. The predicted regions of the *AluY* elements using this method were verified in Section 3.4.3, which also supports the correctness of the computational method proposed. In this section, this method will be applied to two other cases — the *Alu* family generally and the *LINE-1 (L1)* family, to identify the harmful mutation regions lying within their consensus sequences.

### 3.6.1 The *Alu* family

The work in [12] has systematically tested 89 representatives from many *Alu* families and subfamilies, and in Section 3.4, all the *AluY* elements have been examined. In this subsection, the computational method will be applied to a bigger set of elements of the *Alu* family, including 9 *AluJ*, 28 *AluS*, and 52 *AluY*, where their activity fractions are also quantified in [12].

There are a total of 89 elements ($N = 89$) and the length of the *Alu* consensus is $L = 312$. First, pairwise sequence alignment of each of the $N$ *Alu* elements is performed against the *Alu* consensus sequence from Repbase Update to get the mutation data for each element. Given the window size as $wsize = 10$, a mutation matrix, $M(N \times nw)$, is calculated as in Equation 3.3.1, where $nw = L - wsize + 1$. This mutation matrix, representing the number of mutations in each window, is plotted in the heat map as shown in Figure 3.14.

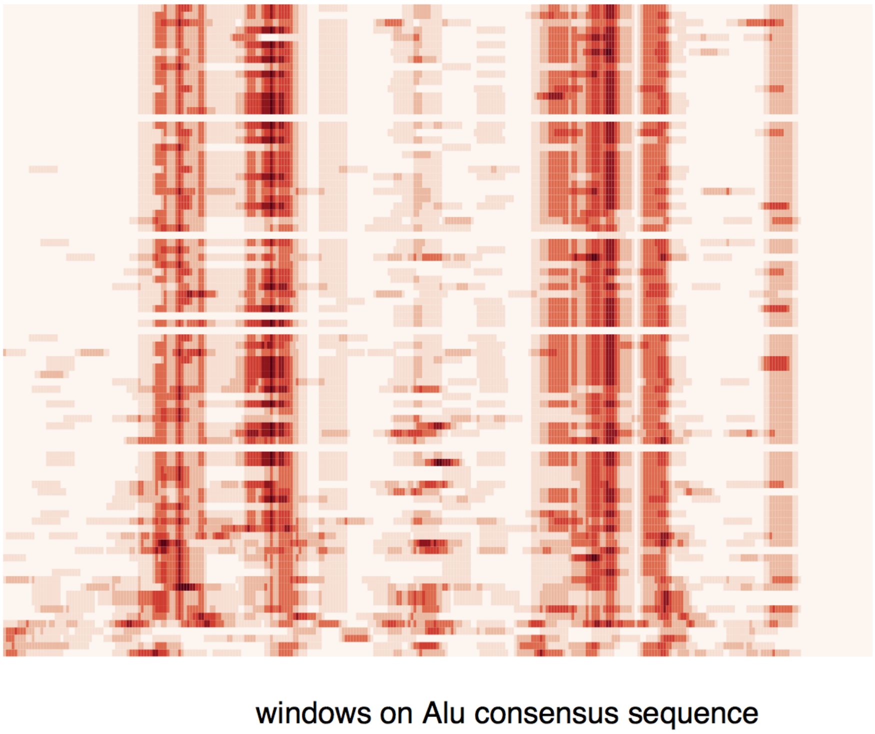


**Figure 3.14:** The number of mutations of the *Alu* elements in each window on the *Alu* consensus ($wsize = 10$), where the x-axis is all windows on the *Alu* consensus sequence and the y-axis is the *Alu* elements sorted by their activity fractions in descending order from the top to the bottom of the chart.

The observed Pearson's coefficient of correlation between the mutations in windows and the activities of the *Alu* elements are calculated using Equation 3.4.1 and is shown in Figure 3.15.

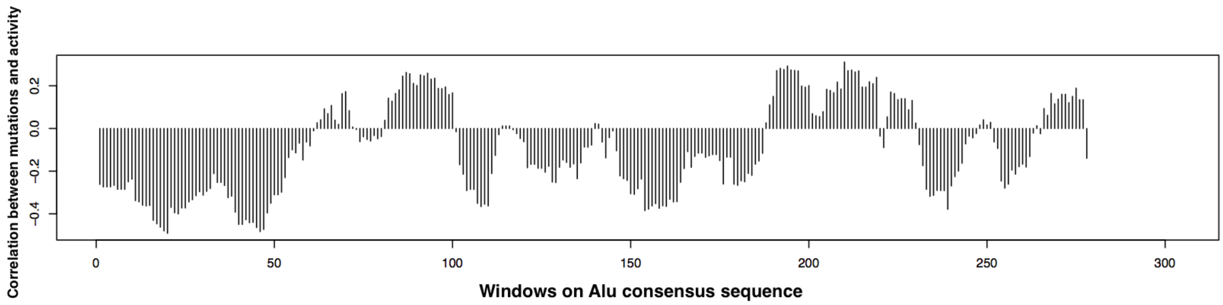


**Figure 3.15:** The Pearson's coefficients of correlation between the number of mutations in each window and the activity fractions of the *Alu* elements. The x-axis gives the windows in order on the *Alu* consensus.

Then the steps in Section 3.4.2 are followed to perform the statistical significance tests on the *Alu* data for $n = 10,000$, and the simulated correlations are calculated. The $p$-value and $q$-value are calculated for each window. The results are shown in Figure 3.16. Finally, the harmful regions in the *Alu* elements are calculated and listed in Table 3.9. In summary, the total length of the harmful mutation regions is 171 bp, which is 54.81% of the *Alu* consensus.

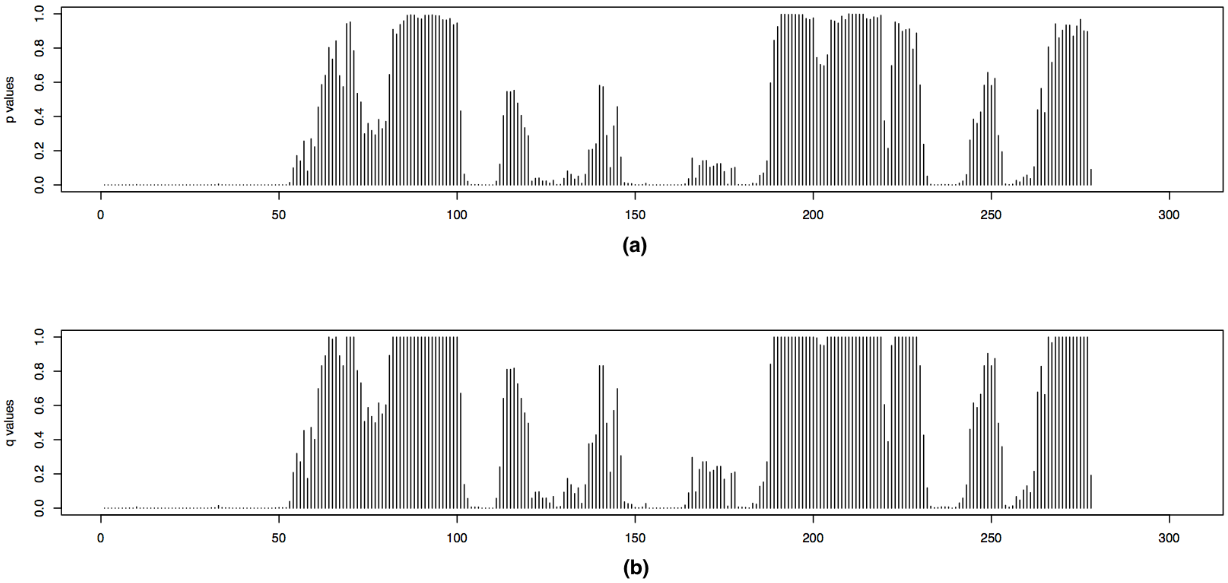


**Figure 3.16:** The $p$-values in (a), and $q$-values in (b) of the *Alu* elements. The x-axis gives the windows in order on the *Alu* consensus.

| region ID | regionStart | regionEnd | average $q$-value |
|---|---|---|---|
| 1 | 1 | 62 | 0.0010 |
| 2 | 104 | 119 | 0.0015 |
| 3 | 126 | 144 | 0.0181 |
| 4 | 147 | 173 | 0.0078 |
| 5 | 176 | 193 | 0.0102 |
| 6 | 233 | 250 | 0.0066 |
| 7 | 254 | 266 | 0.0176 |

**Table 3.9:** The harmful mutation regions in *Alu* elements calculated from correlation analysis ($\lambda = 0.05$).

The harmful mutations regions calculated in the *Alu* elements (Table 3.9) are then compared with the regions in the *AluY* elements (Table 3.2) as listed in Table 3.10. As the *AluY* is a subfamily of *Alu* elements, their consensus sequences are very similar with a percent identity of 88.38%. Moreover, by pairwise aligning the two consensus sequences, there are only two indels in the global alignment, one of length 2 in the *AluY* consensus sequence (positions 62 and 63), the other of length 1 in the *Alu* consensus sequence (position 134). The rareness of indels indicates that the two sequences are aligned very compactly, and their coordinates are almost consistent (off by 1 or 2 in some regions) in the two consensus sequences, so that the positions of harmful regions can be compared directly between the *Alu* and *AluY* families. Calculated from Table 3.10, the number of overlap positions of the harmful regions of the two TE families is 78.35% of the *AluY* harmful regions and 46.63% of the *Alu* harmful regions.

| *AluY* | | *Alu* | |
|---|---|---|---|
| regionStart | regionEnd | regionStart | regionEnd |
| 14 | 34 | 1 | 62 |
| 38 | 57 | 104 | 119 |
| 78 | 87 | 126 | 144 |
| 149 | 172 | 147 | 173 |
| 180 | 190 | 176 | 193 |
| 212 | 222 | 233 | 250 |
| | | 254 | 266 |

**Table 3.10:** Comparison between the harmful mutation regions in *AluY* and *Alu* elements.

### 3.6.2 The *L1* family

*L1* elements comprise 17% of the human genome [84]. An active *L1* is about 6 Kbp in length, and it has been estimated that an average diploid human genome contains approximately 80-100 active *L1s* [17]. In [17], 82 *L1* elements were cloned and each assayed for its ability to retrotranspose in cultured cells. These elements were then compared with the $L1_{RP}$ element to get their quantified retrotranspositional activity fractions in a similar fashion to [12].

Among the 82 *L1* elements in [17], $N = 77$ were retrieved where both their sequences and activity fractions were available. The length of the *L1* consensus sequence (accession no. L19092.1) is 6053 bp ($L = 6053$). Using the computational method in Section 3.4, a mutation matrix $M(N \times nw)$ is generated, and the observed Pearson's coefficient of correlation between the mutations in windows and the activities of the *L1* elements are calculated using Equation 3.4.1. Then the steps in Section 3.4.2 are followed to perform the statistical significance tests on the *L1* data for $n = 10,000$, and the simulated correlations are calculated. The $p$-value and $q$-value for each window are estimated, which gives the harmful regions in the *L1* elements. There are 201 harmful regions calculated from $N = 77$ *L1* elements, and the total length of these regions is 3500 bp in total, which covers 57.82% of the *L1* consensus sequence.

Notice that a large number (38 out of 77) of the *L1* elements have an activity fraction of 0%, and many (25 out of 77) are inactive with activity fractions between 0% to 5%. Since a large number of elements are inactive (more than 80% of the total number of elements), there are mutations occurring in almost every window that contribute to the change of an element's status from active to inactive, which results in most of the windows being identified as harmful windows. Therefore, the same calculation is performed again to only include the elements with non-zero activity fractions ($N = 39$). The observed Pearson's coefficient of correlation between the mutations in windows and the activities of the *L1* elements are calculated using Equation 3.4.1 and shown in Figure 3.17.

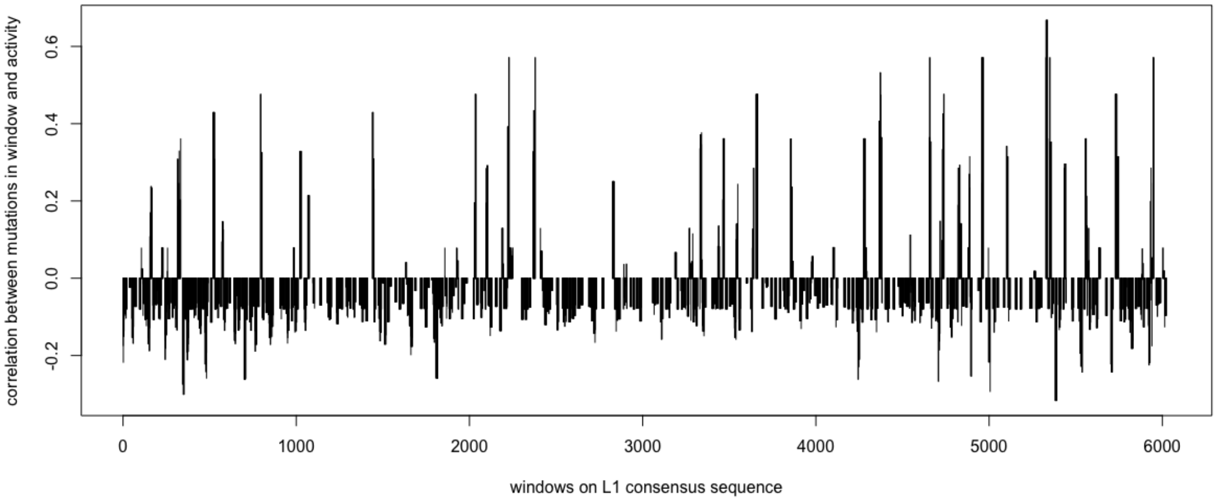


**Figure 3.17:** The Pearson's coefficients of correlation between the number of mutations in each window and the activity fractions of the *L1* elements. The x-axis gives the windows in order on the *L1* consensus.

The $p$-value and $q$-value for each window are shown in Figure 3.18, and the predicted harmful regions with $\lambda = 0.01$ are listed in Table 3.11 (here, $\lambda$ is decreased to 0.01 in order to fit the harmful regions into one page). The total length of these regions is 894 bp, which covers 14.77% of the *L1* consensus.

| region ID | regionStart | regionEnd | average $q$-value | region ID | regionStart | regionEnd | average $q$-value |
|---|---|---|---|---|---|---|---|
| 1 | 19 | 31 | $< 0.0001$ | 33 | 2945 | 2963 | $< 0.0001$ |
| 2 | 94 | 112 | $< 0.0001$ | 34 | 2983 | 2992 | $< 0.0001$ |
| 3 | 182 | 191 | $< 0.0001$ | 35 | 3147 | 3162 | $< 0.0001$ |
| 4 | 314 | 323 | $< 0.0001$ | 36 | 3198 | 3216 | $< 0.0001$ |
| 5 | 353 | 362 | $< 0.0001$ | 37 | 3247 | 3256 | $< 0.0001$ |
| 6 | 364 | 375 | $< 0.0001$ | 38 | 3299 | 3310 | $< 0.0001$ |
| 7 | 381 | 407 | $< 0.0001$ | 39 | 3330 | 3339 | $< 0.0001$ |
| 8 | 474 | 491 | 0.0041 | 40 | 3421 | 3431 | $< 0.0001$ |
| 9 | 505 | 523 | $< 0.0001$ | 41 | 3479 | 3497 | $< 0.0001$ |
| 10 | 530 | 547 | $< 0.0001$ | 42 | 3822 | 3846 | $< 0.0001$ |
| 11 | 588 | 603 | $< 0.0001$ | 43 | 3869 | 3887 | $< 0.0001$ |
| 12 | 661 | 671 | $< 0.0001$ | 44 | 4262 | 4283 | $< 0.0001$ |
| 13 | 698 | 707 | $< 0.0001$ | 45 | 4295 | 4311 | $< 0.0001$ |
| 14 | 854 | 864 | $< 0.0001$ | 46 | 4340 | 4349 | $< 0.0001$ |
| 15 | 925 | 943 | $< 0.0001$ | 47 | 4399 | 4424 | $< 0.0001$ |
| 16 | 1000 | 1020 | $< 0.0001$ | 48 | 4446 | 4464 | $< 0.0001$ |
| 17 | 1046 | 1061 | $< 0.0001$ | 49 | 4613 | 4631 | $< 0.0001$ |
| 18 | 1328 | 1342 | $< 0.0001$ | 50 | 4676 | 4685 | $< 0.0001$ |
| 19 | 1386 | 1398 | $< 0.0001$ | 51 | 4812 | 4827 | $< 0.0001$ |
| 20 | 1455 | 1467 | $< 0.0001$ | 52 | 4899 | 4910 | $< 0.0001$ |
| 21 | 1508 | 1517 | $< 0.0001$ | 53 | 5114 | 5131 | $< 0.0001$ |
| 22 | 1594 | 1612 | $< 0.0001$ | 54 | 5152 | 5170 | $< 0.0001$ |
| 23 | 1935 | 1947 | $< 0.0001$ | 55 | 5179 | 5197 | $< 0.0001$ |
| 24 | 2095 | 2104 | $< 0.0001$ | 56 | 5269 | 5279 | $< 0.0001$ |
| 25 | 2315 | 2330 | $< 0.0001$ | 57 | 5413 | 5424 | $< 0.0001$ |
| 26 | 2332 | 2348 | $< 0.0001$ | 58 | 5426 | 5441 | $< 0.0001$ |
| 27 | 2460 | 2478 | $< 0.0001$ | 59 | 5476 | 5488 | $< 0.0001$ |
| 28 | 2547 | 2556 | $< 0.0001$ | 60 | 5586 | 5596 | $< 0.0001$ |
| 29 | 2591 | 2600 | $< 0.0001$ | 61 | 5713 | 5724 | $< 0.0001$ |
| 30 | 2710 | 2722 | $< 0.0001$ | 62 | 5756 | 5765 | $< 0.0001$ |
| 31 | 2838 | 2855 | $< 0.0001$ | 63 | 5773 | 5787 | $< 0.0001$ |
| 32 | 2889 | 2899 | $< 0.0001$ | 64 | 5816 | 5829 | $< 0.0001$ |

**Table 3.11:** The harmful mutation regions in *L1* elements calculated from correlation analysis ($\lambda = 0.01$).

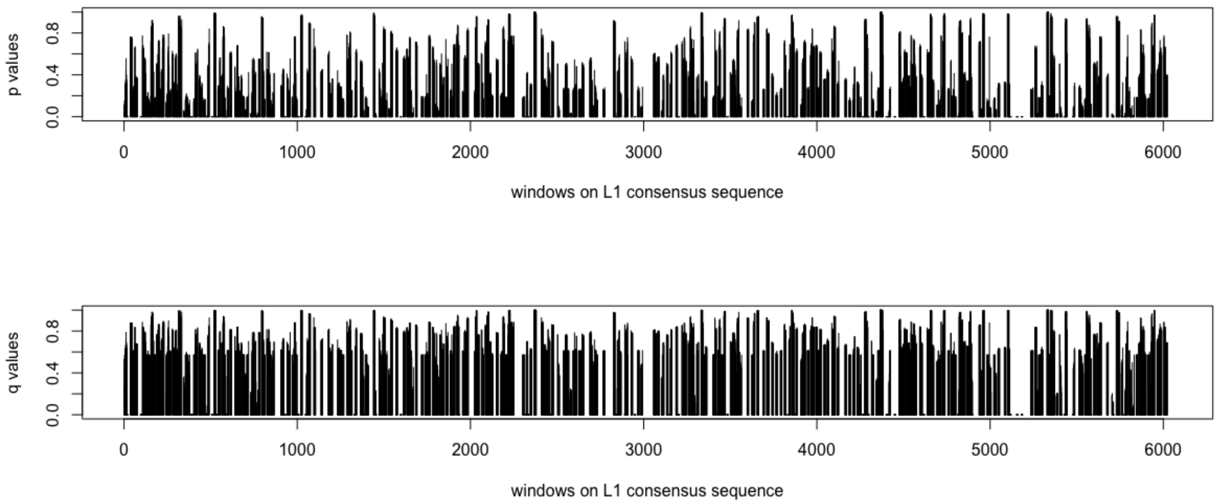


**Figure 3.18:** The $p$-values in (a), and $q$-values in (b) of the *L1* elements. The x-axis gives the windows in order on the *L1* consensus.

### 3.6.3 Perspective

The computational method was inspired by the observation on Figure 3.4 of the *AluY* family that mutations occurring in most of the windows have negative correlation with the mobile activities. In contrast, in Figure 3.15 of the *Alu* family and Figure 3.17 of the *L*1 family, there are a number of windows that have positive correlation with the mobile activity. A positive correlation indicates that the TE mobile activity increases as the number of mutations in a window increases. This might be because of the selection of the consensus sequence, as the mutations are calculated based on the consensus sequence which is assumed to be a "representative" element in that family, and the mutations in younger elements with higher activity relative to the consensus may seem to "increase" the elements' activity. Furthermore, as was previously mentioned, there might be many factors altering elements' activities simultaneously and mutations are only one factor among them. Thus, the reasons that some mutations have positive correlations to mobile activity might be caused by a combination of other unknown factors.

## 3.7 Conclusion

In this chapter, the importance of understanding major factors affecting the mobile activity of transposable elements is discussed. Two computational methods are developed to identify a list of loci such that mutations within those loci changes the activity levels in different ways, which could be used to assess the potential for developing various TE related diseases.

Method I was developed using Pearson's coefficient of correlation analysis and statistical significance tests, and method II uses activity group comparison with statistical significance tests. The results of both methods are verified by using elements with the same percent identity that can have various mobile activities due to mutations that did or did not occur within harmful regions. The identified harmful regions by both methods cover the *AluY* functional sites, which are important for its mobilization, also supporting the fact that mutations in these regions play a significant role in the transposition of active elements. Moreover, the simulation of random coverage proved that the identified harmful regions covering the functional regions is not picked up totally randomly by chance. The correlation analysis method is then applied to a bigger set of elements of the *Alu* family, and then to the *L1* family to identify their harmful mutation regions. To the best of our knowledge, this is the only work in this regard, and there has not been any other studies involving a similar data analysis.

A limitation is that the current data available does not enable the ability to assess the role of other factors influencing the activity of transposable elements, such as a transposition-selection equilibrium, a succession of burst and decay stages [22, 87], or chromatin structure, etc. Though the method was applied to the *Alu* and *L1* families in the human genome only (as they are highly active in the human genome, and the data for activity levels exist), the technique can be easily applied to other families of TEs as well in other organisms once activity levels and sequence data are developed.

Moreover, the harmful regions and their existence in active TEs can be used to inform the simulation of TEs in Chapter 7 in order to understand how TEs activate and deactivate throughout evolution and how genome evolve.

# Chapter 4

# The TE fragment model [1]

In this chapter, a formal model, called the *TE fragment model*, will be created describing notions such as TE families and fragments within genomic sequences. This is created because it is necessary to contribute a suitable foundation and standard nomenclature for future computational analysis; therefore, the models are formal yet compatible with the biological literature on TEs. Furthermore, multiple extensions and problems can be addressed on top of the same TE fragment model. However, this chapter only contains the model itself, which is principally just a set of formal definitions (or a standard markup syntax). Essentially, this is mainly for fixing terminology, which will be used in further chapters. Indeed, the model is used as a base notation within Chapters 5, 6, 7 of this thesis.

The TE fragment model consists of initial definitions of TEs, the set of TEs, and the set of TE fragments. It does not attempt to simulate or capture the molecular operations of TE movement (copying/cutting and pasting throughout a genome). However, the model describes the order and distance between TE fragments in genomic sequences by grouping homologous TEs together. At this level of abstraction, the model can be used to capture and calculate interruptions and their frequencies in a general way. Some of the definitions are also specialized for use when this model is associated with data from the prominent TE database Repbase Update [63] and a common TE identification tool RepeatMasker [131].

## 4.1 Model definition

The purpose of this section is to develop a formal model of TEs and fragments of TEs in order to describe the biological concepts and problems clearly. It will be the starting point in which multiple other problems will be studied.

As a large part of research on bioinformatics is based on the analysis of DNA or amino-acid sequences, a general sequence/string and other mathematical preliminaries will be briefly defined in Definition 8.

[1] The work in this chapter, as well as part of the next chapter, has been published in [58].

**Definition 8.** *Define several terms and notations:*

*An* alphabet $\Sigma$ *is an abstract and finite set of symbols (either nucleic acids or animo acids). A* string *is any finite sequence of characters over* $\Sigma$*. The* length *of* $s$*, denoted by* $|s|$*, is the number of characters in the string. The* empty word *is denoted by* $\lambda$ *and is of length* 0*. The* set of all strings *(including the empty word) over* $\Sigma$ *is denoted by* $\Sigma^*$*. Let* $s = s_1 s_2 \ldots s_n$ *be a string,* $s_i \in \Sigma$*,* $1 \leq i \leq n$*. Let* $j, k$ *satisfy* $1 \leq j \leq k \leq n$*, then the* substring *of* $s$ *which begins at the* $j$*th character and ends at the* $k$*th character is*

$$s(j,k) = s_j s_{j+1} \ldots s_k.$$

*Moreover,* $s(j) = s_j$*, the* $j$*th character alone.*

*Let* $s \in \Sigma^*$*, then* $frag(s)$ *is the set of all possible* fragments (substrings) *of* $s$*. That is,*

$$frag(s) = \{s(p,q) \mid 1 \leq p \leq q \leq |s|\} \cup \{\lambda\}.$$

*Extend this to sets of strings* $S \subseteq \Sigma^*$ *by*

$$frag(S) = \bigcup_{s \in S} frag(s).$$

*Given a set* $X$*,* $|X|$ *is the* number of elements *in* $X$*.*

When talking about a family of transposable elements, scientists usually are referring to a set of similar sequences that evolved from a single TE sequence. Therefore, a *family of transposable elements* is defined to simply be a set of strings (usually these strings will be similar to each other). A *set of TE families*, an *instance of a TE family* and the *consensus TE* are also defined in Definition 9.

**Definition 9.** *A* family of transposable elements (a TE family or a TE) $X$ *is a finite set of strings (usually similar to each other) with* $X \subseteq \Sigma^*$*.*

*An* instance of a TE family $X$ *is an element* $x \in X$*.*

*A* consensus TE *is a consensus sequence of the elements of* $X$*.*

*A* set of TEs $\chi$ *is a finite set of TE families. That is,* $\chi \subseteq 2^{\Sigma^*}$ *(*$2^{\Sigma^*}$ *is the set of all subsets of* $\Sigma^*$*; and so* $\chi$ *is some set of sets of strings),* $\chi$ *is finite and each element of* $\chi$ *is finite.*

Because of different biological contexts, it is also possible to interpret a set of TE families, $\chi$, in multiple ways depending on the purpose. For example, $\chi$ can be used as the set of all TE families and TE instances that are present in a single genome, or as the set of sequences collected in Repbase Update, or any set of sequences that are all similar to the consensus sequences in Repbase Update within a threshold.

Knowing that one TE family contains a number of instances and each instance itself is a string, now a *TE fragments set* can be defined. It is expected to see many fragments of TEs scattered throughout genomes as a family of TEs becomes fragmented within a genome and becomes interrupted by other families of TEs.

**Definition 10.** *Let $\chi$ be a finite set of TEs. Then $\bar{\chi} \subseteq 2^{\Sigma^*}$ is called a* TE fragments set*, if for each element $\bar{X} \in \bar{\chi}$, there exists $X \in \chi$ such that $\bar{X}$ is a subset of $frag(X)$.*

Thus, after picking a set of TEs, a TE fragments set is any set where each element consists of fragments of one TE family in the set of TEs (separate elements in $\bar{\chi}$ could contain fragments from different TEs). Then in principle, any number of fragment sets can be picked for one set of TEs. For example, if $\chi$ is picked to be the set of all TEs in the human genome, then a TE fragments set $\bar{\chi}$ can be the set where each element contains fragments of separate TEs of length at least 50 (in this case, $\bar{\chi} = \{\bar{X} \mid x \in \bar{X}$ implies $|x| = 50,\ \bar{X} \subseteq frag(X),\ X \in \chi$, for some $X\}$).

Although TE fragments sets are defined in a general way, it is also necessary to create a restriction to transposable elements that occur in present-day sequences. RepeatMasker (RM) [131] is a sophisticated program that uses precompiled repeat libraries to find copies of known repeats represented in the libraries (as introduced in Section 2.2.3). The program performs a similarity search on both the "+" and "-" DNA strands based on local alignments, then outputs masked genomic DNA and provides a tabular summary of repeat content (e.g., Table 2.4) detected in both DNA strands. In the following definitions, the general Definition 10 is going to be connected to the fragments reported by RepeatMasker.

The reason that RepeatMasker is chosen to connect with is because RepeatMasker is one of the most accurate tools in detecting TEs, and it is commonly used. It is also possible to replace RepeatMasker with other TE discovery tools or a combination of tools to achieve a higher accuracy in detecting TE fragments in different situations, as long as the reported detailed annotations of each TE fragment is consistent with those defined in Definition 12.

**Definition 11.** *Let $s$ be a string representing some genomic sequence and $\chi_s$ to be the set of TEs existing in $s$, then $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$ is a* RepeatMasker TE fragments set*, running the program with a set of consensus TEs $\chi_s$, against the genomic sequence $s$.*

*In other words, each element of $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$ is a subset of some element of $\bar{\chi}$, where only TE fragments detected by the RepeatMasker program are selected.*

For each TE fragment $z$ of some TE family $X$, and some $\bar{X} \in \bar{\chi}(s \overset{\text{RM}}{\longleftrightarrow} \chi_s)$, a tuple is associated with it in Definition 12. These attributes are referred to frequently in this thesis, and they are also consistent with the output of the RepeatMasker program.

**Definition 12.** *Given a genomic sequence $s$ and a set of TEs $\chi_s$, each* TE fragment *$z$ in each $\bar{X} \in \bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$ is a tuple:*

$$\begin{aligned} info(z) =& (genoName, genoStart, genoEnd, genoLeft, strand, \\ & TEName, TEFamily, TEClass, TEStart, TEEnd, TELeft). \end{aligned} \tag{4.1.1}$$

*The operator "." is used to access the attributes. For example, $z.TEname$ is the name of the TE to which* *fragment z belongs. The definition of each attribute is summarized in the list as follows (from [131]), and described in Example 3.*

$genoName$**:** *The name of the genomic sequence, where the fragment was detected.*

$genoStart$**:** *The start position of the fragment in the genomic sequence.*

$genoEnd$**:** *The end position of the fragment in the genomic sequence.*

$genoLeft$**:** *The opposite of the number of bases after the fragment in the genomic sequence.*

$strand$**:** *Relative orientation: "+" or "-".*

$TEName$**:** *The name of the TE to which the fragment belongs.*

$TEFamily$**:** *The name of the TE family to which the fragment belongs.*

$TEClass$**:** *The class of the TE to which the fragment belongs.*

$TEStart$**:** *The start position of the fragment in the TE consensus sequence to which the fragment belongs, if strand is "+", or the opposite number of bases after the fragment in the TE consensus sequence, if strand is "-".*

$TEEnd$**:** *The end position of the fragment in the TE consensus sequence.*

$TELeft$**:** *The opposite of the number of bases after the fragment in TE consensus sequence, if the strand is "+", or the start position of the fragment in the TE consensus sequence, if the strand is "-".*

Two TE fragments in the same TE family, $\bar{X} \in \bar{\chi}(s \overset{\text{RM}}{\longleftrightarrow} \chi_s)$, have the same *genoName*. Also, a TE fragment can be matched with either the consensus TE or the complement of the consensus TE in the database; however, in both cases, the "+" strand coordinate is used to represent the location where it occurs. Example 3 picks two TE fragments showing the meanings of their attributes visually with respect to a genomic sequence and TE consensus sequences.

**Example 3.** *Compare the human chromosome 1, denoted as s, against the library of human transposable elements in Repbase Update, denoted as $\chi_s$. The two TE fragments, $z_1$ and $z_2$, taken from two separate sets in the RepeatMasker TE fragments set, $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$, are as listed in Table 4.1 with their detailed attributes, and then visualized in Figure 4.1.*

| *Fragment* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEFamily* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | *chr1* | *377414* | *377536* | *-248578886* | *+* | *L1ME1* | *LINE* | *L1* | *6022* | *6151* | *-28* |
| $z_2$ | *chr1* | *388433* | *388732* | *-248567690* | *-* | *AluSg4* | *SINE* | *Alu* | *1* | *297* | *-13* |

**Table 4.1:** A table of two TE fragments on the human chromosome 1.

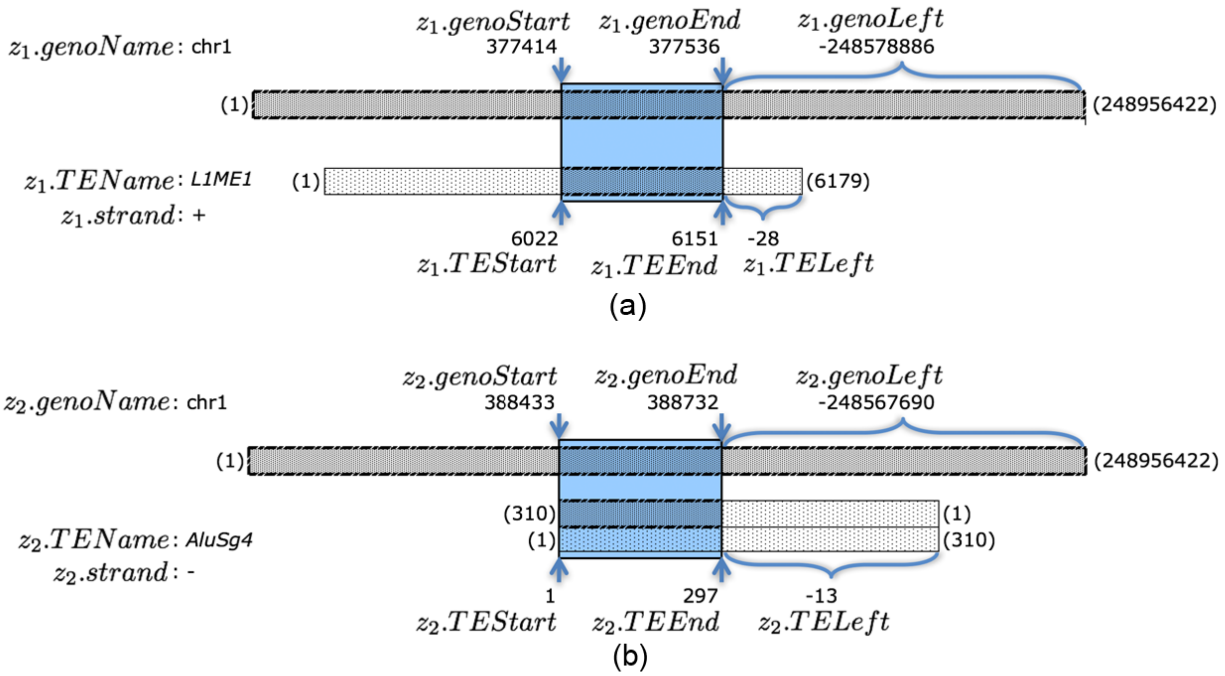


**Figure 4.1:** A conceptual visualization of the TE fragments (in blue shadow) in Table 4.1. (a) visualizes the fragment $z_1$; (b) visualizes the fragment $z_2$. Note that the lengths of the visualized sequences in the figure are not proportional to their actual lengths.

*Fig. 4.1 (a) shows a fragment of the element L1ME1 (L1 family, LINE type), which was detected on the "+" strand of chromosome 1, and Fig. 4.1 (b) shows a fragment of the TE family AluSg4 (Alu family, SINE type), which was detected on the "-" strand of chromosome 1. The two fragments are detected in different strands in the consensus TE as indicated in the figure.*

In the human genome, there exists a huge number of such TE fragments. Table 4.2 lists the the number of TE fragments in each chromosome based on the *hg38* assembly of the human genome.

Most of the present-day copies of TEs are detected by locally aligning the consensus TE sequences against a DNA sequence; thus the DNA sequence is fragmented into segments by the local aligned fragments. Some segments of the DNA sequence are detected as fragments of those TE families, while some are non-transposon DNA sequence. For much of the further analysis in this thesis, only the TE fragments and their positional relationships are of interest, therefore, the non-transposon fragments need to be separated from the TE fragments in the sequence. In essence, this DNA sequence can be "pruned" to present only the TE segments. This process is defined in Definition 13.

**Definition 13.** *Let $s$ be a genomic sequence, $\chi_s$ a fixed ordering of the set of TEs in $s$, where $\chi_s = \{X_1, \ldots, X_m\}$, and $\bar{\chi_s}$ is a set of TE fragments. Assume $s = w_0 z_1 w_1 z_2 w_2\ z_3 \ldots z_k w_k$, with $z_1, \ldots, z_k$ in sets in $\bar{\chi_s}$, and no fragment of $w_0, w_1, \ldots, w_k$ are in sets in $\bar{\chi_s}$. Then a* pruned sequence *$\bar{s}$ of $s$ with respect to $\bar{\chi_s}$ is*

|  | LINE | SINE | LTR | DNA | Total number of TE fragments |
|---|---|---|---|---|---|
| Chromosome 1 | 125,726 | 163,484 | 53,063 | 37,777 | 38,0815 |
| Chromosome 2 | 123,495 | 129,486 | 58,436 | 42,016 | 35,4251 |
| Chromosome 3 | 103,348 | 109,938 | 48,805 | 36,382 | 299,125 |
| Chromosome 4 | 94,966 | 88,618 | 53,582 | 28,930 | 266,743 |
| Chromosome 5 | 91,522 | 92,952 | 46,323 | 30,729 | 262,069 |
| Chromosome 6 | 86,266 | 88,382 | 41,256 | 28,779 | 245,227 |
| Chromosome 7 | 80,022 | 96,350 | 37,609 | 25,355 | 239,841 |
| Chromosome 8 | 75,052 | 79,829 | 37,601 | 22,890 | 215,836 |
| Chromosome 9 | 65,241 | 77,250 | 28,835 | 20,176 | 191,881 |
| Chromosome 10 | 66,746 | 83,330 | 29,978 | 22,813 | 203,247 |
| Chromosome 11 | 72,010 | 84,910 | 30,717 | 21,068 | 209,062 |
| Chromosome 12 | 69,862 | 89,305 | 33,236 | 23,632 | 216,462 |
| Chromosome 13 | 49,838 | 43,589 | 25,716 | 14,726 | 134,189 |
| Chromosome 14 | 46,101 | 54,782 | 22,786 | 14,627 | 138,620 |
| Chromosome 15 | 45,127 | 57,130 | 17,257 | 15,021 | 134,817 |
| Chromosome 16 | 40,469 | 73,211 | 20,245 | 15,435 | 149,522 |
| Chromosome 17 | 40,361 | 76,438 | 15,638 | 14,055 | 146,701 |
| Chromosome 18 | 38,290 | 37,472 | 18,574 | 12,187 | 106,782 |
| Chromosome 19 | 26,867 | 67,882 | 14,397 | 7,556 | 116,751 |
| Chromosome 20 | 34,414 | 46,496 | 16,606 | 13,499 | 111,151 |
| Chromosome 21 | 17,298 | 18,281 | 12,330 | 5,476 | 53,501 |
| Chromosome 22 | 19,467 | 36,238 | 7,888 | 5,471 | 69,128 |
| Chromosome X | 93,431 | 72,352 | 41,028 | 23,315 | 230,560 |
| Chromosome Y | 10,307 | 11,528 | 8,271 | 2,079 | 32,207 |
| Total number in *hg38* | 1,516,226 | 1,779,233 | 720,177 | 483,994 | 4,508,488 |

**Table 4.2:** A summary of the total number of TE fragments in the human genome. It also lists the TEs in four different types and summarized by chromosomes.

$$\bar{s} = \beta_0 z_1 \beta_1 z_2 \beta_2 \ldots z_k \beta_k, \text{ where } \beta_i = |w_i|, \; 0 \leq i \leq k. \tag{4.1.2}$$

*That is, in a pruned sequence, replace all non-TE fragments with their lengths.*

*In addition, from $\bar{s}$ and $\chi_s$, an* order-pruned sequence *$\bar{s}_o$ of $\bar{s}$ is defined as the string over $\{1, \ldots, m\}^*$,*

$$\bar{s}_o = j_1, j_2, \ldots, j_k, \text{ where } z_i \in X_{j_i}, \text{ for all } i, \; 1 \leq i \leq k. \tag{4.1.3}$$

*A pruned sequence can also be extended to a set of pruned sequences. Let $S = \{s_1, \ldots, s_N\}$, then the* set of pruned sequences *of $S$ is $\bar{S} = \{\bar{s_1}, \ldots, \bar{s_N}\}$.*

The reason for extending a pruned sequence to a set of pruned sequences is because a genome usually contains several chromosomes, which is a set of sequences. In the upcoming chapters, not only the pruned sequence/chromosome, but also the set of pruned sequences/chromosomes of a genome, will be used.

Example 4 is an example showing the pruned sequence and the order-pruned sequence of a given genomic sequence segmented by the RepeatMasker detected TE fragments.

**Example 4.** *A piece of the human chromosome 1 from position 33632576 to 33634148, $s$, is compared against the library of human transposable elements in Repbase Update, $\chi_s$, where $\chi_s = \{X_1, X_2, X_3, X_4, X_5\}$, and the names of the TE families $X_1, X_2, X_3, X_4, X_5$ are L2a, L2b, MIR3, MLT1J, MER63A. The TE fragments taken from the RepeatMasker TE fragments set, $\bar{\chi}(s \xleftrightarrow{RM} \chi_s)$, are as listed in Table 4.3 with their detailed attributes.*

| *Fragment* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | *chr1* | *33632576* | *33632977* | *-215617644* | + | *L2a* | *LINE* | *2941* | *3379* | *-47* |
| $z_2$ | *chr1* | *33633163* | *33633226* | *-215617395* | + | *L2b* | *LINE* | *3309* | *3374* | *-1* |
| $z_3$ | *chr1* | *33633332* | *33633389* | *-215617232* | - | *MIR3* | *SINE* | *-19* | *189* | *128* |
| $z_4$ | *chr1* | *33633467* | *33633769* | *-215616852* | - | *L2a* | *LINE* | *-2* | *3424* | *3074* |
| $z_5$ | *chr1* | *33633802* | *33633941* | *-215616680* | + | *MLT1J* | *LTR* | *262* | *389* | *-123* |
| $z_6$ | *chr1* | *33634011* | *33634148* | *-215616473* | - | *MER63A* | *DNA* | *-71* | *139* | *5* |

**Table 4.3:** A table of six TE fragments on the human chromosome 1.

*As in Definition 13, the genomic sequence $s$ is*

$$s = w_0 z_1 w_1 z_2 w_2 z_3 w_3 z_4 w_4 z_5 w_5 z_6 w_6.$$

*This sequence is visualized in Fig. 4.2, where each fragment in the sequence is also marked as the order and the name of the TE family, to which it belongs.*

*From Definition 13, the pruned sequence of $s$ is*

$$\bar{s} = \beta_0 z_1 \beta_1 z_2 \ldots z_6 \beta_6, \text{ where } \beta_i = |w_i|, \; 0 \leq i \leq 6,$$

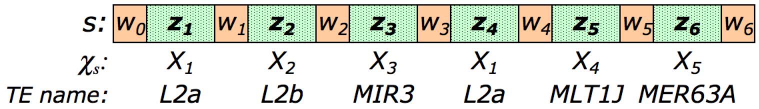


**Figure 4.2:** A conceptual visualization of a genomic sequence (the human chromosome 1 from position 33632576 to 33634148), with the RepeatMasker detected TE fragments in Table 4.3. The TE fragments $z_i$, where $i = 1, \ldots, 6$ in the sequence are also marked with the notation of TE families $X_j \in \chi_s$, where $j = 1, \ldots, 5$, and the names of the TE families to which they belong. Note that the lengths of the visualized sequences in the figure are not proportional to their actual lengths.

*and the order-pruned sequence of s is*

$$\bar{s_o} = 1,\ 2,\ 3,\ 1,\ 4,\ 5,\ \textit{where } z_i \in X_{j_i},\ \textit{for all } i,\ 1 \leq i \leq 6.$$

*That is, the pruned sequence of s is obtained by removing any position of s that is not a fragment of a TE, and instead replacing the part of the sequence between two TE fragments by its length. This is done so that the TE fragments themselves remain (as we are interested in studying them), but the only non-TE aspects of interest for our study of TEs is the length between fragments. The pruned sequences is describing exactly what is needed for our study of TEs.*

So far, some fundamental concepts associated with key biological terms have been defined, such as a family of transposable element, a TE fragment and a pruned sequence, which are also extended to sets within the TE fragment model. These terms were ambiguous and can refer to different concepts in biological literature before they are formally defined in this thesis. This work contributes in setting up a general formal model in using these terms in a consistent way.

## 4.2 Discussion

The TE fragment model serves as a theoretical foundation and helps to describe many TE problems clearly in a precise way. In the next two chapters, two other theoretical models will be proposed based on the TE fragment model: the sequential interruption model (Chapter 5) captures the interruptional activities between every pair of TEs; and the recursive interruption model (Chapter 6) further captures the nested nature of the interruptional activities of older TEs which cannot be represented by the interruptional matrix in the sequential interruption model. A TE simulation in Chapter 7 will also use these definitions.

The sequential and recursive interruption models are used to analyze the interruptional activities between the already annotated/detected TE fragments (as the input), and not to discover these fragments. Thus, in the TE fragment model in this chapter, it is reasonable to replace RepeatMasker with other TE discovery tools or a combination of tools in order to achieve a possibly higher accuracy of detected TE fragments, or to use, e.g., a *de novo* TE discovering tool instead. In essence, any TE detection tools can be used without affecting the definition of the theoretical models in the thesis. However, the accuracy of the chosen TE discovery tool will affect the results of our sequential and recursive interruption models, as it determines the input data of our system.

The notations and descriptions in our system are very detailed. This is because it is useful to have a formal model in order to be precise with further analysis, and for the connection to additional tools such as the linear ordering problem (Chapter 5) and stochastic context-free grammars (Chapter 6). However, this necessitates adding details to the model in order to properly capture all of the detail. For example, the annotations used by RepeatMasker described in Definition 12 are only a small subset of those provided by RepeatMasker and all of those discussed are needed for our models.

# Chapter 5

# The sequential interruption model and the linear ordering problem for sequential interruption analysis [1]

Newer TEs tend to interrupt older TEs, thereby fragmenting older TEs within the single linear sequence. By analyzing that sequence, it is possible to predict where and how often the transpositional interruptions occurred throughout evolution, which can be summarized into a so-called interruption matrix describing how many times each TE interrupts each other TE. A rearrangement of the matrix is attempted in order to minimize the penalty score that is calculated from the non-0's in the upper triangle [47]. This has the effect of predicting an order that those interruptional activity occurred, and further, potentially inferring the ages of these TEs, as discussed in Section 2.3.2. The ordering that was obtained from rearranging the interruption matrix in [47] agreed reasonably with published chronologies. This prediction is made entirely from a single genome, which can therefore be applied in many scenarios.

This interruptional activity can be represented from [47] using the model and algorithms proposed in this chapter. First, the *sequential interruptions* and the *interruption matrix* will be described and defined, then the method of estimating TE ages from [47] will be briefly discussed, and essentially the same matrix that they used for their estimation will be calculated using a specialized model called the *sequential interruptions model*. The model will then be connected to the *linear ordering problem*, a classic matrix optimization problem, whose methods can be used in solving the problem. This reduces the problem that the authors of [47] used to estimate TE ages to an existing well-studied problem (Section 5.3.2) from another area of computer science.

[1] Part of the work in this chapter has been published in [58].

## 5.1 Model of sequential interruptions

To analyze interruptional patterns, only the TE fragments and their relative positions in a genomic sequence are of interest. In the method in [47], an interruption is classified as occurring when one TE fragment is within a certain distance from a fragment on the left and a fragment on the right, where both are from the same TE family, and the two fragments are "close to" continuous within the consensus sequence of the TE family. This information is then compiled into a so-called *interruptional matrix*, giving an estimate on the number of times each TE family interrupted each other. The same analysis can be conducted using the TE fragment model (Chapter 4), and in particular, using pruned sequences of Definition 13 of Chapter 4. This definition provides all that is necessary to calculate the interruptional matrix. Before defining a sequential interruption in Definition 16, *continuous TE fragments* need to be defined first. Examples will also be given to clarify the definition.

**Definition 14.** *Let $s$ be a genomic sequence with a set of TEs $\chi_s$, TE fragment set $\bar{\chi}(s \xleftrightarrow{RM} \chi_s)$ and pruned sequence $\bar{s} = \beta_0 z_1 \beta_1 z_2 \ldots z_i \beta_i \ldots \beta_{j-1} z_j \beta_j \ldots z_k \beta_k$ as in Equation (4.1.2). Then two TE fragments $z_i$ and $z_j$ $(i < j)$ are in the same* transposon region*, if the non-transposon distances between every pair of TE fragments between $z_i$ and $z_j$ is within a threshold $E \in \mathbb{N}$ in the genomic sequence:*

$$\begin{cases} \beta_i \leq E \\ \beta_{i+1} \leq E \\ \vdots \\ \beta_{j-1} \leq E \\ \beta_j \leq E \end{cases}$$

**Definition 15.** *Let $s$ be a genomic sequence with a set of TEs $\chi_s$, TE fragment set $\bar{\chi}(s \xleftrightarrow{RM} \chi_s)$ and pruned sequence $\bar{s} = \beta_0 z_1 \beta_1 z_2 \ldots z_k \beta_k$ as in Equation (4.1.2). Then two TE fragments $z_i$ and $z_j$ $(i < j)$ are* continuous TE fragments*, $z_i \overset{\varepsilon,E}{\sim} z_j$, with distance $\varepsilon \in \mathbb{N}$ (in the consensus sequence) and distance $E \in \mathbb{N}$ (in the genomic sequence), if they satisfy the following conditions:*

1. *they belong to the same TE family:*

$$z_i.TEName = z_j.TEName;$$

2. *they are detected in the same strand:*

$$z_i.strand = z_j.strand;$$

3. *they belong to the same transposon region with the distance threshold $E$;*

*4. they are either separated or overlap* [2] *with a distance less than or equal to $\varepsilon$, with respect to the TE consensus sequence to which family they belong:*

$$\begin{cases} abs(z_j.TEStart - z_i.TEEnd) \leq \varepsilon, & \text{if } z_i \text{ and } z_j \text{ occur in the "+" strand,} \\ abs(z_i.TEStart - z_j.TEEnd) \leq \varepsilon, & \text{if } z_i \text{ and } z_j \text{ occur in the "-" strand.} \end{cases}$$

Notice that continuous TE fragments are not necessarily beside each other in the genomic sequence (there could be other fragments between them), as long as they belong to the same transposon region. Some continuous TE fragments appear to have an overlap of duplication of a portion of the transposon. This is because RepeatMasker [131] often extends the homology match of both fragments to the TE consensus sequence by several base pairs.

**Definition 16.** *Given a genomic sequence $s$, a set of TEs with a fixed ordering on its elements $\chi_s = \{X_1, X_2, \ldots, X_m\}$, a distance $\varepsilon \in \mathbb{N}$ in a TE consensus sequence, a distance $E \in \mathbb{N}$ in a genomic sequence as in Definition 15, as well as a pruned sequence $\bar{s} = \beta_0 z_1 \beta_1 z_2 \ldots z_k \beta_k$, as in Equation (4.1.2). The* sequential interruptions *of $X_j$ by $X_i$ are defined as*

$$\Xi_s^{\varepsilon,E}(X_i, X_j) = \{k \mid z_k \in \bar{X}_i, z_{k-\eta_1}, z_{k+\eta_2} \in \bar{X}_j, z_{k-\eta_1} \overset{\varepsilon,E}{\sim} z_{k+\eta_2}, \text{ and } \eta_1, \eta_2 \in \mathbb{N}\}. \tag{5.1.1}$$

*Thus, the TE family $X_i$ is called* interrupter*, and the TE family $X_j$ is called* interruptee.

*Intuitively, $\Xi_s^{\varepsilon,E}$ gives the set of all positions of $X_i$ interrupting $X_j$ in the genomic sequence $s$. This matches the (more informal) description of the calculation of an interruption matrix in [47].*

Example 5 illustrates how an interruption is identified in a genomic sequence.

**Example 5.** *Table 5.1 is a list of three TE fragments from chromosome 1 position 448062 to 448403 taken from the RepeatMasker TE fragments set, $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$. The fragments belong to two TE families: $L1MD3$ and $AluYc$, where $L1MD3$ is denoted as $X_1$ and $AluYc$ is denoted as $X_2$.*

| *Fragment* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | *chr1* | *448062* | *448139* | *-248802482* | *+* | *L1MD3* | *LINE* | *6988* | *7068* | *-814* |
| $z_2$ | *chr1* | *448150* | *448328* | *-248802293* | *+* | *AluYc* | *SINE* | *122* | *299* | *0* |
| $z_3$ | *chr1* | *448332* | *448403* | *-248802218* | *+* | *L1MD3* | *LINE* | *7068* | *7148* | *-847* |

**Table 5.1:** An example of an interruption.

*As in Definition 13, the genomic sequence $s$ is*

$$s = w_0 z_1 w_1 z_2 w_2 z_3 w_3,$$

[2] The amount that separates them or the amount they overlap is calculated using the *abs*() function to get the absolute value.

*which gives the pruned sequence* $\bar{s}$ *as*

$$s = \beta_0 z_1 \beta_1 z_2 \beta_2 z_3 \beta_3,$$

*where* $z_1, z_3 \in \bar{X}_1$*,* $z_2 \in \bar{X}_2$ $\bar{X}_1, \bar{X}_2 \in \bar{\chi}(s \xleftrightarrow{RM} \chi_s)$*,* $\beta_0, \ldots, \beta_3 \in \mathbb{N}$*.*

*Table 5.2 shows the distances between these TE fragments. Given the distance threshold in genomic sequence as* $E = 20$ *bp (*$\beta_1 \leq E$ *and* $\beta_2 \leq E$*), and the distance threshold in transposon sequence as* $\varepsilon = 10$ *bp (*$0$ *bp* $\leq \varepsilon$*), then* $z_1$ *and* $z_3$ *are continuous TE fragments; that is,* $z_1 \overset{\varepsilon,E}{\sim} z_3$*. These three TE fragments are identified as an interruption, where the interrupter is* $X_2$ *(*$z_2 \in X_2$*), AluYc, and the interruptee is* $X_1$ *(*$z_1, z_3 \in X_1$*), L1MD3.*

| | *distance in genomic sequence* | *distance in transposon sequence* |
|---|---|---|
| $\beta_1$*:*<br>*distance between* $z_1$ *and* $z_2$<br>*in genomic sequence* | $z_2$*.genoStart -* $z_1$*.genoEnd*<br>*= 448150 - 448139*<br>*=11 bp* $\leq E$ | |
| $\beta_2$*:*<br>*distance between* $z_2$ *and* $z_3$<br>*in genomic sequence* | $z_3$*.genoStart -* $z_2$*.genoEnd*<br>*= 448332 - 448328*<br>*= 4 bp* $\leq E$ | |
| $z_1$ *and* $z_3$ | | $z_3$*.TEStart -* $z_1$*.TEEnd*<br>*= 7068 - 7068*<br>*= 0 bp* $\leq \varepsilon$ |

**Table 5.2:** An example of distances between three TE fragments identified as an interruption in Example 5, where the interrupter is $z_2$ (*AluYc*) and the interruptee is $z_1$ and $z_3$ (*L1MD3*).

Using this definition, a large number of sequential interruptions were detected in the human genome (*hg38*) as listed in Table 5.3, using the distance in the genomic sequence $E = 20$ *bp*, and the distance in the transposon sequence $\varepsilon = 10$ *bp*. In this case, the $E$ and $\varepsilon$ are chosen to make the detection of interruptions very strict compared to [47], in which 500 *bp* of "nonrepeat-masked sequence" (this is essentially the sum of $\beta$s in the same transposon region) was used to detect potential transposon clusters and an increasing "repeat index" (this is essentially the $\varepsilon$) up to 50% of the length of the shorter of the two fragments was used to detect continuous fragments.

The frequencies with which the interruptions between different families of TEs occur in the sequence can also show the activity of these TE families. Therefore, the abundance of interruptions is defined to capture the frequencies of interruptions in Definition 17 to represent interruptions in a general way.

**Definition 17.** *Given a genomic sequence s, a set of TEs with a fixed ordering on its elements* $\chi_s = \{X_1, X_2, \ldots, X_m\}$*, the* abundance that $X_i$ interrupts $X_j$ *in s,* $1 \leq i \leq m$*,* $1 \leq j \leq m$*, is defined as the total number of times that* $X_i$ *interrupts* $X_j$*. The abundance is equal to*

$$|\Xi_s^{\varepsilon,E}(X_i, X_j)|.$$

| | Total number of interruptions | Number of interruptions per Mbp |
|---|---|---|
| Chromosome 1 | 33,186 | 133 |
| Chromosome 2 | 28,985 | 120 |
| Chromosome 3 | 23,620 | 119 |
| Chromosome 4 | 20,214 | 106 |
| Chromosome 5 | 20,960 | 115 |
| Chromosome 6 | 19,616 | 115 |
| Chromosome 7 | 20,965 | 132 |
| Chromosome 8 | 17,016 | 117 |
| Chromosome 9 | 16,424 | 119 |
| Chromosome 10 | 17,316 | 129 |
| Chromosome 11 | 16,860 | 125 |
| Chromosome 12 | 18,500 | 139 |
| Chromosome 13 | 10,470 | 92 |
| Chromosome 14 | 12,031 | 112 |
| Chromosome 15 | 12,271 | 120 |
| Chromosome 16 | 13,741 | 152 |
| Chromosome 17 | 14,567 | 175 |
| Chromosome 18 | 8,422 | 105 |
| Chromosome 19 | 13,235 | 226 |
| Chromosome 20 | 9,720 | 151 |
| Chromosome 21 | 4,566 | 98 |
| Chromosome 22 | 6,922 | 136 |
| Chromosome X | 21,400 | 137 |
| Chromosome Y | 2,636 | 46 |
| Human Genome | 383,643 | 124 |

**Table 5.3:** A summary of the number of sequential interruptions ($E = 20$ *bp*, $\varepsilon = 10$ *bp*) detected in the human genome ($hg38$), summarized by chromosomes.

*For a genome $S$ that has chromosomes $s_1, s_2, \ldots, s_N$, the abundance that $X_i$ interrupts $X_j$ for all chromosomes are added up, which is*

$$|\Xi_S^{\varepsilon,E}(X_i, X_j)| = \sum_{n=1}^{N} |\Xi_{s_n}^{\varepsilon,E}(X_i, X_j)|.$$

*The* interruption array *of $X_i$ on $S$, for $1 \leq i \leq m$, is the array*

$$M(i) = [|\Xi_S^{\varepsilon,E}(X_i, X_j)|]_{j=1,\ldots,m}.$$

*The* interruption matrix *on $S$ is an $m \times m$ matrix defined by*

$$M = [|\Xi_S^{\varepsilon,E}(X_i, X_j)|]_{\substack{i=1,\ldots,m \\ j=1,\ldots,m}}.$$

The interruption array and matrix are different ways to structure the abundance by using the ordering on the elements in $\chi_s$. This interruption matrix was calculated in a similar fashion as the interruptional matrix of [47].

Example 6 illustrates how to apply the model of sequential interruptions in a real situation to find sequential interruptions, and calculate the interruptional matrix.

**Example 6.** *Table 5.4 is a list of five TE fragments from chromosome 1 position 448062 to 449273 taken from the RepeatMasker TE fragments set, $\bar{\chi}(s \xleftrightarrow{RM} \chi_s)$, in Example 3. The five fragments belong to three TE families: $X_1$, $X_2$, and $X_3$, where the names of the families $X_1, X_2, X_3$ are $L1MD3$, $AluYc$, $AluSq$.*

| *Fragment* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | *chr1* | *448062* | *448139* | *-248802482* | + | *L1MD3* | *LINE* | *6988* | *7068* | *-814* |
| $z_2$ | *chr1* | *448150* | *448328* | *-248802293* | + | *AluYc* | *SINE* | *122* | *299* | *0* |
| $z_3$ | *chr1* | *448332* | *448403* | *-248802218* | + | *L1MD3* | *LINE* | *7068* | *7148* | *-847* |
| $z_4$ | *chr1* | *448403* | *448710* | *-248801911* | + | *AluSq* | *SINE* | *1* | *313* | *0* |
| $z_5$ | *chr1* | *448710* | *449273* | *-248801348* | + | *L1MD3* | *LINE* | *7149* | *7753* | *-242* |

**Table 5.4:** An example of sequential interruptions on chromosome 1.

*As in Definition 13, the genomic sequence $s$ is*

$$s = w_0 z_1 w_1 z_2 w_2 z_3 w_3 z_4 w_4 z_5 w_5,$$

*as visualized in Figure 5.1.*

*The pruned sequence of $s$ is*

$$\bar{s} = \beta_0 z_1 \beta_1 z_2 \beta_2 z_3 \beta_3 z_4 \beta_4 z_5 \beta_5,$$

*where $z_1, z_3, z_5 \in \bar{X_1}$, $z_2 \in \bar{X_2}$, $z_4 \in \bar{X_3}$, $\bar{X_1}, \bar{X_2}, \bar{X_3} \in \bar{\chi}(s \xleftrightarrow{RM} \chi_s)$, $\beta_0, \ldots, \beta_5 \in \mathbb{N}$, and $z_1 \overset{\varepsilon,E}{\sim} z_3$, $z_3 \overset{\varepsilon,E}{\sim} z_5$, as shown in Figure 5.1.*

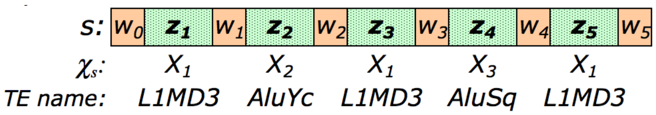


**Figure 5.1:** A conceptual visualization of a genomic sequence (the human chromosome 1 from position 448062 to 449273), with the RepeatMasker detected TE fragments in Table 5.4. Note that the lengths of the visualized sequences in the figure are not proportional to their actual lengths.

*It is possible to see that there are two potential interruptions in $s$: an instance of $X_1$ is present in the sequence, then an instance of $X_2$ and an instance of $X_3$ potentially inserted themselves into the instance of $X_1$ to break it into three segments $z_1$, $z_3$, and $z_5$; that is, $|\Xi_s^{\varepsilon,E}(X_2, X_1)| = 1$ and $|\Xi_s^{\varepsilon,E}(X_3, X_1)| = 1$, where $E = 20$ bp and $\varepsilon = 10$ bp.*

*Given a fixed order of the set of TEs as*

$$\chi_s = \{\ldots, X_1, \ldots, X_2, \ldots, X_3, \ldots\},$$

*the interruption matrix showing only the rows and columns of these TEs is*

$$M = \begin{bmatrix} & \vdots & & \vdots & & \vdots & \\ \ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots \\ & \vdots & & \vdots & & \vdots & \\ \ldots & 1 & \ldots & 0 & \ldots & 0 & \ldots \\ & \vdots & & \vdots & & \vdots & \\ \ldots & 1 & \ldots & 0 & \ldots & 0 & \ldots \\ & \vdots & & \vdots & & \vdots & \end{bmatrix}.$$

*From the analysis on these sequential interruptions, it is reasonable to predict the age of L1MD3 as being older than both AluYc and AluSq, but this provides no clue as to which one of AluYc and AluSq is older, because which one of the two independent interruptions occurred first is unknown.*

The notions in this section transformed the interruptional matrix construction similar to the method described in prose in [47] into a formal model, which is more clear, and can also be used for other purposes, such as the study of recursive patterns, as in Chapter 6.

## 5.2 The method of estimating TE ages from [47]

The interruptional analysis done in [47] was performed using the interruptions between TEs, then rearranged the TEs using a so-called "repositioning" method, in such a way that they hypothesized would order them in the chronological order of TEs (from the oldest to youngest). The method tried to rearrange the interruptional matrix in order to minimize the penalty score that is calculated from the non-0's in the upper triangle of the matrix. This is done because ideally, the matrix that achieves a lowest penalty score corresponds to reordering these TEs families, from those that get interrupted most while getting interrupting the least, to those that interrupt most while getting interrupted least. In this way, in a hypothetical matrix, the TE families are arranged in a predictive chronological order of decreasing age (from oldest to youngest). The interruptional analysis is done as follows:

First, an interruptional matrix was calculated comparable to an interruption matrix in Definition 17, whose rows/columns correspond to a TE ordering, which counts the number of interruptions between each pair of TE families. A method, called the repositioning method, is used to rearrange the TE ordering by repositioning TEs in the ordering to minimize the penalty score. The penalty score is defined as the summation of nonzero entries in the upper triangle of the interruption matrix (the nonzero entries were transformed by a continuous function, $\tau(x) = x$ for $x \leq 3$ and $\tau(x) = 3 + \log(x + 1)/4$ for $x > 3$, before summation). The repositioning method starts at the first TE in the ordering, and moves it to the position that results in the greatest decrease in the penalty score, then rearrange the matrix by placing this TE to its new position. In the rearranged matrix, the first TE is now different, the algorithm then checks the first TE in the new ordering again. When repositioning of the first TE no longer results in a decrease in the penalty score, the algorithm checks and moves to the second TE in the matrix, until the second TE can no longer be repositioned to decrease the penalty score. Then it checks the third TE, and so on until it reaches the last TE. Then the same procedure is iterated multiple times (100,000 times), and every time starts with a random initial ordering of TEs (the initial ordering affects the best solution that the algorithm can find). The final position of each TE in the ordering is represented as the median of the distribution of its positions across all iterations. Note that in some sense, the complexity of the repositioning method is $O(n!)$, where $n$ is the number of TE families. Although [47] did not report how long the repositioning method took to compute the problem, it is likely quite long as they needed to reduce the size of the matrix to a minority fraction of the set of the human TEs.

The formal model of sequential interruptions from Section 5.1 calculates an interruption matrix in essentially the same way as the interruptional matrix in [47]. Next, this matrix will be mapped to a well-studied matrix optimization problem — the linear ordering problem — which rearranges a matrix similarly to the repositioning approach in [47] to predict a potential chronology of these TE families.

## 5.3 Linear ordering problem for sequential interruptions analysis

First of all, a set of matrix rearrangement operations in linear algebra will be examined in the next subsection, in order to describe and compute the linear ordering problem.

### 5.3.1 Preliminaries

When rearranging some objects or values, the act of rearrangement is a permutation as defined in Definition 18.

**Definition 18.** *A* permutation $\pi$ *is a bijective function from* $\{1, 2, \dots, n\}$ *to itself. It will be denoted by an* $n$*-tuple where the number at position* $i$ *is* $\pi(i)$*.*

*A* permutation matrix *is a square* $n \times n$ *binary matrix that has exactly one entry 1 in each row and each column and 0s elsewhere. Specifically, the permutation matrix of a permutation* $\pi$ *is a matrix* $P_\pi$ *whose entries are all 0 except that in row* $i$*, where the entry at column* $\pi(i)$ *equals 1.*

*Each such matrix represents a specific permutation of* $n$ *elements and, when multiplying another* $n \times n$ *matrix* $A$ *with* $P$ *from the left, it permutes the rows of* $A$*. Further, multiplying* $A$ *with the transpose of* $P$*,* $P^T$*, from the right, permutes the columns of* $A$*.*

Example 7 illustrates a permutation of an ordering and its permutation matrix, as well as discussing how to permute a square matrix using this permutation.

**Example 7.** *For an ordering of* $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}$*, a permutation could be* $\pi(1) = 1$*,* $\pi(2) = 4$*,* $\pi(3) = 2$*,* $\pi(4) = 5$*,* $\pi(5) = 3$*, which is written as* $\pi = \begin{pmatrix} 1 & 4 & 2 & 5 & 3 \end{pmatrix}$*.*

*The permutation matrix* $P_\pi$ *of* $\pi$ *is*

$$P_\pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

*As in Definition 18, any square matrix with* $n$ *rows and columns can be rearranged by a permutation of* $n$ *elements, using its permutation matrix.*

*Given a square matrix,*

$$A = \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \\ 21 & 22 & 23 & 24 & 25 \\ 31 & 32 & 33 & 34 & 35 \\ 41 & 42 & 43 & 44 & 45 \\ 51 & 52 & 53 & 54 & 55 \end{bmatrix},$$

*multiplying $A$ with $P_\pi$ from the left permutes the rows of $A$:*

$$P_\pi \times A = \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \\ 41 & 42 & 43 & 44 & 45 \\ 21 & 22 & 23 & 24 & 25 \\ 51 & 52 & 53 & 54 & 55 \\ 31 & 32 & 33 & 34 & 35 \end{bmatrix},$$

*while multiplying $A$ with $P_\pi^T$ from the right permutes the columns of $A$:*

$$A \times P_\pi^T = \begin{bmatrix} 11 & 14 & 12 & 15 & 13 \\ 21 & 24 & 22 & 25 & 23 \\ 31 & 34 & 32 & 35 & 33 \\ 41 & 44 & 42 & 45 & 43 \\ 51 & 54 & 52 & 55 & 53 \end{bmatrix}.$$

*Therefore, $P_\pi \times A \times P_\pi^T$ permutes $A$ with the permutation $\pi = \begin{pmatrix} 1 & 4 & 2 & 3 & 5 \end{pmatrix}$ :*

$$P_\pi \times A \times P_\pi^T = \begin{bmatrix} 11 & 14 & 12 & 15 & 13 \\ 41 & 44 & 42 & 45 & 43 \\ 21 & 24 & 22 & 25 & 23 \\ 51 & 54 & 52 & 55 & 53 \\ 31 & 34 & 32 & 35 & 33 \end{bmatrix}.$$

For each given ordering of $n$ elements, there exists $n!$ possible permutations. Permuting a square matrix in this fashion is the operation used in the linear ordering problem in Section 5.3.2.

### 5.3.2 Linear Ordering Problem

The linear ordering problem is one of the classical combinatorial optimization problems. It was classified as $\mathcal{NP}$-hard in 1979 by Garey and Johnson [45]. This problem is defined using either a graph problem or a matrix problem. The matrix problem is described in [124] as:

Given an $m \times m$ matrix $C$, the *linear ordering problem* is the problem of finding a permutation $\pi$ of the column and row indices $\{1, \cdots, m\}$, such that the value

$$f(\pi) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} C^{(\pi(i),\pi(j))} \tag{5.3.1}$$

is maximized. In other words, the goal is to find a permutation of the columns and rows of $C$ such that the sum of the elements in the upper triangle is maximized.

Analogously, the goal of the TE sequential interruptional analysis is to find a permutation of TE ordering that maximizes the sum of upper triangle of the interruption matrix in Definition 17. The sequential interruption analysis can be described in terms of the linear ordering problem as follows.

Given a set of genomic sequences, $S$, a set of TEs with a fixed ordering on its elements,

$$\chi_s = \{X_1, X_2, \ldots, X_m\},$$

and an interruption matrix of $\chi_s$ on $S$,

$$IM = [|\Xi_S^{\varepsilon,E}(X_i, X_j)|]_{\substack{i=1,\ldots,m\\ j=1,\ldots,m}},$$

the problem is to find a permutation $\pi$ of $\chi_s$, corresponding to the column and row indices $\{1, \cdots, m\}$, such that the value

$$f(\pi) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} IM^{(\pi(i),\pi(j))}, \tag{5.3.2}$$

is maximized. Note that in the LOP, the permutation $\pi$ provides the ordering of both the columns and the rows.

The resultant permutation of $\chi_s$ corresponds to a hypothetical chronological order of TE families in $\chi_s$ of increasing age, as it optimizes essentially the same function used to estimate the ages (in [47], they attempt to find a permutation, corresponding to TE families of decreasing age that minimizes the summation of nonzero entries in the upper triangle of the matrix). The resulting matrix whose rows and columns are rearranged, will have the following features (as shown in Figure 5.2):

1. Each item in the matrix records the number of interruptions that an interrupter (on the vertical axis) has inserted itself into an interruptee (on the horizontal axis);

2. The order of TEs on the vertical axis (from top to bottom) is the same as the order of TEs on the horizontal axis (from left to right), which is arranged in predicted chronological order of increasing in age (from youngest to oldest).

3. The matrix can be divided into four portions:

    - the top-left portion of the matrix represents young TE families interrupting young TE families;

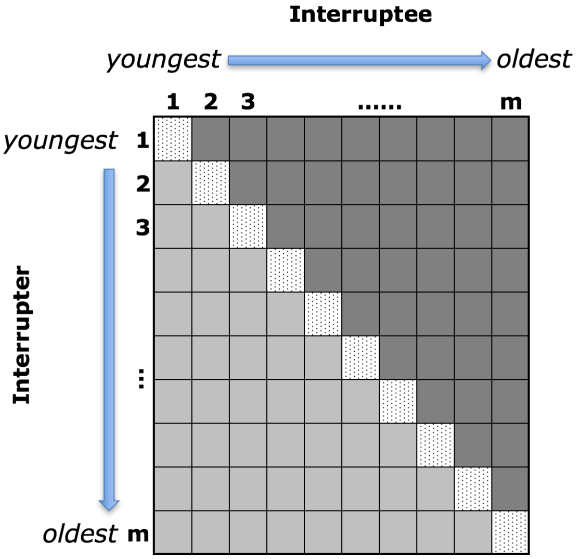


**Figure 5.2:** A conceptual diagram of the permuted interruptional matrix with a maximum sum of the upper triangle. The row and column of the matrix correspond to the resultant permutation of the TE ordering, which is a hypothetical chronological order of TEs of increasing in age.

- the top-right portion of the matrix represents young TE families interrupting old TE families;
- the bottom-left portion of the matrix represents old TE families interrupting young TE families;
- the bottom-right portion of the matrix represents old TE families interrupting old TE families.

4. Ideally, the lower triangle region of the matrix (light grey in Figure 5.2), corresponds to older TE families interrupting younger TE families, should be mainly populated by zeros, meaning that there are no interruptions. Non-zeros in this region might occur possibly because of defragmentation errors, or other mutation events that give the appearance of TE insertion.

5. Most non-zero values should appear in the upper triangle region of the matrix (dark grey in Figure 5.2), which corresponds to young TE families interrupting old TE families.

6. Interruptions of the same families of TEs into themselves (which would be recorded directly on the matrix diagonal) are not scored due to the fact that they are difficult to confidently identify and do not affect the ordering analysis.

The linear ordering problem is $\mathcal{NP}$-hard; this implies that there likely does not exist a polynomial time algorithm for calculating an optimal solution. After computing an interruption matrix of $n$ TEs using the sequential interruption model in Section 5.1, a straightforward method to find the permutation of the problem would be exhaustive search: applying all $n!$ possible permutations to the interruption matrix and the resultant permutation will be the one with which the permuted interruption matrix achieves the maximum score over all $n!$ sum of upper triangle scores. The exhaustive search algorithm has a complexity of $O(n^2 \times n!)$ (the

$n^2$ does the additions of the upper triangle), which is considerably inefficient. In the human genome (*hg38*) and according to Repbase Update, there are 1,080 different TEs existing in the genome, thus the size of the matrix is $n \times n$, where $n = 1080$. As such, it is not feasible to use the exhaustive search on the original matrix (when $n$ is big) to find a permutation. Therefore, non-optimal techniques are required:

1. Because of the sparseness property of the matrix, it is possible to slightly reduce the size of the matrix by removing any transposon (both the corresponding row and column) with zero (or low) interruptions. Lemma 1 shows that it is possible to remove these rows and columns.

   **Lemma 1.** *Let $M$ $(n \times n)$ and $i$, $1 \leq i \leq n$, where row $i$ and column $i$ are all $0$s, and let $M'$ be obtained from $M$ by removing row $i$ and column $i$. $M'$ is $(n-1) \times (n-1)$. The optimal answer for linear ordering problem on $M'$ is the same as $M$.*

   *Proof.* Assume that $\pi = (k_1, k_2, k_3, \ldots, k_n)$ is the order corresponding to the permuted matrix, $M_\pi$, that $\sum_{p=k_1}^{k_n} \sum_{q=p+1}^{k_n} M^{(\pi(p),\pi(q))}$ is maximum among all possible permutations. Given $k_j = i$, remove $k_j$ from $\pi$, then the order becomes $\pi' = \{k_1, \ldots, k_{j-1}, k_{j+1}, \ldots, k_n\}$, and the permuted matrix becomes $M_{\pi'}$, where the row $k_j$ and column $k_j$ in $M_\pi$ are removed from the matrix. As row $i$ and column $i$ are all 0's in $M$, thus row $k_j$ and column $k_j$ are all 0's in $M_\pi$ as well. Thus, the sum of the upper triangle of $M_{\pi'}$ is the same as that of $M_\pi$. Therefore, the optimal answer for the linear ordering problem on $M'$ is the same as $M$. □

   Though Lemma 1 provides a way to reduce $n$, then to reduce the time of computation, the complexity of exhaustive search does not change. After applying Lemma 1 to the overall interruption matrix of the human genome we calculated, the size of the matrix is reduced from 1,080 to 1,015. With $n = 1,015$, an exhaustive search method is still not practical. Therefore, this method can be only used to calculate for a smaller set of TEs (such as a family with perhaps 10 TEs). This brute force method has been implemented only on reduced sized matrix for a small set of TEs. Unfortunately, the removed TEs (that do not have interruption data) were not fit into the calculated order.

2. Furthermore, since the linear ordering problem arises in a variety of applications ranging from archeology and scheduling to economics and even mathematical psychology, algorithms for its efficient solution are required. There are some exact methods that use *Branch-and-Bound* algorithms to solve the problem to (proven) optimality (discussed in [96]), such as the branch-and-bound with partial orderings in [33], the lexicographic search algorithm in [79, 80], and the branch-and-bound approach, where Lagrangian relaxation techniques are used for bound computations in [23]. The branch-and-bound can also be realized in a special way leading to the so-called *Branch-and-Cut* method, which is essentially a branch-and-bound algorithm, where the upper bounds are computed using linear programming relaxations as discussed in [96]. In this thesis, the details of these exact algorithms or their application to the sequential interruption model will not be discussed.

3. Heuristic and meta-heuristic methods attempt to find a good, but not necessarily optimal solution to the problem, which is in contrast to exact methods that guarantee to give an optimum solution. Nevertheless, the time taken to find an optimum solution to a difficult problem by an exact method is often much greater than heuristic and meta-heuristic methods. Thus, heuristic and meta-heuristic methods are often used to solve real optimization problems. Martí and Reinelt [96] summarized many heuristic and meta-heuristic methods to solve the LOP, such as GRASP [40], Tabu search [48], the simulated annealing method [76], variable neighbourhood search [53], scatter search [83], iterated local search [20], etc. A computational comparison of 24 heuristic and meta-heuristic methods for the LOP on 484 instances done in [97], concluded that the meta-heuristics obtain high quality solutions, moreover, the memetic algorithm implementation, `MA`, performs best, followed by iterated local search, `ILS`, and with the Tabu search, `TS`, ranked in third place.

Several of the implemented software for solving the LOP only output the score of the best solution without providing the actual ordering that yields the score, which is important to our problem. Fortunately, we have obtained the source code of one method, Tabu search, from the authors of [97], and by modifying it, the ordering can be output together with its score, which corresponds to the relative ages of TEs. In the next subsection, the general idea of Tabu search [48] will be described, then the result of Tabu search applied in the sequential interruption model will be provided and compared with the published result in [47].

### 5.3.3 Tabu search and results

The word "tabu" comes from a language of Polynesia, Tongan, indicating things that cannot be touched because they are sacred, which accords very well with the idea of Tabu search. Generally speaking, Tabu search keeps a table of solutions that are forbidden to guide the search, so that the selection of solutions is limited according to the table of tabu status.

Tabu search begins in the same way as an ordinary local search, moving from one solution to another repeatedly until a number of global iterations are performed without improving the best solution found so far. If the search space is seen as a huge set of solutions and only a tiny part of the set can be explored, then Tabu search guides the local search process to examine the solution space beyond local optimality. It consists of two search strategies — *intensification* and *diversification* — with complementary objectives to search in the solution set. Intensification favours the exploration of promising areas of the solution space, while diversification moves the search to new regions of the solution space.

As mentioned in the last section, the authors of [97] have provided us with the `C` source code of the Tabu search solving the LOP program. Given an interruption matrix (Definition 17), $IM(1015 \times 1015)$, calculated on the human genome *hg38*, the sum of the overall matrix excluding the sum of diagonal is $\sum_{i=1}^{m} \sum_{j=1}^{m} IM^{(i,j)} -$

$\sum_{i=1}^{m} IM^{(i,i)} = 381,201$. By inputing $IM$ to the Tabu search program, a TE ordering (from predicted youngest TE to the oldest TE) that achieves the best superdiagonal score,

$$f(\pi) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} IM^{(\pi(i),\pi(j))} = 377,417,$$

is calculated. Since the Tabu search is a meta-heuristic algorithm, this score is not guaranteed to be optimal. However, it only took 38 seconds to calculate the best score of a matrix of size 1,015 on a 2.9 GHz Intel Core i5 processor with 16 GB memory. The calculated TE ordering of the interruption matrix, $IM(1015 \times 1015)$, achieving the best score of the LOP is attached in Appendix A. The ability in solving this problem on such a big matrix has made Tabu search outperform the method proposed in [47] in terms of efficiency, which was only able to solve a much smaller matrix also without a guarantee of finding the optimal solution.

The resultant ordering from Tabu search is then compared with the ordering published in [47] (Giordano et. al.) in two different ways. First, as the method in [47] is computationally impractically expensive, though there were $\approx 1,000$ TEs with interruptions, only 405 were selected for calculating their ordering in the paper. Moreover, among these selected 405 TEs, there were 359 of them that are in common with the TEs in the $IM$ that was calculated on the human genome $hg38$. This might be caused by the ongoing updates of the TE names in Repbase Update during these years. The set of the $n = 359$ common TEs are denoted by $\chi_n$. The resultant ordering from Tabu search of $\chi_n$ is denoted as $\pi_{tabu}$, and the ordering published in Giordano et. al. [47] of these TEs is denoted as $\pi_{Giordano}$, then the sub-matrix of $IM$ on $\chi_n$ is denoted as $IM_{\chi_n}$. For the matrix $IM_{\chi_n}$, the sum of the overall matrix excluding its diagonal is

$$\sum_{i=1}^{n} \sum_{j=1}^{n} IM_{\chi_n}^{(i,j)} - \sum_{i=1}^{n} IM_{\chi_n}^{(i,i)} = 169,503 - 1,687 = 167,816.$$

The superdiagonal scores of the two permutations $\pi_{tabu}$ and $\pi_{Giordano}$ on $IM_{\chi_n}$ are

$$f(\pi_{tabu}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} IM_{\chi_n}^{(\pi_{tabu}(i),\pi_{tabu}(j))} = 165,980,$$
$$f(\pi_{Giordano}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} IM_{\chi_n}^{(\pi_{Giordano}(i),\pi_{Giordano}(j))} = 165,591.$$

It can be seen that the ordering calculated by Tabu search achieves a higher score for this (reduced) problem, which indicates that Tabu search is performing better than the method in [47] in terms of the final score.

Second, the similarity between the ordering calculated from Tabu search ($\pi_{tabu}$) and the ordering published in [47] ($\pi_{Giordano}$) are compared with each other using the Pearson's coefficient of correlation calculated as

$$\rho = \frac{cov(\pi_{Giordano}, \pi_{tabu})}{\sigma_{\pi_{tabu}} \sigma_{\pi_{Giordano}}} = 0.943522,$$

which shows a strong positive correlation (agreement) between the two orderings.

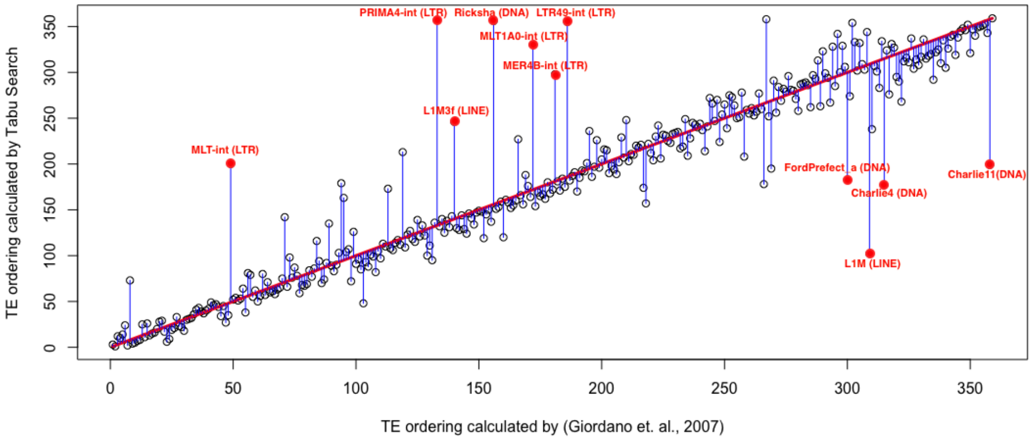


**Figure 5.3:** A comparison between the ordering calculated in [47] ($\pi_{Giordano}$) and the ordering calculated by Tabu search ($\pi_{tabu}$). The thick red diagonal line represents the case when the two orderings are exactly the same. The elements in Table 5.5 are marked in red.

The two orderings are then plotted against each other in Figure 5.3, where the ordering calculated in [47] is shown on the $x$-axis, the ordering calculated by Tabu search is shown on the $y$-axis, and the thick red diagonal line represents the case when the two orderings are exactly the same, namely *100% agreement line.*

As TEs can be active for a period of time, more than one TE can be active at the same time, in parallel. Hence, it is reasonable that the relative age order of these parallelly active TEs are shuffled within a region of the overall ordering, which results in these TEs being "around" the 100% agreement line on the plot. It can be seen from the figure that most of the elements agree in the two orderings very well as they are very "close" to the red line. There are also some "outliers", which indicate that their positions are distant (very different) in the two orderings. The elements that have a distance of more than 100 positions in the two orderings are listed in Table 5.5 and marked with their names and types in brackets in Figure 5.3.

A detailed biological analysis and verifications of their actual positions (actual evolutionary age) of the elements in Table 5.5 relative to other TEs will be left as future work. However, another technique for verification, with the help of a simulation, is created as part of Chapter 7, where this work will be revisited.

## 5.4 Conclusion

In this chapter, a TE sequential interruption model was created based on the abundance of TEs interrupting other TEs, and the problem of predicting TE age, proposed in [47], was formulated by our model as a well-studied matrix problem — the Linear Ordering Problem — which can solve our problem very efficiently. As discussed, though [47] did not report how long it took the repositioning method to solve the problem

| TE_name | TE_type | TE_family | Position in Giodano's method | Position in Tabu search |
|---|---|---|---|---|
| MLT-int | LTR | ERVL-MaLR | 49 | 200 |
| PRIMA4-int | LTR | ERV1 | 133 | 357 |
| L1M3f | LINE | L1 | 140 | 246 |
| Ricksha | DNA | MULE-MuDR | 156 | 356 |
| MLT1A0-int | LTR | ERVL-MaLR | 172 | 330 |
| MER4B-int | LTR | ERV1 | 181 | 298 |
| LTR49-int | LTR | ERV1 | 186 | 355 |
| FordPrefect_a | DNA | hAT-Tip100 | 300 | 183 |
| L1M | LINE | L1 | 309 | 102 |
| Charlie4 | DNA | hAT-Charlie | 315 | 177 |
| Charlie11 | DNA | hAT-Charlie | 358 | 199 |

**Table 5.5:** The TEs that have a distance of more than 100 positions in the two orderings of $\pi_{Giordano}$ and $\pi_{tabu}$.

on the reduced matrix (of size 405), it is likely long, as only a portion of TEs were solved; in contrast, the Tabu search solves the LOP on the full size matrix (of size 1,015) in just 38 seconds, while achieving better results when restricting to the elements common in both method. Therefore, the LOP and Tabu search in particular as per the sequential interruption model is more practical and achieves good results in predicting the relative ages of TEs. Further verification and comparison of the sequential interruption model to another new method will occur in Chapter 7.

# Chapter 6

# The recursive interruption model using stochastic context-free grammars [1]

## 6.1 Introduction

When many insertions occurred throughout the evolution of a genomic sequence, the interruptions can nest in a recursive pattern [81], which cannot be represented entirely with the interruptional matrix approach that only counts the abundance of a TE in-between another TE without storing the hierarchical relationships of interruptions. Indeed, Example 8 shows some nested TEs in real data that occurs in the human genome and cannot be described with the linear model.

**Example 8.** *Table 6.1 contains a list of TE fragments taken from the RepeatMasker TE fragments set,* $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$*, where* $s$ *is the X chromosome of the human genome, and* $\chi_s$ *is the library of human transposable elements in Repbase Update. These seven TE fragments start from the X chromosome position 53437061 to 53438226 that belong to four TE families:* $X_1$*,* $X_2$*,* $X_3$*, and* $X_4$*, where the names of the families of TEs* $X_1, X_2, X_3, X_4$ *are MIR, AluJb, AluSx, AluSq2.*

| *Fragment* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | *chrX* | *53437061* | *53437143* | *-101833417* | + | *MIR* | *SINE* | *3* | *88* | *-174* |
| $z_2$ | *chrX* | *53437143* | *53437277* | *-101833283* | + | *AluJb* | *SINE* | *1* | *132* | *-170* |
| $z_3$ | *chrX* | *53437277* | *53437448* | *-101833112* | + | *AluSx* | *SINE* | *39* | *192* | *-120* |
| $z_4$ | *chrX* | *53437448* | *53437761* | *-101832799* | + | *AluSq2* | *SINE* | *1* | *312* | *0* |
| $z_5$ | *chrX* | *53437761* | *53437887* | *-101832673* | + | *AluSx* | *SINE* | *193* | *312* | *0* |
| $z_6$ | *chrX* | *53437887* | *53438055* | *-101832505* | + | *AluJb* | *SINE* | *133* | *293* | *-9* |
| $z_7$ | *chrX* | *53438055* | *53438226* | *-101832334* | + | *MIR* | *SINE* | *89* | *261* | *-1* |

**Table 6.1:** An example of recursive interruptions on the X chromosome.

*As in Definition 13, the genomic sequence* $s$ *is*

$$s = w_0 z_1 w_1 z_2 w_2 z_3 w_3 \ldots z_7 w_7,$$

[1] Part of the work in this chapter has been published in [59].

*as visualized in Figure 6.1.*

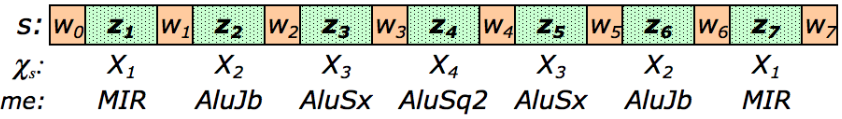

s: w0 z1 w1 z2 w2 z3 w3 z4 w4 z5 w5 z6 w6 z7 w7
χs: X1 X2 X3 X4 X3 X2 X1
TE name: MIR AluJb AluSx AluSq2 AluSx AluJb MIR

**Figure 6.1:** A conceptual visualization of a genomic sequence (the human X chromosome from position 53437061 to 53438226), with the RepeatMasker detected TE fragments in Table 6.1. Note that the lengths of the visualized sequences in the figure are not proportional to their actual lengths.

*The pruned sequence of s is*

$$\bar{s} = \beta_0 z_1 \beta_1 z_2 \beta_2 z_3 \beta_3 z_4 \beta_4 z_5 \beta_5 z_6 \beta_6 z_7 \beta_7,$$

*where* $\beta_0, \ldots, \beta_7 \in \mathbb{N}$, $z_1, z_7 \in \bar{X}_1$, $z_2, z_6 \in \bar{X}_2$, $z_3, z_5 \in \bar{X}_3$, $z_4 \in \bar{X}_4$, $\bar{X}_1, \bar{X}_2, \bar{X}_3, \bar{X}_4 \in \bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$, *and* $z_1 \overset{\varepsilon,E}{\sim} z_7$, $z_2 \overset{\varepsilon,E}{\sim} z_6$, $z_3 \overset{\varepsilon,E}{\sim} z_5$.

*It is possible to see a potential process of nested interruptions described as:*

- *at first, an instance of AluJb inserted itself into an instance of MIR to break it into* $z_1$ *and* $z_7$*;*
- *then an instance of AluSx inserted itself into the instance of AluJb that has already presented in the sequence, to break it into* $z_2$ *and* $z_6$*;*
- *more recently, an instance of AluSq2 (*$z_4$*) inserted itself into the presented AluSx instance to break it into* $z_3$ *and* $z_5$*.*

*From the interruptional analysis above, the age order of the three families of TEs can be predicted from the recursive interruptions as: MIR, AluJb, AluSx, AluSq2, from oldest to youngest.*

The nested nature of the interruptions in Example 8 is not captured by the interruptional matrix as done in Chapter 5, or in [47], because the recursive nesting can "push" fragments so that they are no longer continuous. However, these nested interruptions are informative in predicting the chronological order of when these interruptions occurred in the genomic sequence. Therefore, a new model built on top of the TE fragment model will be created in this chapter to capture this hierarchical nesting feature. First, a specific context-free grammar called the *recursive interruption context-free grammar* will be defined to model the generation of recursive interruptions. Then algorithms that calculate a parse tree of the grammar generating a given *order-pruned sequence* will be given, where the parse tree shows a prediction of the hierarchical structure of TE insertions. Before formally defining the recursive interruption context-free grammar, some basic preliminaries on formal language theory will be introduced in Section 6.2.

## 6.2 Preliminaries on formal language theory

First, the definitions in this section build on the definition of an alphabet and the set of all words over that alphabet as in Definition 8. Also, given an alphabet $\Sigma$, and the set $\Sigma^*$, a language $L$ is any $L \subseteq \Sigma^*$.

In the area of formal language theory, a context-free grammar is a certain type of rewriting system that iteratively rewrites strings as per a set of production rules describing the manner in which letters can become replaced. Formally:

**Definition 19.** *A* context-free grammar (CFG) *is a four tuple $G = (V, T, P, S)$, where*

- *$V$ is the alphabet of variables (usually capital letters),*
- *$T$ is the alphabet of terminals (usually lower case),*
- *$P$ is the finite set of productions, each of the form $A \to w$, where $A \in V$ and $w \in (V \cup T)^*$,*
- *$S$ is the starting variable, where $S \in V$.*

The left-hand side of each production is always a variable, and the right-hand side is any string containing variables and terminals. Each context-free grammar defines a language over the terminal alphabet using either a rewriting mechanism, or an equivalent mechanism using parse trees. We will adopt the latter approach [56].

Derivations can be represented by a tree, called a *parse tree*, that shows clearly how the symbols of a terminal string (e.g., 01010) are derived as leaves of the tree, where the tree is constructed according to the productions.

**Definition 20.** *Let $G = (V, T, P, S)$ be a context-free grammar. A* parse tree *(or derivation tree) of $G$ is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. The parse trees of $G$ are all trees, $t$, with the following conditions (from [56]):*

- *the root is labelled by $S$;*
- *each interior node (non-leaf) is labelled by a variable in $V$;*
- *each leaf is labelled by either a terminal, or $\lambda$. However, if the leaf is labelled $\lambda$, then it must be the only child of its parent;*
- *if an interior node is labelled $A$, and its children are labelled $X_1, X_2, \ldots, X_k$ respectively, from left-to-right, then $A \to X_1 X_2 \ldots X_k$ is a production of $P$. Note that the only time one of the $X$'s can be $\lambda$ is if that is the label of the only child, and $A \to \lambda$ is a production of $G$.*

*The yield of a parse tree $t$ is the string obtained by concatenating all labels on leaves from left to right. This must be a string over $T$. The language generated by $G$, denoted by $L(G)$, is the set of all yields of parse trees of $G$. This is a language over $T$.*

Example 9 shows the context-free grammar of palindromes.

**Example 9.** *A* palindrome *is a string that reads the same forward and backward, such as* `level`*,* `madam`*,* `101101` *etc. Let $G = (V, T, P, S)$ be a context-free grammar, where*

- $V = \{S, X\}$ *is the variable alphabet,*
- $T = \{0, 1\}$ *is the terminal alphabet,*
- $P = \{S \to X,\ X \to \lambda,\ X \to 0,\ X \to 1,\ X \to 0X0,\ X \to 1X1\}$,
- $S$ *is the start symbol.*

*Figure 6.2 shows a parse tree of $G$. The production used at the root is $S \to X$. Then the production used on the first $X$ is $X \to 0X0$, etc. The yield of the tree is* 01010*, a palindrome. It is possible to show that $L(G)$ is the set of all palindromes over $T$.*

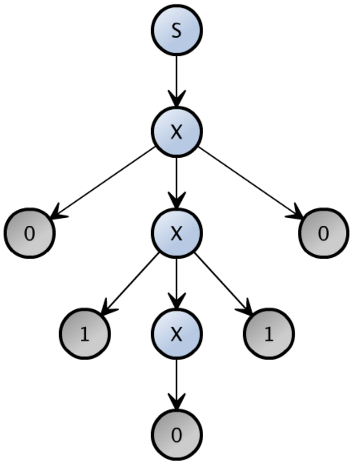


**Figure 6.2:** A parse tree showing the derivation of 01010.

Given a context-free grammar $G$, and a word $w$, the process of finding a parse tree of $G$ generating $w$ is called *parsing*. There is a polynomial time parsing algorithm called the CYK algorithm [56].

A *stochastic context-free grammar* (SCFG) is a context-free grammar, where every production in the grammar has an associated probability value between 0 and 1, such that the probability for all productions from each variable adds to 1. The probability associated with a parse tree is the product of the probabilities of the production instances applied to produce it.

SCFGs can be applied to different areas. For example, it is used in the field of speech recognition in modelling isolated words (words with similar beginning and ending but with different vowels in between) accurately, or in representing the language model component of a speech recognizer [85]. They are also applied in RNA secondary structure problems [36], such as RNA secondary structure prediction for a single sequence, and multiple RNA sequence alignment algorithms that incorporate secondary structure constraints.

## 6.3 Context-free grammar to generate recursive interruptions

In this section, a theoretical model will be proposed to describe the nature of recursive interruptions using a context-free grammar. Ultimately, a probabilistic model will be created.

**Definition 21.** *Given a set of TEs with a fixed order on its elements, $\chi = \{X_1, X_2, \dots, X_m\}$, the* recursive interruption context-free grammar *is a grammar $G = (V, T, P, S)$, where $V = \{S, X_1, X_2, \dots, X_m\}$, $T = \{1, 2, \dots, m\}$, and $P$ contains the following productions:*

$$\begin{aligned} S &\rightarrow X_i S, && 1 \le i \le m, && (6.3.1)\\ S &\rightarrow X_i, && 1 \le i \le m, && (6.3.2)\\ X_i &\rightarrow X_i X_j X_i, && 1 \le i \le m,\ 1 \le j \le m, && (6.3.3)\\ X_i &\rightarrow i, && 1 \le i \le m. && (6.3.4)\end{aligned}$$

This grammar is used to generate strings over $\{1, \dots, m\}^*$ corresponding to TE orders. Intuitively, productions of type 6.3.3 correspond to an instance of $X_j$ inserting itself throughout evolution into an instance of $X_i$, as shown in Figure 6.3, leaving a fragment from $i$, then $j$, then $i$.

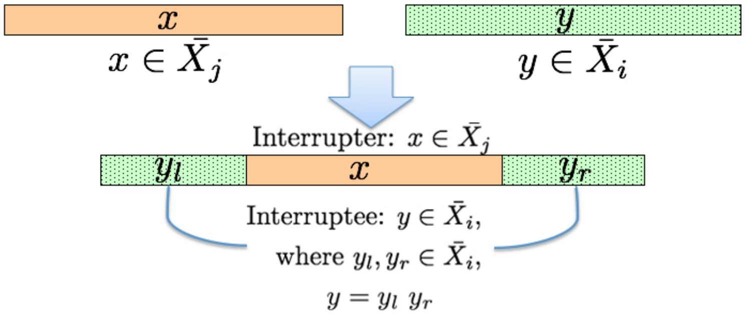

$x$
$y$
$x \in \bar{X}_j$
$y \in \bar{X}_i$
Interrupter: $x \in \bar{X}_j$
$y_l$
$x$
$y_r$
Interruptee: $y \in \bar{X}_i$,
where $y_l, y_r \in \bar{X}_i$,
$y = y_l\ y_r$

**Figure 6.3:** A diagram showing an instance of $X_j$ inserting itself throughout evolution into an instance of $X_i$, corresponding to an application of a production of type 6.3.3.

When constructing a parse tree from the top-down, if there are three consecutive nodes labelled by $X_i X_j X_i$, these can either derive $iji$ (using productions of type 6.3.4) corresponding to that order of TEs, or any of them can be further interrupted (using additional productions of type 6.3.3). Productions of type 6.3.1

correspond to independent positions of the sequence where a TE can insert itself (not a nested insertion, and this can only be produced continuously from the root along the rightmost path of a parse tree). Productions of type 6.3.2 correspond to the final independent position of a TE insertion.

We are therefore interested in parse trees, which correspond to different potential sets of molecular events that could have occurred, and the nesting patterns. However, this context-free grammar is ambiguous (meaning that multiple parse trees can produce the same string). Indeed, it is clear that any non-empty string over $T^*$ can be generated by G by using only productions of types 6.3.1, 6.3.2, and 6.3.4. This would require the application of $2k$ productions to generate a string of length $k$. However, for every application of a production of type 6.3.3, the total number of productions needed to generate a string of length $k$ decreases. If there are $l$ productions of type 6.3.3 applied, the total number of productions needed to generate a string of length $k$ decreases to $2(k - l)$.

Example 10 shows how nested interruptions in a sequence are generated by the grammar as the yield of its possible parse trees.

**Example 10.** *Given a genomic sequence s and a set of TEs with a fixed order on its elements* $\chi_s = \{X_1, X_2, \ldots, X_{10}\}$*, assume*

$$s = w_0 z_1 w_1 \ldots z_{13} w_{13},$$

*as in Equation (4.1.2), with* $z_1, z_4, z_6 \in \bar{X}_2$*,* $z_2, z_8, z_{10}, z_{12} \in \bar{X}_3$*,* $z_5 \in \bar{X}_4$*,* $z_{11}, z_{13} \in \bar{X}_5$*,* $z_3, z_7 \in \bar{X}_6$*,* $z_9 \in \bar{X_{10}}$*. Then an order-pruned sequence*

$$\bar{s}_o = 2,\ 3,\ 6,\ 2,\ 4,\ 2,\ 6,\ 3,\ 10,\ 3,\ 5,\ 3,\ 5$$

*is the yield of the parse tree shown in Figure 6.4.*

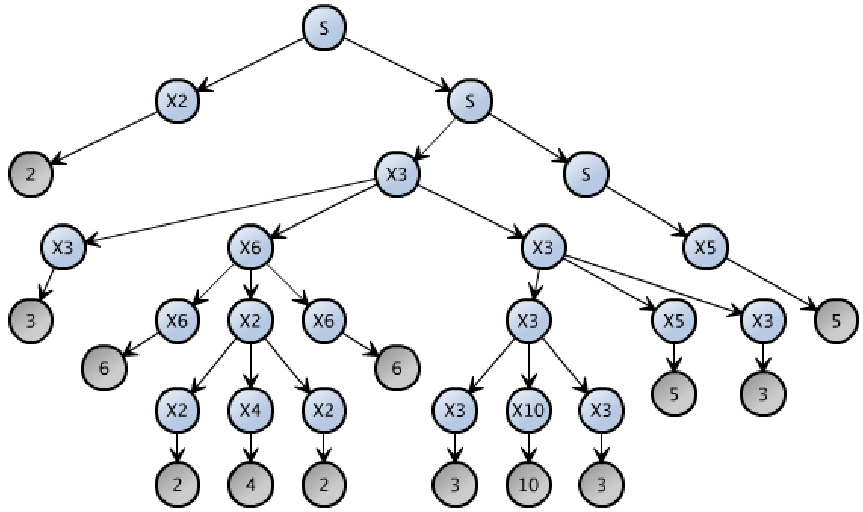


**Figure 6.4:** A parse tree of G from Definition 21 that yields $\bar{s}_o$.

Only the parse tree that maximizes the application of productions of type 6.3.3, or equivalently minimizes the total number of productions applied are of interest. This would correspond to minimizing the number of transpositional events that occurred throughout evolution. Indeed, it is common in phylogeny to prefer the evolutionary pathway that requires the fewest number of changes to be applied [113], which is known as *parsimony*. Hence, minimizing the number of transpositional events is the most parsimonious possibility.

## 6.4 Stochastic CYK algorithm for finding a most likely parse tree

As discussed, given an order-pruned sequence, a parse tree of the grammar that maximizes the applications of productions of type 6.3.3, or minimizes the overall number of productions applied to generate this sequence, is of primary interest. In this subsection, some methods to find such parse trees will be given by converting the recursive interruption context-free grammar into a stochastic context-free grammar.

Considering the context-free grammar in Definition 21, by slightly changing production 6.3.2 and attaching probabilities between 0 and 1, the parse trees that use fewer productions will have a higher probability.

**Definition 22.** *Given a set of TEs with a fixed order on its elements, $\chi = \{X_1, X_2, \ldots, X_m\}$, the* recursive interruption stochastic context-free grammar *is a grammar $G = (V, T, P, S)$, where $V = \{S, X_1, X_2, \ldots, X_m\}$, $T = \{1, 2, \ldots, m\}$, and $P$ contains the following productions where each production is attached with a probability $p$:*

$$S \rightarrow X_i S, \quad 1 \leq i \leq m, \quad p = \frac{1}{m^2}, \tag{6.4.1}$$

$$S \rightarrow X_i X_j, \quad 1 \leq i \leq m,\ 1 \leq j \leq m,\ i \neq j, \quad p = \frac{1}{m^2}, \tag{6.4.2}$$

$$X_i \rightarrow X_i X_j X_i, \quad 1 \leq i \leq m,\ 1 \leq j \leq m,\ i \neq j, \quad p = \frac{1}{m}, \tag{6.4.3}$$

$$X_i \rightarrow i, \quad 1 \leq i \leq m, \quad p = \frac{1}{m}. \tag{6.4.4}$$

Given a word of $w \in T^*$, a *most likely parse tree* for $w$ is defined as a parse tree with a yield of $w$ with the highest probability. This corresponds to the parse tree that has the most productions of type 6.4.3 applied in the recursive interruption context-free grammar. For this grammar, all productions for each variable were given equal weight: for each production of $S$, the probability is $1/m^2$ (there are $m$ productions of type 6.4.1, and $m \times (m-1)$ productions of type 6.4.2 giving $m^2$ productions), and for each production of $X_i$, the probability is $1/m$ (there are $m-1$ productions of type 6.4.3, and 1 production of type 6.4.4). Thus a modified version of the CYK algorithm [36] that takes the probabilities into account can find a most likely parse tree that has a given sequence as yield is of primary interest. In our case, starting with the order-pruned sequence, it can predict a most likely parse tree with it as the yield.

Example 10 shows how nested interruptions in a sequence are generated by the stochastic context-free grammar as the yield of the parse tree that maximized the application of productions of type 6.4.3.

**Example 11.** *Given a genomic sequence* $s$ *and a set of TEs with a fixed order on its elements* $\chi_s = \{X_1, X_2, \ldots, X_{10}\}$*, assume*

$$s = w_0 z_1 w_1 \ldots z_{13} w_{13},$$

*as in Equation (4.1.2), with* $z_1, z_4, z_6 \in \bar{X}_2$*,* $z_2, z_8, z_{10}, z_{12} \in \bar{X}_3$*,* $z_5 \in \bar{X}_4$*,* $z_{11}, z_{13} \in \bar{X}_5$*,* $z_3, z_7 \in \bar{X}_6$*,* $z_9 \in \bar{X_{10}}$*.*

*The stochastic context-free grammar is*

$$\begin{array}{lcllll}
S & \rightarrow & X_i S, & i = 2,3,4,5,6,10 & p{=}0.0278 \\
S & \rightarrow & X_2 X_j, & j = 3,4,5,6,10 & p{=}0.0278 \\
S & \rightarrow & X_3 X_j, & j = 2,4,5,6,10 & p{=}0.0278 \\
S & \rightarrow & X_4 X_j, & j = 2,3,5,6,10 & p{=}0.0278 \\
S & \rightarrow & X_5 X_j, & j = 2,3,4,6,10 & p{=}0.0278 \\
S & \rightarrow & X_6 X_j, & j = 2,3,4,5,10 & p{=}0.0278 \\
S & \rightarrow & X_{10} X_j, & j = 2,3,4,5,6 & p{=}0.0278 \\
X_2 & \rightarrow & X_2 X_j X_2, & j = 3,4,5,6,10 & p{=}0.1667 \\
X_3 & \rightarrow & X_3 X_j X_3, & j = 2,4,5,6,10 & p{=}0.1667 \\
X_4 & \rightarrow & X_4 X_j X_4, & j = 2,3,5,6,10 & p{=}0.1667 \\
X_5 & \rightarrow & X_5 X_j X_5, & j = 2,3,4,6,10 & p{=}0.1667 \\
X_6 & \rightarrow & X_6 X_j X_6, & j = 2,3,4,5,10 & p{=}0.1667 \\
X_{10} & \rightarrow & X_{10} X_j X_{10}, & j = 2,3,4,5,6 & p{=}0.1667 \\
X_i & \rightarrow & i, & i = 2,3,4,5,6,10 & p{=}0.1667
\end{array}$$

*Then an order-pruned sequence*

$$\bar{s}_o = 2,\ 3,\ 6,\ 2,\ 4,\ 2,\ 6,\ 3,\ 10,\ 3,\ 5,\ 3,\ 5$$

*is the yield of the parse tree calculated by the stochastic CYK algorithm shown in Figure 6.4 with probability of* $7.63713361322155e - 18$*.*

*It can be seen that compared to the parse tree in Figure 6.4, the most likely parse tree is the one in Figure 6.5 as it maximizes the number of applications of production 6.4.3 and pushes the nested interruptions deeper into the tree, which is more parsimonious, and therefore a more likely age order than a tree that spreads the interruptions into different branches in the same level.*

The complexity of the (both non-stochastic and stochastic) CYK algorithm is $O(n^3)$ [56], where $n$ is the length of the yield (order-pruned sequence corresponding to the number of TE fragments detected in a genomic sequence). Unlike the standard CYK algorithm that returns all the possible parse trees, the stochastic

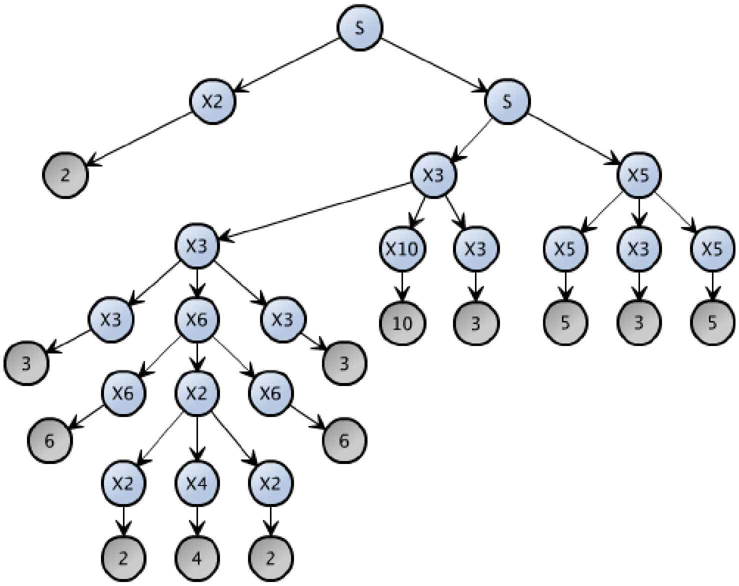


**Figure 6.5:** The most likely parse tree of G from Definition 21 that yields $\bar{s}_o$.

CYK only traces back the productions with the highest probabilities, which can decrease the computing time dramatically in practice. A package in the `Perl language` of a stochastic CYK algorithm has been developed derived from an existing package [123] with polynomial time parsing to calculate the most likely parse tree(s) with the highest probabilities corresponding to the most probable evolutionary events, using the stochastic context-free grammar in Definition 22.

## 6.5 Improvements taking into account positional information between TE fragments

The recursive interruption context-free grammar in Definition 21 is a very simple and general way of capturing the recursive nature of TE interruptions. However, a limitation of the model is that the order-pruned sequence generated by the grammar only contains the TEs (names/order of TEs) to which the detected TE fragments belong. It does not take into account whether two TE fragments are separated by say 1 *bp* or $1,000$ *bp* in the genomic sequence, or where each fragment lies within a TE consensus sequence with the current grammar. It is less clear how one could take the positional information into account to determine whether two TE fragments are continuous fragments (both $E$ and $\varepsilon$ in Definition 15), then further determine the existence of an interruption in an order-pruned sequence.

In this section, one improvement taking into account of the positions of TE fragments in the genomic sequence is incorporated in detecting pruned sequences by utilizing the definition of a transposon region (Definition 14).

Recall the pruned sequence and order-pruned sequence in Definition 13 in Chapter 4. A genomic sequence $s = w_0 z_1 w_1 z_2 w_2 z_3 w_3 z_4 w_4 z_5 w_5 z_6 w_6$ is visualized in Figure 6.6.

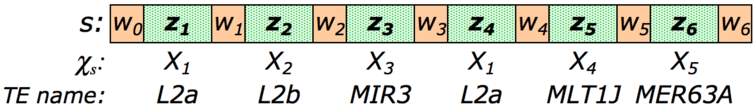


**Figure 6.6:** A conceptual visualization of a genomic sequence (the human chromosome 1 from position 33632576 to 33634148). The TE fragments $z_i$, where $i = 1, \ldots, 6$ in the sequence are marked with the notation of TE families $X_j \in \chi_s$, where $j = 1, \ldots, 5$, and the names of the TE families to which they belong.

According to Definition 13, the pruned sequence of $s$ is

$$\bar{s} = \beta_0 z_1 \beta_1 z_2 \ldots z_6 \beta_6, \text{ where } \beta_i = |w_i|, \; 0 \leq i \leq 6,$$

and the order-pruned sequence of $s$ is

$$\bar{s_o} = 1, \; 2, \; 3, \; 1, \; 4, \; 5, \text{ where } z_i \in X_{j_i}, \text{ for all } i, \; 1 \leq i \leq 6.$$

Similar to the above case, the human genome, or each chromosome, can be represented as a giant pruned sequence that includes all TE fragments in the genome, where the non-repeat part, $\beta_i$, can be very small if two TE fragments are close to each other, or can be very large (or infinity) if two TE fragments are on different chromosomes. However, the fact is that the TE fragments belonging to the same interruption should not be apart for a large genomic distance (they should be on the same chromosome, and very close to each other), which is the why the definition of interruption (Definition 16) takes the two distances (both in genomic sequence and in TE consensus sequence) into account. In light of this property of TE interruption, a pruned sequence can be limited to only represent the segment of genomic sequence where a transposon region is located. Essentially, a transposon region is a region of the genomic sequence where TE fragments are detected and they are close to each other by a distance $E \in \mathbb{N}$.

Given a distance $d \in \mathbb{N}$ (in the genomic sequence), a number of transposon regions can be detected by parsing the genomic sequence, within which potential interruptions may be detected. For example, parsing the genomic sequence of the human genome $hg38$ Y chromosome with $d = 20$ $bp$, there are 2,125 transposon regions detected with the length of their order-pruned sequences ranging from 3 to 65 (the number of TE fragments in the region) as listed in Table 6.2. It can be seen that a majority (78%) of order-pruned sequences are short with a length of less than 10. Each pruned sequence is one yield to feed into the recursive interruption context-free grammar, which corresponds to one (or more) most-likely parse tree.

| Length of the order-pruned sequence | Number of order-pruned sequences of the length | Length of the order-pruned sequence | Number of order-pruned sequences of the length |
|---|---|---|---|
| 3 | 485 | 23 | 11 |
| 4 | 312 | 24 | 5 |
| 5 | 273 | 25 | 12 |
| 6 | 195 | 26 | 4 |
| 7 | 141 | 27 | 8 |
| 8 | 131 | 28 | 3 |
| 9 | 113 | 29 | 3 |
| 10 | 69 | 30 | 5 |
| 11 | 68 | 31 | 3 |
| 12 | 48 | 32 | 1 |
| 13 | 40 | 34 | 3 |
| 14 | 39 | 35 | 2 |
| 15 | 27 | 40 | 2 |
| 16 | 36 | 41 | 1 |
| 17 | 20 | 43 | 1 |
| 18 | 13 | 44 | 1 |
| 19 | 18 | 57 | 1 |
| 20 | 11 | 59 | 1 |
| 21 | 10 | 65 | 1 |
| 22 | 8 | | |

**Table 6.2:** A summary of the order-pruned sequences of the transposon regions detected on the Y chromosome of $hg38$.

Similarly, the order-pruned sequences of transposon regions on every chromosome of the human genome are extracted with the ranges of their length. A summary of the order-pruned sequences in the human genome is listed in Table 6.3.

Note that the total number of order-pruned sequences in the human genome is 327,305, which will obtain around the same number of parse trees (TE interruption trees). The number of order-pruned sequences and the number of parse trees are not necessarily the same. This is because some order-pruned sequences (e.g., $\bar{s_o} = 3, 4, 5$ or $\bar{s_o} = 7, 7, 7, 6$) cannot obtain any interruption tree, and some can obtain multiple most-likely trees with the same probability (e.g., $\bar{s_o} = 3, 6, 3, 6, 3, 6, 10, 6, 3$ can obtain two trees as visualized in Figure 6.7).

| | Number of transposon regions<br>(no. order-pruned sequences) | The length range of order-pruned sequences<br>(minimum length, maximum length) |
|---|---|---|
| Chromosome 1 | 27,747 | (3, 77) |
| Chromosome 2 | 25,173 | (3, 268) |
| Chromosome 3 | 21,057 | (3, 82) |
| Chromosome 4 | 19,006 | (3, 63) |
| Chromosome 5 | 18,794 | (3, 83) |
| Chromosome 6 | 17,470 | (3, 80) |
| Chromosome 7 | 17,423 | (3, 111) |
| Chromosome 8 | 15,348 | (3, 64) |
| Chromosome 9 | 13,682 | (3,78) |
| Chromosome 10 | 14,450 | (3, 72) |
| Chromosome 11 | 14,901 | (3, 95) |
| Chromosome 12 | 16,088 | (3, 74) |
| Chromosome 13 | 9,613 | (3, 79) |
| Chromosome 14 | 10048 | (3, 83) |
| Chromosome 15 | 9,749 | (3, 62) |
| Chromosome 16 | 11,527 | (3, 68) |
| Chromosome 17 | 11,178 | (3, 63) |
| Chromosome 18 | 7,424 | (3, 118) |
| Chromosome 19 | 9,618 | (3, 69) |
| Chromosome 20 | 8,346 | (3, 77) |
| Chromosome 21 | 3,887 | (3, 75) |
| Chromosome 22 | 5,242 | (3, 70) |
| Chromosome X | 17,083 | (3,117) |
| Chromosome Y | 2,451 | (3, 65) |
| Human Genome | 327,305 | (3, 268) |

**Table 6.3:** A summary of the total number of order-pruned sequences and their lengths of regions detected in the human genome (*hg*38), summarized by chromosomes.

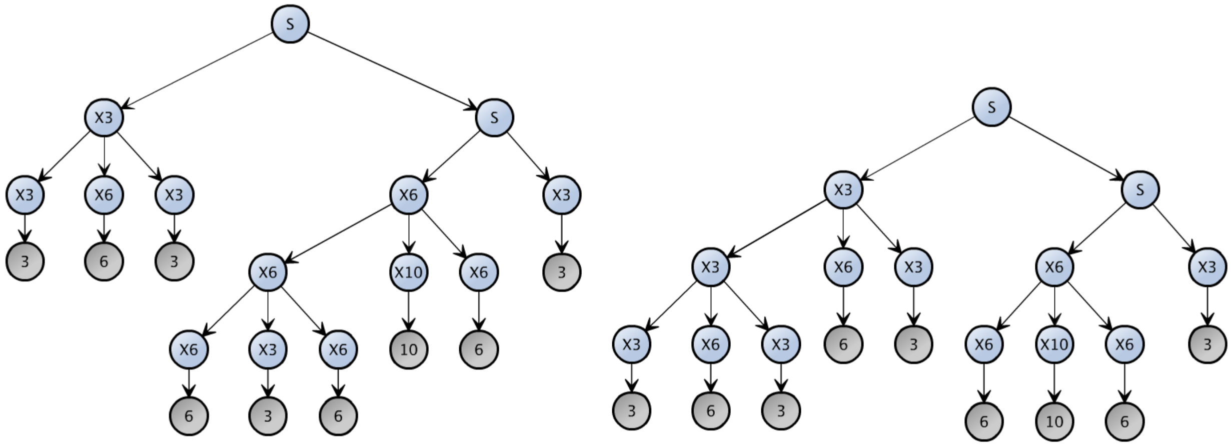


**Figure 6.7:** The two most-likely trees with the same probability of the order-pruned sequence $\bar{s_o} = 3, 6, 3, 6, 3, 6, 10, 6, 3$ on the Y chromosome of $hg38$, where $X_3 = MSTB$, $X_6 = AluSx1$, $X_{10} = AluSx$.

## 6.6 Adjustments to the parse trees

Nested interruptions are captured using a tree by the recursive interruption model, which produces small parse trees representing evolutions of TE interruptions in order-pruned sequences. However, there are redundant nodes and unnecessary "level splits" in the parse trees that are an artefact of the grammar itself (described below). In other words, the parse trees have all the information regarding predicted interruptions, but not encoded in the most natural way. It is not obvious if another stochastic grammar could be instead created that has parse trees that are naturally evolutionary trees (where the parse trees with the highest probabilities correspond to the fewest insertions).

Recall the recursive interruption stochastic context-free grammar in Definition 22: given a set of TEs with a fixed order on its elements, $\chi = \{X_1, X_2, \ldots, X_m\}$, the *recursive interruption stochastic context-free grammar* is a grammar $G = (V, T, P, S)$, where $V = \{S, X_1, X_2, \ldots, X_m\}$, $T = \{1, 2, \ldots, m\}$, and $P$ contains the following productions where each production is attached with a probability $p$:

$$S \rightarrow X_i S, \quad 1 \leq i \leq m, \quad p = \frac{1}{m^2}, \tag{6.4.1}$$

$$S \rightarrow X_i X_j, \quad 1 \leq i \leq m,\ 1 \leq j \leq m,\ i \neq j, \quad p = \frac{1}{m^2}, \tag{6.4.2}$$

$$X_i \rightarrow X_i X_j X_i, \quad 1 \leq i \leq m,\ 1 \leq j \leq m,\ i \neq j, \quad p = \frac{1}{m}, \tag{6.4.3}$$

$$X_i \rightarrow i, \quad 1 \leq i \leq m, \quad p = \frac{1}{m}. \tag{6.4.4}$$

The productions of type 6.4.1 and 6.4.3 determine the generation of interruptions from the left to the right side of the sequence. This places independent interruptions at differing heights of the parse tree — as interruptions occur from left to right sequentially, they move lower and lower down in the parse tree of the grammar. Therefore, the parse trees are not an accurate reflection of the independent nature of these types of interruptions, even though that information is encoded in the tree. In this section, this situation will be addressed by proposing a modification to turn the parse trees into another tree that is more representative of evolutionary trees of interruptions, in order to capture the TE evolution more accurately.

Next, three examples will be analyzed to intuitively explain the conversion. Example 12 illustrates three atomic patterns of interruptions: a single interruption, sequential interruptions, and recursive interruptions. These three patterns can exist by themselves, can nest with themselves, or can mix with other pattern(s) to form more complex interruptions in a genomic sequence. Instead of representing interruptions using a parse tree strictly following the grammar in Definition 22, a simplified form of trees is used in Example 12, showing these interruptions essentially in the same way.

**Example 12.** *Table 6.4 is a list of TE fragments from chromosome 1 taken from the RepeatMasker TE fragments set, $\bar{\chi}(s \overset{RM}{\longleftrightarrow} \chi_s)$. The fragments are grouped into three interruptions sets marked as (a), (b), and (c), corresponding to the trees in Figure 6.8 (a), (b), and (c), where instead of orders of TEs, the nodes of the trees are labelled with the names of TE families to which these fragments belong.*

| *group* | *genoName* | *genoStart* | *genoEnd* | *genoLeft* | *strand* | *TEName* | *TEClass* | *TEStart* | *TEEnd* | *TELeft* |
|---|---|---|---|---|---|---|---|---|---|---|
| *(a)* | *chr1* | *23803* | *24038* | *-249226583* | *+* | *L2b* | *LINE* | *2940* | *3212* | *-175* |
| | *chr1* | *24087* | *24250* | *-249226371* | *+* | *MIR* | *SINE* | *49* | *260* | *-2* |
| | *chr1* | *24254* | *24448* | *-249226173* | *+* | *L2b* | *LINE* | *3213* | *3425* | *-1* |
| *(b)* | *chr1* | *140784* | *141290* | *-249109331* | *+* | *MER21C* | *LTR* | *26* | *527* | *-411* |
| | *chr1* | *141290* | *141597* | *-249109024* | *-* | *AluJb* | *SINE* | *-18* | *294* | *2* |
| | *chr1* | *141597* | *141667* | *-249108954* | *+* | *MER21C* | *LTR* | *528* | *605* | *-333* |
| | *chr1* | *141667* | *141970* | *-249108651* | *+* | *AluJr* | *SINE* | *1* | *302* | *-10* |
| | *chr1* | *141970* | *142271* | *-249108350* | *+* | *MER21C* | *LTR* | *606* | *919* | *-19* |
| *(c)* | *chr1* | *389450* | *389591* | *-248861030* | *+* | *L1ME3D* | *LINE* | *3222* | *3368* | *-2778* |
| | *chr1* | *389589* | *391571* | *-248859050* | *+* | *L1MA8* | *LINE* | *4080* | *6108* | *-183* |
| | *chr1* | *391571* | *392307* | *-248858314* | *-* | *L1MA2* | *LINE* | *-1* | *6303* | *5556* |
| | *chr1* | *392307* | *392431* | *-248858190* | *+* | *L1MA8* | *LINE* | *6109* | *6238* | *-53* |
| | *chr1* | *392465* | *393206* | *-248857415* | *+* | *L1ME3D* | *LINE* | *3352* | *4119* | *-2027* |

**Table 6.4:** An example of three atomic patterns of interruptions. Group (a) is a single interruption; group (b) shows two sequential interruptions; group (c) shows two recursive interruptions. Their corresponding trees are in Figure 6.8.

- *Group (a) is a single interruption, where an instance of MIR inserted itself into an instance of L2b.*

- *Group (b) shows two sequential interruptions (similar to Example 6), where an instance of AluJb and an instance of AluJr inserted themselves into an instance of MER21C and broke MER21C into three fragments.*

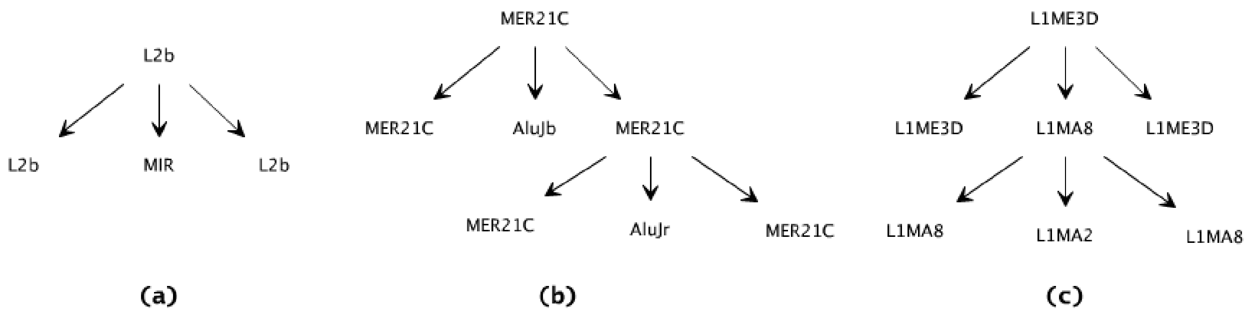


**Figure 6.8:** The corresponding trees in Table 6.4. (a) is a tree of a single interruption; (b) is a tree of two sequential interruptions following the productions of type 6.3.3 of the grammar in Definition 21; (c) is a tree of two recursive interruptions following the productions of type 6.3.3 of the grammar in Definition 21.

- *Group (c) shows two recursive interruptions (similar to Example 8), where an instance of L1MA8 inserted itself into an instance of L1ME3D, then at a later time an instance of L1MA2 inserted itself into an instance of L1MA8.*

*For a single interruption, the root node of the ordered tree represents the interruptee and the children of that node correspond to the fragments of the interruption and their order in the genomic sequence, which are the left fragment of the interruptee, the interrupter, and the right fragment of the interruptee. The interruption shown in Table 6.4 group (a) corresponds to the TE fragments tree in Figure 6.8 (a). Here, the root node is the interruptee, labelled as the name of the TE fragment to which it belongs, L2b, and the three children of the root are (from left to right) the left fragment of the interruptee, L2b, the interrupter, MIR, and the right fragment of the interruptee, L2b.*

*Nested interruptions correspond to higher level TE fragments trees following the same rule, as in Figure 6.8 (b) and (c).*

Notice that the trees in Figure 6.8 capture not only the nested TEs by the levels of the tree, but also the orders of the TE fragments within a genomic sequence. An order-pruned sequence of the genomic sequence can be generated by traversing the leaves of the tree (and mapping TE names with their orders in the TE set), from the left to right of the tree. They contain all the fragments of interruptions as branches. Nevertheless, from the perspective of the relationship between interrupters and interruptees, some of the branches are redundant (such as the left and right fragments of the interruptees), because the interruptee already appears as the parent node. Moreover, the two sequential interruptions in group (2) are split into two levels in Figure 6.8 (b); however, this is simply a side effect of the structure of the parse trees and the rules of the grammar. They are in fact independent. In addition, since only the interruptional phylogeny of TEs is of current interest, the positional order does not matter in this case; thus, an ordered tree is not necessary. Therefore, the tree can be simplified by turning it into an unordered tree, removing the redundant branches and correcting the level split of the sequential interruptions. Definition 23 defines a tree operation, named *condense*, that compresses a tree by reducing the height while still preserving the order of leaves (corresponding to the order-pruned sequence) of the tree.

**Definition 23.** *Given a tree $t$ ($t$ is the root) with a non-root node $\mu$ in $t$, let the* condensing of $\mu$ from $t$ *be the tree obtained from $t$ by replacing $\mu$ with its children, if there are any, and keeping $t$, if there are not. This is drawn in Figure 6.9.*

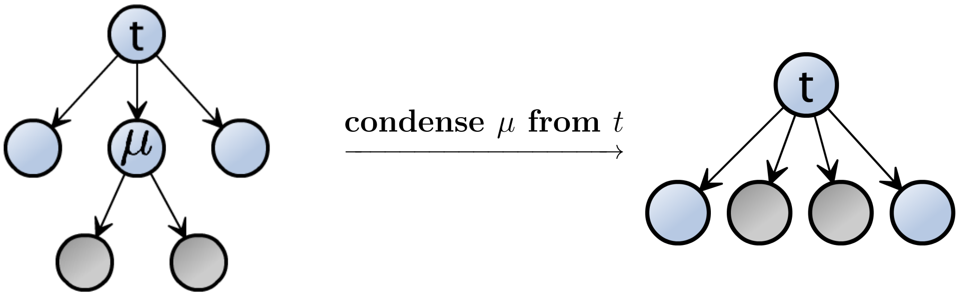


**Figure 6.9:** An example of condensing a non-root node, $\mu$, from a tree $t$.

**Algorithm 1:** Convert a SCFG parse tree to an interruptional evolutionary tree.

```
Data: A SCFG parse tree $t$.
Result: An interruptional evolutionary tree.

/* Step 1: "bring up" the sequential interruptions of the same interruptee to the same
   level of the tree by condensing the inner nodes corresponding to sequential
   interruptions in the tree.                                                        */
for each node $\mu$ of $t$ with children $\mu_1$, $\mu_2$, $\mu_3$ representing a production of the type 6.4.3, from the highest to
  the lowest level of $t$ do
    condense $\mu_1$ and $\mu_3$

/* Step 2: remove all leaves that represent the left and right fragments of
   interruptees.                                                                     */
for each node $\mu$ of $t$ and for each of $\mu$'s children $\mu_1, \mu_2, \ldots, \mu_k$ do
    if the label of $\mu_i$ and $\mu$ are equal, and $\mu_i$ does not have any children then
        remove the child $\mu_i$
return the modified $t$
```

Algorithm 1 turns interruption parse trees, such as the trees in Example 12, into a simplified form of trees, where the redundant branches are removed and the sequential interruptions are moved up into the same level, making them closer to describing evolutionary events.

Example 13 demonstrates how to convert a tree of sequential interruptions in Example 12 into an interruptional evolutionary tree using Algorithm 1. Note that the TE orders, instead of TE names, are used to label the nodes in the next examples, as it is more clear. The nodes representing interrupters of the sequential interruptions are coded in pink in the diagram.

**Example 13.** *Figure 6.10 is an example showing how to convert the context-free grammar parse tree to an interruptional evolutionary tree showing the atomic sequential interruptions pattern using the two steps in Algorithm 1.*

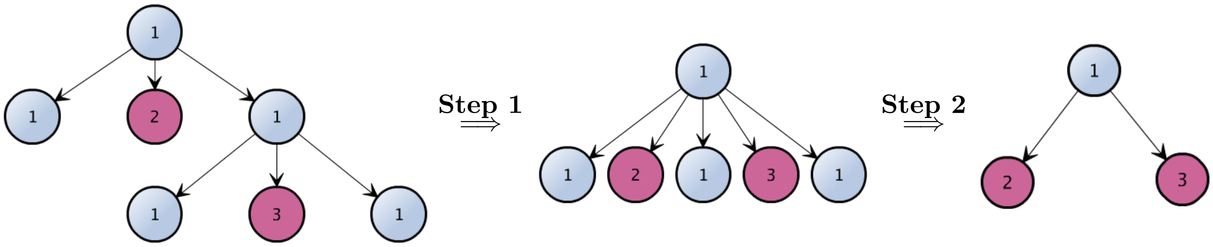


**Figure 6.10:** An example of converting the context-free grammar parse tree to an interruptional evolutionary tree for the atomic sequential interruptions pattern.

In Example 14, there are two more examples of simplifying the trees using Algorithm 1. These two trees are mixed patterns of the atomic patterns of Example 12.

**Example 14.** *Figure 6.11 and Figure 6.12 are two extended examples showing how to simplify the TE interruption trees — where the interruptions are nested in more complex patterns — to interruptional evolutionary trees.*

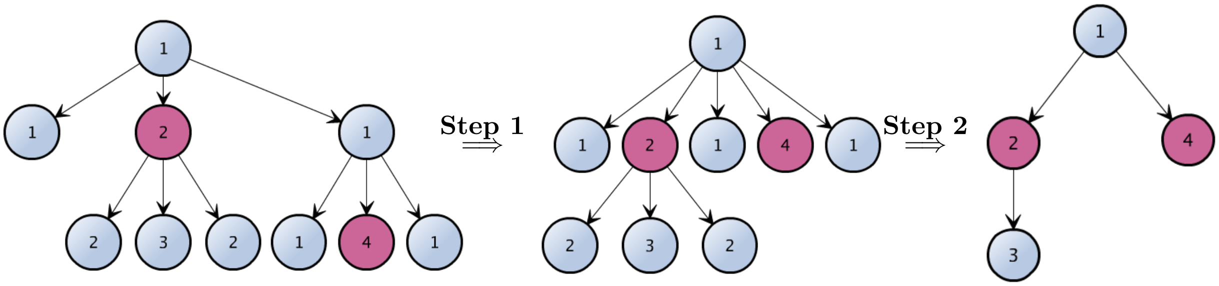


**Figure 6.11:** An example of converting an interruption tree to an interruptional evolutionary tree of mixed interruption patterns.

## 6.7 Construction of the TE-interaction network

Based on the standard formalizations defining TEs and their activities, the next goal is to computationally predict the overall evolution of the TEs and TE activity in the human genome. In this section, an algorithm will be proposed to merge all the small interruptional evolutionary trees obtained from last section and produce a weighted directed graph, called *the overall TE-interaction network*. The TE-interaction network is a rich representation of the interactions between TEs and is a more powerful tool than the individual trees separately to predict the evolution of these TEs.

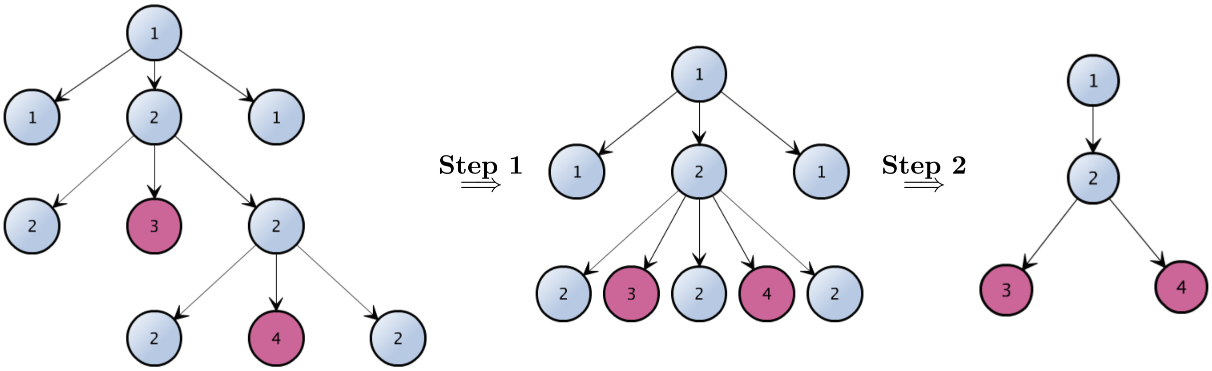


**Figure 6.12:** Another example of converting an interruption tree to an interruptional evolutionary tree of mixed interruption patterns.

Consider the set of all interruptional evolutionary trees generated from the output of Algorithm 1 on the SCFG trees from the recursive interruption model. These simplified interruptional evolutionary trees represent a hypothesis for how younger TEs interrupt older TEs in the human genome. Each parse tree is generated from an order-pruned sequence of a transposon region of the genome, where the TE fragments are close to each other in the genomic sequence, therefore, instead of a “global” evolutionary relationship, these simplified evolutionary trees each represents a “local” pattern of TE interruptions. Here, another model will be created to show the overall TE interactions globally in a genome. Informally, all the local interruptional trees will be merged into a global structure; however in so doing, a graph is required instead of a tree. In the area of graph theory, a *weighted directed graph*, or a *weighted directed network*, is a graph that is a set of vertices connected by edges, where the edges have a direction and a weight associated with them. By merging the simplified evolutionary trees (output by Algorithm 1) into a weighted directed graph, the TE-interaction network is constructed that shows the overall TE interactions globally in a genome.

The network construction is described as follows:

- Let $G = (V, E, \pi)$ be a weighted graph, where the edges have non-negative integer weights, and the vertices are all the TEs that appear in the trees, denoted as $V = \{TE_1, TE_2, \ldots, TE_n\}$.

- The edges are built iteratively. For each tree, one at a time (no matter in any order until all trees are merged), add all the edges from the tree into the graph either with a weight of 1 if it is new, or if the edge already exists, add 1 to the weight of that edge. For example, if $(TE_i, TE_j)$ is an edge of the tree, then add $(TE_i, TE_j)$ as an edge to $G$ with weight 1, or if the edge already exists in the graph, increase the weight of the edge by 1.

At the end, the graph shows all the edges present in any tree, and the weights show how many times the edge is used in the trees. Moreover, each TE occurs exactly once in the graph if it is involved in any interruption. This weighted directed graph is called the *TE-interaction network*, which encodes the interactions between

TEs that interrupt each other. The network itself is an overall representation of the evolution of the TEs. One would expect that TEs that occur "early" in paths represent older TEs, whereas TEs that occur "later" in paths represent newer TEs. If the graph were acyclic, then a linear age order of the TEs would be some listing of the vertices whereby, for each edge $(u, v)$ of the graph, $u$ comes before $v$ in the listing. This is known as a *topological sort* on the graph. However, it is reasonable to assume that the network contains cycles. This is because the lifespan of TEs could overlap over time in reality, which means multiple TEs might be active in parallel. For example, if two TEs, $TE_i$ and $TE_j$, are active at the same time, then it is possible that in some region where $TE_i$ already exists in some genome, $TE_j$ inserted itself into $TE_i$ (a simplified tree $TE_i \rightarrow TE_j$), or vice versa (a simplified tree $TE_j \rightarrow TE_i$). Then by merging these trees together, there is a cycle $TE_i \rightarrow TE_j \rightarrow TE_i$ in the network. They can also be more complex with intermediate TEs.

Moreover, in the network, the nodes that only contain outgoing edges are informally referred to as *upstream nodes* corresponding to the TEs with the oldest predicted age or a TE with relatively old age and a short lifespan (so that it did not have time to interrupt others when it was active), and the nodes that only have incoming edges are called *downstream nodes* corresponding to the TEs with the youngest predicted age or a TE with relatively young age and a long lifespan (when it has lots of chances to interrupt others). Moreover, the inner nodes in the network correspond to the TEs with intermediate predicted age, among which the ones that are "closer" to the upstream nodes are predicted to be relatively older, while the ones that are "closer" to the downstream nodes are predicted to be relatively younger.

Next, the TE-interaction network of two examples will be examined, with a comparison to the sequential interruption model in one example and same validation using biological literature. A more systematic comparison of the two approaches will be done in Chapter 7.

Example 15 illustrates the construction of a TE-interaction network on a small set of TEs (20 TEs) on the human Y chromosome, and some predictions on the ages and lifespans of these TEs are made from the network, along with some verification on the predicted ages using biological literature.

**Example 15.** *Select 20 TEs in the human genome, denoted as* $\chi_s = \{X_1, X_2, \ldots, X_{20},\}$*, where the information of these TEs are listed in Table 6.5.*

*All the order-pruned sequences only containing TEs* $\in \chi_s$ *were identified on the Y chromosome. There are a total of 37 order-pruned sequences as listed in Table 6.6.*

*Figure 6.13 is the TE-interaction network of* $\chi_s$*, generated by merging the small trees in Table 6.6.*

*The TEs in* $\chi_s$ *belong to two TE types, SINE and LTR, which are colour coded in the figure, where the elements of SINE type are marked in grey and LTR are in blue. Moreover, each element is marked with its name and type. The network is drawn so that upstream TEs are towards the top and downstream TEs are towards the bottom.*

| *Notation* | *TE* | *family/subfamily* | *type* |
|---|---|---|---|
| $X_1$ | *MLT1C* | *ERVL − MaLR* | *LTR* |
| $X_2$ | *MLT1A0* | *ERVL − MaLR* | *LTR* |
| $X_3$ | *MSTB* | *ERVL − MaLR* | *LTR* |
| $X_4$ | *MSTB1* | *ERVL − MaLR* | *LTR* |
| $X_5$ | *MLT1D* | *ERVL − MaLR* | *LTR* |
| $X_6$ | *AluSx1* | *AluS* | *SINE* |
| $X_7$ | *AluSg4* | *AluS* | *SINE* |
| $X_8$ | *FLAM_C* | *Alu* | *SINE* |
| $X_9$ | *MER74A* | *ERVL* | *LTR* |
| $X_{10}$ | *AluSx* | *AluS* | *SINE* |
| $X_{11}$ | *AluJb* | *AluJ* | *SINE* |
| $X_{12}$ | *FRAM* | *Alu* | *SINE* |
| $X_{13}$ | *MER74B* | *ERVL* | *LTR* |
| $X_{14}$ | *AluSp* | *AluS* | *SINE* |
| $X_{15}$ | *AluSq* | *AluS* | *SINE* |
| $X_{16}$ | *AluY* | *AluY* | *SINE* |
| $X_{17}$ | *MER67C* | *ERV1* | *LTR* |
| $X_{18}$ | *AluSc8* | *AluS* | *SINE* |
| $X_{19}$ | *AluJo* | *AluJ* | *SINE* |
| $X_{20}$ | *AluYc* | *AluY* | *SINE* |

**Table 6.5:** List of TE names along with known information.

| | *order-pruned sequence detected on the Y chromosome of hg*38 | *Edges of the tree* |
|---|---|---|
| *1* | *19 6 19* | 19 → 6 |
| *2* | *11 10 11* | 11 → 10 |
| *3* | *6 16 6* | 6 → 16 |
| *4* | *2 15 2* | 2 → 15 |
| *5* | *11 6 11* | 11 → 6 |
| *6* | *19 10 19* | 19 → 10 |
| *7* | *2 20 2* | 2 → 20 |
| *8* | *2 19 2* | 2 → 19 |
| *9* | *2 11 2* | 2 → 11 |
| *10* | *5 19 5* | 5 → 19 |
| *11* | *11 6 11* | 11 → 6 |
| *12* | *6 16 6* | 6 → 16 |
| *13* | *4 14 4* | 4 → 14 |
| *14* | *19 16 19* | 19 → 16 |
| *15* | *19 16 19* | 19 → 16 |
| *16* | *16 11 16* | 16 → 11 |
| *17* | *11 10 6 10* | 10 → 6 |
| *18* | *7 7 7 6* | |
| *19* | *6 10 11 6* | |
| *20* | *1 6 1 1* | 1 → 6 |
| *21* | *1 6 1 1* | 1 → 6 |
| *22* | *5 10 16 10* | 10 → 16 |
| *23* | *2 10 2 14* | 2 → 10 |
| *24* | *11 15 11 6* | 11 → 15 |
| *25* | *1 1 8 1* | 1 → 8 |
| *26* | *2 2 2 6 2* | 2 → 6 |
| *27* | *1 1 5 6 5* | 5 → 6 |
| *28* | *19 19 19 11 19* | 19 → 11 |
| *29* | *10 10 11 10 11* | 10 → 11, 11 → 10 |
| *30* | *5 1 1 1 1 1* | |
| *31* | *1 16 1 1 1 1* | 1 → 16 |
| *32* | *16 19 11 11 7 11* | 11 → 7 |
| *33* | *10 10 15 10 6 10 15 15 15* | 10 → 6, 15 → 10 |
| *34* | *1 1 17 18 17 16 17 10 16* | 17 → 18, 17 → 16 |
| *35* | *3 6 3 6 3 6 10 6 3* | 3 → 6, 6 → 3, 6 → 10 |
| *36* | *1 2 2 3 3 3 3 3 3 3 3 3 3 4 3 4 3 4 3 3 3 3 3 3 4 3 3 3 3 4 3 4 3 4 4 3 3 3 3 3 3 3 3 3 3 3 4 4 3 4 3 3 3 3 3 3 3 3 4 3* | 3 → 4,4 → 3 |
| *37* | *5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 7 8 9 10 11 8 12 6 13 14 15 16 6 17 16 17 18 18 19 11 20* | 16 → 17 |

**Table 6.6:** The order-pruned sequences only containing $\chi_s$ in Table 6.5 on the human Y chromosome.

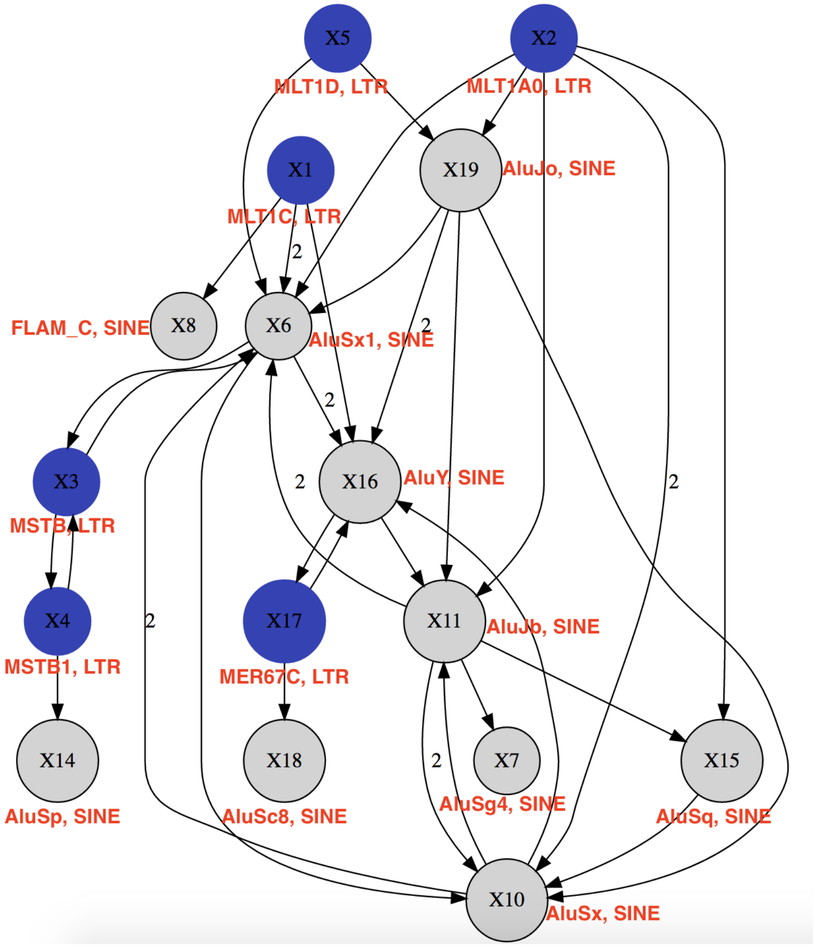


**Figure 6.13:** The directed graph generated by merging the interruption trees in Table 6.6, and visualized using Graphviz [44]. The edge weights greater than 1 are marked on the edges. The elements of SINE type are marked in grey and LTR are in blue. Each vertex in the graph is marked with its corresponding TE name and type.

*The predicted interruptional evolution of these TEs is encoded in the network:*

- *upstream elements in the graph are predicted to be the oldest in age, e.g.,* $X_1$ *(MLT1C, ERVL-MaLR, LTR),* $X_2$ *(MLT1A0, ERVL-MaLR, LTR),* $X_5$ *(MLT1D, ERVL-MaLR, LTR).*
- *downstream elements in the graph are predicted to be the youngest in age, e.g.,* $X_7$ *(AluSg4, AluS, SINE),* $X_8$ *(FLAM_C, Alu, SINE),* $X_{14}$ *(AluSp, AluS, SINE),* $X_{15}$ *(AluSq, AluS, SINE),* $X_{18}$ *(AluSc8, AluS, SINE).*
- *elements in the same cycle in the graph indicate that they have overlapped lifespans:*

**cycle 1:** $X_6$ *(AluSx1, AluS, SINE)* → $X_{10}$ *(AluSx, AluS, SINE)* → $X_{11}$ *(AluJb, AluJ, SINE)* → $X_6$ *(AluSx1, AluS, SINE);*

**cycle 2:** $X_6$ *(AluSx1, AluS, SINE)* → $X_{16}$ *(AluY, AluY, SINE)* → $X_{11}$ *(AluJb, AluJ, SINE)* → $X_6$

*(AluSx1, AluS, SINE);*

**cycle 3:** $X_3$ *(MSTB, ERVL-MaLR, LTR)* → $X_4$ *(MSTB1, ERVL-MaLR, LTR)* → $X_3$ *(MSTB, ERVL-MaLR, LTR);*

**cycle 4:** $X_{16}$ *(AluY, AluY, SINE)* → $X_{17}$ *(MER67C, ERV1, LTR)* → $X_{16}$ *(AluY, AluY, SINE).*

- *paths in the network indicate orders of TE ages. The elements marked in "⟦ ⟧" are in the same cycle in the network, which indicates that these elements have overlapped lifespans.*

**path 1:** $X_2$ *(MLT1A0, ERVL-MaLR, LTR)* → $X_{19}$ *(AluJo, AluJ, SINE)* → ⟦ $X_6$ *(AluSx1, AluS, SINE)* → $X_{10}$ *(AluSx, AluS, SINE)* → $X_{16}$ *(AluY, AluY, SINE)* → $X_{17}$ *(MER67C, ERV1, LTR)* ⟧ → $X_{18}$ *(AluSc8, AluS, SINE);*

**path 2:** $X_1$ *(MLT1C, ERVL-MaLR, LTR)* → ⟦ $X_6$ *(AluSx1, AluS, SINE)* → $X_{16}$ *(AluY, AluY, SINE)* → $X_{11}$ *(AluJb, AluJ, SINE)* ⟧ → $X_{15}$ *(AluSq, AluS, SINE)*

**path 3:** $X_5$ *(MLT1D, ERVL-MaLR, LTR)* → $X_{19}$ *(AluJo, AluJ, SINE)* → $X_{11}$ *(AluJb, AluJ, SINE)* → $X_7$ *(AluSg4, AluS, SINE)*

**path 4:** ⟦ $X_3$ *(MSTB, ERVL-MaLR, LTR)* → $X_4$ *(MSTB1, ERVL-MaLR, LTR)* ⟧ → $X_{14}$ *(AluSp, AluS, SINE)*

*As previous mentioned in Chapter 2, LTR elements are known to be relatively old elements that are probably distinct in the human genome, while Alus in SINEs are relatively young elements, which matches the above predicted relative age of the upstream and downstream elements. It was also mentioned in Chapter 3 that there are three Alu subfamilies. AluJ is the most ancient (about 65 million years old), and is thought to be functionally extinct; the second oldest is the AluS subfamily, which became active approximately 30 million years ago, and only some intact elements were found to be active in humans; AluY is the youngest subfamily, and most elements of this subfamily are currently active. The history of Alu elements matches with the above predicted relative ages as well, e.g., paths 1 to 4. The predicted evolution of the TEs in* $\chi_s$ *in the network also matches with relative order published in [47] as shown in Figure 2.8, in terms of the cycles (overlapped lifespan) of the three subfamilies of Alu elements.*

In the area of graph theory, a directed graph is *strongly connected* if there is a path between all pairs of vertices. A *strongly connected component* of a directed graph is a maximal strongly connected subgraph. It should be noted that some cycles in the network can form into a strongly connected component, such as cycle 1 and cycle 2 in Example 15. However, cycles are different from strongly connected components in predicting lifespans of TEs. For example, $X_6$ is predicted to have a longer lifespan in cycle 1 and cycle 2 in Example 15 than in the strongly connected component consisted of cycle 1 and cycle 2. Therefore, no further attempts were made to construct strongly connected components here.

Furthermore, the *adjacency matrix* equivalent to the graph of a TE-interaction network contains the number of times (the weight of the edge) each TE inserts into each other TE. Hence, this matrix can also be reordered using the Linear Ordering Problem from Chapter 5, and indeed the Tabu search program. This predicts a linear age order of the TEs in $\chi_s$ that were involved in interactions.

There are two major differences between the adjacency matrix (recursive interruption model) and the interruption matrix (sequential interruption model):

- first, the interruption-detection techniques are different. The detection of interruptions in the sequential problem, or with the interruption model has more conditions because they are detected by comparing the positions of TE remnants on both the genomic sequence and the TE consensus as defined in Definition 16, while the interruptions in the recursive interruption model are detected by parsing the order-pruned sequences of the genomic sequence using the SCFG in Definition 22. Then, the recursive model includes all nested interruptions as well;

- second, the values in the interruption matrix in the sequential interruption model represent the number of times younger elements (interrupters on rows) interrupted older elements (interruptees on columns), while the adjacency matrix converted from the TE-interaction network represent that older elements has edges in the evolutionary tree pointing to younger elements. Hence, if feeding the two matrices to the LOP, the age order calculated from interruption matrix is from the youngest to oldest TEs, and from the adjacency matrix is in contrast from the oldest to youngest TEs.

The result of applying the Tabu search algorithm to the TE-interaction network of Figure 6.13 is listed in Table 6.7.

It can be seen from Table 6.7 that by applying the Tabu search algorithm for LOP to the data detected by the recursive interruption model, the TEs are ordered from LTR to SINE (except for $MER67C$), and the three major subfamilies of $Alu$ are ordered by their known ages from $AluJ$ to $AluS$ to $AluY$. The TEs in Table 6.7 are then compared with the age order of the entire set of human TEs (Appendix A) predicted by the sequential interruption model using Tabu search. The comparison is listed in Table 6.8, where the age orders are from the oldest to youngest, and the TEs are sorted by the order calculated on interruption matrix.

From the comparison of the same method applied to the two different matrices, it can be seen that the predicted age calculated on the interruption matrix moved $MER67C$ to an older age, so that all TEs of the LTR type are older than those of the SINE type. In addition, the $FLAM\_C$ element is also moved up to be the oldest Alu element in the table. The ordering on the interruption matrix also agree with the facts that LTR are old elements and SINEs are young TEs.

This single result supports the interruptional analysis method. However, though a linear age order is a clear

| Age order | Notation | TE | family/subfamily | type |
|---|---|---|---|---|
| from oldest to youngest | | | | |
| 1 | $X_1$ | $MLT1C$ | $ERVL-MaLR$ | LTR |
| 2 | $X_4$ | $MSTB1$ | $ERVL-MaLR$ | LTR |
| 3 | $X_2$ | $MLT1A0$ | $ERVL-MaLR$ | LTR |
| 4 | $X_3$ | $MSTB$ | $ERVL-MaLR$ | LTR |
| 5 | $X_5$ | $MLT1D$ | $ERVL-MaLR$ | LTR |
| 6 | $X_{19}$ | $AluJo$ | $AluJ$ | SINE |
| 7 | $X_{11}$ | $AluJb$ | $AluJ$ | SINE |
| 8 | $X_8$ | $FLAM\_C$ | $Alu$ | SINE |
| 9 | $X_{15}$ | $AluSq$ | $AluS$ | SINE |
| 10 | $X_{10}$ | $AluSx$ | $AluS$ | SINE |
| 11 | $X_6$ | $AluSx1$ | $AluS$ | SINE |
| 12 | $X_7$ | $AluSg4$ | $AluS$ | SINE |
| 13 | $X_{17}$ | $MER67C$ | $ERV1$ | LTR |
| 14 | $X_{14}$ | $AluSp$ | $AluS$ | SINE |
| 15 | $X_{18}$ | $AluSc8$ | $AluS$ | SINE |
| 16 | $X_{16}$ | $AluY$ | $AluY$ | SINE |

**Table 6.7:** The age order of TEs in the TE-interaction network in Figure 6.13 predicted by Tabu search on the adjacency matrix of the network.

| TE name | TE family | TE type | Age order calculated on adjacency matrix | Position in Appendix A | Age order calculated on interruption matrix |
|---|---|---|---|---|---|
| MER67C | ERV1 | LTR | 13 | 595 | 1 |
| MLT1D | ERVL-MaLR | LTR | 5 | 571 | 2 |
| MLT1C | ERVL-MaLR | LTR | 1 | 563 | 3 |
| MLT1A0 | ERVL-MaLR | LTR | 3 | 427 | 4 |
| MSTB1 | ERVL-MaLR | LTR | 2 | 331 | 5 |
| MSTB | ERVL-MaLR | LTR | 4 | 282 | 6 |
| FLAM_C | Alu | SINE | 8 | 226 | 7 |
| AluJo | AluJ | SINE | 6 | 224 | 8 |
| AluJb | AluJ | SINE | 7 | 208 | 9 |
| AluSx | AluS | SINE | 10 | 122 | 10 |
| AluSx1 | AluS | SINE | 11 | 100 | 11 |
| AluSq | AluS | SINE | 9 | 85 | 12 |
| AluSg4 | AluS | SINE | 12 | 68 | 13 |
| AluSp | AluS | SINE | 14 | 67 | 14 |
| AluSc8 | AluS | SINE | 15 | 54 | 15 |
| AluY | AluY | SINE | 16 | 22 | 16 |

**Table 6.8:** Comparison between the age orders of TEs in Figure 6.13 predicted by Tabu search of LOP on adjacency matrix and on interruption matrix.

representation, the lifespans of TEs are not predicted.

Example 16 describes another example of a TE-interaction network of a bigger set of TEs (41 TEs) on the human X chromosome. The TE-interaction network in this example is much more complex than that in Figure 6.13, whereas it is only based on the order-pruned sequences containing solely the 41 TEs on the one chromosome.

**Example 16.** *These select 41 TEs in the human genome, denoted as* $\chi_s = \{X_1, X_2, \ldots, X_{41},\}$*, where the information of these TEs are listed in Table 6.9.*

*The order-pruned sequences only containing TEs in* $\chi_s$ *on the X chromosome of* $hg38$ *are identified. There are a total of 174 order-pruned sequences that have interruptions encoded in them.*

*Figure 6.14 is the TE-interaction network of* $\chi_s$ *constructed by merging the interruption trees from the 174 order-pruned sequences.*

*The evolution of these TEs is encoded in the network:*

- *upstream elements in the graph are predicted to be the oldest, e.g.,* $X_{41}$ *(L1ME3E, L1, LINE),* $X_{13}$ *(LOR1-int, ERV1, LTR),* $X_{26}$ *(MER92-int, ERV1, LTR),* $X_4$ *(LTR41, ERVL, LTR),* $X_{30}$ *(LTR49, ERV1, LTR),* $X_{38}$ *(Tigger2TcMar-Tigger, DNA), etc.*

  *All of the upstream TEs belong to LTR or DNA transposons (except* $X_{41}$*), which are considered to be very old.* $X_{41}$ *belongs to the LINE type and* $L1ME$ *is one of the oldest TE families in the human genome as shown in Figure 2.8.*

- *downstream elements in the graph are predicted to be the youngest, e.g.,* $X_{20}$ *(AluSc8, AluS, SINE),* $X_{10}$ *(AluSx3, AluS, SINE),* $X_{37}$ *(LTR10F, ERV1, LTR),* $X_2$ *(AluY, AluY, SINE) etc.*

  *As previously mentioned, the AluS and AluY families are young TE families in the human genome as shown in Figure 2.8. The LTR10F appears as a downstream node in the network, but the TEs of the LTR type are probably old. This might be because this element (even with an old age) probably has a small number of copies in the human genome, so that younger elements had little chance to insert into it. We could not find any literature to verify the age of this element.*

- *elements in the same cycle in the graph indicate that they have overlapped lifespans:*

  **cycle 1:** $X_{39}$ *(AluJo, AluSJ, SINE)* → $X_{12}$ *(AluSq2, AluS, SINE)* → $X_6$ *(L1M5, L1, LINE)* → $X_{39}$ *(AluJo, AluJ, SINE);*

  **cycle 2:** $X_{12}$ *(AluSq2, AluS, SINE)* → $X_6$ *(L1M5, L1, LINE)* → $X_7$ *(AluJr, AluJ, SINE)* → $X_{23}$ *(LTR49-int, ERV1, LTR)* → $X_12$ *(AluSq2, AluS, SINE);*

| *Notation* | *TE* | *family/subfamily* | *type* |
|---|---|---|---|
| $X_1$ | $AluSg7$ | $Alu$ | *SINE* |
| $X_2$ | $AluY$ | $Alu$ | *SINE* |
| $X_3$ | $ERVL-int$ | $ERVL$ | *LTR* |
| $X_4$ | $LTR41$ | $ERVL$ | *LTR* |
| $X_5$ | $AluJb$ | $Alu$ | *SINE* |
| $X_6$ | $L1M5$ | $L1$ | *LINE* |
| $X_7$ | $AluJr$ | $Alu$ | *SINE* |
| $X_8$ | $AluSx1$ | $Alu$ | *SINE* |
| $X_9$ | $AluJr4$ | $Alu$ | *SINE* |
| $X_{10}$ | $AluSx3$ | $Alu$ | *SINE* |
| $X_{11}$ | $MLT1E3$ | $ERVL-MaLR$ | *LTR* |
| $X_{12}$ | $AluSq2$ | $Alu$ | *SINE* |
| $X_{13}$ | $LOR1-int$ | $ERV1$ | *LTR* |
| $X_{14}$ | $MER49$ | $ERV1$ | *LTR* |
| $X_{15}$ | $AluSx$ | $Alu$ | *SINE* |
| $X_{16}$ | $LTR26$ | $ERV1$ | *LTR* |
| $X_{17}$ | $AluSz6$ | $Alu$ | *SINE* |
| $X_{18}$ | $MER90$ | $ERV1$ | *LTR* |
| $X_{19}$ | $Tigger5b$ | $TcMar-Tigger$ | *DNA* |
| $X_{20}$ | $AluSc8$ | $Alu$ | *SINE* |
| $X_{21}$ | $Alu$ | $Alu$ | *SINE* |
| $X_{22}$ | $MER94$ | $hAT-Blackjack$ | *DNA* |
| $X_{23}$ | $LTR49-int$ | $ERV1$ | *LTR* |
| $X_{24}$ | $AluSz$ | $Alu$ | *SINE* |
| $X_{25}$ | $MER41B$ | $ERV1$ | *LTR* |
| $X_{26}$ | $MER92-int$ | $ERV1$ | *LTR* |
| $X_{27}$ | $LTR29$ | $ERV1$ | *LTR* |
| $X_{28}$ | $AluYb8$ | $Alu$ | *SINE* |
| $X_{29}$ | $MER34B-int$ | $ERV1$ | *LTR* |
| $X_{30}$ | $LTR49$ | $ERV1$ | *LTR* |
| $X_{31}$ | $THE1D$ | $ERVL-MaLR$ | *LTR* |
| $X_{32}$ | $LTR54$ | $ERV1$ | *LTR* |
| $X_{33}$ | $Tigger2b_Pri$ | $TcMar-Tigger$ | *DNA* |
| $X_{34}$ | $LTR2$ | $ERV1$ | *LTR* |
| $X_{35}$ | $Harlequin-int$ | $ERV1$ | *LTR* |
| $X_{36}$ | $MER50$ | $ERV1$ | *LTR* |
| $X_{37}$ | $LTR10F$ | $ERV1$ | *LTR* |
| $X_{38}$ | $Tigger2$ | $TcMar-Tigger$ | *DNA* |
| $X_{39}$ | $AluJo$ | $Alu$ | *SINE* |
| $X_{40}$ | $THE1D-int$ | $ERVL-MaLR$ | *LTR* |
| $X_{41}$ | $L1ME3E$ | $L1$ | *LINE* |

**Table 6.9:** List of TE names along with their information.

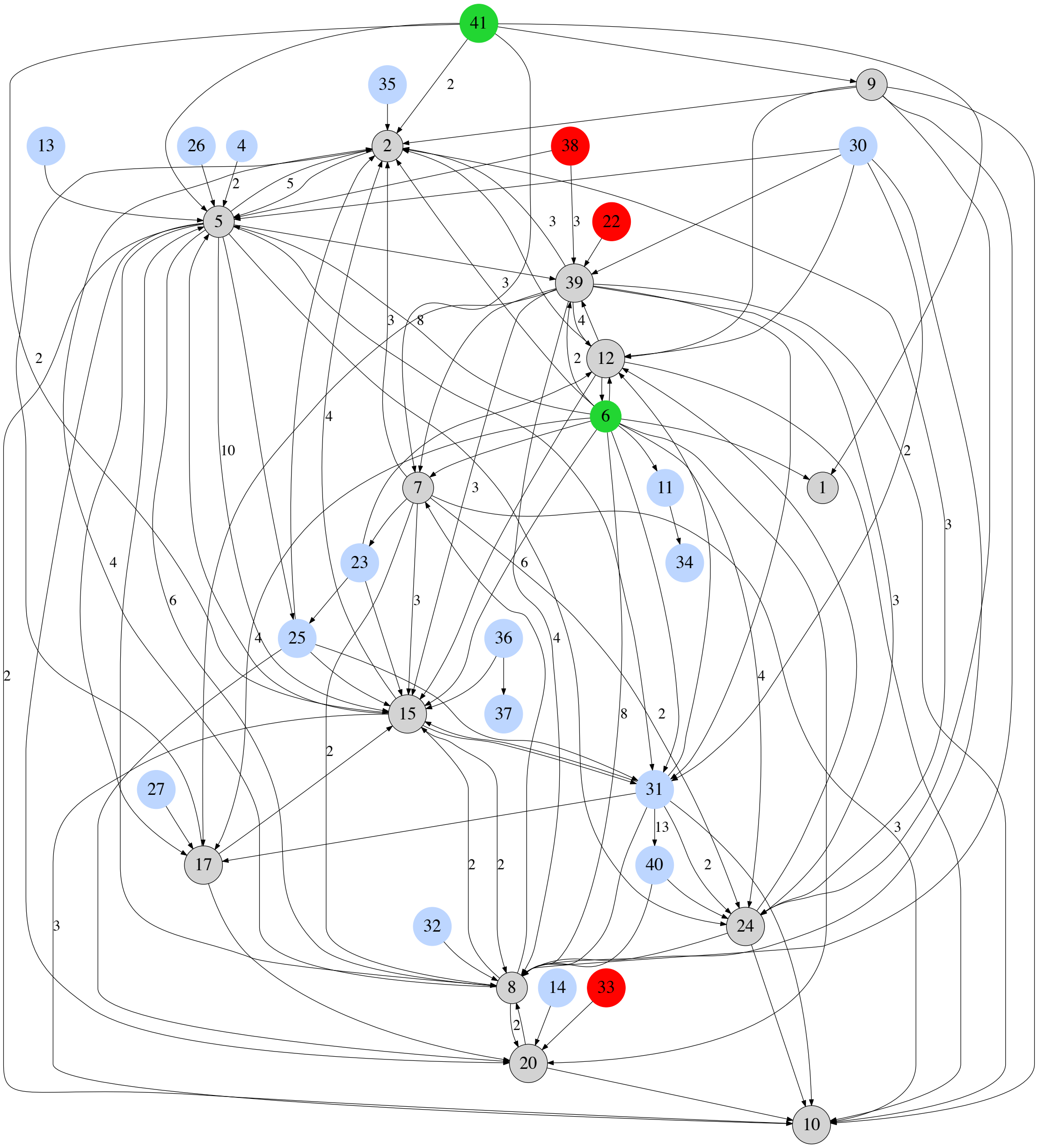


**Figure 6.14:** The directed graph generated by merging the interruption trees of the TEs in Table 6.9 using Graphviz [44]. The edge weights greater than 1 are marked on the edges. The elements of SINE type are marked in grey, LTR are in blue, LINEs are in green, and DNA are in red.

**cycle 3:** $X_{15}$ *(AluSx, AluS, SINE)* → $X_{31}$ *(THE1D, ERVL-MaLR, LTR)* → $X_{17}$ *(AluSz6, AluS, SINE)* → $X_{15}$ *(AluSx, AluS, SINE);*

**cycle 4:** $X_{18}$ *(MER90, ERV1, LTR)* → $X_{20}$ *(AluSc8, AluS, SINE)* → $X_{18}$ *(MER90, ERV1, LTR).*

- *paths in the network indicate age orders. The elements marked in "⟦ ⟧" are in the same cycle in the network, which indicates that these elements have overlapped lifespans.*

  **path 1:** $X_{38}$ *(Tigger2, TcMar-Tigger, DNA)* → ⟦$X_{39}$ *(AluJo, AluJ, SINE)* → $X_{12}$ *(AluSq2, AluS, SINE)* → $X_6$ *(L1M5, L1, LINE)*⟧ → $X_8$ *(AluSx1, AluS, SINE)* → $X_{20}$ *(AluSc8, AluS, SINE)* → $X_{10}$ *(AluSx3, AluS, SINE);*

  **path 2:** $X_{41}$ *(L1ME3E, L1, LINE)* → $X_7$ *(AluJr, AluJ, SINE)* → ⟦$X_{15}$ *(AluSx, AluS, SINE)* → $X_{31}$ *(THE1D, ERVL-MaLR, LTR)* ⟧ → $X_{24}$ *(AluSz, AluS, SINE)* → $X_{10}$ *(AluSx3, AluS, SINE)*

  *It can be seen in Figure 2.8 that the Tigger elements of DNA transposon and the L1ME family of LINEs are very old elements, which matches the predicted age in the above paths.*

## 6.8 The TE-interaction network of the human genome

The network of the whole set of human TEs on the entire human genome (1,080 TEs) is also constructed. As the network is very complicated to visualize and analyze, it is represented as an adjacency matrix which is then fed to Tabu search. The predicted age order of the whole set of human TEs calculated by Tabu search based on the adjacency matrix is attached in Appendix B.

## 6.9 Discussion

So far, three theoretical models have been proposed: the TE fragment model helps to describe the TE problems clearly in a precise way; the sequential interruption model captures the interruptional activities between every pair of TEs; and the recursive interruption model further captures the nested nature of the interruptional activities of older TEs which cannot be represented by the interruptional matrix in the sequential interruption model.

The TE-interaction network is a richer representation of the phylogeny of TEs than only the interruption trees, as it shows the overall interactions globally and the evolutionary history of all TEs rather than a set of more local interruptional information. The network can also be represented as an adjacency matrix which is similar to the interruption matrix discussed in the sequential interruption model in Chapter 5. However, unlike the interruptional matrix, the TE-interaction network encodes both sequential interruptions with their abundance, and also the recursive interruptions with the interruptional evolution encoded in the hierarchy of the trees. As was mentioned previously, a topological sort would represent a prediction of the TEs, if the graph had been acyclic. It should be possible to remove some edges from the graph to break the cycles. This can be done by using the *minimum feedback arc set problem*, which is classified as $\mathcal{NP}$-complete [21], but this is beyond the scope of this thesis. Nevertheless, the cycles in the graphs are also informative, as they can perhaps indicate some notion of the lifespan of TEs. Therefore, we made no attempt to remove the cycles.

The model is used on the human genome, but as a standard method, it can be easily applied to other genomes to construct the TE-interaction network by including the TEs and their interruptions in that genome. Furthermore, it can even be applied to multiple related genomes (e.g., several primate genomes, or several plant genomes) where common TEs exist in them to study the interactions between both TEs and the genomes.

Last but not least, the network could possibly serve as a visualization tool to show the interactive history between TEs. The weights on outgoing edges represent the number of times each TE got interrupted by another TE (how much older the TE is compared to another), which further implies the age of that TE — the bigger outdegree, the older age. The weights on incoming edges capture the number of times each TE interrupt another TE, which further implies the lifespan of a TE — the bigger indegree, the longer lifespan. A Python implementation of this chapter can adjust parse trees to obtain interruptional evolutionary trees, then construct the network from the adjusted trees.

In the next chapter, there will be further use of this model on simulated data. A systematic comparison between the two approaches (models from Chapter 5 and 6), together with a new technique to measure the accuracy, will be given.

# Chapter 7

# Simulation of the TE transpositions through sequence evolution

TEs have existed for billions of years, as they predate eukaryotes. They are ubiquitous in the human genome and many other genomes, and their impacts on genome size, genome structure, plasticity, and evolution are substantial. In the previous chapters, given the TE remnants in a genome, theoretical models have been created predicting the history of TE interaction which reflects the age of TEs and their activities throughout evolution. In this chapter, a simulation of TE transpositions is created to imitate the simplification of evolutionary history of the propagation of TEs. By simulating the TE activities through evolution, the remnants of TEs with their positions in a simulated genome can be generated. Such a simulation is an important tool for understanding transpositions and the evolution of genomes generally, since TEs are such an important factor in their form. Furthermore, a simulation can be used as a verification tool for TE prediction problems. Ideally, the known ages and lifespans of TEs from a simulation, and the predicted age order calculated by the models in Chapter 5 and 6 using the simulated remnants data should be identical, which serves as an *in silico* verification to these theoretical models. Using simulations of sequence evolution as a means of verification has also been used similarly in other areas, such as to compare multiple sequence alignment algorithms [135].

## 7.1 Introduction

As introduced in Chapter 2, the typical lifespan of a transposon starts from the activation of the transposon, followed by a burst of transposition activity. For both Class I and Class II elements, a transposition event generates a duplicated copy of the transposon, and inserts it into a new genomic site with a sequence that is identical to the original copy. The copies of this transposon accumulate mutations independently, and as their divergence increases, their mobile activity slows down with time. The transposon then ebbs further until it is deactivated once all of its copies become inactive. The inactive elements become relics and fade into the genomic sequence while accumulating mutations at the neutral mutation rate as surrounding DNA loci, which results in older TEs becoming more divergent than younger TEs. At the same time, the old elements (both active or inactive copies) can get interrupted by the transpositions of younger active elements [47], where a younger element replicates itself at the site of insertion within an old element.

The simulation begins with a set of TEs and their ages (when they first appear) as an input. It starts from a point in evolutionary time (e.g., 200 MYA), and simulates the mutations in a genome (using an existing tool, called `PhyloSim`) and the insertion and degrading activities of TEs. As time progresses, TEs are activated when the "current" time matches their input ages. The activated TEs start their transpositional activities while also accumulating mutations at the same time. The mutations in TEs decrease the activity levels of these TEs, until they become inactive. The simulation can imitate the activity of the entire lifespan of these transposons in a genome of a molecular sequence.

In the next sections, more details of the simulation will be introduced including the background of a similar simulation of sequence evolution, key parameters, existing substitution models, and the workflow of the simulation.

## 7.2 Background

### The human mutation rate

Human genomes differ from each other in a number of ways, such as single nucleotide mutations, insertions and deletions, repeat polymorphisms, and larger-scale rearrangements. A *mutation rate* is defined as a measure of the rate at which various types of mutations occur in the genome over time, which is characterized using a measure such as mutations per base pair per cell division, per gene per generation, or per genome per generation, etc. In most studies, mutation rates are based on single nucleotide mutations because they are comparatively easy to quantify. The mutation rate in the human genome was estimated in different units as:

- $\mu_1 = 0.17\%$ mutations per base pair per Mys in [73];
- $\mu_2 = 10^{-10}$ mutations per replication per base pair in [107];
- $\mu_3 = 2.5 \times 10^{-8}$ per site per generation [108];
- $\mu_4 = 160$ mutations per dipoid genome per generation [108].

These rates in different units are roughly consistent by converting to each other. For example,

$$\begin{aligned}
\mu_4 &= 160 \text{ mutations (per human genome) (per generation)} \\
&\approx \frac{\mu_1}{10^6 \text{ years per Mys}} \times 3.2 \times 10^9 \text{ (bp) (per human genome)} \times 30 \text{ (years) (per generation)} \\
&= \frac{0.17\% \text{ mutations (per bp) (per Mys)}}{10^6 \text{ years per Mys}} \times 3.2 \times 10^9 \text{ (bp) (per human genome)} \times 30 \text{ (years) (per generation)} \\
&= 163.2 \text{ mutations (per human genome) (per generation)} \\
\mu_3 &= 2.5 \times 10^{-8} \text{ per site per generation} \\
&= \frac{\mu_4}{6.4 \times 10^9 \text{ (per bp per diplod genome)}} \\
&= \frac{160 \text{ mutations per dipoid genome per generation}}{6.4 \times 10^9 \text{ (per bp per diplod genome)}} \\
&= 2.5 \times 10^{-8} \text{ per site per generation}
\end{aligned}$$

## Transposition rates of *L1* and *Alu* in the human genome

Because of their continued activity and accumulation in the genome over years, $L1$ and *Alu* elements have had a huge impact on the evolution of primate genomes. To assess this impact, the amplification/transposition rates of the $L1$ and *Alu* elements are considered. A *transposition rate* is defined as the frequency that retrotranspositions occur in the germline [1] over time.

The current transposition rate of $L1$ retrotransposition has been estimated as approximately 1 insertion for every 140 births in humans [38], and the current rate of *Alu* retrotransposition has been estimated as approximately 1 insertion for every 20 births in humans [31]. The rates are calculated based both on the frequency of disease-causing *de novo* insertions compared with nucleotide substitutions and on comparisons between the human and chimpanzee genomes and between multiple human genome sequences [147].

## Models of DNA substitutions

The process of a sequence of nucleotides changing into another sequence of nucleotides can be described using a DNA substitution model, which is a phenomenological description of the DNA sequence evolution as a string of four discrete states. Mutation events whose occurrence at each site of the DNA sequence can be mathematically modelled by a continuous-time Markov chain, which is defined by matrices containing the relative probabilities of changing from any nucleotide to any other nucleotide at any site over any period of

[1] In biology and genetics, the *germline* of a mature or developing individual is the line (sequence) of germ cells that have genetic material that may be passed to a child.

evolutionary time. Note that most modern simulation software allows simulating sequence regions evolving under different models/parameters.

The most common DNA substitution models includes JC69 (Jukes and Cantor, 1969) [62], K80 (Kimura, 1980) [75], F81 (Felsenstein 1981) [39], HKY85 (Hasegawa, Kishino and Yano 1985) [54], T92 (Tamura 1992) [138], TN93 (Tamura and Nei 1993) [139], GTR (generalised time-reversible) [140], etc., among which JC69 is the simplest model based on assumptions such as equal base frequencies and equal mutation rates.

### The `PhyloSim` simulation package

The `PhyloSim` [129] (License: GNU General Public License Version 3) is an object-oriented framework of Monte Carlo simulation[2] of sequence evolution written in `R`, which simulates the evolution of DNA or protein sequences by using substitution models of the type of the sequence following an input-phylogenetic tree. The framework builds on and complements the `APE` (Analyses of Phylogenetics and Evolution) package [3].

`PhyloSim` uses the *Gillespie algorithm* [46] as a unified framework for simulating the actions of many concurrent processes such as substitutions, insertions, and deletions in sequence evolution. The Gillespie algorithm is a method of stochastic simulation, which tracks the evolution of variables that can change randomly with some probabilities. It was first used to simulate reactions of chemical or biochemical systems efficiently and accurately, and is now heavily used in the area of computational systems biology.

In `PhyloSim`, the simulation of sequence evolution is guided by an input-phylogenetic tree, where the branch lengths represent evolutionary time length. Every branch (in terms of evolutionary time) of the tree is simulated in two steps iterated repeatedly: first, randomly generate the time of occurrence of the next event; second, modify the sequence object according to a randomly selected event. The event is chosen by selecting the highest event rate, which is estimated by the substitution models attached to the sites in the sequence. These steps are repeated until the available time on the current branch is exhausted. At the end, the genomic sequences of the simulated species (at the tips of the phylogenetic tree) are related to each other according to the (input) phylogenetic tree.

The key features offered by `PhyloSim` includes (adapted from [129]): the evolution of a set of discrete characters can be simulated with arbitrary states evolving by a Markov process (JC69, HKY, GTR, etc.)

[2] Monte Carlo simulation is a numerical experimentation technique to obtain the statistics of the output variables of a computational model, given the statistics of the input variables [94].

[3] The `APE` package provides functions for reading, writing, plotting, and manipulating phylogenetic trees, analyses of comparative data in a phylogenetic framework, ancestral character analyses, analyses of diversification and macroevolution, computing distances from DNA sequences, reading and writing nucleotide sequences as well as importing from BioConductor.

with an arbitrary rate matrix; the evolution can be simulated by a combination of different substitution processes with any rate matrices on the same site; popular models and patterns of among-sites rate variation can be simulated; different site- and process-specific parameters are allowed for every site, which permits for any number of partitions in the simulated data.

## 7.3 Methodology

Our simulation of TE transpositions through sequence evolution is written in the `R` language and built on top of the `PhyloSim` package. This is because `PhyloSim` provides functions that simulate random substitutions through sequence evolution, and the TE transposition simulation imitates the replication and insertion of active TEs on a dynamically changing genomic sequence through evolution created by `PhyloSim`. Therefore, the `PhyloSim` package provides the necessary generality on which to extend, rather than rewriting its functionality. Unlike `PhyloSim` that simulates sequence evolution of multiple species under the guidance of a phylogenetic tree, the TE transposition simulation currently only simulates one genomic sequence. The workflow of the TE transposition simulation starts from an evolutionary time $T$ and an original genomic sequence. As time is being consumed, mutations are introduced into the genome randomly using the functions provided by `PhyloSim` following a substitution model and the mutation rate; at the same time, TEs are being activated when the current time reaches the input ages of these TEs, and transpositions occur through evolution. This is repeated until time is exhausted. The modified genomic sequence at the end of the simulation represents the current-day genome, which encodes the history of TE activities by their positions of insertions and their interruption patterns, similar to the current-day human genome.

The simulation is subject to a number of parameters such as mutation rate, transposition rates, substitution model etc., and a set of input data, which is listed and explained in details in the next subsections, as well as a discussion of the limitations of the approach.

### Parameters, inputs, and limitations

There are some parameters in Table 7.1, which are useful for the simulation, followed by justifications of these parameters.

Though mutation rates may differ between species and even between different regions of the genome of a single species, in this chapter, the main objective is to simulate the TE transpositions instead of the evolution of the sequence itself, so an assumption that any site in the sequence has the same neutral mutation rate is made for simplicity. More dynamic rates, including for example, epigenetic effects on transposition is left as future work. Similarly, the same substitution model is applied to each site in the nucleotide sequence for

| Name | Notation | Values |
|---|---|---|
| Mutation rate | $\mu$ | 0.17% (per site) (per Mys) |
| Substitution model | $p$ | JC69 |
| Evolutionary time to simulate | $T$ | 200 MYA |
| Length of initial genomic sequence | $seq.len$ | 10,000 bp |
| Transposition rate | $Tr.rate$ | 10 site mutations per insertion |
| Threshold of percent identity to deactivate a TE | $PID$ | 90% |

**Table 7.1:** Parameters of the simulation

simplicity as well. In our simulation, the JC69 model is applied to the genomic sequence. A more extensive simulation taking into account sequence regions evolving under different models/parameters is also left as future work.

As previously mentioned in Chapter 3, no *Alu* elements with more than 10% mutations were active in the cell culture in [12]. Therefore, the threshold of percent identity to deactivate a TE in the simulation is set to be

$$PID = 90\%.$$

Recall that the transposition rate, denoted as $Tr.rate$, for the *Alu* elements has been estimated as approximately 1 insertion for every 20 births in humans. We assume that transposons have the same mutation rate as their surrounding DNA loci after they inserted into the genome. We also assume that both the transposition rates and mutation rate are constant over time. Given a neutral mutation rate of 160 mutations per diploid genome per generation in human [108], the transposition rates of *Alu* elements can then be converted and represented relative to the sequence evolution in our simulation as:

$$\begin{aligned} Tr.rate &= \frac{\text{160 mutations (per diploid genome) (per generation)}}{\text{1/20 } Alu \text{ insertion (per diploid genome) (per generation)}} \\ &= 3,200 \text{ mutations (per } Alu \text{ insertion).} \end{aligned}$$

However, to make the time taken to execute the simulation practical and the TE-interaction network easy to read, only a small set of 20 TEs (2% of the TEs in the human genome) will be simulated on a small genome of length 10,000 bp ( $3.33 \times 10^{-4}\%$ of the human genome size). Moreover, in order to generate a large number of insertions and interruptions in a practical amount of time in the simulation, the transposition rate is set to $Tr.rate = 10$ mutations (per insertion) for simplicity.

It should be noted that the simulation in this thesis only imitates the transposition of retrotransposons (transpose using copy-and-paste mechanism), not DNA transposons (transpose using cut-and-paste mechanism). This is because each insertion of retrotransposons is stable through evolutionary time, and is a “fossil” of a unique transposition event. The transposition of DNA transposons involves not only insertions, but also excisions, which is more complicated.

The simulation is based on known TEs and their properties, therefore, there are known data inputs, including consensus sequences, ages, and harmful regions of these TEs.

1. TE consensus sequences: a list of $n$ TEs ($n = 20$) whose propagation will be simulated along with their consensus sequences. The consensus is randomly generated with a length of 30 bp (10% of the length of *Alu*).

2. TE ages: the list of $n$ TEs are input together with their age of activities (ranging from 200 Mys to 30 Mys). The TEs will be activated once the current time reaches their age (when the TEs start appearing in the genome).

3. Harmful regions: the genomic positions of harmful regions in the consensus sequences of the TEs. If mutations occur within the harmful regions of a TE, the activity fraction of this TE will be decreased in the simulation. According to the harmful regions of *AluY* elements calculated in Table 3.2, the harmful regions covered 32.9% of the *AluY* consensus sequence. Therefore, in the simulation, 30% of the consensus sequences are marked as harmful regions, where the positions of the regions are randomly generated.

## Initializations

The simulation starts with some preprocessing and initialization steps. This includes certain values that are needed for a verification of the sequential interruption model and the recursive interruption model in previous chapters.

1. construct the database of TEs. Each record is one transposon, denoted by $\mathcal{TE}$, with its attributes as

   $$info(\mathcal{TE}) = (name, consensus, length, status, age, harmfulRG, SCFGterminal, t_{activate}, t_{deactivate}).$$

   The definitions of the attributes are as follows:

   *name*: the name of the transposon;

   *consensus*: the consensus sequence of the transposon from Repbase Update;

   *length*: the number of nucleotides in the consensus sequence;

*status*: the activity status of the transposon. The status is set to 0 when the simulation starts:

$$status = \begin{cases} 0 & \text{if the transposon has never been activated;} \\ 1 & \text{if the current time equals to the age of this transposon, it is activated;} \\ 2 & \text{if this TE is currently active;} \\ 3 & \text{if the TE is no longer active (but stays in the genome).} \end{cases}$$

*age*: the age of the transposon, meaning the time when it became active;

$harmfulRG$: start and end positions of the harmful regions in the consensus sequence;

$SCFGterminal$: corresponding terminal letter (a label) of this element used for the context-free grammar (explained below);

$t_{activate}$: time when the element is activated;

$t_{deactivate}$: time when the element is deactivated.

2. Initialize an interruptional matrix, $IM$, with all 0s, where rows represent interrupters, columns represent interruptees. This matrix will store the sequential interruptions that occur during the simulation, and will be used input to the sequential interruption model developed in Chapter 5 as a verification.

3. Create a randomly generated nucleotide sequence, *geno.seq*, of length *seq.len*, and attach a substitution model, $p$, to the sites in the sequence (using functions provided in `PhyloSim`).

4. Initialize a TE fragment list, $TEfragments$, where each element in the list is one TE fragment $z$ with its attributes as:

$$info(z) = (genoName, genoStart, genoEnd, genoLeft, strand, TEName, TEClass,$$
$$TEStart, TEEnd, TELeft, \mathit{segmented}, \mathit{updated}, \mathit{activeFrac}, \mathit{pid}, YearInsertion).$$

Besides the attributes defined in the TE fragment model (Chapter 4) in Definition 12, there are five additional attributes for the sake of the simulation:

- *segmented*: “yes” if the fragment is segmented by another TE; “no” otherwise.

- *updated*: “1” if the coordinate of the fragment is updated (when another fragment was inserted into the genomic sequence at a position before the current fragment, the coordinate of the fragment needs to be updated); “0” the fragment has already been updated, or there is no need to update the coordinate (if another fragment was inserted at a position after the current fragment, the coordinate of the fragment does not change).

- *activeFrac*: the active fraction of the fragment. 0% means inactive; 100% means the most active when the TE is just activated.
- *pid*: the percent identity of the fragment compared to the consensus sequence of this transposon.
- $YearInsertion$: time when the fragment was inserted into the genomic sequence.

## General steps of the simulation

Briefly, the TE transposition simulation is based upon the sequence evolution simulation, where random mutations (simulated by the substitution model) are introduced into the sequence for each time step iteratively. For every $Tr.rate$ (a transposition rate described in terms of the number of mutations) mutations, introduce a TE insertion. The inserted TE is replicated from either a newly activated TE from the TE database (when the current time matches the age of that TE) or from a randomly selected active TE copy (from its activity fraction) that already existed in the genome. Each active TE existing in the genome has an attribute called activity fraction, denoted as *activeFrac*, which is dynamically calculated by the current percent identity, the number of mutations occurred within the harmful regions, and the lifespan of that TE. The mutations and insertions are repeated until the simulation time is exhausted.

Starting from a time in evolution $t \leftarrow T$ (e.g., 200 million years ago), for every time step, do the following operations until $t$ is exhausted.

**Step 1** Introduce a mutation into the sequence subject to the substitution model $p$ attached to the sequence using functions provided in the `PhyloSim` package.

**Step 2** After accumulating $Tr.rate$ substitutions, check whether there are active TEs to transpose:

**case 1:** if there are newly activated TE ($TE.age \leq t$ and $TE.status == 0$), prepare to introduce an insertion from the TEs in the TE database by doing the following steps, then go to **Step 3**. If there are more than one TEs that are newly activated, randomly choose one from them. The newly activated TE that is chosen to be inserted is denoted by $\mathcal{TE}$.

(a) In the TE database, update the activation status of the chosen TE, $\mathcal{TE}.status = 1$ (activate the TE now) and its year of activation, $\mathcal{TE}.t_{activate} = t$;

(b) Create a copy of $\mathcal{TE}.consensus$ as *seqToInsert*;

**case 2:** if there does not exist any newly activated TE, check if there are existing active TE copies in the genomic sequence, and prepare to introduce a TE insertion from an active copy by doing the following steps, then go to **Step 3**.

For every intact TE fragment (that did not get interrupted by other TEs) $z$ in the $TEfragments$ list, do the following steps:

(a) Calculate the percent identity of the fragment, *z.pid*, by comparing the sequence of this fragment to the consensus sequence.

(b) If there are mutations that occurred within the harmful regions of the TE, calculate the number of mutations in the harmful regions, denoted as $n$. The more mutations that occurred in harmful regions, the less active this TE will be.

(c) Calculate the lifespan of the TE family ($\mathcal{TE}$) that the fragment belongs to, denoted as $span = t - \mathcal{TE}.age$. The longer this TE family has existed in the genome, the less active it tends to be.

(d) update the activity fraction $z.activeFrac$:

- If $z$ diverges to a threshold ($z.pid \leq PID$), change it to the inactive status ($z.activeFrac \leftarrow 0\%$);

- Otherwise, calculate a new activity fraction subject to the percent identity of the fragment, the number of mutations in the harmful regions, and the lifespan since the TE of this fragment has become active:

$$z.activeFrac \leftarrow z.pid \times \frac{1}{n+1} \times \frac{1}{span} \times z.activeFrac.$$

(e) select the active TE fragment with the highest activity fraction (if there are multiple fragments with the same activity fraction, randomly select one), and create a copy of its sequence as $seqToInsert$ (the new insert will have the same sequence and same activity fraction as the original fragment).

**case 3:** if there are no active TEs to transpose, skip **Step 3** and **Step 4**, and go to **Step 5** directly.

**Step 3** Insert $seqToInsert$ from the last step into the genomic sequence:

(a) Randomly generate a position of the genomic site, $pos$.

(b) Create a fragment object that inherits the attributes from the original copy of the element (either a newly activated TE or an existing active TE copy in the genomic sequence) with positions in the genome and append it to the $TEfragments$ list.

(c) Insert $seqToInsert$ at $pos$.

(d) check if the insertion interrupts other existing fragments in $TEfragments$ list (if *pos* is within any fragments). Once an interruption is detected, deactivate the interruptee ($z.segmented \leftarrow yes$, $z.activeFrac \leftarrow 0\%$), and update the interruptional matrix $IM$.

(e) Update coordinates of all fragments afterwards.

(f) Update the percent identity of each intact TE fragment (that did not get interrupted by others) in the TE fragment list. If it is less than $PID$, make this fragment inactive ($z.activeFrac \leftarrow 0\%$).

**Step 4** Check the activity fractions of all TEs in the list of $TEfragments$. If no copies of the family $\mathcal{TE}$ are still active, deactivate this family in the TE database and record the deactivation time ( $\mathcal{TE}.status \leftarrow 3$, $\mathcal{TE}.t_{deactivate} \leftarrow t$), otherwise, $\mathcal{TE}.status \leftarrow 2$.

**Step 5** Sample a *time.interval* with a fixed *mean* calculated from the mutation rate $\mu$ and the current length of the sequence $seq.len'$,

$$mean = \frac{\frac{1}{\mu}}{seq.len'}.$$

**Step 6** Decrease the current time $t$ by *time.interval*, and repeat from **Step 1** until $t$ is exhausted.

## Output

After $t$ is exhausted, the simulation is finished, and the final genomic sequence is output. In addition, the $TEfragments$ list with detailed information regarding each fragment in the genome is output. Furthermore, there are two outputs that connect to the theoretical models created in previous chapters:

- An interruptional matrix of the sequential interruptions that are detected from the simulated genomic sequence using the sequential interruption model in Chapter 5 will be output and fed to the Tabu search of the Linear Ordering Problem to predict a linear age order of these TEs. This is used to verify the accuracy of the sequential interruptional analysis work, as the correct answer is known for the simulation.

- A list of order-pruned sequences of the generated TE remnants will be output and fed into the recursive interruptions model in Chapter 6 to predict the local TE interruption trees of evolution.

- A list of TEs with their lifespans in the simulation will be output. All the interruption trees from the last step will be adjusted and merged into a TE-interaction network, and will be visualized whereby the node size of each node in the network, determined by the lifespan of that TE in the simulation.

## 7.4 Results

Although the simulation will be run 10 times and aggregate statistics will be collected, one simulation will be described in detail first as an example. This allows for a more detailed discussion.

A simulation of 20 TEs was run for $T = 200$ Mys using the parameters listed in Table 7.1. The consensus and harmful regions (cover 30% of the length of the consensus) of these TEs are randomly generated. The activation and deactivation time and the lifespan of each TE in the simulation are in Table 7.2 (from oldest to youngest). Note that the TEs are labelled intentionally to be consistent with their age for simplicity; for example, $TE1$ is the oldest, and $TE20$ is the youngest. Moreover, the column of input age order in the table is marked in red, which will be compared to the predicted age orders later.

| TE name | Input age order (oldest to youngest) | Year of activation (MYA) | Year of deactivation (MYA) | Lifespan (Mys) | Number of fragments in genome |
|---|---|---|---|---|---|
| TE1 | 1 | 199 | 138 | 61 | 80 |
| TE2 | 2 | 195 | 171 | 23 | 6 |
| TE3 | 3 | 189 | 168 | 22 | 3 |
| TE4 | 4 | 185 | 151 | 34 | 1 |
| TE5 | 5 | 180 | 112 | 68 | 77 |
| TE6 | 6 | 170 | 119 | 50 | 13 |
| TE7 | 7 | 160 | 89 | 71 | 58 |
| TE8 | 8 | 150 | 86 | 64 | 59 |
| TE9 | 9 | 140 | 69 | 71 | 19 |
| TE10 | 10 | 130 | 55 | 74 | 122 |
| TE11 | 11 | 120 | 97 | 22 | 1 |
| TE12 | 12 | 110 | 62 | 48 | 7 |
| TE13 | 13 | 100 | 63 | 37 | 9 |
| TE14 | 14 | 90 | 13 | 76 | 129 |
| TE15 | 15 | 80 | 39 | 41 | 5 |
| TE16 | 16 | 70 | 8 | 61 | 22 |
| TE17 | 17 | 60 | 0 | 60 | 60 |
| TE18 | 18 | 50 | 0 | 50 | 53 |
| TE19 | 19 | 40 | 0 | 40 | 33 |
| TE20 | 20 | 30 | 0 | 30 | 29 |

**Table 7.2:** The input age order, activation time, deactivation time and lifespans of TEs in the simulation. The column of input age order is marked in red, which will be compared to the predicted age orders later.

### Verification of the theoretical models using the simulated data

As mentioned in Section 7.3, there are three outputs from the simulation: an interruption matrix of sequential interruptions, a set of order-pruned sequences of TE remnants with $E = 5\ bp$ in transposon regions in the genomic sequence, and the lifespans of the TEs in the simulation. The output data are fed into both the sequential interruption model and recursive interruption model of the previous chapters, as illustrated in the flow chart in Figure 7.1, to verify the sequential and recursive interruption models created in Chapters 5 and 6.

First, the interruption matrix (IM) detected using the sequential interruption model is fed into the Tabu search of the LOP to predict an age order from IM; second, the order-pruned sequences are fed into the stochastic CYK to generate the most-likely parse trees of the SCFG of recursive interruption model; then the most-likely parse trees are adjusted and merged into an TE-interruption network where the node sizes of the TEs in the network are determined by the lifespans of TEs in simulation (these are not inferred); meanwhile, the adjacency matrix (AM) of the TE-interruption network is fed into the Tabu search of the LOP again to predict another age order from AM; last (not shown on the workflow), the predicted age orders from IM (the blue order) and from AM (the green order) are compared to the input age order of TEs (the red order), and the correlations of each comparison will be reported reflecting the accuracy of the predicted ages against the input known ages. Moreover, the two predicted orders will be compared to each other to report how much they agree with each other.

As discussed in Lemma 1, the position of transposons with zero interruptions does not affect the optimal solution of the LOP, hence, it is reasonable to remove these TEs from the interruption matrix so that they will not appear in the predicted order. The predicted age order from the interruption matrix is calculated and shown in Table 7.3. Note that TE4 and TE11 were not involved in any interruptions, so their relative ages were not predicted, and the predicted age order of the sequential interruption model is marked in the same colour (blue) as the predicted order of IM in Figure 7.1.

To quantify how much the predicted order is correlated with the input order, the Pearson's coefficient of correlation is calculated between the blue order (predicted age order from IM) and the red order (input TE age order) as:

$$\rho_{\text{blue vs. red}} = 0.9401445,$$

which indicates that the predicted order from IM and the input order have strong positive correlation (a correlation of 1 means that the two orders are identical). Furthermore, Figure 7.2 shows the comparison between the blue order (predicted age order from IM) and the red order (input TE age order). As previously mentioned in Chapter 5, it is reasonable that the two orders in comparison are distributed "around" the diagonal line, as TEs have overlapped lifespans. It can be seen that the predicted age order calculated by

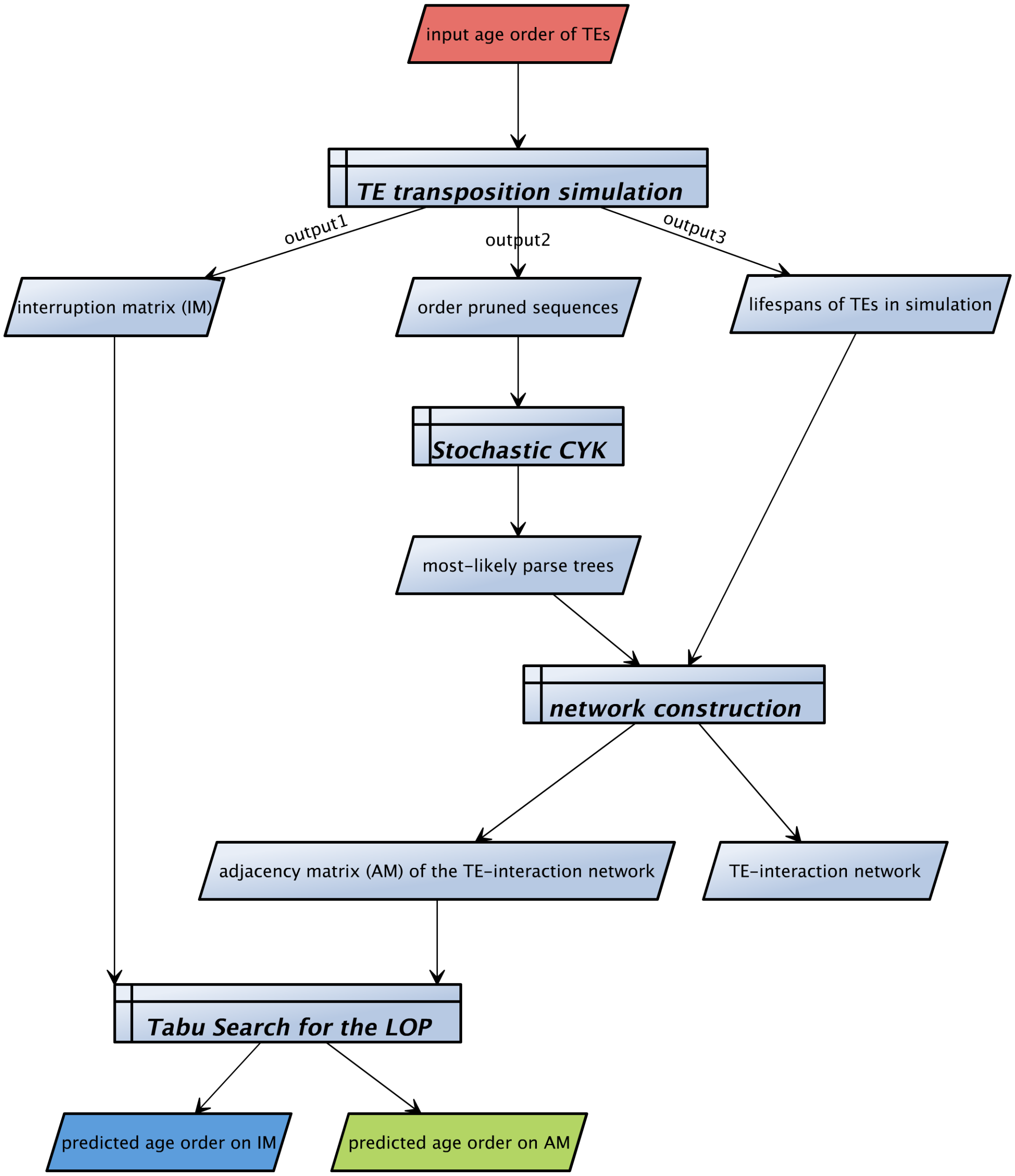


**Figure 7.1:** A workflow of simulated data verifying the theoretical models in previous chapters. The three age orders are colour coded in the chart, where the input age order in red corresponds to the red column in Table 7.2, the predicted age order on IM in blue corresponds to the blue column in Table 7.3, and the predicted age order on AM in green corresponds to the green column in Table 7.5.

| TE name | Predicted age order<br>calculated by Tabu search from IM<br>(oldest to youngest) |
|---|---|
| TE1 | 3 |
| TE2 | 5 |
| TE3 | 1 |
| TE5 | 2 |
| TE6 | 6 |
| TE7 | 7 |
| TE8 | 4 |
| TE9 | 9 |
| TE10 | 8 |
| TE12 | 12 |
| TE13 | 13 |
| TE14 | 10 |
| TE15 | 11 |
| TE16 | 14 |
| TE17 | 15 |
| TE18 | 18 |
| TE19 | 17 |
| TE20 | 16 |

**Table 7.3:** The relative age order calculated by Tabu search from the interruption matrix of the sequential interruption model. Note that TE4 and TE11 were not involved in any interruptions, so their relative ages are not predicted. The column of the predicted age order is marked in the same colour as the predicted order of IM in Figure 7.1.

Tabu search agrees well with the input age order.

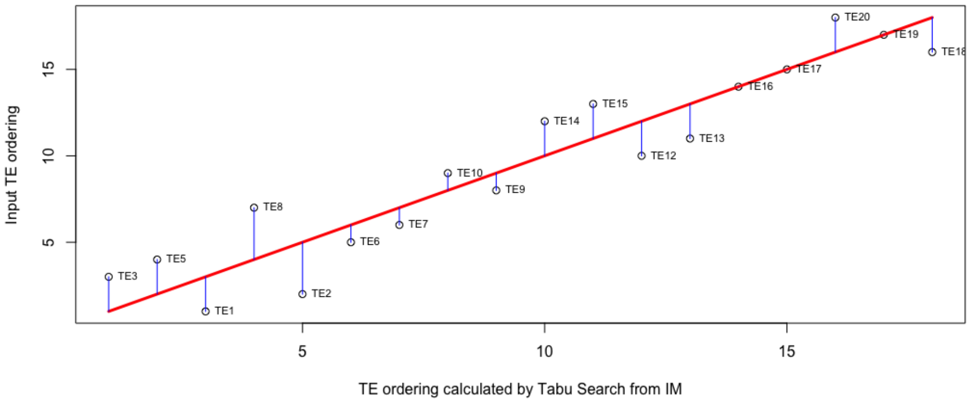


**Figure 7.2:** A comparison between the input age order and the predicted age order calculated by Tabu search from the simulated interruption matrix. The x-axis is the predicted relative age order, the y-axis is the input relative age order, and the diagonal line marked in red represents all points that agree between the predicted and actual order.

Next, the order-pruned sequences output from the simulation are fed into the recursive interruption model to calculate the most-likely parse trees. The parse trees are then adjusted to obtain interruptional evolutionary trees using the implementation of Algorithm 1, and the interruptional evolutionary trees are merged into a TE-interaction network as shown in Figure 7.3.

Note that each node in the network represents one TE, and the TE numbering matches the numbering of the variables in the grammar, for example, the node $X1$ in the network represents TE1 for simplicity. The edge weights greater than one are marked on the edges. As previously discussed in Chapter 6, the lifespans of TEs also influences the number of interruptions in which the TEs can be involved. In light of this, some improvements on visualizing the network have been made to include the information of lifespans. The size of each node in Figure 7.3 is subjected to the lifespan of the corresponding TE output from the simulation, where the size of the node is the logarithm (base 10) of the lifespan of that TE. The reason for converting the lifespan into the logarithm scale is for the clarity, as otherwise the big nodes will make the directions and weights on the edges hard to see. Furthermore, the TEs are aligned with their activation time (timeline on the left of the graph) in the simulation.

The network shows the interruptions between TEs visually. It can be seen that most of the edges point down, which suggests younger TEs interrupting older TEs in the simulation; whereas there are some edges that point up, which suggests overlapped lifespans between the TEs connected by the cycles. Some properties of the predicted interruptional evolution of these TEs are encoded in the network:

- upstream elements in the graph are predicted to be the oldest in age, e.g., $X_2$, $X_3$.

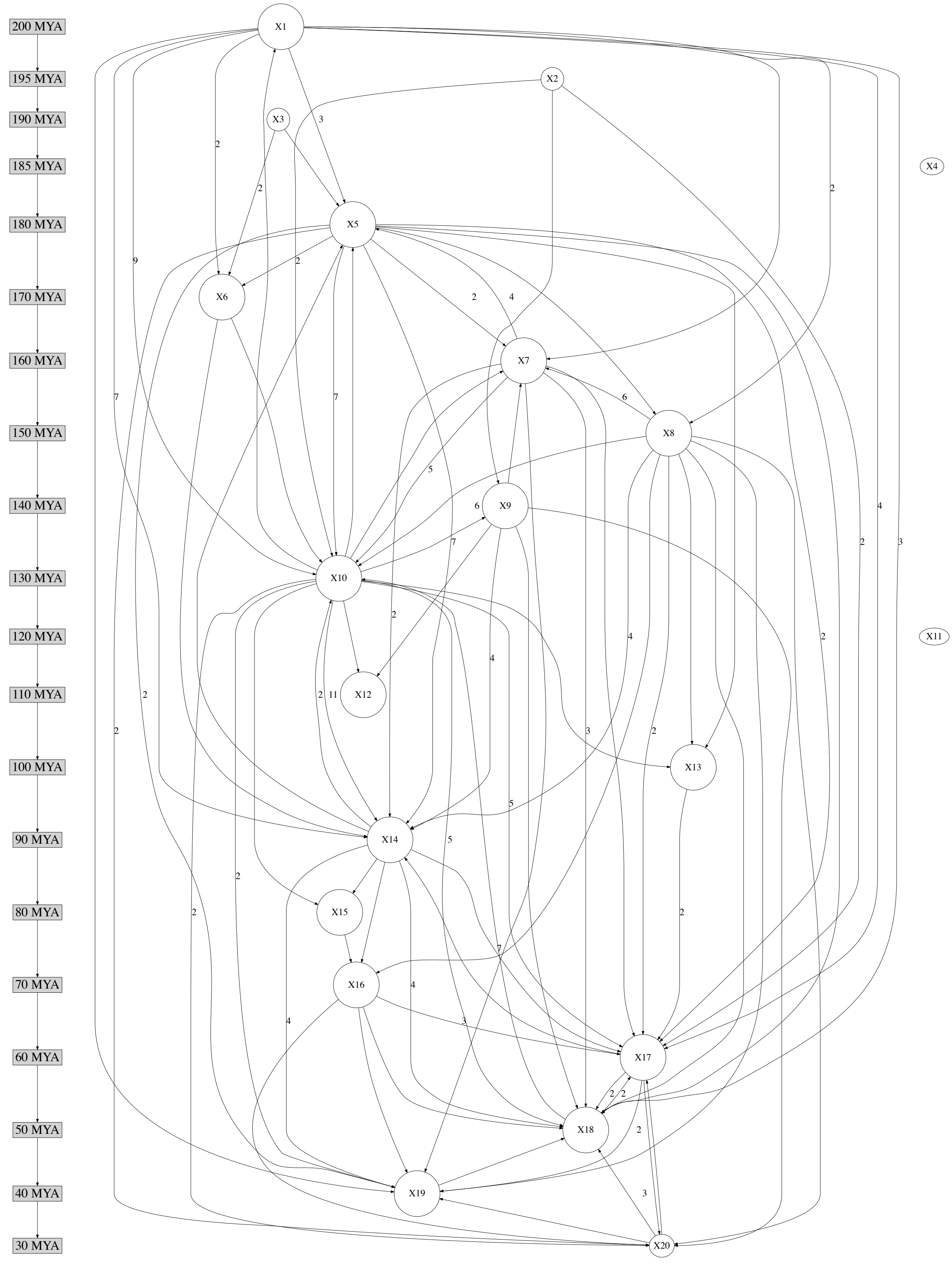


**Figure 7.3:** The TE-interaction network calculated from the recursive interruption model on the simulated data, with each TE aligned with its year of activation.

- downstream elements in the graph are predicted to be the youngest in age, e.g., $X_{12}$.

  Though we know that $X_{12}$ is not the youngest in age, it only has 7 fragments in the genome as in Table 7.2, therefore, it has less chance to be interrupted by younger TEs than the TEs with a large number of copies in the genome.

- elements in the same cycle in the graph indicate that they have overlapped lifespans:

  **cycle 1:** $X_5 \to X_{10} \to X_5$;

  **cycle 2:** $X_7 \to X_{10} \to X_9 \to X_7$;

  **cycle 3:** $X_{14} \to X_{16} \to X_{17} \to X_{14}$;

  **cycle 4:** $X_{17} \to X_{20} \to X_{18} \to X_{17}$.

  The activation and deactivation time in Table 7.2 show that the TEs in the same cycle have overlapped lifespans.

- paths in the network indicate orders of TE ages. The elements marked in "$\llbracket\ \rrbracket$" are in the same cycle in the network, which indicates that these elements have overlapped lifespans. As the input age order is encoded in the TE names, a small number in the name suggests an old age, and vice versa.

  **path 1:** $X_1 \to \llbracket X_5 \to\ X_{10} \rrbracket \to X_{12}$;

  **path 2:** $X_8 \to \llbracket\ X_7 \to X_{10} \rrbracket \to X_{14} \to X_{18}$

  **path 3:** $X_{13} \to\ \llbracket X_{17} \to X_{20} \to X_{19} \to X_{18} \rrbracket$

  **path 4:** $X_{14} \to X_{15} \to X_{16} \to X_{19}$

  The TEs in a path with inverted numbering also indicate overlapped lifespans, such as $X_8 \to X_7$ in path 2, and $X_{20} \to X_{19} \to X_{18}$ in path 3.

Table 7.4 summarizes the indegree and outdegree of each node in Figure 7.3. As discussed, the old TEs tend to have high outdegree and low indegree, while the young TEs tend to have high indegree and low outdegree.

Next, the adjacency matrix of the TE-interaction network in Figure 7.3 is fed into the Tabu search of the LOP, and a predicted age order from AM (the green order in Figure 7.1) is calculated and listed in Table 7.5, where the column of the predicted order is coloured in green to match with the data in Figure 7.1.

Similarly, the Pearson's coefficient of correlation is calculated between the green order (predicted age order

| TE name | Outdegree | Indegree |
|---|---|---|
| TE1 | 32 | 1 |
| TE2 | 4 | 0 |
| TE3 | 3 | 0 |
| TE4 | 0 | 0 |
| TE5 | 27 | 10 |
| TE6 | 2 | 6 |
| TE7 | 16 | 11 |
| TE8 | 23 | 3 |
| TE9 | 8 | 2 |
| TE10 | 32 | 32 |
| TE11 | 0 | 0 |
| TE12 | 0 | 2 |
| TE13 | 2 | 3 |
| TE14 | 20 | 37 |
| TE15 | 1 | 2 |
| TE16 | 6 | 3 |
| TE17 | 6 | 31 |
| TE18 | 3 | 25 |
| TE19 | 1 | 15 |
| TE20 | 5 | 8 |

**Table 7.4:** The indegree and outdegree of each node in the network in Figure 7.3.

| TE name | Predicted age order calculated by Tabu search from AM (oldest to youngest) |
|---|---|
| TE1 | 1 |
| TE2 | 2 |
| TE3 | 4 |
| TE5 | 7 |
| TE6 | 8 |
| TE7 | 6 |
| TE8 | 3 |
| TE9 | 5 |
| TE10 | 9 |
| TE12 | 16 |
| TE13 | 14 |
| TE14 | 10 |
| TE15 | 11 |
| TE16 | 12 |
| TE17 | 15 |
| TE18 | 18 |
| TE19 | 17 |
| TE20 | 13 |

**Table 7.5:** The relative age order calculated by Tabu search from the adjacency matrix (AM) of the TE-interaction network of the recursive interruption model. Note that TE4 and TE11 were not involved in any interruptions (they are not connected to any other nodes in Figure 7.3), so their relative ages are not predicted. The column of predicted age order is marked in the same colour as the predicted order of AM in Figure 7.1.

from AM) and the red order (input TE age order) as:

$$\rho_{\text{green vs. red}} = 0.8658411,$$

which indicates that the predicted order from AM and the input order have strong positive correlation as well, but it is less strong than the correlation between the predicted order from IM and the input order ($\rho_{\text{blu vs. red}} = 0.9401445$) in this particular simulation. Furthermore, Figure 7.4 shows the comparison between the green order (predicted age order from AM) and the red order (input TE age order).

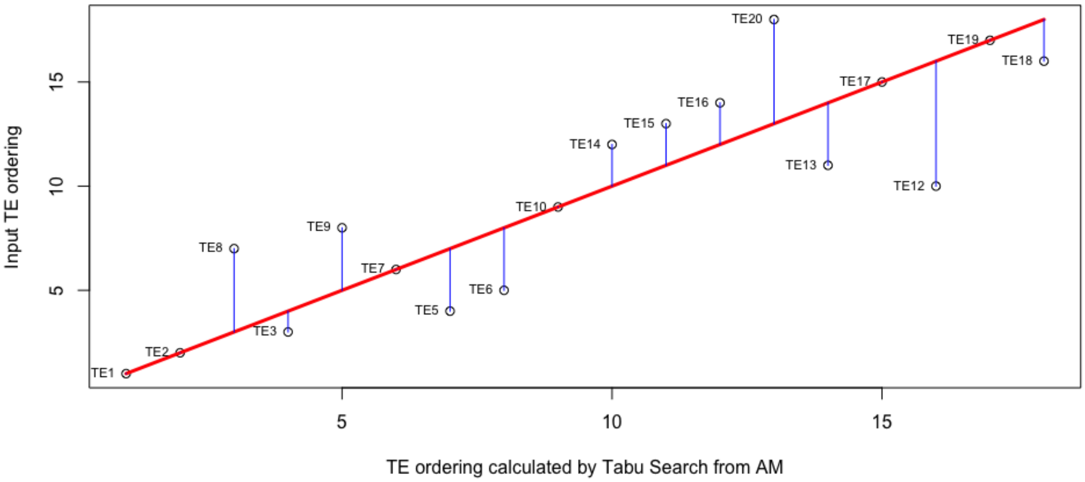


**Figure 7.4:** A comparison between the input age order and the predicted age order calculated by Tabu search from adjacency matrix. The x-axis is the predicted relative age, the y-axis is the input relative age, and the diagonal line marked in red represents all points that agree between the predicted and actual order.

Lastly, the two predicted orders are compared to each other, and the Pearson's coefficient of correlation between the two orders is calculated as

$$\rho_{\text{green vs. blue}} = 0.8968008,$$

which indicates that the two predicted orders are also positively correlated with each other.

To further show that the two theoretical models created in this thesis achieves good results in predicting the relative ages of TEs in general, the simulation was run for 10 times with the same parameters as in Table 7.1 for 20 TEs for $T = 200$ Mys using the same data input (TE consensus, age order, and harmful regions). The same workflow of Figure 7.1 is applied to all the simulated data to compare the predicted orders from both the sequential interruption model and the recursive interruption model to the input order. The correlations are listed in Table 7.6.

The correlations in the table suggest that the recursive interruption model seems to perform better in gen-

|  | $\rho_{\text{blue vs. red}}$ | $\rho_{\text{green vs. red}}$ |
|---|---|---|
|  | (ordering from IM vs. input order) | (ordering from AM vs. input order) |
| simulation1 | 0.936 | 0.962 |
| simulation2 | 0.889 | 0.898 |
| simulation3 | 0.686 | 0.950 |
| simulation4 | 0.611 | 0.991 |
| simulation5 | 0.641 | 0.989 |
| simulation6 | 0.734 | 0.833 |
| simulation7 | 0.719 | 0.956 |
| simulation8 | 0.787 | 0.929 |
| simulation9 | 0.692 | 0.950 |
| simulation10 | 0.681 | 0.934 |
| average | 0.738 | 0.939 |

**Table 7.6:** Correlations calculated from ten simulations.

eral, because it achieves a better correlation on average, in contrast to the one example above, where the sequential model performed better. Indeed, on average, the Pearson's coefficient of correlation is 0.939 for the recursive interruption model, versus the considerably lower 0.738 for the sequential interruption model. Further simulations across longer timespans, more TEs, larger sequence, and varied parameters is left as future work.

## 7.5 Conclusion and discussion

In this chapter, a simulation is developed in the `R` language, which simulates the evolutionary process of how TEs propagate through time, built on top of the existing `PhyloSim` package that simulates sequence evolution. It is based on several assumptions, such as that the mutation rate is constant both through all genomic sites including inserted TEs, as well as through evolution. It also assumes that the transposition rate is constant for all the TEs in the simulation. The simulation is general for all TE families or all genomes.

After a simulation, the TE remnants in the simulated genome are used to verify the theoretical models in previous chapters. First, the predicted relative age calculated by Tabu search based on the sequential interruption model shows a strong positive correlation with the input age order of TEs, which supports the accuracy of the sequential interruption model in predicting TE ages. Then a TE-interaction network is created using the recursive interruption model which shows the interactions between TEs in evolution graphically. The directions of edges generally pointing downward in the graph agrees with the known TE activity in the simulation. The adjacency matrix of the TE-interaction network is then used by Tabu search to predict another relative age order of the TEs, which also shows a strong positive correlation with the input order. Both the direction of edges and the agreement between predicted order and input order support the accuracy of the result of the recursive interruption model. Moreover, the two predicted orders from the sequential interruption model and the recursive interruption model are positively correlated with each other as well, meaning that both methods aiming to find the same solution. The average correlation from the two models suggests that the recursive interruption model, with an average Pearson's coefficient of correlation of 0.939, achieves a better result than the sequential interruption model, with an average Pearson's coefficient of correlation of 0.738, in predicting a linear relative age order of TEs. Moreover, the recursive interruption model is more useful than the sequential interruption model in terms of visualizing the TE interactions in the entire genome as a whole, which further can show the lifespans of the TEs, although more work on visualization is left as future work.

# Chapter 8

# Conclusion and future work

Although traditionally viewed as "junk DNA", many discoveries have shown that transposable elements, especially retrotransposons (SINEs and LINEs), have played a fundamental role in primate evolution, including the evolution of our own genome. Furthermore, TEs not only contributed to the formation of novel genes and gene transcription networks, but also play a role in human disease such as cancers and Alzheimer's. They remain active in the human central nervous system throughout life and act as a driver of human intellect [86]. However, comparatively little work or analysis is currently being undertaken on TEs. Therefore, it is important to understand how the activity of TEs changes throughout their lives, how they shape the genome, and how the positions and patterns of existing fossil elements can infer aspects of genome evolution.

In this chapter, an overall conclusion will tie each of the previous chapters together with respect to the research goals proposed in Section 1.2, and then some future directions will be proposed.

## 8.1 Conclusion

First, recall the four major research goals proposed in Chapter 1:

**Goal 1** Create a model that describes TEs and remnants of TEs formally.

**Goal 2** Understand the factors that affect the activities of active TEs, and understand how activity is affected.

**Goal 3** Predict the age, lifespan, and activity of TEs in the human genome from the remnants of these elements in the genome.

**Goal 4** Understand the dynamics of TEs transpositions through evolution.

The research in this thesis has provided information on not only current active/existing TEs, but also the evolution of TEs over time inferred by their interaction with each other, potentially contributing to both our understanding of human health and the evolution of species. The major work of this thesis can be divided into three topics: the prediction of harmful mutation regions within active TEs, the prediction of evolutionary relationships between different TEs, and a simulation of propagation of TEs throughout evolution.

The first topic, the prediction of harmful mutations regions in Chapter 3, consists of the two predicting methods, the correlation method and the group comparison method, verifications and applications. By using the statistical methods, a negative correlation between the positions of mutations within a TE and the change of activity of that TE has been revealed, which accomplished **Goal 2** proposed in the objectives of the thesis. Understanding how activity is influenced by the sequence of TEs can be useful for understanding susceptibility to different diseases caused by their activity. In addition, the results of the predicted regions are useful information for the last topic of simulating the TE activity throughout evolution.

In the second topic, three theoretical models have been created in order to predict the evolutionary relationships (the age, lifespan, and interactions) between different TEs from the remnants of these elements in the genome from Chapter 4 to Chapter 6.

- The first model, the TE fragment model in Chapter 4, set up a foundation (high-level abstraction) that consists of an initial definition of TEs, TE fragments, and interruptions between TEs, etc., which serves as a baseline in describing the other models in the thesis. Such a model, if used, helps in using unambiguous terminology consistently. This has accomplished the **Goal 1** proposed in the objectives of the thesis that describes TEs and their remnants formally.

- The second model, the sequential interruption model in Chapter 5, is formalized from the problem in [47] using the TE fragment model, and is connected and reduced to the well-studied Linear Ordering Problem. It can be solved by the existing meta-heuristic method of Tabu search for LOP incredibly efficiently while achieving a better result than the method in [47]. This is used to predict the order of all 1,080 TEs as opposed to the 405 TEs that were ordered in [47].

- The third model, the recursive interruption model in Chapters 6, is built upon the TE fragment model and is complementary to the sequential interruption model. By capturing the interruptions and nested interruptions of TEs existing in the genome using the recursive interruption stochastic context-free grammar, the parse trees of the grammar can illustrate the relationships of TE interruptions using the levels of the trees. The stochastic CYK algorithm is implemented and determines the most-likely parse trees, and is applied to the entire human genome. Then by merging the small interruptional evolutionary trees into a TE-interaction network, TE evolution and interactions can be visualized as a whole.

All the models under this topic have accomplished **Goal 3** proposed in the objectives of the thesis that predicts the age and lifespan of TEs in a single genome. Both of the latter two models do the entire prediction from only a single genomic sequence. This is somewhat amazing what can be determined from a single sequence rather than requiring multiple genomes.

The third topic is the simulation of the TE activity and propagation throughout evolution in Chapter 7. A simulation has been created to imitate the activation, propagation, interaction, and deactivation of a number of TEs in a simulated genome throughout evolution, using the data of harmful regions calculated from the first topic. After the simulation, the genome represents a "current-day" genome containing the remnants of TEs, similar to the current-day human genome. By generating an interruption matrix and order-pruned sequences from the simulated genome, a linear age order of TEs is calculated from the sequential interruption model, and a TE-interaction network is constructed from the recursive interruption model. These are compared to the input age order and the activity of TEs in the simulation. The results of predicted age order by the models on the simulated data agree very well with the input data (with an average Pearson's coefficient of correlation of 0.939 for the recursive interruption model, and 0.738 for the sequential interruption model), which serves as a verification to the theoretical models in the second topic. The simulation accomplishes **Goal 4** proposed in the objectives of the thesis.

In conclusion, all four objectives proposed at the beginning of the thesis have been accomplished, in predicting the harmful mutations regions within a TE, and both a linear age order and an interaction network of TE evolution. This helps improve the understanding of transposable elements generally and helps to understand the evolution of genomes, for which TEs are a major influence.

## 8.2 Future work

The major contributions of this thesis are the theoretical models that predict the relative age order of TEs in the human genome, the computational methods for predicting harmful mutation regions in active TEs, as well as the simulation of the evolutionary process of TE propagation. Beyond this thesis, there are some avenues for further work on various aspects of transposable elements.

The computational models were developed and applied only on the human genome and human TEs in the thesis. The models can be further applied to other related genomes to analyze the evolution of TEs both within and across those genomes, which will help in understanding the roles that TEs play in phylogeny. Also, many organisms, such as many plant species, have very dynamic genomes with many active transposable elements. Methods developed here could end up playing an important role in improving our understanding of these species, not only from the perspective of evolution, but also present-day and future possibilities for them as well, such as susceptibility to disease, or speciation.

Second, regarding active TEs in the human genome, it is important to understand factors that affect TE activity. Computational methods have been developed that predict harmful mutation regions affecting the activity of TEs. However, limited by the current data, the roles of factors other than mutations, such as chromatin structure, influencing the activity of TEs were unable to be assessed. By collaborating with experts on biology of TEs, a thorough analysis of TE activities can be conducted computationally, aiming for detection of more factors affecting TE activities and their relationship with human disease.

In the two theoretical models in predicting TE insertion evolution, two different interruption detection techniques were used. An interruption is detected using the positional information (positions in both genomic sequence and TE consensus sequence) of the TE fragments by several conditions in the sequential interruption model. In the recursive interruption model, interruptions are detected using the stochastic context-free grammar by parsing the order-pruned sequences, which only encode the positional information in the genomic sequence. Hence, the technique of the sequential interruption model is superior in one respect to that of the recursive interruption model. In contrast, the recursive interruption model is superior to the sequential interruption model in detecting nested interruptions which cannot be represented by sequential interruptions. One future direction is to combine the two techniques. For example, one could first preprocess the order-pruned sequences to incorporate conditions of sequential interruptions, then detect interruptions using the SCFG that is useful in discovering the nested interruptions.

As a visualization tool, more methods can be investigated for drawing the TE-interaction network in laying out the nodes in a clear way to show the relationships between interactions and the predicted ages. One attempt was made in Chapter 7 to visualize the nodes by different sizes subject to the known lifespans in simulation. Other useful attributes can be studied to adjust the node sizes in a TE-interaction network from actual TE data to infer the lifespans of TEs.

There are many possibilities for future work on TE simulation. First, additional biological verifications can be made to validate the predicted age orders and lifespans of human TEs calculated from the models of the thesis. The efficiency of the simulation can be optimized, so that it can be used to simulate a larger genome (e.g., comparable to the size of the human genome) and TE activities that are closer to a real situation. Additional parameters can be systematically studied. Also, starting with properties of existing real genomes, it is of interest to study the types of parameters and likelihood of obtaining those properties via simulation. This is especially interesting given the variance of TE activity in different organisms. It is also of interest to understand why certain taxa have such high activity, whereas others have low activity, and to understand why this might be the case.

# References

[1] Pankaj Agarwal et al. The Repeat Pattern Toolkit (RPT): analyzing the structure and evolution of the *C. elegans* genome. In *International Conference on Intelligent Systems for Molecular Biology; ISMB.*, volume 2, pages 1–9, 1993.

[2] Bronwen L Aken, Sarah Ayling, Daniel Barrell, Laura Clarke, Valery Curwen, Susan Fairley, Julio Fernandez Banet, Konstantinos Billis, Carlos García Girón, Thibaut Hourlier, et al. The ensembl gene annotation system. *Database: the Journal of Biological Databases and Curation*, 2016.

[3] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[4] Kateřina Ambrožová, Terezie Mandáková, Petr Bureš, Pavel Neumann, Ilia J Leitch, Andrea Koblížková, Jiří Macas, and Martin A Lysak. Diverse retrotransposon families and an AT-rich satellite DNA revealed in giant genomes of Fritillaria lilies. *Annals of Botany*, 107(2):255–268, 2010.

[5] P.A. Apoil, E. Kuhlein, A. Robert, H. Rubie, and A. Blancher. HIGM syndrome caused by insertion of an AluYb8 element in exon 1 of the CD40LG gene. *Immunogenetics*, 59(1):17–23, 2007.

[6] Z. Bao and S.R. Eddy. Automated *de novo* identification of repeat sequence families in sequenced genomes. *Genome Research*, 12(8):1269–1276, 2002.

[7] M.A. Batzer, P.L. Deininger, U. Hellmann-Blumberg, Jerzy Jurka, D. Labuda, C.M. Rubin, C.W. Schmid, E. Zietkiewicz, and E. Zuckerkandl. Standardized Nomenclature for Alu Repeats. *Journal of Molecular Evolution*, 42(1):3–6, 1996.

[8] Mark A. Batzer and Prescott L. Deininger. Alu repeats and human genomic diversity. *Nature Reviews Genetics*, 3(5):370–379, 2002.

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

[10] V.P. Belancio, A.M. Roy-Engel, and P.L. Deininger. All y'all need to know 'bout retroelements in cancer. In *Seminars in Cancer Biology*, volume 20, pages 200–210. Elsevier, 2010.

[11] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[12] E. Andrew Bennett, Heiko Keller, Ryan E. Mills, Steffen Schmidt, John V. Moran, Oliver Weichenrieder, and Scott E. Devine. Active Alu retrotransposons in the human genome. *Genome Research*, 18(12):1875–1883, 2008.

[13] C.M. Bergman and H. Quesneville. Discovering and detecting transposable elements in genome sequences. *Briefings in Bioinformatics*, 8(6):382–392, 2007.

[14] Christian Biémont and Cristina Vieira. Genetics: junk DNA as an evolutionary force. *Nature*, 443(7111):521–524, 2006.

[15] C. Bouchet, S. Vuillaumier-Barrot, M. Gonzales, S. Boukari, C. Le Bizec, C. Fallet, A.L. Delezoide, H. Moirot, A. Laquerriere, F. Encha-Razavi, et al. Detection of an Alu insertion in the POMT1 gene from three French Walker Warburg syndrome families. *Molecular Genetics and Metabolism*, 90(1):93–96, 2007.

[16] Lindell Bromham and David Penny. The modern molecular clock. *Nature Reviews Genetics*, 4(3):216–224, 2003.

[17] Brook Brouha, Joshua Schustak, Richard M. Badge, Sheila Lutz-Prigge, Alexander H. Farley, John V. Moran, and Haig H. Kazazian. Hot L1s account for the bulk of retrotransposition in the human population. *Proceedings of the National Academy of Sciences*, 100(9):5280–5285, 2003.

[18] D. Campagna, C. Romualdi, N. Vitulo, M. Del Favero, M. Lexa, N. Cannata, and G. Valle. Rap: a new computer program for *de novo* identification of repeated sequences in whole genomes. *Bioinformatics*, 21(5):582, 2005.

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

[20] Guadalupe Castilla Castilla Valdez and Shulamith Samantha Bastiani Medina. Iterated local search for the linear ordering problem. *International Journal of Combinatorial Optimization Problems and Informatics*, 3(1):12, 2012.

[21] Pierre Charbit, Stéphan Thomassé, and Anders Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability and Computing*, 16(01):1–4, 2007.

[22] Brian Charlesworth and Deborah Charlesworth. The population dynamics of transposable elements. *Genetical Research*, 42(01):1–27, 1983.

[23] Irène Charon and Olivier Hudry. A branch-and-bound algorithm to solve the linear ordering problem for weighted tournaments. *Discrete Applied Mathematics*, 154(15):2097–2116, 2006.

[24] Jian-Min Chen, Emmanuelle Masson, Milan Macek, Odile Raguénès, Tereza Piskackova, Brigitte Fercot, Libor Fila, David N Cooper, Marie-Pierre Audrézet, and Claude Férec. Detection of two Alu insertions in the CFTR gene. *Journal of Cystic Fibrosis*, 7(1):37–43, 2008.

[25] Jichao Chen, Amir Rattner, and Jeremy Nathans. Effects of L1 retrotransposon insertion on transcript processing, localization and accumulation: lessons from the retinal degeneration 7 mouse and implications for the genomic ecology of L1 elements. *Human Molecular Genetics*, 15(13):2146–2156, 2006.

[26] Wei Chen, Benjamin J Swanson, and Wendy L Frankel. Molecular genetics of microsatellite-unstable colorectal cancer for pathologists. *Diagnostic Pathology*, 12(1):24, 2017.

[27] Benoît Chénais, Aurore Caruso, Sophie Hiard, and Nathalie Casse. The impact of transposable elements on eukaryotic genomes: from genome size increase to genetic adaptation to stressful environments. *Gene*, 509(1):7–15, 2012.

[28] John Douglas Cleary and Laura PW Ranum. Repeat associated non-ATG (RAN) translation: new starts in microsatellite expansion disorders. *Current Opinion in Genetics & Development*, 26:6–15, 2014.

[29] Richard Cordaux. The human genome in the LINE of fire. *Proceedings of the National Academy of Sciences*, 105(49):19033–19034, 2008.

[30] Richard Cordaux and Mark A. Batzer. The impact of retrotransposons on human genome evolution. *Nature Reviews Genetics*, 10(10):691–703, 2009.

[31] Richard Cordaux, Dale J. Hedges, Scott W. Herke, and Mark A. Batzer. Estimating the retrotransposition rate of human Alu elements. *Gene*, 373:134–137, 2006.

[32] A.P. Jason de Koning, Wanjun Gu, Todd A. Castoe, Mark A. Batzer, and David D. Pollock. Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genetics*, 7(12):e1002384, 2011.

[33] John S. Decani. A branch and bound algorithm for maximum likelihood paired comparison ranking. *Biometrika*, 59(1):131–135, 1972.

[34] Prescott L. Deininger and Mark A. Batzer. Alu repeats and human disease. *Molecular Genetics and Metabolism*, 67(3):183–193, 1999.

[35] Marie Dewannieux, Cécile Esnault, and Thierry Heidmann. LINE-mediated retrotransposition of marked Alu sequences. *Nature Genetics*, 35(1):41–48, 2003.

[36] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1998.

[37] R.C. Edgar and E.W. Myers. Piler: identification and classification of genomic repeats. *Bioinformatics*, 21(suppl 1):i152–i158, 2005.

[38] Adam D. Ewing and Haig H. Kazazian. High-throughput sequencing reveals extensive variation in human-specific L1 content in individual human genomes. *Genome Research*, 20(9):1262–1270, 2010.

[39] Joseph Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

[40] Thomas A. Feo and Mauricio G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[41] C. Feschotte, N. Jiang, and S.R. Wessler. Plant transposable elements: where genetics meets genomics. *Nature Reviews Genetics*, 3(5):329–341, 2002.

[42] C. Feschotte and E.J. Pritham. DNA transposons and the evolution of eukaryotic genomes. *Annual Review of Genetics*, 41:331–368, 2007.

[43] David J. Finnegan. Eukaryotic transposable elements and genome evolution. *Trends in Genetics*, 5:103–107, 1989.

[44] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software Practice and Experience*, 30(11):1203–1233, 2000.

[45] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP Completeness*, volume 174. Freeman San Francisco, CA, San Francisco, 1979.

[46] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[47] J. Giordano, Y. Ge, Y. Gelfand, G. Abrusán, G. Benson, and P.E. Warburton. Evolutionary history of mammalian transposons determined by genome-wide defragmentation. *PLoS Computational Biology*, 3(7):e137, 2007.

[48] Fred Glover and Manuel Laguna. *Tabu Search**. Springer, New York, 2013.

[49] John L. Goodier and Haig H. Kazazian. Retrotransposons revisited: the restraint and rehabilitation of parasites. *Cell*, 135(1):23–35, 2008.

[50] Dan Graur. *Molecular and Genome Evolution*. Sinauer Associates, Incorporated, Massachusetts, USA, 2016.

[51] J. Greilhuber, T. Borsch, K. Müller, A. Worberg, S. Porembski, and W. Barthlott. Smallest angiosperm genomes found in lentibulariaceae, with chromosomes of bacterial size. *Plant Biology*, 8(6):770–777, 2006.

[52] YanHong Gu, Hiroko Kodama, Shigero Watanabe, Nobuyuki Kikuchi, Ineo Ishitsuka, Hiroshi Ozawa, Chie Fujisawa, and Katsuaki Shiga. The first reported case of Menkes disease caused by an Alu insertion mutation. *Brain and Development*, 29(2):105–108, 2007.

[53] Pierre Hansen and Nenad Mladenović. *Variable Neighborhood Search*. Springer, New York, 2003.

[54] Masami Hasegawa, Hirohisa Kishino, and Taka-aki Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174, 1985.

[55] Julien Häsler and Katharina Strub. Alu elements as regulators of gene expression. *Nucleic Acids Research*, 34(19):5491–5497, 2006.

[56] John E. Hopcroft. *Introduction to Automata Theory, Languages, and Computation, 3/E*. Pearson Education India, 2008.

[57] Lingling Jin. Multiple sequence alignment augmented by expert user constraints. Master's thesis, University of Saskatchewan, 2010.

[58] Lingling Jin and Ian McQuillan. Computational modelling of the interruptional activities between transposable elements. In *Lecture Notes in Computer Science*, volume 8273, pages 108–120. Springer, 2013.

[59] Lingling Jin and Ian McQuillan. Computational modelling of interruptional activities between transposable elements using grammars and the linear ordering problem. *Soft Computing*, 20(1):19–35, 2016.

[60] Lingling Jin, Ian McQuillan, and Longhai Li. Computationally identify harmful regions that influence transpositions of active transposable elements in the human genome. In *IEEE International Conference on Bioinformatics & Biomedicine, 2016. Proceedings.* IEEE, 2016.

[61] Lingling Jin, Ian McQuillan, and Longhai Li. Computational identification of harmful mutation regions to the activity of transposable elements, 2017. Submitted to BMC Genomics.

[62] Thomas H Jukes, Charles R Cantor, et al. Evolution of protein molecules. *Mammalian Protein Metabolism*, 3(21):132, 1969.

[63] Jerzy Jurka, V.V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. Repbase update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research*, 110(1-4):462–467, 2005.

[64] Jerzy Jurka, P. Klonowski, V. Dagman, and P. Pelton. CENSOR–a program for identification and elimination of repetitive elements from DNA sequences. *Computers & Chemistry*, 20(1):119–121, 1996.

[65] Cristina M. Justice, Zhining Den, Son V. Nguyen, Mark Stoneking, Prescott L. Deininger, Mark A. Batzer, and Bronya J.B. Keats. Phylogenetic analysis of the Friedreich ataxia GAA trinucleotide repeat. *Journal of Molecular Evolution*, 52(3):232–238, 2001.

[66] A. Kalyanaraman and S. Aluru. Efficient algorithms and software for detection of full-length LTR retrotransposons. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. IEEE*, pages 56–64. IEEE, 2005.

[67] V. Kapitonov and J. Jurkal. The Age of Alu Subfamilies. *Journal of Molecular Evolution*, 42(1):59–65, 1996.

[68] Vladimir V. Kapitonov and Jerzy Jurka. A universal classification of eukaryotic transposable elements implemented in repbase. *Nature Reviews Genetics*, 9(5):411–412, 2008.

[69] Haig H. Kazazian. An estimated frequency of endogenous insertional mutations in humans. *Nature Genetics*, 22(2):130–130, 1999.

[70] Haig H. Kazazian. Mobile elements: Drivers of genome evolution. *Science*, 303(5664):1626–1632, 2004.

[71] Haig H. Kazazian. *Mobile DNA: Finding Treasure in Junk.* FT Press, United States, 2011.

[72] Yogeshwar D. Kelkar, Svitlana Tyekucheva, Francesca Chiaromonte, and Kateryna D. Makova. The genome-wide determinants of human and chimpanzee microsatellite evolution. *Genome Research*, 18(1):30–38, 2008.

[73] Hameed Khan, Arian Smit, and Stéphane Boissinot. Molecular evolution and tempo of amplification of human LINE-1 retrotransposons since the origin of primates. *Genome Research*, 16(1):78–87, 2006.

[74] M. Kimura. Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences*, 78(1):454, 1981.

[75] Motoo Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120, 1980.

[76] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[77] Miriam K. Konkel, Jerilyn A. Walker, Ashley B. Hotard, Megan C. Ranck, Catherine C. Fontenot, Jessica Storer, Chip Stewart, Gabor T. Marth, Mark A. Batzer, 1000 Genomes Consortium, et al. Sequence analysis and characterization of active human Alu Subfamilies Based on the 1000 Genomes Pilot Project. *Genome Biology and Evolution*, 7(9):2608–2622, 2015.

[78] I. Korf, M. Yandell, and J. Bedell. *BLAST.* O'Reilly Media, Inc., Sebastopol, CA, USA, 2003.

[79] B. Korte and Walter Oberhofer. Zwei algorithmen zur lösung eines komplexen reihenfolgeproblems. *Unternehmensforschung Operations Research-Recherche Opérationnelle*, 12:217–231, 1968.

[80] Bernhard Korte and Walter Oberhofer. Triangularizing input-output matrices and the structure of production. *European Economic Review*, 1(4):482–511, 1970.

[81] Brent A. Kronmiller and Roger P. Wise. Tenest: automated chronological annotation and visualization of nested plant transposable elements. *Plant Physiology*, 146(1):45–59, 2008.

[82] S. Kurtz and C. Schleiermacher. Reputer: fast computation of maximal repeats in complete genomes. *Bioinformatics*, 15(5):426, 1999.

[83] Manuel Laguna, Rafael Marti, and Rafael Cunquero Martí. *Scatter Search: Methodology and Implementations in C*, volume 24. Springer, New York, 2003.

[84] E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[85] Karim Lari and Steve J Young. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech & Language*, 5(3):237–257, 1991.

[86] Peter A Larsen, Michael W Lutz, Kelsie E Hunnicutt, Mirta Mihovilovic, Ann M Saunders, Anne D Yoder, and Allen D Roses. The Alu neurodegeneration hypothesis: A primate-specific mechanism for neuronal transcription noise, mitochondrial dysfunction, and manifestation of neurodegenerative disease. *Alzheimer's & Dementia*, 2017.

[87] Arnaud Le Rouzic and Pierre Capy. Population genetics models of competition between transposable element subfamilies. *Genetics*, 174(2):785–793, 2006.

[88] E. Lerat. Identifying repeats and transposable elements in sequenced genomes: how to find your way through the dense forest of programs. *Heredity*, 104(6):520–533, 2009.

[89] Xuemin Li, William A. Scaringe, Kathleen A. Hill, Stacy Roberts, April Mengos, Diane Careri, Miguel Tezanos Pinto, Carol K. Kasper, and Steve S. Sommer. Frequency of recent retrotransposition events in the human factor ix gene. *Human Mutation*, 17(6):511–519, 2001.

[90] Zhanjiang John Liu. *Bioinformatics in Aquaculture: Principles and Methods.* John Wiley & Sons, 2017.

[91] Tiago Loureiro, Rui Camacho, Jorge Vieira, and Nuno A. Fonseca. Improving the performance of transposable elements detection tools. *Journal of Integrative Bioinformatics*, 10(3):231, 2013.

[92] J.F. Lucier, J. Perreault, J.F. Noël, G. Boire, and J.P. Perreault. RTAnalyzer: a web application for finding new retrotransposons and detecting L1 retrotransposition signatures. *Nucleic Acids Research*, 35(suppl 2):W269–W274, 2007.

[93] B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440, 2002.

[94] Sankaran Mahadevan. Monte carlo simulation. *Mechanical Engineering-New York and Basel-Marcel Dekker-*, pages 123–146, 1997.

[95] Licínio Manco, Luís Relvas, C. Silva Pinto, Janet Pereira, A. Bessa Almeida, and M. Letícia Ribeiro. Molecular characterization of five portuguese patients with pyrimidine 5'-nucleotidase deficient hemolytic anemia showing three new p5'ni mutations. *Haematologica*, 91(2):266–267, 2006.

[96] Rafael Martí and Gerhard Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, volume 175. Springer, New York, 2011.

[97] Rafael Martí, Gerhard Reinelt, and Abraham Duarte. A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, 51(3):1297–1317, 2012.

[98] E.M. McCarthy and J.F. McDonald. LTR_STRUC: a novel search and identification program for LTR retrotransposons. *Bioinformatics*, 19(3):362–367, 2003.

[99] James T McClave and Terry Sincich. *Statistics.* Prentice Hall, One Lake Street, Upper Saddle River, NJ, 07458, United States, 2005.

[100] John Francis McDonald. *Transposable elements and evolution*, volume 1. Springer Science & Business Media, New York, 2012.

[101] P. Medstrand, L.N. Van De Lagemaat, and D.L. Mager. Retroelement distributions in the human genome: variations associated with age and proximity to genes. *Genome Research*, 12(10):1483–1495, 2002.

[102] Blake C. Meyers, Scott V. Tingey, and Michele Morgante. Abundance, distribution, and transcriptional activity of repetitive elements in the maize genome. *Genome Research*, 11(10):1660–1676, 2001.

[103] Yoshio Miki, Isamu Nishisho, Akira Horii, Yasuo Miyoshi, Joji Utsunomiya, Kenneth W. Kinzler, Bert Vogelstein, and Yusuke Nakamura. Disruption of the APC gene by a retrotransposal insertion of L1 sequence in a colon cancer. *Cancer Research*, 52(3):643–645, 1992.

[104] Ryan E. Mills, E. Andrew Bennett, Rebecca C. Iskow, and Scott E. Devine. Which transposable elements are active in the human genome? *Trends in Genetics*, 23(4):183–191, 2007.

[105] Ryan E. Mills, E. Andrew Bennett, Rebecca C. Iskow, Christopher T. Luttig, Circe Tsui, W. Stephen Pittard, and Scott E. Devine. Recently mobilized transposons in the human and chimpanzee genomes. *The American Journal of Human Genetics*, 78(4):671–679, 2006.

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

[107] LA Moran. Estimating human mutation rate: Biochemical method. *Sandwalk: Strolling with a sceptical biochemist*, 2013.

[108] Michael W Nachman and Susan L Crowell. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 156(1):297–304, 2000.

[109] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[110] Minoru Okubo, Asako Horinishi, Mieko Saito, Tetsu Ebara, Yoriko Endo, Kohei Kaku, Toshio Murase, and Masaaki Eto. A novel complex deletion–insertion mutation mediated by Alu repetitive elements leads to lipoprotein lipase deficiency. *Molecular Genetics and Metabolism*, 92(3):229–233, 2007.

[111] Eric M. Ostertag and Haig H. Kazazian Jr. Biology of mammalian L1 retrotransposons. *Annual Review of Genetics*, 35(1):501–538, 2001.

[112] John K. Pace and Cédric Feschotte. The Evolutionary History of Human DNA Transposons: Evidence for Intense Activity in the Primate Lineage. *Genome Research*, 17(4):422–432, 2007.

[113] Jonathan Pevsner. *Bioinformatics and functional genomics*. John Wiley & Sons, New York City, United States, 2005.

[114] Jonathan Pevsner. *Bioinformatics and functional genomics*. John Wiley & Sons, New York City, United States, 2015.

[115] P.A. Pevzner, H. Tang, and G. Tesler. De novo repeat classification and fragment assembly. *Genome Research*, 14(9):1786–1796, 2004.

[116] Benoit Piegu, Romain Guyot, Nathalie Picault, Anne Roulin, Abhijit Saniyal, Hyeran Kim, Kristi Collura, Darshan S Brar, Scott Jackson, Rod A Wing, et al. Doubling genome size without polyploidization: dynamics of retrotransposition-driven genomic expansions in Oryza australiensis, a wild relative of rice. *Genome Research*, 16(10):1262–1269, 2006.

[117] A.L. Price, N.C. Jones, and P.A. Pevzner. De novo identification of repeat families in large genomes. *Bioinformatics*, 21(suppl 1):i351–i358, 2005.

[118] A. Rambaut and L. Bromham. Estimating divergence dates from molecular sequences. *Molecular Biology and Evolution*, 15(4):442–448, 1998.

[119] Guy-Franck Richard, Alix Kerrest, and Bernard Dujon. Comparative genomics and molecular dynamics of DNA repeats in eukaryotes. *Microbiology and Molecular Biology Reviews*, 72(4):686–727, 2008.

[120] Kate R Rosenbloom, Joel Armstrong, Galt P Barber, Jonathan Casper, Hiram Clawson, Mark Diekhans, Timothy R Dreszer, Pauline A Fujita, Luvina Guruvadoo, Maximilian Haeussler, et al. The ucsc genome browser database: 2015 update. *Nucleic Acids Research*, 43(D1):D670–D681, 2015.

[121] S. Saha, S. Bridges, Z.V. Magbanua, and D.G. Peterson. Computational approaches and tools used in identification of dispersed repetitive DNA sequences. *Tropical Plant Biology*, 1(1):85–96, 2008.

[122] Phillip SanMiguel, Brandon S Gaut, Alexander Tikhonov, Yuko Nakajima, and Jeffrey L Bennetzen. The paleontology of intergene retrotransposons of maize. *Nature Genetics*, 20(1):43–45, 1998.

[123] Anoop Sarkar. An implementation of the CYK algorithm. `https://github.com/anoopsarkar`.

[124] T. Schiavinotto and T. Stützle. The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 3(4):367–402, 2004.

[125] Patrick S. Schnable, Doreen Ware, Robert S. Fulton, Joshua C. Stein, Fusheng Wei, Shiran Pasternak, Chengzhi Liang, Jianwei Zhang, Lucinda Fulton, Tina A. Graves, et al. The B73 maize genome: complexity, diversity, and dynamics. *Science*, 326(5956):1112–1115, 2009.

[126] Emma C. Scott, Eugene J. Gardner, Ashiq Masood, Nelson T. Chuang, Paula M. Vertino, and Scott E. Devine. A hot L1 retrotransposon evades somatic repression and initiates human colorectal cancer. *Genome Research*, 2016.

[127] The Chimpanzee Sequencing, Analysis Consortium, et al. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature*, 437(7055):69–87, 2005.

[128] Daniel Sinnett, Chantal Richer, Jean-Marc Deragon, and D. Labuda. Alu RNA secondary structure consists of two independent 7 SL RNA-like folding units. *Journal of Biological Chemistry*, 266(14):8675–8678, 1991.

[129] Botond Sipos, Tim Massingham, Gregory E Jordan, and Nick Goldman. PhyloSim-Monte Carlo simulation of sequence evolution in the R statistical computing environment. *BMC Bioinformatics*, 12(1):104, 2011.

[130] A.F.A. Smit. *Structure and Evolution of Mammalian Interspersed Repeats*. PhD thesis, University of Southern California, 1996.

[131] A.F.A. Smit, Hubley R., and Green P. RepeatMasker Open-3.0, 1996-2010. http://www.repeatmasker.org.

[132] A.F.A. Smit, G. Toth, A.D. Riggs, and Jerzy Jurka. Ancestral, mammalian-wide subfamilies of LINE-1 repetitive sequences. *Journal of Molecular Biology*, 246(3):401–417, 1995.

[133] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[134] John D. Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

[135] Amarendran R Subramanian, Jan Weyer-Menkhoff, Michael Kaufmann, and Burkhard Morgenstern. Dialign-t: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics*, 6(1):66, 2005.

[136] Cheng Sun, Donald B. Shepard, Rebecca A. Chong, Jose Lopez Arriaza, Kathryn Hall, Todd A. Castoe, Cedric Feschotte, David D. Pollock, and Rachel Lockridge Mueller. LTR retrotransposons contribute to genomic gigantism in plethodontid salamanders. *Genome Biology and Evolution*, 4(2):168–183, 2012.

[137] S.T. Szak, O.K. Pickeral, W. Makalowski, M.S. Boguski, D. Landsman, J.D. Boeke, et al. Molecular archeology of l1 insertions in the human genome. *Genome Biol*, 3(10), 2002.

[138] Koichiro Tamura. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+ C-content biases. *Molecular Biology and Evolution*, 9(4):678–687, 1992.

[139] Koichiro Tamura and Masatoshi Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular Biology and Evolution*, 10(3):512–526, 1993.

[140] Simon Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[141] Z. Tu, S. Li, and C. Mao. The changing tails of a novel short interspersed element in aedes aegypti. *Genetics*, 168(4):2037, 2004.

[142] Toru Udaka, Nobuhiko Okamoto, Michihiko Aramaki, Chiharu Torii, Rika Kosaki, Noboru Hosokai, Toshiyuki Hayakawa, Naoyuki Takahata, Takao Takahashi, and Kenjiro Kosaki. An Alu retrotransposition-mediated deletion of CHD7 in a patient with CHARGE syndrome. *American Journal of Medical Genetics Part A*, 143(7):721–726, 2007.

[143] Graham S Warren et al. *Plant biotechnology: comprehensive biotechnology. Second supplement/volume editor, Michael W. Fowler & Graham S. Warren; editor-in-chief, Murray Moo-Young.* Pergamon Press, Oxford, New York, Seoul, Tokyo, 1992.

[144] Ronald L Wasserstein and Nicole A Lazar. The ASA's statement on p-values: context, process, and purpose. *Am Stat*, 70(2):129–133, 2016.

[145] Thomas Wicker, François Sabot, Aurélie Hua-Van, Jeffrey L Bennetzen, Pierre Capy, Boulos Chalhoub, Andrew Flavell, Philippe Leroy, Michele Morgante, Olivier Panaud, et al. A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics*, 8(12):973–982, 2007.

[146] John D. Storey with contributions from Andrew J. Bass, Alan Dabney, and David Robinson. *qvalue: Q-value estimation for false discovery rate control*, 2015. R package version 2.2.2.

[147] Jinchuan Xing, Yuhua Zhang, Kyudong Han, Abdel Halim Salem, Shurjo K. Sen, Chad D. Huff, Qiong Zhou, Ewen F. Kirkness, Samuel Levy, Mark A Batzer, et al. Mobile elements create structural variation: Analysis of a complete human genome. *Genome Research*, 19(9):1516–1526, 2009.

[148] Andrea Zuccolo, Aswathy Sebastian, Jayson Talag, Yeisoo Yu, HyeRan Kim, Kristi Collura, Dave Kudrna, and Rod A. Wing. Transposable element distribution, abundance and role in genome size variation in the genus *Oryza*. *BMC Evolutionary Biology*, 7(1):152, 2007.

[149] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.

# Appendix A

## The TE ordering in the human genome predicted from the sequential interruption model by applying Tabu search on interruption matrix

| Age order<br>(youngest to oldest) | TE name | TE family | TE type |
|---|---|---|---|
| 1 | AluYa5 | Alu | SINE |
| 2 | AluYk3 | Alu | SINE |
| 3 | AluYc3 | Alu | SINE |
| 4 | HERV1_LTRb | ERV1 | LTR |
| 5 | AluYe5 | Alu | SINE |
| 6 | LTR6B | ERV1 | LTR |
| 7 | AluYk2 | Alu | SINE |
| 8 | AluYi6 | Alu | SINE |
| 9 | AluYd8 | Alu | SINE |
| 10 | LTR7A | ERV1 | LTR |
| 11 | hAT-16_Crp | hAT-Charlie | DNA |
| 12 | AluYc | Alu | SINE |
| 13 | AluYk4 | Alu | SINE |
| 14 | LTR5A | ERVK | LTR |
| 15 | AluYb8 | Alu | SINE |
| 16 | MER11B | ERVK | LTR |
| 17 | AluYb9 | Alu | SINE |
| 18 | LTR12C | ERV1 | LTR |
| 19 | Alu | Alu | SINE |
| 20 | LTR12E | ERV1 | LTR |
| 21 | MER9a2 | ERVK | LTR |
| 22 | AluY | Alu | SINE |
| 23 | AluYa8 | Alu | SINE |
| 24 | LTR14 | ERVK | LTR |
| 25 | LTR13 | ERVK | LTR |
| 26 | AluYm1 | Alu | SINE |
| 27 | MER11D | ERVK | LTR |
| 28 | LTR13_ | ERVK | LTR |
| 29 | LTR3B | ERVK | LTR |
| 30 | LTR22C2 | ERVK | LTR |
| 31 | LTR10G | ERV1 | LTR |
| 32 | AluYi6_4d | Alu | SINE |
| 33 | LTR6A | ERV1 | LTR |
| 34 | AluYe6 | Alu | SINE |
| 35 | AluYf1 | Alu | SINE |
| 36 | LTR7Y | ERV1 | LTR |
| 37 | LTR5_Hs | ERVK | LTR |
| 38 | AluYj4 | Alu | SINE |

| | | | |
|---|---|---|---|
| 39 | MER9a1 | ERVK | LTR |
| 40 | LTR17 | ERV1 | LTR |
| 41 | LTR7 | ERV1 | LTR |
| 42 | MER11C | ERVK | LTR |
| 43 | LTR21A | ERV1 | LTR |
| 44 | AluYg6 | Alu | SINE |
| 45 | LTR76 | ERV1 | LTR |
| 46 | LTR22E | ERVK | LTR |
| 47 | LTR1F1 | ERV1 | LTR |
| 48 | LTR12D | ERV1 | LTR |
| 49 | LTR22B1 | ERVK | LTR |
| 50 | LTR22 | ERVK | LTR |
| 51 | LTR18A | ERVL | LTR |
| 52 | LTR14C | ERVK | LTR |
| 53 | LTR12F | ERV1 | LTR |
| 54 | AluSc8 | Alu | SINE |
| 55 | HERV1_LTRa | ERV1 | LTR |
| 56 | LTR12 | ERV1 | LTR |
| 57 | LTR5B | ERVK | LTR |
| 58 | L1PA2 | L1 | LINE |
| 59 | LTR2 | ERV1 | LTR |
| 60 | AluYk11 | Alu | SINE |
| 61 | LTR18C | ERVL | LTR |
| 62 | LTR12B | ERV1 | LTR |
| 63 | AmnSINE2 | tRNA-Deu | SINE |
| 64 | AluYh3 | Alu | SINE |
| 65 | LTR7B | ERV1 | LTR |
| 66 | MER9a3 | ERVK | LTR |
| 67 | AluSp | Alu | SINE |
| 68 | AluSg4 | Alu | SINE |
| 69 | LTR12_ | ERV1 | LTR |
| 70 | AluSg | Alu | SINE |
| 71 | AluYk12 | Alu | SINE |
| 72 | MER11A | ERVK | LTR |
| 73 | LTR14B | ERVK | LTR |
| 74 | AluYh9 | Alu | SINE |
| 75 | LTR3A | ERVK | LTR |
| 76 | LTR10F | ERV1 | LTR |
| 77 | AluSq4 | Alu | SINE |
| 78 | AluSq10 | Alu | SINE |
| 79 | AluSc5 | Alu | SINE |
| 80 | LTR2B | ERV1 | LTR |
| 81 | AluSg7 | Alu | SINE |
| 82 | LTR2C | ERV1 | LTR |
| 83 | AluSc | Alu | SINE |
| 84 | LTR10B1 | ERV1 | LTR |
| 85 | AluSq | Alu | SINE |
| 86 | LTR10B2 | ERV1 | LTR |
| 87 | LTR13A | ERVK | LTR |
| 88 | LTR10D | ERV1 | LTR |
| 89 | AluSx4 | Alu | SINE |
| 90 | LTR3 | ERVK | LTR |
| 91 | AluSx3 | Alu | SINE |
| 92 | AluYh3a3 | Alu | SINE |

| | | | |
|---|---|---|---|
| 93 | LTR22A | ERVK | LTR |
| 94 | L1HS | L1 | LINE |
| 95 | AluSq2 | Alu | SINE |
| 96 | LTR3B_ | ERVK | LTR |
| 97 | LTR27 | ERV1 | LTR |
| 98 | LTR15 | ERV1 | LTR |
| 99 | LTR22B | ERVK | LTR |
| 100 | AluSx1 | Alu | SINE |
| 101 | LTR4 | ERV1 | LTR |
| 102 | L1PA3 | L1 | LINE |
| 103 | LTR1B1 | ERV1 | LTR |
| 104 | LTR10A | ERV1 | LTR |
| 105 | LTR1A2 | ERV1 | LTR |
| 106 | L1PA4 | L1 | LINE |
| 107 | LTR1C3 | ERV1 | LTR |
| 108 | MER85 | PiggyBac | DNA |
| 109 | LTR19B | ERV1 | LTR |
| 110 | LTR46 | ERV1 | LTR |
| 111 | LTR30 | ERV1 | LTR |
| 112 | L1P1 | L1 | LINE |
| 113 | LTR22B2 | ERVK | LTR |
| 114 | L1PA5 | L1 | LINE |
| 115 | LTR7C | ERV1 | LTR |
| 116 | LTR10E | ERV1 | LTR |
| 117 | LTR1B0 | ERV1 | LTR |
| 118 | LTR25 | ERV1 | LTR |
| 119 | MER52D | ERV1 | LTR |
| 120 | LTR10C | ERV1 | LTR |
| 121 | MER48 | ERV1 | LTR |
| 122 | AluSx | Alu | SINE |
| 123 | LTR71A | ERV1 | LTR |
| 124 | THE1-int | ERVL-MaLR | LTR |
| 125 | LTR28B | ERV1 | LTR |
| 126 | LTR19A | ERV1 | LTR |
| 127 | AluSz | Alu | SINE |
| 128 | L1P2 | L1 | LINE |
| 129 | LTR66 | ERVL | LTR |
| 130 | LTR9C | ERV1 | LTR |
| 131 | L1PA6 | L1 | LINE |
| 132 | MLT2A1 | ERVL | LTR |
| 133 | LTR9A1 | ERV1 | LTR |
| 134 | MER61F | ERV1 | LTR |
| 135 | MER52C | ERV1 | LTR |
| 136 | MER51A | ERV1 | LTR |
| 137 | LTR22C0 | ERVK | LTR |
| 138 | HERV1_LTRd | ERV1 | LTR |
| 139 | L1PA7 | L1 | LINE |
| 140 | LTR61 | ERV1 | LTR |
| 141 | MER51B | ERV1 | LTR |
| 142 | LTR71B | ERV1 | LTR |
| 143 | LTR1D | ERV1 | LTR |
| 144 | LTR18B | ERVL | LTR |
| 145 | MER9B | ERVK | LTR |
| 146 | AluSz6 | Alu | SINE |

|  |  |  |  |
|---|---|---|---|
| 147 | LTR1A1 | ERV1 | LTR |
| 148 | LTR9B | ERV1 | LTR |
| 149 | MER61A | ERV1 | LTR |
| 150 | LTR77 | ERV1 | LTR |
| 151 | L1PA8 | L1 | LINE |
| 152 | LTR10B | ERV1 | LTR |
| 153 | LTR5 | ERVK | LTR |
| 154 | MER57A1 | ERV1 | LTR |
| 155 | LTR9 | ERV1 | LTR |
| 156 | MER75A | PiggyBac | DNA |
| 157 | THE1A | ERVL-MaLR | LTR |
| 158 | HERV1_I-int | ERV1 | LTR |
| 159 | MER61E | ERV1 | LTR |
| 160 | LTR1F2 | ERV1 | LTR |
| 161 | THE1A-int | ERVL-MaLR | LTR |
| 162 | LTR2752 | ERV1 | LTR |
| 163 | PABL_A | ERV1 | LTR |
| 164 | LTR1E | ERV1 | LTR |
| 165 | PABL_B | ERV1 | LTR |
| 166 | MER61C | ERV1 | LTR |
| 167 | MADE1 | TcMar-Mariner | DNA |
| 168 | MER47C | TcMar-Tigger | DNA |
| 169 | THE1B | ERVL-MaLR | LTR |
| 170 | LTR14A | ERVK | LTR |
| 171 | MER30B | hAT-Charlie | DNA |
| 172 | LTR1 | ERV1 | LTR |
| 173 | LTR19C | ERV1 | LTR |
| 174 | MER61D | ERV1 | LTR |
| 175 | MER52A | ERV1 | LTR |
| 176 | MER51C | ERV1 | LTR |
| 177 | LTR21C | ERVL | LTR |
| 178 | MER84 | ERV1 | LTR |
| 179 | LTR28C | ERV1 | LTR |
| 180 | MER83C | ERV1 | LTR |
| 181 | LTR21B | ERV1 | LTR |
| 182 | LTR28 | ERV1 | LTR |
| 183 | LTR1D1 | ERV1 | LTR |
| 184 | LTR22C | ERVK | LTR |
| 185 | MLT2A2 | ERVL | LTR |
| 186 | MER75B | PiggyBac | DNA |
| 187 | MER51E | ERV1 | LTR |
| 188 | LTR27C | ERV1 | LTR |
| 189 | LTR9D | ERV1 | LTR |
| 190 | MER61B | ERV1 | LTR |
| 191 | LTR1B | ERV1 | LTR |
| 192 | MER4E | ERV1 | LTR |
| 193 | MER41A | ERV1 | LTR |
| 194 | LTR1F | ERV1 | LTR |
| 195 | MER4A1 | ERV1 | LTR |
| 196 | HERV1_LTRc | ERV1 | LTR |
| 197 | THE1C | ERVL-MaLR | LTR |
| 198 | HERV-Fc1_LTR3 | ERV1 | LTR |
| 199 | LTR1C1 | ERV1 | LTR |
| 200 | MER50B | ERV1 | LTR |

| | | | |
|---|---|---|---|
| 201 | LTR27E | ERV1 | LTR |
| 202 | LTR1C | ERV1 | LTR |
| 203 | LTR27D | ERV1 | LTR |
| 204 | THE1D | ERVL-MaLR | LTR |
| 205 | MER30 | hAT-Charlie | DNA |
| 206 | MER1A | hAT-Charlie | DNA |
| 207 | LTR43 | ERV1 | LTR |
| 208 | AluJb | Alu | SINE |
| 209 | MER1B | hAT-Charlie | DNA |
| 210 | MER83 | ERV1 | LTR |
| 211 | MER4A1_ | ERV1 | LTR |
| 212 | L1P3 | L1 | LINE |
| 213 | LTR8 | ERV1 | LTR |
| 214 | AluYh7 | Alu | SINE |
| 215 | L1PA8A | L1 | LINE |
| 216 | L1P3b | L1 | LINE |
| 217 | L1PA10 | L1 | LINE |
| 218 | LTR27B | ERV1 | LTR |
| 219 | THE1D-int | ERVL-MaLR | LTR |
| 220 | MER66B | ERV1 | LTR |
| 221 | LTR75_1 | ERV1 | LTR |
| 222 | MER75 | PiggyBac | DNA |
| 223 | MSTA | ERVL-MaLR | LTR |
| 224 | AluJo | Alu | SINE |
| 225 | LTR38 | ERV1 | LTR |
| 226 | FLAM_C | Alu | SINE |
| 227 | L1P4d | L1 | LINE |
| 228 | LTR47B4 | ERVL | LTR |
| 229 | L1P | L1 | LINE |
| 230 | FRAM | Alu | SINE |
| 231 | LTR35A | ERV1 | LTR |
| 232 | MER41B | ERV1 | LTR |
| 233 | MER4E1 | ERV1 | LTR |
| 234 | MER57B1 | ERV1 | LTR |
| 235 | L1PB1 | L1 | LINE |
| 236 | MST-int | ERVL-MaLR | LTR |
| 237 | MER41E | ERV1 | LTR |
| 238 | MER107 | hAT-Charlie | DNA |
| 239 | MER66A | ERV1 | LTR |
| 240 | PRIMA4_LTR | ERV1 | LTR |
| 241 | HSMAR1 | TcMar-Mariner | DNA |
| 242 | AluJr | Alu | SINE |
| 243 | L1P4e | L1 | LINE |
| 244 | THE1B-int | ERVL-MaLR | LTR |
| 245 | LTR43B | ERV1 | LTR |
| 246 | THE1C-int | ERVL-MaLR | LTR |
| 247 | MER41C | ERV1 | LTR |
| 248 | LTR32 | ERVL | LTR |
| 249 | LTR45C | ERV1 | LTR |
| 250 | MLT2B3 | ERVL | LTR |
| 251 | MER50 | ERV1 | LTR |
| 252 | L1PA14 | L1 | LINE |
| 253 | MER4B | ERV1 | LTR |
| 254 | MER66D | ERV1 | LTR |

| | | | |
|---|---|---|---|
| 255 | LTR62 | ERVL | LTR |
| 256 | L1PA11 | L1 | LINE |
| 257 | LTR8B | ERV1 | LTR |
| 258 | L1PA13 | L1 | LINE |
| 259 | MER87B | ERV1 | LTR |
| 260 | MER4A | ERV1 | LTR |
| 261 | Charlie3 | hAT-Charlie | DNA |
| 262 | LTR26E | ERV1 | LTR |
| 263 | AluJr4 | Alu | SINE |
| 264 | MER66C | ERV1 | LTR |
| 265 | L1PREC2 | L1 | LINE |
| 266 | MER83B | ERV1 | LTR |
| 267 | LTR60B | ERV1 | LTR |
| 268 | MLT2B5 | ERVL | LTR |
| 269 | LTR35B | ERV1 | LTR |
| 270 | MER87 | ERV1 | LTR |
| 271 | LOR1b | ERV1 | LTR |
| 272 | MER50-int | ERV1 | LTR |
| 273 | LTR8A | ERV1 | LTR |
| 274 | L1PA12 | L1 | LINE |
| 275 | MER8 | TcMar-Tigger | DNA |
| 276 | L1PB2 | L1 | LINE |
| 277 | MER21A | ERVL | LTR |
| 278 | Tigger3d | TcMar-Tigger | DNA |
| 279 | L1PB | L1 | LINE |
| 280 | L1MA1 | L1 | LINE |
| 281 | LTR26 | ERV1 | LTR |
| 282 | MSTB | ERVL-MaLR | LTR |
| 283 | MER41D | ERV1 | LTR |
| 284 | L1M | L1 | LINE |
| 285 | MSTA-int | ERVL-MaLR | LTR |
| 286 | LTR45 | ERV1 | LTR |
| 287 | LTR34 | ERV1 | LTR |
| 288 | MER72B | ERV1 | LTR |
| 289 | LTR36 | ERV1 | LTR |
| 290 | LOR1a | ERV1 | LTR |
| 291 | L1MA2 | L1 | LINE |
| 292 | LTR51 | ERV1 | LTR |
| 293 | L1PB3 | L1 | LINE |
| 294 | LTR24 | ERV1 | LTR |
| 295 | L1P4 | L1 | LINE |
| 296 | L1P4a | L1 | LINE |
| 297 | L1PA15 | L1 | LINE |
| 298 | LTR26D | ERV1 | LTR |
| 299 | LTR26B | ERV1 | LTR |
| 300 | MER72 | ERV1 | LTR |
| 301 | LTR59 | ERV1 | LTR |
| 302 | LTR38C | ERV1 | LTR |
| 303 | LTR73 | ERV1 | LTR |
| 304 | MER4C | ERV1 | LTR |
| 305 | LTR23 | ERV1 | LTR |
| 306 | LTR26C | ERV1 | LTR |
| 307 | LTR45B | ERV1 | LTR |
| 308 | Tigger1a_Art | TcMar-Tigger | DNA |

| | | | |
|---|---|---|---|
| 309 | MER4CL34 | ERV1 | LTR |
| 310 | L1MA3 | L1 | LINE |
| 311 | MER4D1 | ERV1 | LTR |
| 312 | LTR38A1 | ERV1 | LTR |
| 313 | FAM | Alu | SINE |
| 314 | LTR47A | ERVL | LTR |
| 315 | FLAM_A | Alu | SINE |
| 316 | Ricksha_0 | MULE-MuDR | DNA |
| 317 | L1PA16 | L1 | LINE |
| 318 | MER4D | ERV1 | LTR |
| 319 | LTR72 | ERV1 | LTR |
| 320 | LTR35 | ERV1 | LTR |
| 321 | LTR47B3 | ERVL | LTR |
| 322 | MER6C | TcMar-Tigger | DNA |
| 323 | LTR48 | ERV1 | LTR |
| 324 | LTR42 | ERVL | LTR |
| 325 | MER41G | ERV1 | LTR |
| 326 | LTR72B | ERV1 | LTR |
| 327 | MER34D | ERV1 | LTR |
| 328 | L1PA17 | L1 | LINE |
| 329 | LTR24C | ERV1 | LTR |
| 330 | L1PB4 | L1 | LINE |
| 331 | MSTB1 | ERVL-MaLR | LTR |
| 332 | MER6A | TcMar-Tigger | DNA |
| 333 | MER4D0 | ERV1 | LTR |
| 334 | LTR56 | ERV1 | LTR |
| 335 | HSMAR2 | TcMar-Mariner | DNA |
| 336 | LTR24B | ERV1 | LTR |
| 337 | LTR64 | ERV1 | LTR |
| 338 | L1P5 | L1 | LINE |
| 339 | Tigger3a | TcMar-Tigger | DNA |
| 340 | LTR38B | ERV1 | LTR |
| 341 | PrimLTR79 | ERV1 | LTR |
| 342 | LTR29 | ERV1 | LTR |
| 343 | MER49 | ERV1 | LTR |
| 344 | MER101 | ERV1 | LTR |
| 345 | Tigger2b_Pri | TcMar-Tigger | DNA |
| 346 | LTR06 | ERV1 | LTR |
| 347 | LTR48B | ERV1 | LTR |
| 348 | MER57B2 | ERV1 | LTR |
| 349 | LTR47A2 | ERVL | LTR |
| 350 | MER65A | ERV1 | LTR |
| 351 | Tigger2a | TcMar-Tigger | DNA |
| 352 | MER65D | ERV1 | LTR |
| 353 | MSTB2 | ERVL-MaLR | LTR |
| 354 | MER39B | ERV1 | LTR |
| 355 | MER51D | ERV1 | LTR |
| 356 | LTR57 | ERVL | LTR |
| 357 | LTR47B | ERVL | LTR |
| 358 | MSTA1 | ERVL-MaLR | LTR |
| 359 | MSTC | ERVL-MaLR | LTR |
| 360 | MER135 | DNA | DNA |
| 361 | MER6B | TcMar-Tigger | DNA |
| 362 | MER65C | ERV1 | LTR |

| | | | |
|---|---|---|---|
| 363 | L1MA5A | L1 | LINE |
| 364 | MER2B | TcMar-Tigger | DNA |
| 365 | L1MA4A | L1 | LINE |
| 366 | MER101B | ERV1 | LTR |
| 367 | MER47A | TcMar-Tigger | DNA |
| 368 | MER65B | ERV1 | LTR |
| 369 | Tigger3c | TcMar-Tigger | DNA |
| 370 | MER47B | TcMar-Tigger | DNA |
| 371 | LTR39 | ERV1 | LTR |
| 372 | LTR69 | ERVL | LTR |
| 373 | LTR49 | ERV1 | LTR |
| 374 | Merlin1_HS | Merlin | DNA |
| 375 | LTR60 | ERV1 | LTR |
| 376 | LTR47B2 | ERVL | LTR |
| 377 | LTR54 | ERV1 | LTR |
| 378 | Tigger4b | TcMar-Tigger | DNA |
| 379 | Charlie12 | hAT-Charlie | DNA |
| 380 | Tigger3 | TcMar-Tigger | DNA |
| 381 | L1MA4 | L1 | LINE |
| 382 | MER76-int | ERVL | LTR |
| 383 | HERVK11D-int | ERVK | LTR |
| 384 | Tigger2 | TcMar-Tigger | DNA |
| 385 | HERV-Fc1_LTR1 | ERV1 | LTR |
| 386 | MER54A | ERVL | LTR |
| 387 | Tigger5 | TcMar-Tigger | DNA |
| 388 | MER6 | TcMar-Tigger | DNA |
| 389 | HERVL32-int | ERVL | LTR |
| 390 | Tigger5b | TcMar-Tigger | DNA |
| 391 | Tigger4a | TcMar-Tigger | DNA |
| 392 | MER57C2 | ERV1 | LTR |
| 393 | L1M2a1 | L1 | LINE |
| 394 | Tigger1 | TcMar-Tigger | DNA |
| 395 | L1MA5 | L1 | LINE |
| 396 | OldhAT1 | hAT-Ac | DNA |
| 397 | Tigger3b | TcMar-Tigger | DNA |
| 398 | MER2 | TcMar-Tigger | DNA |
| 399 | MER57C1 | ERV1 | LTR |
| 400 | MER44A | TcMar-Tigger | DNA |
| 401 | MSTD | ERVL-MaLR | LTR |
| 402 | MER39 | ERV1 | LTR |
| 403 | MER44B | TcMar-Tigger | DNA |
| 404 | Ricksha_c | MULE-MuDR | DNA |
| 405 | MER44C | TcMar-Tigger | DNA |
| 406 | MER96 | hAT-Tip100 | DNA |
| 407 | MER73 | ERVL | LTR |
| 408 | Tigger7 | TcMar-Tigger | DNA |
| 409 | MER57D | ERV1 | LTR |
| 410 | LTR31 | ERV1 | LTR |
| 411 | UCON132b | hAT-Tip100 | DNA |
| 412 | MER57E1 | ERV1 | LTR |
| 413 | MER21B | ERVL | LTR |
| 414 | MamRep1161 | TcMar-Tigger | DNA |
| 415 | MLT1A1 | ERVL-MaLR | LTR |
| 416 | MER44D | TcMar-Tigger | DNA |

| | | | |
|---|---|---|---|
| 417 | LTR70 | ERV1 | LTR |
| 418 | MER54B | ERVL | LTR |
| 419 | LTR54B | ERV1 | LTR |
| 420 | MER34B | ERV1 | LTR |
| 421 | L1PBa | L1 | LINE |
| 422 | MER70B | ERVL | LTR |
| 423 | LTR109A2 | ERV1 | LTR |
| 424 | LTR44 | ERV1 | LTR |
| 425 | LTR58 | ERV1 | LTR |
| 426 | MLT2B2 | ERVL | LTR |
| 427 | MLT1A0 | ERVL-MaLR | LTR |
| 428 | L1PBb | L1 | LINE |
| 429 | L1MA6 | L1 | LINE |
| 430 | MER34C_ | ERV1 | LTR |
| 431 | MER67A | ERV1 | LTR |
| 432 | LTR75B | ERVL | LTR |
| 433 | MER34 | ERV1 | LTR |
| 434 | MER34C2 | ERV1 | LTR |
| 435 | MER94B | hAT-Blackjack | DNA |
| 436 | MER57F | ERV1 | LTR |
| 437 | MER95 | ERV1 | LTR |
| 438 | LTR108e_Mam | ERVL | LTR |
| 439 | Tigger1a_Mars | TcMar-Tigger | DNA |
| 440 | X8_LINE | CR1 | LINE |
| 441 | L1P4b | L1 | LINE |
| 442 | L1MA7 | L1 | LINE |
| 443 | HERVS71-int | ERV1 | LTR |
| 444 | Tigger4 | TcMar-Tigger | DNA |
| 445 | MER74B | ERVL | LTR |
| 446 | MER34C | ERV1 | LTR |
| 447 | HERV15-int | ERV1 | LTR |
| 448 | MER70C | ERVL | LTR |
| 449 | MLT2B1 | ERVL | LTR |
| 450 | HERV1_LTRe | ERV1 | LTR |
| 451 | UCON79 | DNA? | DNA? |
| 452 | MLT1A | ERVL-MaLR | LTR |
| 453 | MER57E2 | ERV1 | LTR |
| 454 | MSTB-int | ERVL-MaLR | LTR |
| 455 | MER45A | hAT-Tip100 | DNA |
| 456 | CR1-L3A_Croc | CR1 | LINE |
| 457 | MLT1N2-int | ERVL-MaLR | LTR |
| 458 | UCON8 | DNA | DNA |
| 459 | Ricksha_a | MULE-MuDR | DNA |
| 460 | L1P4c | L1 | LINE |
| 461 | Penelope1_Vert | Penelope | LINE |
| 462 | HERV9NC-int | ERV1 | LTR |
| 463 | MER83A-int | ERV1 | LTR |
| 464 | MER57E3 | ERV1 | LTR |
| 465 | MLT1E-int | ERVL-MaLR | LTR |
| 466 | HERV9-int | ERV1 | LTR |
| 467 | HERVFH19-int | ERV1 | LTR |
| 468 | MamRep488 | hAT-Tip100 | DNA |
| 469 | UCON78 | DNA | DNA |
| 470 | X2_LINE | CR1 | LINE |

|  |  |  |  |
|---|---|---|---|
| 471 | LTR91 | ERVL | LTR |
| 472 | PRIMAX-int | ERV1 | LTR |
| 473 | MER126 | DNA | DNA |
| 474 | MER34A | ERV1 | LTR |
| 475 | MER50C | ERV1 | LTR |
| 476 | MLT2B4 | ERVL | LTR |
| 477 | MLT1E1A-int | ERVL-MaLR | LTR |
| 478 | UCON33 | TcMar-Tigger | DNA |
| 479 | MSTB2-int | ERVL-MaLR | LTR |
| 480 | X5B_LINE | CR1 | LINE |
| 481 | HERVK22-int | ERVK | LTR |
| 482 | MER83B-int | ERV1 | LTR |
| 483 | MamGypsy2-LTR | Gypsy | LTR |
| 484 | MER70A | ERVL | LTR |
| 485 | MSTA1-int | ERVL-MaLR | LTR |
| 486 | MLT1E1-int | ERVL-MaLR | LTR |
| 487 | LTR38-int | ERV1 | LTR |
| 488 | MLT1A1-int | ERVL-MaLR | LTR |
| 489 | L1M1 | L1 | LINE |
| 490 | Charlie30a | hAT-Charlie | DNA |
| 491 | Charlie4 | hAT-Charlie | DNA |
| 492 | MADE2 | TcMar-Mariner | DNA |
| 493 | X6B_LINE | CR1 | LINE |
| 494 | Ricksha_b | MULE-MuDR | DNA |
| 495 | L1M3de | L1 | LINE |
| 496 | MER121 | hAT? | DNA |
| 497 | HERVFH21-int | ERV1 | LTR |
| 498 | L1PBa1 | L1 | LINE |
| 499 | UCON107 | hAT-Tag1 | DNA |
| 500 | HERVL66-int | ERVL | LTR |
| 501 | LTR53B | ERVL | LTR |
| 502 | MER131 | DNA? | DNA? |
| 503 | LTR108b_Mam | ERVL | LTR |
| 504 | UCON29 | PiggyBac? | DNA |
| 505 | MER31A | ERV1 | LTR |
| 506 | MER127 | TcMar-Tigger | DNA |
| 507 | X6A_LINE | CR1 | LINE |
| 508 | MER21C | ERVL | LTR |
| 509 | Mam_R4 | Dong-R4 | LINE |
| 510 | Chompy-7_Croc | PIF-Harbinger | DNA |
| 511 | L1M2 | L1 | LINE |
| 512 | AmnSINE1 | 5S-Deu-L2 | SINE |
| 513 | LTR81AB | Gypsy | LTR |
| 514 | HERVK14-int | ERVK | LTR |
| 515 | FordPrefect_a | hAT-Tip100 | DNA |
| 516 | X9_LINE | L1 | LINE |
| 517 | HERV17-int | ERV1 | LTR |
| 518 | MER88 | ERVL | LTR |
| 519 | L1MA8 | L1 | LINE |
| 520 | L1MA9 | L1 | LINE |
| 521 | LTR19-int | ERV1 | LTR |
| 522 | HERV30-int | ERV1 | LTR |
| 523 | MLT1-int | ERVL-MaLR | LTR |
| 524 | MLT2D | ERVL | LTR |

| | | | |
|---|---|---|---|
| 525 | L1MB1 | L1 | LINE |
| 526 | MSTB1-int | ERVL-MaLR | LTR |
| 527 | L1M2c | L1 | LINE |
| 528 | L1M2b | L1 | LINE |
| 529 | Tigger17b | TcMar-Tigger | DNA |
| 530 | Charlie10b | hAT-Charlie | DNA |
| 531 | MSTC-int | ERVL-MaLR | LTR |
| 532 | L1M3b | L1 | LINE |
| 533 | MLT1G-int | ERVL-MaLR | LTR |
| 534 | MER34A1 | ERV1 | LTR |
| 535 | MER89 | ERV1 | LTR |
| 536 | LTR108a_Mam | ERVL | LTR |
| 537 | HERVIP10F-int | ERV1 | LTR |
| 538 | ERV24B_Prim-int | ERV1 | LTR |
| 539 | MER92A | ERV1 | LTR |
| 540 | LFSINE_Vert | tRNA | SINE |
| 541 | MER67B | ERV1 | LTR |
| 542 | MER67D | ERV1 | LTR |
| 543 | L1MA10 | L1 | LINE |
| 544 | MLT1B | ERVL-MaLR | LTR |
| 545 | ERV24_Prim-int | ERV1 | LTR |
| 546 | MER34B-int | ERV1 | LTR |
| 547 | LTR57-int | ERVL | LTR |
| 548 | MLT1E2-int | ERVL-MaLR | LTR |
| 549 | LTR55 | ERVL? | LTR |
| 550 | LTR53 | ERVL | LTR |
| 551 | MER74C | ERVL | LTR |
| 552 | MER31B | ERV1 | LTR |
| 553 | L1MC1 | L1 | LINE |
| 554 | LTR33C | ERVL | LTR |
| 555 | MER97d | hAT-Tip100 | DNA |
| 556 | LTR108c_Mam | ERVL | LTR |
| 557 | MER58D | hAT-Charlie | DNA |
| 558 | L1M3c | L1 | LINE |
| 559 | MLT1C-int | ERVL-MaLR | LTR |
| 560 | Tigger17a | TcMar-Tigger | DNA |
| 561 | MER74A | ERVL | LTR |
| 562 | MER92D | ERV1 | LTR |
| 563 | MLT1C | ERVL-MaLR | LTR |
| 564 | LTR107_Mam | LTR | LTR |
| 565 | Charlie11 | hAT-Charlie | DNA |
| 566 | HERVKC4-int | ERVK | LTR |
| 567 | MLT-int | ERVL-MaLR | LTR |
| 568 | MLT2C2 | ERVL | LTR |
| 569 | MER77B | ERVL | LTR |
| 570 | MER77 | ERVL | LTR |
| 571 | MLT1D | ERVL-MaLR | LTR |
| 572 | MLT1E1 | ERVL-MaLR | LTR |
| 573 | LTR53-int | ERVL | LTR |
| 574 | L1MB2 | L1 | LINE |
| 575 | MLT1E | ERVL-MaLR | LTR |
| 576 | MER92B | ERV1 | LTR |
| 577 | Tigger17c | TcMar-Tigger | DNA |
| 578 | MER68C | ERVL | LTR |

| | | | |
|---|---|---|---|
| 579 | MER97b | hAT-Tip100 | DNA |
| 580 | LTR16D1 | ERVL | LTR |
| 581 | MER92C | ERV1 | LTR |
| 582 | Tigger6b | TcMar-Tigger | DNA |
| 583 | hAT-N1_Mam | hAT-Tip100? | DNA |
| 584 | LTR52 | ERVL | LTR |
| 585 | LTR75 | ERVL | LTR |
| 586 | LTR16D2 | ERVL | LTR |
| 587 | MLT1E2 | ERVL-MaLR | LTR |
| 588 | MER81 | hAT-Blackjack | DNA |
| 589 | CR1-16_AMi | CR1 | LINE |
| 590 | L1PA15-16 | L1 | LINE |
| 591 | MER68B | ERVL | LTR |
| 592 | MER76 | ERVL | LTR |
| 593 | MER106A | hAT-Charlie | DNA |
| 594 | L1MB3 | L1 | LINE |
| 595 | MER67C | ERV1 | LTR |
| 596 | MER106B | hAT-Charlie | DNA |
| 597 | MER68 | ERVL | LTR |
| 598 | MLT1B-int | ERVL-MaLR | LTR |
| 599 | MER70-int | ERVL | LTR |
| 600 | LTR108d_Mam | ERVL | LTR |
| 601 | MER45B | hAT-Tip100 | DNA |
| 602 | L1MB4 | L1 | LINE |
| 603 | L1M3d | L1 | LINE |
| 604 | LTR40A1 | ERVL | LTR |
| 605 | L1MC2 | L1 | LINE |
| 606 | MER53 | hAT | DNA |
| 607 | LTR40b | ERVL | LTR |
| 608 | LTR68 | ERV1 | LTR |
| 609 | MER45C | hAT-Tip100 | DNA |
| 610 | MLT1E3 | ERVL-MaLR | LTR |
| 611 | L1M3e | L1 | LINE |
| 612 | MSTD-int | ERVL-MaLR | LTR |
| 613 | MLT1E3-int | ERVL-MaLR | LTR |
| 614 | MER90 | ERV1 | LTR |
| 615 | Tigger6a | TcMar-Tigger | DNA |
| 616 | MLT2C1 | ERVL | LTR |
| 617 | ORSL | hAT-Tip100 | DNA |
| 618 | MLT1E1A | ERVL-MaLR | LTR |
| 619 | MER20 | hAT-Charlie | DNA |
| 620 | X7D_LINE | CR1 | LINE |
| 621 | MER58A | hAT-Charlie | DNA |
| 622 | LTR16B2 | ERVL | LTR |
| 623 | LTR40a | ERVL | LTR |
| 624 | MLT1F-int | ERVL-MaLR | LTR |
| 625 | MamRep1894 | hAT | DNA |
| 626 | Tigger9b | TcMar-Tigger | DNA |
| 627 | L1M3 | L1 | LINE |
| 628 | MER63A | hAT-Blackjack | DNA |
| 629 | MLT1G3-int | ERVL-MaLR | LTR |
| 630 | Charlie14a | hAT-Charlie | DNA |
| 631 | MER91B | hAT-Tip100 | DNA |
| 632 | LTR41B | ERVL | LTR |

| | | | |
|---|---|---|---|
| 633 | LTR16B | ERVL | LTR |
| 634 | MLT1F2 | ERVL-MaLR | LTR |
| 635 | LTR41 | ERVL | LTR |
| 636 | LTR52-int | ERVL | LTR |
| 637 | MER90a | ERV1 | LTR |
| 638 | LTR86A1 | ERVL | LTR |
| 639 | UCON55 | TcMar-Tigger | DNA |
| 640 | L5 | RTE-X | LINE |
| 641 | LTR41C | ERVL | LTR |
| 642 | L1MB8 | L1 | LINE |
| 643 | MER96B | hAT-Tip100 | DNA |
| 644 | MER58C | hAT-Charlie | DNA |
| 645 | LTR40c | ERVL | LTR |
| 646 | MER58B | hAT-Charlie | DNA |
| 647 | MLT1L-int | ERVL-MaLR | LTR |
| 648 | MER99 | hAT? | DNA |
| 649 | LTR102_Mam | ERVL | LTR |
| 650 | L1M3f | L1 | LINE |
| 651 | MLT1F | ERVL-MaLR | LTR |
| 652 | L1MB5 | L1 | LINE |
| 653 | ERV3-16A3_LTR | ERVL | LTR |
| 654 | Tigger17 | TcMar-Tigger | DNA |
| 655 | L1MB7 | L1 | LINE |
| 656 | MER63B | hAT-Blackjack | DNA |
| 657 | MLT1G1-int | ERVL-MaLR | LTR |
| 658 | MER3 | hAT-Charlie | DNA |
| 659 | MER110A | ERV1 | LTR |
| 660 | LTR80B | ERVL | LTR |
| 661 | LTR37A | ERV1 | LTR |
| 662 | LTR16B1 | ERVL | LTR |
| 663 | MLT2F | ERVL | LTR |
| 664 | MLT1F1 | ERVL-MaLR | LTR |
| 665 | MLT1G1 | ERVL-MaLR | LTR |
| 666 | MLT1H1 | ERVL-MaLR | LTR |
| 667 | MER94 | hAT-Blackjack | DNA |
| 668 | LTR16 | ERVL | LTR |
| 669 | LTR88b | Gypsy? | LTR |
| 670 | FordPrefect | hAT-Tip100 | DNA |
| 671 | MER124 | DNA? | DNA? |
| 672 | MER119 | hAT-Charlie | DNA |
| 673 | MER104 | TcMar-Tc2 | DNA |
| 674 | MER97a | hAT-Tip100 | DNA |
| 675 | MER110-int | ERV1 | LTR |
| 676 | MER5A1 | hAT-Charlie | DNA |
| 677 | DNA1_Mam | TcMar | DNA |
| 678 | L1MC | L1 | LINE |
| 679 | MLT1I-int | ERVL-MaLR | LTR |
| 680 | Charlie10 | hAT-Charlie | DNA |
| 681 | Charlie5 | hAT-Charlie | DNA |
| 682 | MERX | TcMar-Tigger | DNA |
| 683 | MLT1F1-int | ERVL-MaLR | LTR |
| 684 | MER45R | hAT-Tip100 | DNA |
| 685 | L1M2a | L1 | LINE |
| 686 | Tigger12c | TcMar-Tigger | DNA |

|  |  |  |  |
|---|---|---|---|
| 687 | L1MEg2 | L1 | LINE |
| 688 | Arthur1C | hAT-Tip100 | DNA |
| 689 | LTR16A2 | ERVL | LTR |
| 690 | L1MD3 | L1 | LINE |
| 691 | L1M4 | L1 | LINE |
| 692 | Charlie6 | hAT-Charlie | DNA |
| 693 | L1MD | L1 | LINE |
| 694 | L1MD1 | L1 | LINE |
| 695 | Charlie1b | hAT-Charlie | DNA |
| 696 | L1MC3 | L1 | LINE |
| 697 | L1M4a2 | L1 | LINE |
| 698 | L1MD2 | L1 | LINE |
| 699 | Charlie1a | hAT-Charlie | DNA |
| 700 | MER33 | hAT-Charlie | DNA |
| 701 | Looper | PiggyBac | DNA |
| 702 | MLT1G3 | ERVL-MaLR | LTR |
| 703 | HERVIP10B3-int | ERV1 | LTR |
| 704 | L1MCc | L1 | LINE |
| 705 | MER5C1 | hAT-Charlie | DNA |
| 706 | Charlie1 | hAT-Charlie | DNA |
| 707 | MER63C | hAT-Blackjack | DNA |
| 708 | MER5C | hAT-Charlie | DNA |
| 709 | MER5A | hAT-Charlie | DNA |
| 710 | Charlie10a | hAT-Charlie | DNA |
| 711 | MER105 | hAT-Charlie | DNA |
| 712 | X7C_LINE | CR1 | LINE |
| 713 | MLT1G | ERVL-MaLR | LTR |
| 714 | LTR86B2 | ERVL | LTR |
| 715 | LTR16A | ERVL | LTR |
| 716 | LTR105_Mam | ERVL | LTR |
| 717 | LTR80A | ERVL | LTR |
| 718 | Charlie7a | hAT-Charlie | DNA |
| 719 | Zaphod3 | hAT-Tip100 | DNA |
| 720 | L1ME2z | L1 | LINE |
| 721 | MER63D | hAT-Blackjack | DNA |
| 722 | MLT1H2-int | ERVL-MaLR | LTR |
| 723 | LTR83 | ERVL | LTR |
| 724 | Kanga1a | TcMar-Tc2 | DNA |
| 725 | MER91A | hAT-Tip100 | DNA |
| 726 | LTR37B | ERV1 | LTR |
| 727 | MER5B | hAT-Charlie | DNA |
| 728 | LTR87 | ERVL? | LTR |
| 729 | LTR86C | ERVL | LTR |
| 730 | MLT1H | ERVL-MaLR | LTR |
| 731 | MLT1F2-int | ERVL-MaLR | LTR |
| 732 | LTR50 | ERVL | LTR |
| 733 | LTR16E1 | ERVL | LTR |
| 734 | LTR16A1 | ERVL | LTR |
| 735 | L1MEb | L1 | LINE |
| 736 | LTR46-int | ERV1 | LTR |
| 737 | MER91C | hAT-Tip100 | DNA |
| 738 | MLT1K-int | ERVL-MaLR | LTR |
| 739 | MLT2E | ERVL | LTR |
| 740 | Charlie9 | hAT-Charlie | DNA |

| | | | |
|---|---|---|---|
| 741 | Kanga1c | TcMar-Tc2 | DNA |
| 742 | Cheshire | hAT-Charlie | DNA |
| 743 | MLT1J1 | ERVL-MaLR | LTR |
| 744 | L1ME3 | L1 | LINE |
| 745 | MamRep38 | hAT | DNA |
| 746 | Charlie17a | hAT-Charlie | DNA |
| 747 | LTR81B | Gypsy | LTR |
| 748 | LTR81A | Gypsy | LTR |
| 749 | MLT1H2 | ERVL-MaLR | LTR |
| 750 | LTR16D | ERVL | LTR |
| 751 | LTR16C | ERVL | LTR |
| 752 | LTR16E2 | ERVL | LTR |
| 753 | L1M4a1 | L1 | LINE |
| 754 | MER110 | ERV1 | LTR |
| 755 | MER4B-int | ERV1 | LTR |
| 756 | LTR103_Mam | ERV1? | LTR |
| 757 | MamGypLTR1c | Gypsy | LTR |
| 758 | MamGypLTR1a | Gypsy | LTR |
| 759 | LTR33A | ERVL | LTR |
| 760 | MER115 | hAT-Tip100 | DNA |
| 761 | CR1-3_Croc | CR1 | LINE |
| 762 | Charlie4a | hAT-Charlie | DNA |
| 763 | MER21-int | ERVL | LTR |
| 764 | MLT1I | ERVL-MaLR | LTR |
| 765 | X7B_LINE | CR1 | LINE |
| 766 | Charlie19a | hAT-Charlie | DNA |
| 767 | Arthur1A | hAT-Tip100 | DNA |
| 768 | L2-1_AMi | L2 | LINE |
| 769 | LTR33 | ERVL | LTR |
| 770 | Arthur1 | hAT-Tip100 | DNA |
| 771 | LTR82A | ERVL | LTR |
| 772 | Arthur1B | hAT-Tip100 | DNA |
| 773 | MamRTE1 | RTE-BovB | LINE |
| 774 | Zaphod2 | hAT-Tip100 | DNA |
| 775 | L1MEg1 | L1 | LINE |
| 776 | LTR79 | ERVL | LTR |
| 777 | MER97c | hAT-Tip100 | DNA |
| 778 | LTR104_Mam | Gypsy | LTR |
| 779 | L1M3a | L1 | LINE |
| 780 | Charlie26a | hAT-Charlie | DNA |
| 781 | LTR33B | ERVL | LTR |
| 782 | MER46C | TcMar-Tigger | DNA |
| 783 | MamGypLTR1d | Gypsy | LTR |
| 784 | MER102a | hAT-Charlie | DNA |
| 785 | MER102c | hAT-Charlie | DNA |
| 786 | MamGypLTR2c | Gypsy | LTR |
| 787 | Tigger14a | TcMar-Tigger | DNA |
| 788 | LTR88a | Gypsy? | LTR |
| 789 | Charlie17 | hAT-Charlie | DNA |
| 790 | Kanga1b | TcMar-Tc2 | DNA |
| 791 | MLT1J2 | ERVL-MaLR | LTR |
| 792 | L1MC4a | L1 | LINE |
| 793 | LTR82B | ERVL | LTR |
| 794 | L1MC4 | L1 | LINE |

|  |  |  |  |
|---|---|---|---|
| 795 | Charlie4z | hAT-Charlie | DNA |
| 796 | LTR33A_ | ERVL | LTR |
| 797 | HERVL18-int | ERVL | LTR |
| 798 | L1ME3A | L1 | LINE |
| 799 | LTR85c | Gypsy? | LTR |
| 800 | MER117 | hAT-Charlie | DNA |
| 801 | Kanga1 | TcMar-Tc2 | DNA |
| 802 | Kanga1d | TcMar-Tc2 | DNA |
| 803 | MamRep1879 | hAT-Tip100? | DNA |
| 804 | MLT1J | ERVL-MaLR | LTR |
| 805 | HAL1M8 | L1 | LINE |
| 806 | Tigger18a | TcMar-Tigger | DNA |
| 807 | L1MC5 | L1 | LINE |
| 808 | Plat_L3 | CR1 | LINE |
| 809 | Tigger16a | TcMar-Tigger | DNA |
| 810 | MamTip1 | hAT-Tip100 | DNA |
| 811 | MamSINE1 | tRNA-RTE | SINE |
| 812 | MLT1J2-int | ERVL-MaLR | LTR |
| 813 | Tigger9a | TcMar-Tigger | DNA |
| 814 | MER112 | hAT-Charlie | DNA |
| 815 | HERVL74-int | ERVL | LTR |
| 816 | LTR67B | ERVL | LTR |
| 817 | HERVK13-int | ERVK | LTR |
| 818 | MIR | MIR | SINE |
| 819 | MamGyp-int | Gypsy | LTR |
| 820 | MER102b | hAT-Charlie | DNA |
| 821 | MER20B | hAT-Charlie | DNA |
| 822 | LTR81C | Gypsy | LTR |
| 823 | L1ME5 | L1 | LINE |
| 824 | Charlie15a | hAT-Charlie | DNA |
| 825 | Charlie7 | hAT-Charlie | DNA |
| 826 | LTR106_Mam | LTR | LTR |
| 827 | HERVE-int | ERV1 | LTR |
| 828 | MLT1L | ERVL-MaLR | LTR |
| 829 | Charlie16a | hAT-Charlie | DNA |
| 830 | LTR103b_Mam | ERV1? | LTR |
| 831 | HUERS-P2-int | ERV1 | LTR |
| 832 | LTR84b | ERVL | LTR |
| 833 | HUERS-P1-int | ERV1 | LTR |
| 834 | L1MEh | L1 | LINE |
| 835 | LTR86A2 | ERVL | LTR |
| 836 | HERVK-int | ERVK | LTR |
| 837 | Charlie22a | hAT-Charlie | DNA |
| 838 | HERVK3-int | ERVK | LTR |
| 839 | L1M4c | L1 | LINE |
| 840 | MER113A | hAT-Charlie | DNA |
| 841 | MLT1H1-int | ERVL-MaLR | LTR |
| 842 | L1MEa | L1 | LINE |
| 843 | MIR1_Amn | MIR | SINE |
| 844 | ERVL-int | ERVL | LTR |
| 845 | MamRep4096 | hAT-Tip100 | DNA |
| 846 | LTR85a | Gypsy? | LTR |
| 847 | MLT1K | ERVL-MaLR | LTR |
| 848 | MER68-int | ERVL | LTR |

|  |  |  |  |
|---|---|---|---|
| 849 | Tigger12A | TcMar-Tigger | DNA |
| 850 | CR1_Mam | CR1 | LINE |
| 851 | UCON74 | DNA? | DNA? |
| 852 | LTR89 | ERVL? | LTR |
| 853 | LTR90A | LTR | LTR |
| 854 | Chap1_Mam | hAT-Charlie | DNA |
| 855 | L1MEg | L1 | LINE |
| 856 | LTR101_Mam | ERVL | LTR |
| 857 | L1M5 | L1 | LINE |
| 858 | Charlie13b | hAT-Charlie | DNA |
| 859 | L1MDa | L1 | LINE |
| 860 | Charlie21a | hAT-Charlie | DNA |
| 861 | MamGypLTR1b | Gypsy | LTR |
| 862 | L1MCb | L1 | LINE |
| 863 | L1MCa | L1 | LINE |
| 864 | MLT1A0-int | ERVL-MaLR | LTR |
| 865 | L1MEf | L1 | LINE |
| 866 | L1MEc | L1 | LINE |
| 867 | L1ME1 | L1 | LINE |
| 868 | L1M4b | L1 | LINE |
| 869 | LTR78B | ERV1 | LTR |
| 870 | L1MC5a | L1 | LINE |
| 871 | L1ME3E | L1 | LINE |
| 872 | L1ME2 | L1 | LINE |
| 873 | LTR65 | ERV1 | LTR |
| 874 | Charlie2a | hAT-Charlie | DNA |
| 875 | LTR85b | Gypsy? | LTR |
| 876 | MLT1N2 | ERVL-MaLR | LTR |
| 877 | MLT1M | ERVL-MaLR | LTR |
| 878 | BLACKJACK | hAT-Blackjack | DNA |
| 879 | L2a | L2 | LINE |
| 880 | L1MEi | L1 | LINE |
| 881 | LTR78 | ERV1 | LTR |
| 882 | Charlie18a | hAT-Charlie | DNA |
| 883 | L1ME4b | L1 | LINE |
| 884 | MER101-int | ERV1 | LTR |
| 885 | Zaphod | hAT-Tip100 | DNA |
| 886 | Charlie15b | hAT-Charlie | DNA |
| 887 | Charlie17b | hAT-Charlie | DNA |
| 888 | L1ME3B | L1 | LINE |
| 889 | MamRep137 | TcMar-Tigger | DNA |
| 890 | Charlie29a | hAT-Charlie | DNA |
| 891 | L1M7 | L1 | LINE |
| 892 | L1ME3F | L1 | LINE |
| 893 | Charlie2b | hAT-Charlie | DNA |
| 894 | L1ME3D | L1 | LINE |
| 895 | HAL1b | L1 | LINE |
| 896 | LTR88c | Gypsy? | LTR |
| 897 | MamGypLTR3a | Gypsy | LTR |
| 898 | Tigger13a | TcMar-Tigger | DNA |
| 899 | MLT1O | ERVL-MaLR | LTR |
| 900 | MLT1J-int | ERVL-MaLR | LTR |
| 901 | Charlie24 | hAT-Charlie | DNA |
| 902 | ORSL-2a | hAT-Tip100 | DNA |

| | | | |
|---|---|---|---|
| 903 | MamGypLTR2b | Gypsy | LTR |
| 904 | MamRep1151 | LTR? | LTR? |
| 905 | Kanga11a | TcMar-Tc2 | DNA |
| 906 | MamRep1527 | LTR | LTR |
| 907 | MLT1D-int | ERVL-MaLR | LTR |
| 908 | LTR84a | ERVL | LTR |
| 909 | Charlie8 | hAT-Charlie | DNA |
| 910 | L2b | L2 | LINE |
| 911 | MER84-int | ERV1 | LTR |
| 912 | MER113B | hAT-Charlie | DNA |
| 913 | MLT1H-int | ERVL-MaLR | LTR |
| 914 | L1MEd | L1 | LINE |
| 915 | L1M8 | L1 | LINE |
| 916 | Charlie23a | hAT-Charlie | DNA |
| 917 | LTR90B | LTR | LTR |
| 918 | LTR81 | Gypsy | LTR |
| 919 | L1MDb | L1 | LINE |
| 920 | HERV4_I-int | ERV1 | LTR |
| 921 | HERV35I-int | ERV1 | LTR |
| 922 | MamGypLTR3 | Gypsy | LTR |
| 923 | MamRep434 | TcMar-Tigger | DNA |
| 924 | Kanga2_a | TcMar-Tc2 | DNA |
| 925 | MLT1J1-int | ERVL-MaLR | LTR |
| 926 | ORSL-2b | hAT-Tip100 | DNA |
| 927 | MamTip3 | hAT-Tip100 | DNA |
| 928 | MIRb | MIR | SINE |
| 929 | L2 | L2 | LINE |
| 930 | Tigger11a | TcMar-Tigger | DNA |
| 931 | L1ME3G | L1 | LINE |
| 932 | Tigger20a | TcMar-Tigger | DNA |
| 933 | MER103C | hAT-Charlie | DNA |
| 934 | L1M6 | L1 | LINE |
| 935 | MamRep605 | LTR? | LTR? |
| 936 | L4_A_Mam | RTE-X | LINE |
| 937 | MER113 | hAT-Charlie | DNA |
| 938 | Tigger16b | TcMar-Tigger | DNA |
| 939 | HAL1 | L1 | LINE |
| 940 | HAL1ME | L1 | LINE |
| 941 | Charlie25 | hAT-Charlie | DNA |
| 942 | MLT1A-int | ERVL-MaLR | LTR |
| 943 | MamTip2 | hAT-Tip100 | DNA |
| 944 | LTR86B1 | ERVL | LTR |
| 945 | L1M6B | L1 | LINE |
| 946 | L1ME3Cz | L1 | LINE |
| 947 | Tigger8 | TcMar-Tigger | DNA |
| 948 | L2c | L2 | LINE |
| 949 | HERVE_a-int | ERV1 | LTR |
| 950 | L1ME4a | L1 | LINE |
| 951 | L3b | CR1 | LINE |
| 952 | Tigger19a | TcMar-Tigger | DNA |
| 953 | X7A_LINE | CR1 | LINE |
| 954 | Tigger15a | TcMar-Tigger | DNA |
| 955 | L1MEj | L1 | LINE |
| 956 | MIR3 | MIR | SINE |

| | | | |
|---|---|---|---|
| 957 | L4_B_Mam | RTE-X | LINE |
| 958 | MIRc | MIR | SINE |
| 959 | L1ME4c | L1 | LINE |
| 960 | L3 | CR1 | LINE |
| 961 | Tigger12 | TcMar-Tigger | DNA |
| 962 | Tigger10 | TcMar-Tigger | DNA |
| 963 | Tigger19b | TcMar-Tigger | DNA |
| 964 | Harlequin-int | ERV1 | LTR |
| 965 | HERVL-int | ERVL | LTR |
| 966 | HERVH-int | ERV1 | LTR |
| 967 | MamGypsy2-I | Gypsy | LTR |
| 968 | L1ME3C | L1 | LINE |
| 969 | ERV3-16A3_I-int | ERVL | LTR |
| 970 | MER92-int | ERV1 | LTR |
| 971 | MER31-int | ERV1 | LTR |
| 972 | HUERS-P3b-int | ERV1 | LTR |
| 973 | LTR49-int | ERV1 | LTR |
| 974 | MER61-int | ERV1 | LTR |
| 975 | HERVK11-int | ERVK | LTR |
| 976 | HERV9N-int | ERV1 | LTR |
| 977 | Ricksha | MULE-MuDR | DNA |
| 978 | L4_C_Mam | RTE-X | LINE |
| 979 | PABL_A-int | ERV1 | LTR |
| 980 | LTR43-int | ERV1 | LTR |
| 981 | MER89-int | ERV1 | LTR |
| 982 | MER65-int | ERV1 | LTR |
| 983 | PRIMA4-int | ERV1 | LTR |
| 984 | L2-3_Crp | L2 | LINE |
| 985 | HERV3-int | ERV1 | LTR |
| 986 | HERVK9-int | ERVK | LTR |
| 987 | HERVK14C-int | ERVK | LTR |
| 988 | MER66-int | ERV1 | LTR |
| 989 | HERVIP10FH-int | ERV1 | LTR |
| 990 | MER34-int | ERV1 | LTR |
| 991 | HERVI-int | ERV1 | LTR |
| 992 | LTR25-int | ERV1 | LTR |
| 993 | ERVL47-int | ERVL | LTR |
| 994 | Charlie13a | hAT-Charlie | DNA |
| 995 | PABL_B-int | ERV1 | LTR |
| 996 | HERV16-int | ERVL | LTR |
| 997 | LTR39-int | ERV1 | LTR |
| 998 | PRIMA41-int | ERV1 | LTR |
| 999 | Charlie20a | hAT-Charlie | DNA |
| 1000 | MER52-int | ERV1 | LTR |
| 1001 | LTR37-int | ERV1 | LTR |
| 1002 | HERVP71A-int | ERV1 | LTR |
| 1003 | MER4-int | ERV1 | LTR |
| 1004 | MER57A-int | ERV1 | LTR |
| 1005 | HERVL40-int | ERVL | LTR |
| 1006 | MER51-int | ERV1 | LTR |
| 1007 | LOR1-int | ERV1 | LTR |
| 1008 | LTR23-int | ERV1 | LTR |
| 1009 | HUERS-P3-int | ERV1 | LTR |
| 1010 | ERVL-B4-int | ERVL | LTR |

| | | | |
|---|---|---|---|
| 1011 | MARNA | TcMar-Mariner | DNA |
| 1012 | MER57-int | ERV1 | LTR |
| 1013 | ERVL-E-int | ERVL | LTR |
| 1014 | MER41-int | ERV1 | LTR |
| 1015 | HERVH48-int | ERV1 | LTR |

# Appendix B

# The TE ordering in the human genome predicted from the recursive interruption model by applying Tabu search on the adjacency matrix of the TE-interaction network

| Age order<br>(youngest to oldest) | TE name | TE family | TE type |
|---|---|---|---|
| 1 | AluYc | SINE | Alu |
| 2 | LTR5A | LTR | ERVK |
| 3 | AluYk4 | SINE | Alu |
| 4 | AluYa5 | SINE | Alu |
| 5 | MER9a2 | LTR | ERVK |
| 6 | AluYb8 | SINE | Alu |
| 7 | AluYe5 | SINE | Alu |
| 8 | AluYc3 | SINE | Alu |
| 9 | LTR5_Hs | LTR | ERVK |
| 10 | AluY | SINE | Alu |
| 11 | L1PA2 | LINE | L1 |
| 12 | AluYk2 | SINE | Alu |
| 13 | HERV1_LTRb | LTR | ERV1 |
| 14 | LTR13 | LTR | ERVK |
| 15 | HERVE_a-int | LTR | ERV1 |
| 16 | AluYk3 | SINE | Alu |
| 17 | LTR2 | LTR | ERV1 |
| 18 | AluYm1 | SINE | Alu |
| 19 | AluYf1 | SINE | Alu |
| 20 | AluYb9 | SINE | Alu |
| 21 | HERV9NC-int | LTR | ERV1 |
| 22 | LTR7 | LTR | ERV1 |
| 23 | LTR21A | LTR | ERV1 |
| 24 | LTR12C | LTR | ERV1 |
| 25 | AluYh3 | SINE | Alu |
| 26 | AluYj4 | SINE | Alu |
| 27 | LTR7B | LTR | ERV1 |
| 28 | LTR14 | LTR | ERVK |
| 29 | MER9a1 | LTR | ERVK |
| 30 | MER11C | LTR | ERVK |
| 31 | LTR6A | LTR | ERV1 |
| 32 | AluSp | SINE | Alu |
| 33 | LTR13_ | LTR | ERVK |
| 34 | AluSc8 | SINE | Alu |
| 35 | AluSg | SINE | Alu |

| | | | |
|---|---|---|---|
| 36 | AluSg4 | SINE | Alu |
| 37 | AluSq10 | SINE | Alu |
| 38 | AluSc5 | SINE | Alu |
| 39 | AluSg7 | SINE | Alu |
| 40 | MER11B | LTR | ERVK |
| 41 | LTR5B | LTR | ERVK |
| 42 | AluYg6 | SINE | Alu |
| 43 | HERVK11D-int | LTR | ERVK |
| 44 | MER11D | LTR | ERVK |
| 45 | AluYd8 | SINE | Alu |
| 46 | AluSx3 | SINE | Alu |
| 47 | AluYi6 | SINE | Alu |
| 48 | L1HS | LINE | L1 |
| 49 | L1PA5 | LINE | L1 |
| 50 | LTR12B | LTR | ERV1 |
| 51 | MER11A | LTR | ERVK |
| 52 | AluSq4 | SINE | Alu |
| 53 | HERV9N-int | LTR | ERV1 |
| 54 | AluSq | SINE | Alu |
| 55 | LTR22C2 | LTR | ERVK |
| 56 | HERV9-int | LTR | ERV1 |
| 57 | LTR10F | LTR | ERV1 |
| 58 | LTR10G | LTR | ERV1 |
| 59 | LTR17 | LTR | ERV1 |
| 60 | AluSq2 | SINE | Alu |
| 61 | LTR12F | LTR | ERV1 |
| 62 | LTR14C | LTR | ERVK |
| 63 | LTR12 | LTR | ERV1 |
| 64 | LTR12_ | LTR | ERV1 |
| 65 | LTR5 | LTR | ERVK |
| 66 | LTR7A | LTR | ERV1 |
| 67 | LTR12D | LTR | ERV1 |
| 68 | Alu | SINE | Alu |
| 69 | LTR22B | LTR | ERVK |
| 70 | AluYe6 | SINE | Alu |
| 71 | AluYk12 | SINE | Alu |
| 72 | LTR2B | LTR | ERV1 |
| 73 | LTR2C | LTR | ERV1 |
| 74 | LTR3A | LTR | ERVK |
| 75 | LTR6B | LTR | ERV1 |
| 76 | LTR10D | LTR | ERV1 |
| 77 | LTR14B | LTR | ERVK |
| 78 | AluSx4 | SINE | Alu |
| 79 | AluSc | SINE | Alu |
| 80 | AluSx1 | SINE | Alu |
| 81 | L1P1 | LINE | L1 |
| 82 | L1PA4 | LINE | L1 |
| 83 | AluYi6_4d | SINE | Alu |
| 84 | LTR13A | LTR | ERVK |
| 85 | LTR1B1 | LTR | ERV1 |
| 86 | LTR27 | LTR | ERV1 |
| 87 | LTR7C | LTR | ERV1 |
| 88 | LTR18A | LTR | ERVL |
| 89 | AluYk11 | SINE | Alu |

|  |  |  |  |
|---|---|---|---|
| 90 | LTR30 | LTR | ERV1 |
| 91 | LTR3B_ | LTR | ERVK |
| 92 | AluSz6 | SINE | Alu |
| 93 | AluYh3a3 | SINE | Alu |
| 94 | LTR3B | LTR | ERVK |
| 95 | MER9a3 | LTR | ERVK |
| 96 | LTR10A | LTR | ERV1 |
| 97 | LTR76 | LTR | ERV1 |
| 98 | AluSz | SINE | Alu |
| 99 | L1PA3 | LINE | L1 |
| 100 | HERVI-int | LTR | ERV1 |
| 101 | HERV15-int | LTR | ERV1 |
| 102 | LTR15 | LTR | ERV1 |
| 103 | HERV1_I-int | LTR | ERV1 |
| 104 | LTR10C | LTR | ERV1 |
| 105 | LTR1A2 | LTR | ERV1 |
| 106 | HERVIP10B3-int | LTR | ERV1 |
| 107 | L1PA6 | LINE | L1 |
| 108 | HERV1_LTRa | LTR | ERV1 |
| 109 | LTR10B1 | LTR | ERV1 |
| 110 | LTR19C | LTR | ERV1 |
| 111 | HERVK22-int | LTR | ERVK |
| 112 | LTR12E | LTR | ERV1 |
| 113 | LTR22B1 | LTR | ERVK |
| 114 | LTR61 | LTR | ERV1 |
| 115 | LTR22B2 | LTR | ERVK |
| 116 | LTR7Y | LTR | ERV1 |
| 117 | LTR10E | LTR | ERV1 |
| 118 | LTR10B2 | LTR | ERV1 |
| 119 | LTR22E | LTR | ERVK |
| 120 | LTR22A | LTR | ERVK |
| 121 | AluSx | SINE | Alu |
| 122 | LTR22 | LTR | ERVK |
| 123 | LTR4 | LTR | ERV1 |
| 124 | Penelope1_Vert | LINE | Penelope |
| 125 | LTR22C0 | LTR | ERVK |
| 126 | LTR3 | LTR | ERVK |
| 127 | THE1-int | LTR | ERVL-MaLR |
| 128 | L1PA8 | LINE | L1 |
| 129 | L1PA7 | LINE | L1 |
| 130 | HERVFH19-int | LTR | ERV1 |
| 131 | LTR19A | LTR | ERV1 |
| 132 | MER61C | LTR | ERV1 |
| 133 | LTR9D | LTR | ERV1 |
| 134 | THE1B | LTR | ERVL-MaLR |
| 135 | LTR9A1 | LTR | ERV1 |
| 136 | LTR46 | LTR | ERV1 |
| 137 | THE1A-int | LTR | ERVL-MaLR |
| 138 | LTR71B | LTR | ERV1 |
| 139 | MER61E | LTR | ERV1 |
| 140 | LTR66 | LTR | ERVL |
| 141 | HERVH48-int | LTR | ERV1 |
| 142 | MLT2A1 | LTR | ERVL |
| 143 | MER48 | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 144 | LTR25 | LTR | ERV1 |
| 145 | LTR9B | LTR | ERV1 |
| 146 | PABL_A | LTR | ERV1 |
| 147 | MER61B | LTR | ERV1 |
| 148 | LTR2752 | LTR | ERV1 |
| 149 | MER52C | LTR | ERV1 |
| 150 | LTR1F1 | LTR | ERV1 |
| 151 | MADE1 | DNA | TcMar-Mariner |
| 152 | MER52D | LTR | ERV1 |
| 153 | THE1A | LTR | ERVL-MaLR |
| 154 | MER51B | LTR | ERV1 |
| 155 | LTR14A | LTR | ERVK |
| 156 | LTR18B | LTR | ERVL |
| 157 | LTR19B | LTR | ERV1 |
| 158 | LTR71A | LTR | ERV1 |
| 159 | LTR1B | LTR | ERV1 |
| 160 | LTR9 | LTR | ERV1 |
| 161 | LTR77 | LTR | ERV1 |
| 162 | MER4A1 | LTR | ERV1 |
| 163 | LTR1E | LTR | ERV1 |
| 164 | LTR1D | LTR | ERV1 |
| 165 | MER61A | LTR | ERV1 |
| 166 | THE1C | LTR | ERVL-MaLR |
| 167 | MER57A1 | LTR | ERV1 |
| 168 | MER41A | LTR | ERV1 |
| 169 | MER51C | LTR | ERV1 |
| 170 | MER61F | LTR | ERV1 |
| 171 | LTR9C | LTR | ERV1 |
| 172 | MER51A | LTR | ERV1 |
| 173 | HERV1_LTRc | LTR | ERV1 |
| 174 | LTR27C | LTR | ERV1 |
| 175 | MER61D | LTR | ERV1 |
| 176 | MER50B | LTR | ERV1 |
| 177 | LTR28 | LTR | ERV1 |
| 178 | PABL_B | LTR | ERV1 |
| 179 | MER4A1_ | LTR | ERV1 |
| 180 | MER4E | LTR | ERV1 |
| 181 | MER52A | LTR | ERV1 |
| 182 | LTR28C | LTR | ERV1 |
| 183 | LTR1A1 | LTR | ERV1 |
| 184 | MER9B | LTR | ERVK |
| 185 | LTR8 | LTR | ERV1 |
| 186 | MLT2A2 | LTR | ERVL |
| 187 | LTR27D | LTR | ERV1 |
| 188 | LTR1D1 | LTR | ERV1 |
| 189 | THE1D | LTR | ERVL-MaLR |
| 190 | MER30 | DNA | hAT-Charlie |
| 191 | FRAM | SINE | Alu |
| 192 | LTR43 | LTR | ERV1 |
| 193 | MER123 | DNA? | DNA? |
| 194 | MER1B | DNA | hAT-Charlie |
| 195 | LTR38C | LTR | ERV1 |
| 196 | MER85 | DNA | PiggyBac |
| 197 | LTR1B0 | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 198 | HERV-Fc1-int | LTR | ERV1 |
| 199 | MER83 | LTR | ERV1 |
| 200 | CR1-L3B_Croc | LINE | CR1 |
| 201 | AluJo | SINE | Alu |
| 202 | LTR1 | LTR | ERV1 |
| 203 | MER41B | LTR | ERV1 |
| 204 | PRIMA4_LTR | LTR | ERV1 |
| 205 | L1PA10 | LINE | L1 |
| 206 | MER51E | LTR | ERV1 |
| 207 | MER107 | DNA | hAT-Charlie |
| 208 | LTR22C | LTR | ERVK |
| 209 | LTR10B | LTR | ERV1 |
| 210 | UCON107 | DNA | hAT-Tag1 |
| 211 | MER66B | LTR | ERV1 |
| 212 | MER83B | LTR | ERV1 |
| 213 | LTR1C3 | LTR | ERV1 |
| 214 | LTR47B4 | LTR | ERVL |
| 215 | AluJb | SINE | Alu |
| 216 | HERVL18-int | LTR | ERVL |
| 217 | L1P3 | LINE | L1 |
| 218 | LTR27E | LTR | ERV1 |
| 219 | LTR18C | LTR | ERVL |
| 220 | LTR28B | LTR | ERV1 |
| 221 | Charlie3 | DNA | hAT-Charlie |
| 222 | FLAM_C | SINE | Alu |
| 223 | LTR27B | LTR | ERV1 |
| 224 | MER75 | DNA | PiggyBac |
| 225 | MER4E1 | LTR | ERV1 |
| 226 | MER83C | LTR | ERV1 |
| 227 | MER75A | DNA | PiggyBac |
| 228 | LTR1C | LTR | ERV1 |
| 229 | AluJr | SINE | Alu |
| 230 | MER41C | LTR | ERV1 |
| 231 | L1P2 | LINE | L1 |
| 232 | L1P5 | LINE | L1 |
| 233 | LTR21B | LTR | ERV1 |
| 234 | MER30B | DNA | hAT-Charlie |
| 235 | L1PB | LINE | L1 |
| 236 | HERV-Fc1_LTR3 | LTR | ERV1 |
| 237 | HERV1_LTRd | LTR | ERV1 |
| 238 | AluYa8 | SINE | Alu |
| 239 | L1PB1 | LINE | L1 |
| 240 | HSMAR1 | DNA | TcMar-Mariner |
| 241 | L1PA8A | LINE | L1 |
| 242 | L1PA11 | LINE | L1 |
| 243 | MER1A | DNA | hAT-Charlie |
| 244 | MSTA | LTR | ERVL-MaLR |
| 245 | MER4A | LTR | ERV1 |
| 246 | LOR1b | LTR | ERV1 |
| 247 | L1PA12 | LINE | L1 |
| 248 | MST-int | LTR | ERVL-MaLR |
| 249 | MLT2B3 | LTR | ERVL |
| 250 | MER50 | LTR | ERV1 |
| 251 | LTR26 | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 252 | L1PA14 | LINE | L1 |
| 253 | MER4D1 | LTR | ERV1 |
| 254 | MER57C2 | LTR | ERV1 |
| 255 | L1PBb | LINE | L1 |
| 256 | LTR72 | LTR | ERV1 |
| 257 | L1PBa | LINE | L1 |
| 258 | FLAM_A | SINE | Alu |
| 259 | MER57B1 | LTR | ERV1 |
| 260 | THE1D-int | LTR | ERVL-MaLR |
| 261 | L1PB2 | LINE | L1 |
| 262 | AluJr4 | SINE | Alu |
| 263 | L1PREC2 | LINE | L1 |
| 264 | L1PA13 | LINE | L1 |
| 265 | MER4B | LTR | ERV1 |
| 266 | LTR8A | LTR | ERV1 |
| 267 | MER84 | LTR | ERV1 |
| 268 | LTR60B | LTR | ERV1 |
| 269 | MER66C | LTR | ERV1 |
| 270 | Tigger3d | DNA | TcMar-Tigger |
| 271 | HERVKC4-int | LTR | ERVK |
| 272 | PrimLTR79 | LTR | ERV1 |
| 273 | LTR1C1 | LTR | ERV1 |
| 274 | HERVIP10FH-int | LTR | ERV1 |
| 275 | LTR35A | LTR | ERV1 |
| 276 | L1MA1 | LINE | L1 |
| 277 | hAT-16_Crp | DNA | hAT-Charlie |
| 278 | MER75B | DNA | PiggyBac |
| 279 | LTR36 | LTR | ERV1 |
| 280 | MER4C | LTR | ERV1 |
| 281 | LTR57 | LTR | ERVL |
| 282 | LTR8B | LTR | ERV1 |
| 283 | L1MA2 | LINE | L1 |
| 284 | MER41D | LTR | ERV1 |
| 285 | MSTB-int | LTR | ERVL-MaLR |
| 286 | LTR32 | LTR | ERVL |
| 287 | MER66A | LTR | ERV1 |
| 288 | MSTB | LTR | ERVL-MaLR |
| 289 | LOR1a | LTR | ERV1 |
| 290 | MER21A | LTR | ERVL |
| 291 | AluYh7 | SINE | Alu |
| 292 | L1PB3 | LINE | L1 |
| 293 | L1PA15 | LINE | L1 |
| 294 | L1M | LINE | L1 |
| 295 | LTR45C | LTR | ERV1 |
| 296 | L1MA3 | LINE | L1 |
| 297 | LTR38A1 | LTR | ERV1 |
| 298 | MER72 | LTR | ERV1 |
| 299 | MER66D | LTR | ERV1 |
| 300 | LTR35B | LTR | ERV1 |
| 301 | MER4D0 | LTR | ERV1 |
| 302 | LTR38B | LTR | ERV1 |
| 303 | LTR73 | LTR | ERV1 |
| 304 | LTR47A | LTR | ERVL |
| 305 | FAM | SINE | Alu |

|  |  |  |  |
|---|---|---|---|
| 306 | L1PA16 | LINE | L1 |
| 307 | LTR44 | LTR | ERV1 |
| 308 | MER49 | LTR | ERV1 |
| 309 | LTR43B | LTR | ERV1 |
| 310 | MER4D | LTR | ERV1 |
| 311 | L1PA17 | LINE | L1 |
| 312 | LTR62 | LTR | ERVL |
| 313 | MER51D | LTR | ERV1 |
| 314 | LTR1F2 | LTR | ERV1 |
| 315 | LTR39 | LTR | ERV1 |
| 316 | MSTB1-int | LTR | ERVL-MaLR |
| 317 | L1P4 | LINE | L1 |
| 318 | LTR26C | LTR | ERV1 |
| 319 | LTR26E | LTR | ERV1 |
| 320 | MER70A | LTR | ERVL |
| 321 | L1PB4 | LINE | L1 |
| 322 | MER41E | LTR | ERV1 |
| 323 | MSTB1 | LTR | ERVL-MaLR |
| 324 | LTR75_1 | LTR | ERV1 |
| 325 | MER87 | LTR | ERV1 |
| 326 | LTR38 | LTR | ERV1 |
| 327 | LTR24 | LTR | ERV1 |
| 328 | MER87B | LTR | ERV1 |
| 329 | LTR26D | LTR | ERV1 |
| 330 | LTR47B2 | LTR | ERVL |
| 331 | LTR24C | LTR | ERV1 |
| 332 | LTR64 | LTR | ERV1 |
| 333 | AmnSINE1 | SINE | 5S-Deu-L2 |
| 334 | MER65C | LTR | ERV1 |
| 335 | LTR47B3 | LTR | ERVL |
| 336 | MER72B | LTR | ERV1 |
| 337 | LTR34 | LTR | ERV1 |
| 338 | MER39B | LTR | ERV1 |
| 339 | LTR51 | LTR | ERV1 |
| 340 | LTR59 | LTR | ERV1 |
| 341 | LTR06 | LTR | ERV1 |
| 342 | LTR45B | LTR | ERV1 |
| 343 | LTR56 | LTR | ERV1 |
| 344 | LTR29 | LTR | ERV1 |
| 345 | MER57B2 | LTR | ERV1 |
| 346 | L1M2c | LINE | L1 |
| 347 | L1P4a | LINE | L1 |
| 348 | MER8 | DNA | TcMar-Tigger |
| 349 | LTR47A2 | LTR | ERVL |
| 350 | L1MA5A | LINE | L1 |
| 351 | LTR35 | LTR | ERV1 |
| 352 | LTR48B | LTR | ERV1 |
| 353 | HSMAR2 | DNA | TcMar-Mariner |
| 354 | Tigger2 | DNA | TcMar-Tigger |
| 355 | MER65A | LTR | ERV1 |
| 356 | Tigger2b_Pri | DNA | TcMar-Tigger |
| 357 | Tigger2a | DNA | TcMar-Tigger |
| 358 | MSTA1-int | LTR | ERVL-MaLR |
| 359 | MSTA1 | LTR | ERVL-MaLR |

| | | | |
|---|---|---|---|
| 360 | L1PBa1 | LINE | L1 |
| 361 | LTR72B | LTR | ERV1 |
| 362 | LTR49 | LTR | ERV1 |
| 363 | MER34D | LTR | ERV1 |
| 364 | L1M3f | LINE | L1 |
| 365 | MSTB2-int | LTR | ERVL-MaLR |
| 366 | MER54A | LTR | ERVL |
| 367 | MSTB2 | LTR | ERVL-MaLR |
| 368 | MER101 | LTR | ERV1 |
| 369 | MER4CL34 | LTR | ERV1 |
| 370 | L1MA4 | LINE | L1 |
| 371 | MSTC-int | LTR | ERVL-MaLR |
| 372 | MSTC | LTR | ERVL-MaLR |
| 373 | MER47A | DNA | TcMar-Tigger |
| 374 | Tigger4b | DNA | TcMar-Tigger |
| 375 | LTR42 | LTR | ERVL |
| 376 | LTR23 | LTR | ERV1 |
| 377 | MLT1A1-int | LTR | ERVL-MaLR |
| 378 | LTR45 | LTR | ERV1 |
| 379 | MER6A | DNA | TcMar-Tigger |
| 380 | LTR24B | LTR | ERV1 |
| 381 | MER6B | DNA | TcMar-Tigger |
| 382 | MER65D | LTR | ERV1 |
| 383 | MER50C | LTR | ERV1 |
| 384 | MER6 | DNA | TcMar-Tigger |
| 385 | Tigger3a | DNA | TcMar-Tigger |
| 386 | L1MA5 | LINE | L1 |
| 387 | MER57C1 | LTR | ERV1 |
| 388 | L1MA4A | LINE | L1 |
| 389 | LTR54 | LTR | ERV1 |
| 390 | LTR1F | LTR | ERV1 |
| 391 | THE1C-int | LTR | ERVL-MaLR |
| 392 | Tigger1 | DNA | TcMar-Tigger |
| 393 | MER2B | DNA | TcMar-Tigger |
| 394 | MER2 | DNA | TcMar-Tigger |
| 395 | Tigger3 | DNA | TcMar-Tigger |
| 396 | Tigger4a | DNA | TcMar-Tigger |
| 397 | Tigger3c | DNA | TcMar-Tigger |
| 398 | MER44B | DNA | TcMar-Tigger |
| 399 | MER44A | DNA | TcMar-Tigger |
| 400 | LTR26B | LTR | ERV1 |
| 401 | Tigger3b | DNA | TcMar-Tigger |
| 402 | MSTD-int | LTR | ERVL-MaLR |
| 403 | MER41G | LTR | ERV1 |
| 404 | LTR48 | LTR | ERV1 |
| 405 | MSTD | LTR | ERVL-MaLR |
| 406 | MER39 | LTR | ERV1 |
| 407 | MER21B | LTR | ERVL |
| 408 | MLT1A1 | LTR | ERVL-MaLR |
| 409 | MER44C | DNA | TcMar-Tigger |
| 410 | MER47B | DNA | TcMar-Tigger |
| 411 | LTR54B | LTR | ERV1 |
| 412 | Tigger7 | DNA | TcMar-Tigger |
| 413 | MER47C | DNA | TcMar-Tigger |

|  |  |  |  |
|---|---|---|---|
| 414 | Tigger5 | DNA | TcMar-Tigger |
| 415 | L1P | LINE | L1 |
| 416 | MER73 | LTR | ERVL |
| 417 | MER44D | DNA | TcMar-Tigger |
| 418 | MER65B | LTR | ERV1 |
| 419 | MLT1A0 | LTR | ERVL-MaLR |
| 420 | Tigger1a_Mars | DNA | TcMar-Tigger |
| 421 | MER31A | LTR | ERV1 |
| 422 | MER34B | LTR | ERV1 |
| 423 | MER54B | LTR | ERVL |
| 424 | MER101B | LTR | ERV1 |
| 425 | MLT2B1 | LTR | ERVL |
| 426 | MER57E1 | LTR | ERV1 |
| 427 | THE1B-int | LTR | ERVL-MaLR |
| 428 | MER70B | LTR | ERVL |
| 429 | LTR31 | LTR | ERV1 |
| 430 | MER88 | LTR | ERVL |
| 431 | MER34C | LTR | ERV1 |
| 432 | Tigger5b | DNA | TcMar-Tigger |
| 433 | MER34 | LTR | ERV1 |
| 434 | MER34C_ | LTR | ERV1 |
| 435 | MER57F | LTR | ERV1 |
| 436 | MLT2B2 | LTR | ERVL |
| 437 | MLT1A | LTR | ERVL-MaLR |
| 438 | MER6C | DNA | TcMar-Tigger |
| 439 | MER21C | LTR | ERVL |
| 440 | MLT2B5 | LTR | ERVL |
| 441 | L1M1 | LINE | L1 |
| 442 | L1MA6 | LINE | L1 |
| 443 | Ricksha_c | DNA | MULE-MuDR |
| 444 | LTR69 | LTR | ERVL |
| 445 | L1MA7 | LINE | L1 |
| 446 | MLT2B4 | LTR | ERVL |
| 447 | L1M3c | LINE | L1 |
| 448 | LTR58 | LTR | ERV1 |
| 449 | L1M3e | LINE | L1 |
| 450 | MER34A | LTR | ERV1 |
| 451 | L1P3b | LINE | L1 |
| 452 | MER57D | LTR | ERV1 |
| 453 | Tigger4 | DNA | TcMar-Tigger |
| 454 | L1MA8 | LINE | L1 |
| 455 | L1MA9 | LINE | L1 |
| 456 | MER34A1 | LTR | ERV1 |
| 457 | MER45A | DNA | hAT-Tip100 |
| 458 | MLT2D | LTR | ERVL |
| 459 | MER67D | LTR | ERV1 |
| 460 | MLT1B | LTR | ERVL-MaLR |
| 461 | MER67B | LTR | ERV1 |
| 462 | LTR47B | LTR | ERVL |
| 463 | Tigger17c | DNA | TcMar-Tigger |
| 464 | LTR108a_Mam | LTR | ERVL |
| 465 | MLT2C2 | LTR | ERVL |
| 466 | MER74C | LTR | ERVL |
| 467 | MER31B | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 468 | LTR53B | LTR | ERVL |
| 469 | Tigger17a | DNA | TcMar-Tigger |
| 470 | MER95 | LTR | ERV1 |
| 471 | MER57E2 | LTR | ERV1 |
| 472 | LTR55 | LTR | ERVL? |
| 473 | L1MB2 | LINE | L1 |
| 474 | MER81 | DNA | hAT-Blackjack |
| 475 | MLT1C | LTR | ERVL-MaLR |
| 476 | MER89 | LTR | ERV1 |
| 477 | MLT1D | LTR | ERVL-MaLR |
| 478 | MER106B | DNA | hAT-Charlie |
| 479 | MER106A | DNA | hAT-Charlie |
| 480 | MER53 | DNA | hAT |
| 481 | MADE2 | DNA | TcMar-Mariner |
| 482 | CR1-16_AMi | LINE | CR1 |
| 483 | UCON29 | DNA | PiggyBac? |
| 484 | LTR75 | LTR | ERVL |
| 485 | L2-1_Crp | LINE | L2 |
| 486 | X2_LINE | LINE | CR1 |
| 487 | X1_LINE | LINE | CR1 |
| 488 | Charlie15a | DNA | hAT-Charlie |
| 489 | Chompy-6_Croc | DNA | PIF-Harbinger |
| 490 | MER67A | LTR | ERV1 |
| 491 | MER96 | DNA | hAT-Tip100 |
| 492 | MER68-int | LTR | ERVL |
| 493 | MER20 | DNA | hAT-Charlie |
| 494 | MLT1N2-int | LTR | ERVL-MaLR |
| 495 | X5B_LINE | LINE | CR1 |
| 496 | UCON14 | DNA? | DNA? |
| 497 | UCON7 | DNA? | DNA? |
| 498 | Eulor10 | DNA? | DNA? |
| 499 | X7D_LINE | LINE | CR1 |
| 500 | MLT1E1A-int | LTR | ERVL-MaLR |
| 501 | MER127 | DNA | TcMar-Tigger |
| 502 | X8_LINE | LINE | CR1 |
| 503 | L1P4d | LINE | L1 |
| 504 | MER68 | LTR | ERVL |
| 505 | MER131 | DNA? | DNA? |
| 506 | Eulor2B | DNA? | DNA? |
| 507 | LTR108c_Mam | LTR | ERVL |
| 508 | MamRep1894 | DNA | hAT |
| 509 | LTR75B | LTR | ERVL |
| 510 | UCON11 | DNA | TcMar-Tigger |
| 511 | UCON62 | DNA? | DNA? |
| 512 | Tigger17b | DNA | TcMar-Tigger |
| 513 | Ricksha_b | DNA | MULE-MuDR |
| 514 | MER5C1 | DNA | hAT-Charlie |
| 515 | MER94B | DNA | hAT-Blackjack |
| 516 | UCON49 | LINE | L2 |
| 517 | HERV-Fc2-int | LTR | ERV1 |
| 518 | Merlin1_HS | DNA | Merlin |
| 519 | MER133A | DNA? | DNA? |
| 520 | Eulor6A | DNA? | DNA? |
| 521 | X6B_LINE | LINE | CR1 |

| | | | |
|---|---|---|---|
| 522 | UCON81 | DNA | hAT-Charlie |
| 523 | LTR108e_Mam | LTR | ERVL |
| 524 | LTR108b_Mam | LTR | ERVL |
| 525 | UCON89 | DNA | hAT? |
| 526 | MER132 | DNA | TcMar-Pogo |
| 527 | MER134 | DNA? | DNA? |
| 528 | MER57E3 | LTR | ERV1 |
| 529 | Ricksha | DNA | MULE-MuDR |
| 530 | MER121B | DNA | hAT? |
| 531 | Eulor12 | DNA? | DNA? |
| 532 | UCON69 | DNA | hAT? |
| 533 | MER34C2 | LTR | ERV1 |
| 534 | Eulor2A | DNA? | DNA? |
| 535 | UCON39 | DNA | TcMar-Tigger |
| 536 | Charlie10a | DNA | hAT-Charlie |
| 537 | UCON86 | LINE | L2 |
| 538 | UCON21 | DNA? | DNA? |
| 539 | Eulor7 | DNA? | DNA? |
| 540 | MER68C | LTR | ERVL |
| 541 | MER92D | LTR | ERV1 |
| 542 | Eulor3 | DNA? | DNA? |
| 543 | Ricksha_a | DNA | MULE-MuDR |
| 544 | HERV1_LTRe | LTR | ERV1 |
| 545 | Eulor5A | DNA? | DNA? |
| 546 | UCON51 | LTR? | LTR? |
| 547 | Eulor2C | DNA? | DNA? |
| 548 | MLT1E-int | LTR | ERVL-MaLR |
| 549 | MLT1E | LTR | ERVL-MaLR |
| 550 | LTR109A2 | LTR | ERV1 |
| 551 | Eulor11 | DNA | DNA |
| 552 | MER68B | LTR | ERVL |
| 553 | MER125 | DNA | DNA |
| 554 | MLT1E1 | LTR | ERVL-MaLR |
| 555 | UCON83 | SINE? | SINE? |
| 556 | MER58A | DNA | hAT-Charlie |
| 557 | UCON80 | DNA | hAT? |
| 558 | MER58D | DNA | hAT-Charlie |
| 559 | LTR60 | LTR | ERV1 |
| 560 | Eulor6D | DNA? | DNA? |
| 561 | MER133B | DNA? | DNA? |
| 562 | MER74B | LTR | ERVL |
| 563 | MamRep1161 | DNA | TcMar-Tigger |
| 564 | L1M3 | LINE | L1 |
| 565 | CR1-11_Crp | LINE | CR1 |
| 566 | Eulor9B | DNA | DNA |
| 567 | LTR81AB | LTR | Gypsy |
| 568 | MER126 | DNA | DNA |
| 569 | L1MC | LINE | L1 |
| 570 | MER70C | LTR | ERVL |
| 571 | LTR70 | LTR | ERV1 |
| 572 | HERV-Fc2_LTR | LTR | ERV1 |
| 573 | L2-3_AMi | LINE | L2 |
| 574 | L1M2a | LINE | L1 |
| 575 | UCON97 | DNA | DNA |

| | | | |
|---|---|---|---|
| 576 | HERV-Fc1_LTR2 | LTR | ERV1 |
| 577 | L1MB3 | LINE | L1 |
| 578 | PRIMAX-int | LTR | ERV1 |
| 579 | MLT1E1-int | LTR | ERVL-MaLR |
| 580 | CRP1 | SINE? | tRNA |
| 581 | UCON79 | DNA? | DNA? |
| 582 | MER76 | LTR | ERVL |
| 583 | AluYh9 | SINE | Alu |
| 584 | MER130 | DNA | DNA |
| 585 | MER121 | DNA | hAT? |
| 586 | Eulor9C | DNA | DNA |
| 587 | L1M2 | LINE | L1 |
| 588 | UCON23 | DNA | hAT-Tip100? |
| 589 | Eulor1 | DNA | DNA |
| 590 | MER92C | LTR | ERV1 |
| 591 | Eulor8 | DNA | TcMar? |
| 592 | UCON8 | DNA | DNA |
| 593 | MER92B | LTR | ERV1 |
| 594 | L1MC1 | LINE | L1 |
| 595 | MER113A | DNA | hAT-Charlie |
| 596 | UCON100 | DNA? | DNA? |
| 597 | MER67C | LTR | ERV1 |
| 598 | L1M3d | LINE | L1 |
| 599 | MER77B | LTR | ERVL |
| 600 | L1MA10 | LINE | L1 |
| 601 | UCON78 | DNA | DNA |
| 602 | MER74A | LTR | ERVL |
| 603 | Eulor6B | DNA? | DNA? |
| 604 | MER77 | LTR | ERVL |
| 605 | L1MB1 | LINE | L1 |
| 606 | L1MB4 | LINE | L1 |
| 607 | L1MC2 | LINE | L1 |
| 608 | LTR53 | LTR | ERVL |
| 609 | LTR40b | LTR | ERVL |
| 610 | MLT1E3 | LTR | ERVL-MaLR |
| 611 | L1MB5 | LINE | L1 |
| 612 | LTR52-int | LTR | ERVL |
| 613 | MLT2C1 | LTR | ERVL |
| 614 | HERVFH21-int | LTR | ERV1 |
| 615 | LTR23-int | LTR | ERV1 |
| 616 | MLT1E2 | LTR | ERVL-MaLR |
| 617 | UCON2 | DNA? | DNA? |
| 618 | HERV-Fc1_LTR1 | LTR | ERV1 |
| 619 | MLT1E1A | LTR | ERVL-MaLR |
| 620 | LTR41C | LTR | ERVL |
| 621 | LTR21C | LTR | ERVL |
| 622 | Charlie12 | DNA | hAT-Charlie |
| 623 | LTR40a | LTR | ERVL |
| 624 | L1M4a2 | LINE | L1 |
| 625 | MLT1F-int | LTR | ERVL-MaLR |
| 626 | MLT1A0-int | LTR | ERVL-MaLR |
| 627 | LTR52 | LTR | ERVL |
| 628 | LTR41B | LTR | ERVL |
| 629 | MLT1F2 | LTR | ERVL-MaLR |

| | | | |
|---|---|---|---|
| 630 | Chompy-7_Croc | DNA | PIF-Harbinger |
| 631 | MER45R | DNA | hAT-Tip100 |
| 632 | Eulor5B | DNA? | DNA? |
| 633 | Eulor6E | DNA? | DNA? |
| 634 | MER63A | DNA | hAT-Blackjack |
| 635 | MER136 | DNA | DNA |
| 636 | L1MB8 | LINE | L1 |
| 637 | MER58C | DNA | hAT-Charlie |
| 638 | L1MB7 | LINE | L1 |
| 639 | LTR16B2 | LTR | ERVL |
| 640 | MER96B | DNA | hAT-Tip100 |
| 641 | ORSL | DNA | hAT-Tip100 |
| 642 | CR1-13_AMi | LINE | CR1 |
| 643 | Tigger9b | DNA | TcMar-Tigger |
| 644 | MER3 | DNA | hAT-Charlie |
| 645 | MLT1G-int | LTR | ERVL-MaLR |
| 646 | MER58B | DNA | hAT-Charlie |
| 647 | L1MD | LINE | L1 |
| 648 | MER90a | LTR | ERV1 |
| 649 | MLT1G | LTR | ERVL-MaLR |
| 650 | MLT1F1-int | LTR | ERVL-MaLR |
| 651 | MER45B | DNA | hAT-Tip100 |
| 652 | LTR37A | LTR | ERV1 |
| 653 | MLT1F1 | LTR | ERVL-MaLR |
| 654 | MER92A | LTR | ERV1 |
| 655 | LTR80B | LTR | ERVL |
| 656 | L1MC3 | LINE | L1 |
| 657 | LTR68 | LTR | ERV1 |
| 658 | L1P4b | LINE | L1 |
| 659 | Tigger6a | DNA | TcMar-Tigger |
| 660 | LTR41 | LTR | ERVL |
| 661 | LTR65 | LTR | ERV1 |
| 662 | MLT-int | LTR | ERVL-MaLR |
| 663 | MLT1F | LTR | ERVL-MaLR |
| 664 | UCON55 | DNA | TcMar-Tigger |
| 665 | MER90 | LTR | ERV1 |
| 666 | MLT1-int | LTR | ERVL-MaLR |
| 667 | L1MD2 | LINE | L1 |
| 668 | MER63B | DNA | hAT-Blackjack |
| 669 | MER110A | LTR | ERV1 |
| 670 | MER63D | DNA | hAT-Blackjack |
| 671 | LTR16B | LTR | ERVL |
| 672 | LTR40c | LTR | ERVL |
| 673 | MLT1G1-int | LTR | ERVL-MaLR |
| 674 | MER135 | DNA | DNA |
| 675 | Charlie10b | DNA | hAT-Charlie |
| 676 | MLT1G1 | LTR | ERVL-MaLR |
| 677 | MER97d | DNA | hAT-Tip100 |
| 678 | MER119 | DNA | hAT-Charlie |
| 679 | L1MD3 | LINE | L1 |
| 680 | MER89-int | LTR | ERV1 |
| 681 | Charlie1 | DNA | hAT-Charlie |
| 682 | Charlie10 | DNA | hAT-Charlie |
| 683 | L1M2a1 | LINE | L1 |

| | | | |
|---|---|---|---|
| 684 | Charlie1a | DNA | hAT-Charlie |
| 685 | MER33 | DNA | hAT-Charlie |
| 686 | MLT1E3-int | LTR | ERVL-MaLR |
| 687 | MER63C | DNA | hAT-Blackjack |
| 688 | MER5A1 | DNA | hAT-Charlie |
| 689 | MLT1H1-int | LTR | ERVL-MaLR |
| 690 | LTR83 | LTR | ERVL |
| 691 | MER5A | DNA | hAT-Charlie |
| 692 | LTR37B | LTR | ERV1 |
| 693 | MER105 | DNA | hAT-Charlie |
| 694 | LTR86C | LTR | ERVL |
| 695 | LTR33C | LTR | ERVL |
| 696 | L1M4c | LINE | L1 |
| 697 | MLT2F | LTR | ERVL |
| 698 | LTR16B1 | LTR | ERVL |
| 699 | LTR80A | LTR | ERVL |
| 700 | ERV3-16A3_LTR | LTR | ERVL |
| 701 | MER94 | DNA | hAT-Blackjack |
| 702 | Charlie1b | DNA | hAT-Charlie |
| 703 | LTR16A | LTR | ERVL |
| 704 | LTR107_Mam | LTR | LTR |
| 705 | LTR33A | LTR | ERVL |
| 706 | L1PA15-16 | LINE | L1 |
| 707 | Tigger6b | DNA | TcMar-Tigger |
| 708 | L1M4 | LINE | L1 |
| 709 | L1MC4a | LINE | L1 |
| 710 | MLT1H1 | LTR | ERVL-MaLR |
| 711 | Tigger17 | DNA | TcMar-Tigger |
| 712 | Arthur1C | DNA | hAT-Tip100 |
| 713 | MLT1G3-int | LTR | ERVL-MaLR |
| 714 | LTR16 | LTR | ERVL |
| 715 | LTR90B | LTR | LTR |
| 716 | LTR16D | LTR | ERVL |
| 717 | MER99 | DNA | hAT? |
| 718 | MLT1H | LTR | ERVL-MaLR |
| 719 | L1MD1 | LINE | L1 |
| 720 | MLT1G3 | LTR | ERVL-MaLR |
| 721 | MLT1J1-int | LTR | ERVL-MaLR |
| 722 | L1M4a1 | LINE | L1 |
| 723 | MER104 | DNA | TcMar-Tc2 |
| 724 | Charlie17a | DNA | hAT-Charlie |
| 725 | MLT1K-int | LTR | ERVL-MaLR |
| 726 | MLT1D-int | LTR | ERVL-MaLR |
| 727 | Zaphod3 | DNA | hAT-Tip100 |
| 728 | MER97a | DNA | hAT-Tip100 |
| 729 | L1ME1 | LINE | L1 |
| 730 | L1ME2z | LINE | L1 |
| 731 | LTR91 | LTR | ERVL |
| 732 | MLT1L-int | LTR | ERVL-MaLR |
| 733 | MERX | DNA | TcMar-Tigger |
| 734 | MER5B | DNA | hAT-Charlie |
| 735 | LTR16A1 | LTR | ERVL |
| 736 | DNA1_Mam | DNA | TcMar |
| 737 | MER5C | DNA | hAT-Charlie |

| | | | |
|---|---|---|---|
| 738 | MLT1H2-int | LTR | ERVL-MaLR |
| 739 | LTR16E2 | LTR | ERVL |
| 740 | MamRep38 | DNA | hAT |
| 741 | Kanga1 | DNA | TcMar-Tc2 |
| 742 | MER115 | DNA | hAT-Tip100 |
| 743 | MLT1H2 | LTR | ERVL-MaLR |
| 744 | LTR16C | LTR | ERVL |
| 745 | Arthur1A | DNA | hAT-Tip100 |
| 746 | MamRep488 | DNA | hAT-Tip100 |
| 747 | X9_LINE | LINE | L1 |
| 748 | LTR33 | LTR | ERVL |
| 749 | LTR16E1 | LTR | ERVL |
| 750 | LTR40A1 | LTR | ERVL |
| 751 | LTR33B | LTR | ERVL |
| 752 | L1M2b | LINE | L1 |
| 753 | MER91A | DNA | hAT-Tip100 |
| 754 | LTR67B | LTR | ERVL |
| 755 | LTR104_Mam | LTR | Gypsy |
| 756 | MER102c | DNA | hAT-Charlie |
| 757 | LTR87 | LTR | ERVL? |
| 758 | LTR16A2 | LTR | ERVL |
| 759 | Tigger19b | DNA | TcMar-Tigger |
| 760 | Arthur1B | DNA | hAT-Tip100 |
| 761 | L1MEb | LINE | L1 |
| 762 | Charlie4 | DNA | hAT-Charlie |
| 763 | Kanga1c | DNA | TcMar-Tc2 |
| 764 | L2-3_Crp | LINE | L2 |
| 765 | LTR82A | LTR | ERVL |
| 766 | Charlie4a | DNA | hAT-Charlie |
| 767 | Cheshire | DNA | hAT-Charlie |
| 768 | L1MC5 | LINE | L1 |
| 769 | MLT1J-int | LTR | ERVL-MaLR |
| 770 | MLT1J | LTR | ERVL-MaLR |
| 771 | LTR85c | LTR | Gypsy? |
| 772 | HAL1M8 | LINE | L1 |
| 773 | MLT2E | LTR | ERVL |
| 774 | LTR43-int | LTR | ERV1 |
| 775 | L1M3a | LINE | L1 |
| 776 | Charlie7a | DNA | hAT-Charlie |
| 777 | MLT1I-int | LTR | ERVL-MaLR |
| 778 | MER117 | DNA | hAT-Charlie |
| 779 | MLT1J2 | LTR | ERVL-MaLR |
| 780 | MLT1J1 | LTR | ERVL-MaLR |
| 781 | MLT1I | LTR | ERVL-MaLR |
| 782 | Looper | DNA | PiggyBac |
| 783 | Kanga1b | DNA | TcMar-Tc2 |
| 784 | MER112 | DNA | hAT-Charlie |
| 785 | MER97b | DNA | hAT-Tip100 |
| 786 | MamGypLTR1a | LTR | Gypsy |
| 787 | MamGypLTR1d | LTR | Gypsy |
| 788 | L1MC5a | LINE | L1 |
| 789 | Kanga1a | DNA | TcMar-Tc2 |
| 790 | MamRep1879 | DNA | hAT-Tip100? |
| 791 | LTR50 | LTR | ERVL |

| | | | |
|---|---|---|---|
| 792 | OldhAT1 | DNA | hAT-Ac |
| 793 | MER45C | DNA | hAT-Tip100 |
| 794 | LTR33A_ | LTR | ERVL |
| 795 | MER46C | DNA | TcMar-Tigger |
| 796 | Tigger19a | DNA | TcMar-Tigger |
| 797 | L2-1_AMi | LINE | L2 |
| 798 | LTR102_Mam | LTR | ERVL |
| 799 | LFSINE_Vert | SINE | tRNA |
| 800 | MIR | SINE | MIR |
| 801 | LTR46-int | LTR | ERV1 |
| 802 | MER91B | DNA | hAT-Tip100 |
| 803 | MER102b | DNA | hAT-Charlie |
| 804 | MER102a | DNA | hAT-Charlie |
| 805 | LTR106_Mam | LTR | LTR |
| 806 | MamRep137 | DNA | TcMar-Tigger |
| 807 | MamRep1527 | LTR | LTR |
| 808 | ERVL-int | LTR | ERVL |
| 809 | LTR16D2 | LTR | ERVL |
| 810 | MamGypLTR2c | LTR | Gypsy |
| 811 | HERVP71A-int | LTR | ERV1 |
| 812 | L1M5 | LINE | L1 |
| 813 | L1MCb | LINE | L1 |
| 814 | LTR108d_Mam | LTR | ERVL |
| 815 | L1MCa | LINE | L1 |
| 816 | Charlie14a | DNA | hAT-Charlie |
| 817 | hAT-N1_Mam | DNA | hAT-Tip100? |
| 818 | L1M4b | LINE | L1 |
| 819 | MamRep434 | DNA | TcMar-Tigger |
| 820 | L1MEc | LINE | L1 |
| 821 | CR1_Mam | LINE | CR1 |
| 822 | L1ME3 | LINE | L1 |
| 823 | LTR86A2 | LTR | ERVL |
| 824 | MLT1J2-int | LTR | ERVL-MaLR |
| 825 | UCON74 | DNA? | DNA? |
| 826 | L1MDa | LINE | L1 |
| 827 | L1ME2 | LINE | L1 |
| 828 | LTR86B2 | LTR | ERVL |
| 829 | Zaphod | DNA | hAT-Tip100 |
| 830 | L1MEg1 | LINE | L1 |
| 831 | MER110 | LTR | ERV1 |
| 832 | L1ME3A | LINE | L1 |
| 833 | L1ME3G | LINE | L1 |
| 834 | MLT1L | LTR | ERVL-MaLR |
| 835 | Charlie17 | DNA | hAT-Charlie |
| 836 | Charlie16a | DNA | hAT-Charlie |
| 837 | L1ME5 | LINE | L1 |
| 838 | Charlie6 | DNA | hAT-Charlie |
| 839 | Kanga1d | DNA | TcMar-Tc2 |
| 840 | LTR101_Mam | LTR | ERVL |
| 841 | Charlie24 | DNA | hAT-Charlie |
| 842 | MamGypLTR1b | LTR | Gypsy |
| 843 | Tigger9a | DNA | TcMar-Tigger |
| 844 | MLT1N2 | LTR | ERVL-MaLR |
| 845 | Charlie4z | DNA | hAT-Charlie |

| | | | |
|---|---|---|---|
| 846 | Chap1_Mam | DNA | hAT-Charlie |
| 847 | LTR86A1 | LTR | ERVL |
| 848 | MER113 | DNA | hAT-Charlie |
| 849 | MamTip1 | DNA | hAT-Tip100 |
| 850 | MIRb | SINE | MIR |
| 851 | MER91C | DNA | hAT-Tip100 |
| 852 | MLT1E2-int | LTR | ERVL-MaLR |
| 853 | LTR84a | LTR | ERVL |
| 854 | LTR86B1 | LTR | ERVL |
| 855 | L1ME4b | LINE | L1 |
| 856 | LTR84b | LTR | ERVL |
| 857 | Tigger16a | DNA | TcMar-Tigger |
| 858 | ORSL-2a | DNA | hAT-Tip100 |
| 859 | MamRep4096 | DNA | hAT-Tip100 |
| 860 | L1MC4 | LINE | L1 |
| 861 | Plat_L3 | LINE | CR1 |
| 862 | BLACKJACK | DNA | hAT-Blackjack |
| 863 | MIR1_Amn | SINE | MIR |
| 864 | LTR90A | LTR | LTR |
| 865 | Charlie18a | DNA | hAT-Charlie |
| 866 | LTR88c | LTR | Gypsy? |
| 867 | MLT1C-int | LTR | ERVL-MaLR |
| 868 | LTR79 | LTR | ERVL |
| 869 | LTR53-int | LTR | ERVL |
| 870 | L2a | LINE | L2 |
| 871 | Charlie2a | DNA | hAT-Charlie |
| 872 | L1M7 | LINE | L1 |
| 873 | Kanga11a | DNA | TcMar-Tc2 |
| 874 | L1MEh | LINE | L1 |
| 875 | L1MEg | LINE | L1 |
| 876 | Charlie29a | DNA | hAT-Charlie |
| 877 | Kanga2_a | DNA | TcMar-Tc2 |
| 878 | L1ME3F | LINE | L1 |
| 879 | Tigger13a | DNA | TcMar-Tigger |
| 880 | L1ME3Cz | LINE | L1 |
| 881 | L1ME4a | LINE | L1 |
| 882 | LTR16D1 | LTR | ERVL |
| 883 | L1ME3D | LINE | L1 |
| 884 | Tigger8 | DNA | TcMar-Tigger |
| 885 | MER103C | DNA | hAT-Charlie |
| 886 | MamTip3 | DNA | hAT-Tip100 |
| 887 | ORSL-2b | DNA | hAT-Tip100 |
| 888 | LTR105_Mam | LTR | ERVL |
| 889 | HAL1b | LINE | L1 |
| 890 | Charlie19a | DNA | hAT-Charlie |
| 891 | AmnSINE2 | SINE | tRNA-Deu |
| 892 | X7B_LINE | LINE | CR1 |
| 893 | L2c | LINE | L2 |
| 894 | LTR57-int | LTR | ERVL |
| 895 | L1ME3E | LINE | L1 |
| 896 | Charlie11 | DNA | hAT-Charlie |
| 897 | LTR82B | LTR | ERVL |
| 898 | MER101-int | LTR | ERV1 |
| 899 | X7C_LINE | LINE | CR1 |

|  |  |  |  |
|---|---|---|---|
| 900 | L1P4c | LINE | L1 |
| 901 | Tigger15a | DNA | TcMar-Tigger |
| 902 | LTR89 | LTR | ERVL? |
| 903 | MLT1M | LTR | ERVL-MaLR |
| 904 | L4_A_Mam | LINE | RTE-X |
| 905 | L1ME3B | LINE | L1 |
| 906 | L4_B_Mam | LINE | RTE-X |
| 907 | MamGypLTR3 | LTR | Gypsy |
| 908 | Charlie22a | DNA | hAT-Charlie |
| 909 | X6A_LINE | LINE | CR1 |
| 910 | MLT1B-int | LTR | ERVL-MaLR |
| 911 | MamRTE1 | LINE | RTE-BovB |
| 912 | Tigger11a | DNA | TcMar-Tigger |
| 913 | MIRc | SINE | MIR |
| 914 | MLT1O | LTR | ERVL-MaLR |
| 915 | L1M8 | LINE | L1 |
| 916 | Charlie17b | DNA | hAT-Charlie |
| 917 | LTR81A | LTR | Gypsy |
| 918 | Tigger20a | DNA | TcMar-Tigger |
| 919 | MER20B | DNA | hAT-Charlie |
| 920 | HERV17-int | LTR | ERV1 |
| 921 | HERVL74-int | LTR | ERVL |
| 922 | Charlie9 | DNA | hAT-Charlie |
| 923 | Charlie20a | DNA | hAT-Charlie |
| 924 | MamGypsy2-LTR | LTR | Gypsy |
| 925 | LTR81B | LTR | Gypsy |
| 926 | LTR78 | LTR | ERV1 |
| 927 | MER21-int | LTR | ERVL |
| 928 | FordPrefect_a | DNA | hAT-Tip100 |
| 929 | Mam_R4 | LINE | Dong-R4 |
| 930 | MIR3 | SINE | MIR |
| 931 | Charlie25 | DNA | hAT-Charlie |
| 932 | L5 | LINE | RTE-X |
| 933 | ERVL47-int | LTR | ERVL |
| 934 | L1M3b | LINE | L1 |
| 935 | L1ME3C | LINE | L1 |
| 936 | Tigger14a | DNA | TcMar-Tigger |
| 937 | CR1-3_Croc | LINE | CR1 |
| 938 | L1MEi | LINE | L1 |
| 939 | L3b | LINE | CR1 |
| 940 | HERVK14C-int | LTR | ERVK |
| 941 | HERVK-int | LTR | ERVK |
| 942 | MamSINE1 | SINE | tRNA-RTE |
| 943 | Tigger18a | DNA | TcMar-Tigger |
| 944 | Charlie15b | DNA | hAT-Charlie |
| 945 | L1MCc | LINE | L1 |
| 946 | MER129 | LTR? | LTR? |
| 947 | MER34B-int | LTR | ERV1 |
| 948 | LTR85a | LTR | Gypsy? |
| 949 | MamTip2 | DNA | hAT-Tip100 |
| 950 | L2b | LINE | L2 |
| 951 | LTR103_Mam | LTR | ERV1? |
| 952 | FordPrefect | DNA | hAT-Tip100 |
| 953 | Charlie26a | DNA | hAT-Charlie |

| | | | |
|---|---|---|---|
| 954 | MER113B | DNA | hAT-Charlie |
| 955 | L1ME4c | LINE | L1 |
| 956 | Tigger12 | DNA | TcMar-Tigger |
| 957 | L2 | LINE | L2 |
| 958 | Tigger12A | DNA | TcMar-Tigger |
| 959 | CR1-L3A_Croc | LINE | CR1 |
| 960 | HAL1ME | LINE | L1 |
| 961 | CR1-12_AMi | LINE | CR1 |
| 962 | LTR78B | LTR | ERV1 |
| 963 | L4_C_Mam | LINE | RTE-X |
| 964 | Zaphod2 | DNA | hAT-Tip100 |
| 965 | UCON132b | DNA | hAT-Tip100 |
| 966 | L3 | LINE | CR1 |
| 967 | Eulor9A | DNA | DNA |
| 968 | UCON99 | DNA | DNA |
| 969 | Charlie30a | DNA | hAT-Charlie |
| 970 | MamGypLTR1c | LTR | Gypsy |
| 971 | MamGyp-int | LTR | Gypsy |
| 972 | ERV24_Prim-int | LTR | ERV1 |
| 973 | MER76-int | LTR | ERVL |
| 974 | Charlie7 | DNA | hAT-Charlie |
| 975 | Tigger1a_Art | DNA | TcMar-Tigger |
| 976 | PABL_B-int | LTR | ERV1 |
| 977 | L1MDb | LINE | L1 |
| 978 | MamGypLTR3a | LTR | Gypsy |
| 979 | Eulor6C | DNA? | DNA? |
| 980 | LTR88b | LTR | Gypsy? |
| 981 | MER83B-int | LTR | ERV1 |
| 982 | MER41-int | LTR | ERV1 |
| 983 | L1MEd | LINE | L1 |
| 984 | HERVL40-int | LTR | ERVL |
| 985 | PRIMA41-int | LTR | ERV1 |
| 986 | LTR85b | LTR | Gypsy? |
| 987 | LTR39-int | LTR | ERV1 |
| 988 | HERVIP10F-int | LTR | ERV1 |
| 989 | MER110-int | LTR | ERV1 |
| 990 | HERVL-int | LTR | ERVL |
| 991 | L1MEg2 | LINE | L1 |
| 992 | LTR103b_Mam | LTR | ERV1? |
| 993 | Charlie5 | DNA | hAT-Charlie |
| 994 | Charlie13b | DNA | hAT-Charlie |
| 995 | X7A_LINE | LINE | CR1 |
| 996 | MER4B-int | LTR | ERV1 |
| 997 | ERVL-E-int | LTR | ERVL |
| 998 | CR1-8_Crp | LINE | CR1 |
| 999 | Charlie21a | DNA | hAT-Charlie |
| 1000 | L1M6B | LINE | L1 |
| 1001 | MER70-int | LTR | ERVL |
| 1002 | HERV35I-int | LTR | ERV1 |
| 1003 | ERVL-B4-int | LTR | ERVL |
| 1004 | MER61-int | LTR | ERV1 |
| 1005 | Tigger12c | DNA | TcMar-Tigger |
| 1006 | Charlie23a | DNA | hAT-Charlie |
| 1007 | HERVS71-int | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 1008 | ERV3-16A3_I-int | LTR | ERVL |
| 1009 | MER124 | DNA? | DNA? |
| 1010 | ERV24B_Prim-int | LTR | ERV1 |
| 1011 | Harlequin-int | LTR | ERV1 |
| 1012 | MER31-int | LTR | ERV1 |
| 1013 | MER57A-int | LTR | ERV1 |
| 1014 | MER34-int | LTR | ERV1 |
| 1015 | Tigger10 | DNA | TcMar-Tigger |
| 1016 | HERVK14-int | LTR | ERVK |
| 1017 | MLT1H-int | LTR | ERVL-MaLR |
| 1018 | PRIMA4-int | LTR | ERV1 |
| 1019 | HERVK11-int | LTR | ERVK |
| 1020 | HERVE-int | LTR | ERV1 |
| 1021 | LTR49-int | LTR | ERV1 |
| 1022 | MER51-int | LTR | ERV1 |
| 1023 | MamGypsy2-I | LTR | Gypsy |
| 1024 | MamGypLTR2b | LTR | Gypsy |
| 1025 | MARNA | DNA | TcMar-Mariner |
| 1026 | MLT1A-int | LTR | ERVL-MaLR |
| 1027 | HAL1 | LINE | L1 |
| 1028 | MER52-int | LTR | ERV1 |
| 1029 | MER84-int | LTR | ERV1 |
| 1030 | LTR88a | LTR | Gypsy? |
| 1031 | HERVK9-int | LTR | ERVK |
| 1032 | HUERS-P2-int | LTR | ERV1 |
| 1033 | HUERS-P1-int | LTR | ERV1 |
| 1034 | PABL_A-int | LTR | ERV1 |
| 1035 | HERVK3-int | LTR | ERVK |
| 1036 | MER92-int | LTR | ERV1 |
| 1037 | HERV3-int | LTR | ERV1 |
| 1038 | MLT1O-int | LTR | ERVL-MaLR |
| 1039 | MER57-int | LTR | ERV1 |
| 1040 | LTR81C | LTR | Gypsy |
| 1041 | MamRep1151 | LTR? | LTR? |
| 1042 | L1MEf | LINE | L1 |
| 1043 | MLT1K | LTR | ERVL-MaLR |
| 1044 | HERVK13-int | LTR | ERVK |
| 1045 | HERV16-int | LTR | ERVL |
| 1046 | L1M6 | LINE | L1 |
| 1047 | MSTA-int | LTR | ERVL-MaLR |
| 1048 | MamRep605 | LTR? | LTR? |
| 1049 | MER4-int | LTR | ERV1 |
| 1050 | Charlie13a | DNA | hAT-Charlie |
| 1051 | LOR1-int | LTR | ERV1 |
| 1052 | HERV4_I-int | LTR | ERV1 |
| 1053 | MER97c | DNA | hAT-Tip100 |
| 1054 | LTR81 | LTR | Gypsy |
| 1055 | Charlie8 | DNA | hAT-Charlie |
| 1056 | UCON33 | DNA | TcMar-Tigger |
| 1057 | LTR37-int | LTR | ERV1 |
| 1058 | Ricksha_0 | DNA | MULE-MuDR |
| 1059 | HUERS-P3-int | LTR | ERV1 |
| 1060 | MLT1F2-int | LTR | ERVL-MaLR |
| 1061 | LTR38-int | LTR | ERV1 |

| | | | |
|---|---|---|---|
| 1062 | Arthur1 | DNA | hAT-Tip100 |
| 1063 | L1MEa | LINE | L1 |
| 1064 | L1P4e | LINE | L1 |
| 1065 | MER66-int | LTR | ERV1 |
| 1066 | MER83A-int | LTR | ERV1 |
| 1067 | Charlie2b | DNA | hAT-Charlie |
| 1068 | Tigger16b | DNA | TcMar-Tigger |
| 1069 | LTR19-int | LTR | ERV1 |
| 1070 | HERVL32-int | LTR | ERVL |
| 1071 | LTR25-int | LTR | ERV1 |
| 1072 | MER65-int | LTR | ERV1 |
| 1073 | MER50-int | LTR | ERV1 |
| 1074 | HERVL66-int | LTR | ERVL |
| 1075 | HUERS-P3b-int | LTR | ERV1 |
| 1076 | L1MEj | LINE | L1 |
| 1077 | HERVH-int | LTR | ERV1 |
| 1078 | HERV30-int | LTR | ERV1 |
| 1079 | L1M3de | LINE | L1 |