8-1-2009

Proximity based association rules for spatial data mining in genomes

Surya Saha

PROXIMITY BASED ASSOCIATION RULES FOR SPATIAL DATA MINING IN GENOMES

By

Surya Saha

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

August 2009

PROXIMITY BASED ASSOCIATION RULES FOR SPATIAL DATA MINING IN GENOMES

By

Surya Saha

Approved:

Susan Bridges
Professor of Computer Science and Engineering
(Co-Director of Dissertation)

Daniel G. Peterson
Associate Professor of Plant and Soil Science
(Co-Director of Dissertation)

Julia Hodges
Professor of Computer Science and Engineering
(Committee Member)

Eric Hansen
Associate Professor of Computer Science and Engineering
(Committee Member)

Yoginder Dandass
Assistant Professor of Computer Science and Engineering
(Committee Member)

Andy Perkins
Assistant Professor of Computer Science and Engineering
(Committee Member)

Edward B. Allen
Associate Professor of Computer Science and Engineering and Graduate Coordinator

Sarah A. Rajala
Dean of the Bagley College of Engineering

Name: Surya Saha

Date of Degree: Aug 8 , 2009

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Susan Bridges

Title of Study: PROXIMITY BASED ASSOCIATION RULES FOR SPATIAL DATA MINING IN GENOMES.

Pages in Study: 139

Candidate for Degree of Doctor of Philosophy

Our knowledge discovery algorithm employs a combination of association rule mining and graph mining to identify frequent spatial proximity relationships in genomic data where the data is viewed as a one-dimensional space. We apply mining techniques and metrics from association rule mining to identify frequently co-occurring features in genomes followed by graph mining to extract sets of co-occurring features.

Using a case study of *ab initio* repeat finding, we have shown that our algorithm, ProxMiner, can be successfully applied to identify weakly conserved patterns among features in genomic data. The application of pairwise spatial relationships increases the sensitivity of our algorithm while the use of a confidence threshold based on false discovery rate reduces the noise in our results. Unlike available defragmentation algorithms, ProxMiner discovers associations among *ab initio* repeat families to identify larger more complete repeat families. ProxMiner will increase the effectiveness of repeat discovery techniques for newly sequenced genomes where *ab initio* repeat finders are only able to identify partial repeat families.

In this dissertation, we provide two detailed examples of ProxMiner-discovered novel repeat families and one example of a known rice repeat family that has been extended by ProxMiner. These examples encompass some of the different types of repeat families that can be discovered by our algorithm. We have also discovered many other potentially interesting novel repeat families that can be further studied by biologists.

Keywords: association rule mining, spatial rules, repeat, defragmentation, graph mining, novel repeat regions, DNA

DEDICATION

I would like to dedicate this dissertation to my parents, Subhash and Dolly Saha, and my fiancée, Sujata Singh.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to everyone who helped bring this dissertation to completion. First and foremost, I would like to acknowledge my major professor and dissertation director, Dr. Susan Bridges, for her constant guidance, invaluable suggestions and endless patience throughout my academic term here at Mississippi State University (MSU). This work could not have been completed without her support. I would also like to thank Dr. Daniel Peterson, who co-directed my research, for his continuous encouragement and for countless delightful discussions about research and life in general. I also express my appreciation to the other members of my committee for their guidance and advice – specifically, Dr. Julia Hodges, Dr. Eric Hansen, Dr. Yoginder Dandass, and Dr. Andy Perkins. I especially appreciate the help of Dr. Zenaida Magbanua for providing valuable expertise and suggestions during the analysis of results. I would also like to thank other researchers in the Computational Biology group at CSE, especially Brandon Malone for setting up a local GBrowse for the analysis of our results. Dr. Susan McCouch deserves a special mention for graciously providing the DNA samples used for molecular biology experiments in this study.

I would like to acknowledge the Mississippi Genome Exploration Laboratory and the Department of Computer Science and Engineering at MSU for providing the infrastructure and resources needed to perform this work. I am also thankful to Dr. Julia

Hodges for providing me with financial support as teaching assistant in the Department of Computer Science and Engineering. I would like to also acknowledge the friendly staff at the Department of Computer Science and Engineering (Jo Coleson, Courtney Blaylock, Keri Chisolm, Russ Ward, Brenda Collins, and Brandi Velcek), who were always approachable and helpful with anything and everything that I needed during my time at MSU.

On a personal note, I am eternally grateful to my parents, Subhash and Dolly Saha, for their endless love and support throughout my life, and my uncle, Dr. Sukumar Saha, who was instrumental in my decision to pursue this PhD and has been a continuous source of support during my time here.

Finally, I gratefully acknowledge the following agencies and organizations that have partially funded my research at MSU during different times over the last few years:

- Hearin Fellowship from the Bagley College of Engineering at MSU
- National Science Foundation (DBI-0421717, EPSCoR grant EPS-0556308)
- LSBI grant number 325928
- MSU Office of Research grant 250731
- USDA Forest Service (SRS 03-CA-11330126-263)

TABLE OF CONTENTS

CHAPTER

APPENDIX

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Technological breakthroughs in DNA sequencing during the last decade [2] have caused an exponential increase in the rate at which genome sequence is being generated [3, 4]. These “second generation” sequencing technologies are already taxing the limits of current data mining methods and systems, and with “third generation” sequencing technologies on the horizon, this problem will only be exacerbated [2, 3, 5]. Genomic sequence data by itself, however, is of limited value [6]. Substantive value is derived from sequence by identifying functional elements and developing an understanding of how these elements contribute to the evolution, survival, and propagation of an organism [6, 7].

The focus of this dissertation is the development of algorithms to discover consistently recurring spatial patterns among features in genomic sequence. Genome sequence may be viewed computationally as a string of letters from a 4-letter alphabet of nucleotides (A,T,C,G) and therefore as a one-dimensional search space. Although there has been substantial research in the development of algorithms for identifying different constituents of the *genome* such as genes [8, 9], small non-coding RNAs [10], promoters [11-13], repeats [14, 15], etc., there has been little previous research investigating spatial relationships among these features. Our approach for finding such relationships among groups of features in the genome uses a combination of association rule mining [16] and graph-based mining [17] as shown in Figure 1.1. We apply association rule mining to discover frequently occurring proximity relationships among features of the genome. Association rule metrics [16] are used to measure the strength of discovered spatial relationships. All qualifying association rules, each denoting a spatial relationship between two features, are mapped into a directed graph and related features are identified by extracting all the connected components. Each set of related features represents a potentially interesting recurring spatial pattern of features.

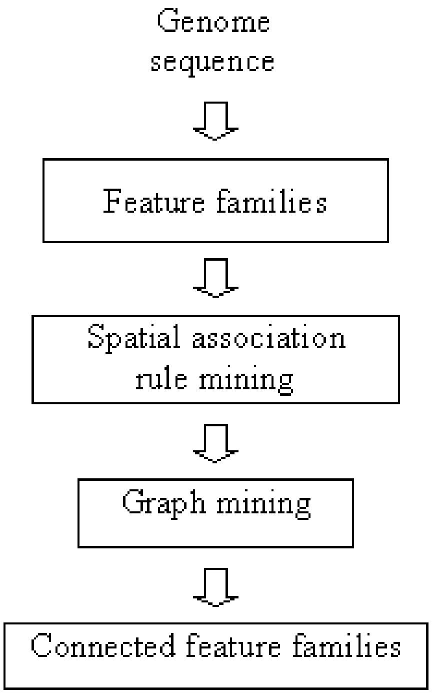


Figure 1.1

Hybrid spatial association rule mining approach

Possible applications for our data mining method include discovery of proximity relationships among gene families, gene and regulatory elements, microRNAs and genes, etc. We have elected to demonstrate the utility of our new data mining approach in the domain of *ab initio* repeat finding in genomes. Repetitive DNA sequences constitute a major portion of most genomes, but repeats are one of the less explored features of genomes as compared to genes [18, 19]. Recent research indicates that repeats serve a number of evolutionary roles in the genome [20, 21]. Their computational discovery and analysis has become increasingly important in genomics today [14, 15, 22]. We apply our spatial data mining approach to the problem of finding novel fragmented repeat families. Our approach is particularly useful for newly sequenced genomes where little is known about the repeat families found in the organism.

1.1 Background

The computational problem we are addressing is the discovery of proximity relationships in a one-dimensional space—the genome. Extensive related research has been conducted in other domains that mine one-dimensional data including intrusion detection in data streams to discover attack patterns [23-26] and in text mining to discover relationships among concepts [27-31].

In the remainder of this section, we address the rationale for applying our hybrid data mining technique for identification of repeat families in DNA. We follow with a brief overview of current computational repeat finding algorithms. Although the predominant focus in genomics has been on identification and characterization of proteins and genes, recent results from the ENCODE project [32-34] show that a substantially larger part of the genome is active than previously thought and new types of features are being discovered all the time [20, 35]. One area that has been largely ignored and is now receiving increased attention is the discovery and characterization of repeat regions in the genome [14, 22]. Once overlooked as “junk DNA” [18], repetitive regions recently have been found to have important regulatory functions in eukaryotic genomes [20, 21, 36, 37]. Additionally, it has long been known that repetitive DNA can cause insertions, deletions, or rearrangements in DNA that can alter gene structure and regulation [21, 36, 37]. It has also been reported that the tremendous increase in repeat activity in populations under extreme environmental stress may be a means of rapidly increasing diversity in DNA through mutation [38-40].

Repetitive DNA in the genomes of living organisms consists of active repeats and highly diverged ancestral repeats. All repeats, as they exist in the genomes today, originated through the replication of a DNA sequence and movement of the copy to another site within the genome [14]. Over time periods ranging from thousands to millions of years, sequences have evolved, diverged and undergone multiple rounds of replication so that extant instances may ultimately bear little similarity to each other [14]. Moreover, repetitive DNA is typically subject to higher rates of mutation compared to genes [38, 41]. One goal of repeat finding is to identify groups of repeats that are derived from a common ancestor (*repeat families*). Each repeat in a repeat family is an *element* of that family. Analysis of sequenced genomes has shown that a majority of repetitive DNA is old and represents diverged elements of inactive repeat families [42, 43]. Identifying these remnant elements of ancient repeat families is challenging because they are highly diverged and fragmented and may have limited similarity to each other [22]. Figure 1.2 illustrates the types of variation that may occur among elements of a repeat family.

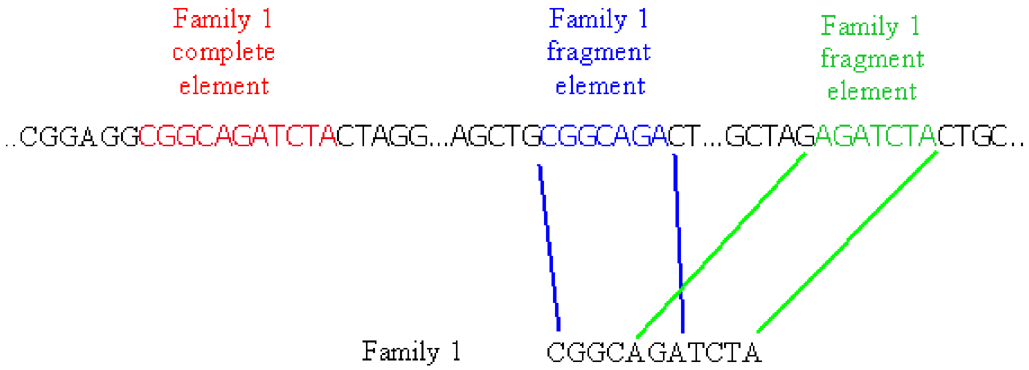


Figure 1.2

Illustration of types of diverged elements of a repeat family

1.2 Current approaches for repeat identification

There are two major classes of repeat identification algorithms: signature-based and *ab initio* [14, 22, 44]. Signature-based systems identify repetitive sequences by comparing input against a database of known repeats or by identifying sequence features indicative of known repeats in input sequences [45, 46]. Although these algorithms are widely used for repeat detection and characterization, only a limited number of repeats can be identified with such techniques since repeat databases represent a minute proportion of all repeat families present in living organisms. Therefore *ab initio* prediction-based systems are becoming increasingly important [14, 15, 22, 44, 47].

Ab initio algorithms identify repeat families without using any prior knowledge of existing repeats and thus can be used to find unknown and uncharacterized repeat families [44, 47]. Most *ab initio* algorithms use a two-step process for repeat identification [14, 44, 48]. In the first step, sequences that potentially contain the same repetitive sequence are identified and clustered using approaches such as *k*-mer mining, spaced seeds and periodicity-based detection. In the second step, a description of the repeat family and/or a *consensus sequence* is extracted from each cluster using various techniques such as secondary clustering, graph mining and string matching/extension. A consensus sequence for a repeat region or repeat family is a common representation of all elements belonging to that repeat family. We have recently completed an extensive review and empirical evaluation of *ab initio* repeat finders [14, 15]. We found large differences in the performance of the algorithms and utility of the information produced by these tools. We also report that repeat discovery tools available today are not effective for identification of ancient repeat regions.

1.3 Motivation

Our knowledge discovery algorithm employs a combination of association rule mining and graph mining to identify proximity relationships in genomic data. As a proof of concept, we utilize our algorithm to identify diverged repeat families in DNA sequence data.

Our empirical evaluation of repeat finders [15] and research by others [22] indicate that the repeat families discovered by *ab initio* tools are actually fragments of larger diverged repeat families. We take advantage of the fact that, although elements of diverged repeats retain only weak similarity along their entire length with the original repeat, certain sections of elements belonging to the same family retain tend to be conserved. If we consider repeat family members that exist as residual fragments in the genome today, the spatial arrangement or order of these fragments is largely preserved. Previous research has addressed the problem of *defragmenting* known repeats [22, 49] . Defragmentation is a procedure to assemble or cluster the fragments of a sequence feature such as a repeat region. Although these approaches are somewhat effective for genomes of well-studied species such as human, they cannot be readily applied to most newly sequenced genomes where little is known about repeat families. Therefore, in the absence of a comprehensive repeat database, the discovery and characterization of repeat families is a necessary component for understanding of the organism's genome.

1.4 Spatial proximity mining in genomic sequence

We describe a data mining technique for discovering spatial proximity relationships among features in genomic sequence. As a case study, we investigate the identification of novel repeat families. An overall view of our approach is shown in Figure 1.1. We apply association rule mining followed by graph mining to find relationships among partial but conserved portions of highly diverged repeat families. We build representations of hypothetical ancestral repeat families from the graph of spatial relationships that were identified by association rule mining.

First, we employ an *ab initio* repeat finder, RepeatScout [44], to extract initial descriptions of repeat families from a DNA string. We then record the start and end coordinates of all instances of each repeat family on the genome. The coordinates are mined to discover association rules representing spatial proximity relationships among elements of all repeat families in a single pass through the list of associations. Discovered association rules represent the spatial relationships Upstream (before) or Downstream (after). We use the association rule metric of *confidence* [16] coupled with Monte Carlo simulation [50, 51] to assess the significance of spatial relations that are discovered.

All qualifying relationships are modeled into a directed graph with edge weights equal to confidence for the relationship and with related repeat families as vertices. Connected components are extracted from this digraph. Each connected component represents a set of repeat families found to co-occur consistently in the same spatial configuration. Each set of repeat families is deemed to represent a spatially conserved larger repeat region that can be analyzed for further characterization.

1.5 Hypothesis

Hypothesis: A combination of association rule mining and graph mining can be used to discover new and interesting spatial patterns among features in genomic data. Specifically, we demonstrate the utility of our knowledge discovery approach for identifying novel and diverged repeat families in genomes. Our results are assessed and validated by biologists for contribution to repeat discovery in genomes.

1.6 Contributions

1. We describe a new algorithm that combines association rule mining with graph mining for discovery of spatial proximity relationships in genomes. A Monte Carlo-based simulation method is used to evaluate confidence in the proximity association rules. We demonstrate application of our algorithm for effective extraction of models of diverged repeat families.
2. Compared to other *ab initio* repeat discovery methods, our method is shown to discover highly diverged repeat families with only small conserved sections using no prior knowledge of the structure or sequence of the repeats. Unlike other repeat defragmentation approaches, our method works at the repeat family level rather than defragmenting individual instances of repeats. Our algorithm can defragment *ab initio* repeats as well as previously known repeats while other defragmentation algorithms work only with known repeats.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORK

Association rule mining in biological datasets has emerged as an important area of research in the last few years. We use locations of families of genome features as input to an algorithm that applies association rule mining and graph mining to find consistent spatial relations among genomic features. As a proof of concept, we apply this algorithm to identify spatial relationships among families of repetitive DNA identified by a computational repeat finder with the goal of defragmenting the repeat families. This chapter is organized as follows. First we discuss basic concepts of association rule mining followed by a discussion of spatial association rule mining. We then present a brief overview of the applications of spatial data mining and graph mining in computational biology. Finally, we close the chapter with a review of algorithms used for computational repeat finding.

2.1 Association rule mining

Agrawal *et al.* first proposed association rule mining [16, 52] to discover rules describing the co-occurrence of items in transaction data. Let $i = i_1, i_2,\ldots\ i_s$ be a set of binary attributes representing items. Each transaction t, in this case, is a binary vector and represents a sales record that contains items that have been purchased in a single transaction. Here, $t[p] = 1$ if item i_p was purchased in this transaction, and $t[p] = 0$ otherwise. Each transaction is a tuple in a database. Let I be a set of some items in i called an itemset in association rule mining terminology. A transaction t satisfies I if for all items i_p in I, $t[p] = 1$. The "classical" association rule problem is generation of association rules from transaction data that meet certain minimal thresholds. Subsequently, association rule mining has been applied in many different domains and to many different types of data including spatial data, temporal and time series data, the world-wide web, etc. [30, 53-56].

An association rule is a probabilistic implication of the form X → Y, which states that if X is observed, then Y is likely to be observed. X is called the antecedent and Y is called the consequent. There are two commonly used measures of the "interestingness" of rules. For association rule X → Y, where X and Y are itemsets:

$$Confidence(X \rightarrow Y) = P(Y \mid X) = \frac{P(X,Y)}{P(X)} = \frac{|\,Transactions(X \bigcap Y)}{|\,Transactions(X)\,|} \tag{2.1}$$

$$Support(X \rightarrow Y) = P(X,Y) = \frac{|\,Transactions(X \bigcap Y)}{|\,Transactions\,|} \tag{2.2}$$

Confidence is the measure of the conditional probability of a rule and quantifies the proportion of items containing X that also contain Y. *Support* gives the fraction of records in the database that contain both X and Y. In the context of a retail business [16], rules having high support and strong confidence are desirable.

Lift [57] is another popular measure of interestingness of association rules. It is also known as interest [58] or strength [59]. Lift is a combined measure of confidence and support.

$$Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{Support(X \rightarrow Y)} \tag{2.3}$$

The problem of association rule mining can be divided into essentially two sub-problems.

- Generation of sets containing items having support higher than a minimum threshold. These sets are called large or *frequent itemsets*.
- Computing association rules from the frequent itemsets with a minimum confidence constraint.

In this section, we introduce some basic algorithms for association rule mining within the context of mining traditional transactional datasets and then summarize selected extensions to the basic algorithm. We provide an overview of spatial data mining research and applications of association rule mining for spatial data mining. Finally, we present a summary of applications of association rule mining to biological problems.

2.1.1 Algorithms for association rule mining

Agrawal *et al.* [16] proposed the AIS (Agrawal, Imielinski, Swami) algorithm in one of the seminal works in association rule mining research. This technique is limited to discovering rules with one *item* in the consequent. AIS makes multiple passes over the database and adds items to large *itemsets* at every iteration. The first pass is used to determine items (itemsets of size 1) that are frequent in the database (i.e., that meet a specified support criterion). Subsequent passes construct itemsets with progressively more items that still meet the support criterion. The itemsets for which support is calculated during each pass are called *candidate itemsets*. A pruning and estimation mechanism is used to discard itemsets with low support. Major drawbacks of this approach include too many passes over the database and high memory usage and runtimes due to generation of candidate itemsets that are eventually discarded.

The Apriori algorithm, again developed by Agrawal *et al.* [60], addressed many of the problems of the AIS algorithm and is one of the most widely used association rule mining algorithms. Apriori employs a new candidate itemset generation method and pruning technique. Apriori utilizes the property that all sub-itemsets of a frequent itemset must be frequent. Only large and frequent itemsets from the previous pass over the database are considered in subsequent passes. Consequently, the number of candidate itemsets to be retained for future passes through the database becomes progressively smaller. Apriori still has the disadvantage of scanning the entire database many times [60]. A number of algorithms have been derived from the basic Apriori algorithm. Most of these algorithms follow one of three approaches: reducing the total number of passes over the entire database [60, 61]; replacing the entire database with a subset based on the current frequent itemsets [61, 62]; and exploring different kinds of pruning techniques to reduce the number of candidate itemsets [63, 64].

The FP-Tree algorithm [65] by Han *et al.* [65] overcomes many of the bottlenecks of the Apriori algorithm by constructing an FP-Tree and generating frequent patterns from the FP-Tree [65-67]. Other extensions include multiple concept level association rule mining [68-70], multi dimensional association rule mining [71-74] and the use of additional constraints such as knowledge based constraints [75, 76], data constraints [77], dimensional constraints [78], interestingness constraints or rule formation constraints [79, 80]. Association rules have also been utilized for classification problems where a set of high quality rules is selected for prediction [81, 82]. Zaki *et al.* [83] provide an excellent survey of parallel and distributed algorithms for association rule mining and classify algorithms based on load-balancing strategy, architecture and type of parallelism used.

2.1.2 Association rule mining from spatial data

Spatial data such as geographical information is extensively used for knowledge discovery [56]. Wide application of remote sensing technology, automated collection of climate data and the importance of geographic information systems (GIS) have all contributed to development of very large spatial databases [84] and new technologies for data mining and knowledge discovery [53].

Spatial relationships denote distance-based relationships (e.g., within, near, very near, adjacent, etc.) among entities. Spatial data mining integrates spatial relationships and properties of items participating in the relationship to derive interesting rules. For example, in the spatial association rule Is_a(X,house)^Close_to(X,Beach)→

Is_expensive(X), the location of the house and its price are combined with the proximity relationships Close_to.

A substantial body of research exists in the area of spatial association rule mining of geographical data. Koperski *et al.* [53, 85] present research in mining geo-spatial databases using the Apriori algorithm. They use a top-down, progressive deepening approach that searches for frequent rules at a high concept level. A deeper search is then used for lower level concepts for frequent rules discovered in the previous step. The authors [86] also proposed a two step method to classify entities stored in GIS databases into spatial and non-spatial classes using decision trees. This work was then extended by Malerba *et al.* [54] to mine census data. Malerba *et al.* utilize the formalism of Datalog from the area of inductive logic programming (ILP) to represent spatial information and background knowledge required for analysis of spatial relations. Han *et al.* adapt their DBMiner relational data mining system [87] to develop GeoMiner [88] for discovering rules from a geo-spatial database. Han *et al.* also designed a spatial data mining language called Geo-Mining Query Language (GMQL), an extension to Spatial SQL [89], as a part of this work.

A graph-based spatial relationship mining engine forms the core of a neighborhood discovery approach proposed by Ester *et al.* [90]. A neighborhood graph is built where a node represents an entity in the spatial database. An edge between two nodes denotes a spatial relation and short paths are used to discover sets of co-located entities that are likely to influence each other.

Rinzivillo *et al.*[91] put forward a general approach to extract spatial transaction data from GIS and mine for frequent spatial association rules. They have implemented the Apriori algorithm [60] as a case study. Knorr *et al.* [92] present an interesting technique to mine the aggregate proximity relationships between entities on a map. In their example, clusters are groups of houses, and the features are other structures on the map such as schools and parks represented as polygons. They identify the features that are located in proximity to clusters in two-dimensional space. Braunmuller *et al.* [93] introduce a parallel spatial data mining approach and demonstrate reductions in I/O cost as well as processing time.

A genetic algorithm-based spatial association rule mining technique (ARMNGA) was proposed by Dai *et al.* [94] to mine databases of images. They demonstrate that ARMNGA outperforms the classical Apriori algorithm [60], especially in tests where a low support parameter is used.

2.1.3 Association rule mining in biology

The exponential increase in the number and size of biological datasets has made the application of knowledge discovery methods such as association rule mining an attractive proposition.

Satou *et al.* [95] applied association rule mining to discover previously unknown correlations among sequence, structure and function in endopeptidase proteins [96]. The authors used data from public protein databases and their PACADE database system for the association rule mining analysis.

Microarray experiments are a common source of very large datasets in biology. These experiments are used to compare the expression or activity of genes in different tissues or under different conditions. One of the first applications of association rule mining to mine microarray datasets was performed by Becquet *et al.* [97]. The authors extracted association rules relating genes from human gene expression data using the Min-Ex algorithm [98] and the results were found to correspond to co-regulated genes reported in the literature. Georgii *et al.* [99] investigated and validated the utility of quantitative association rules based on half-spaces (linear combinations of variables with constant thresholds [100]) to identify co-regulated genes from microarray results. Quantitative association rules based on half-spaces can be defined as follows: "If the weighted sum of some variables is greater than a threshold, then, with high probability, a different weighted sum of variables is greater than a second threshold [100]". The authors show that, compared to regular discrete interval-based association rules, these rules can better model desired biological inferences from the continuous numerical data produced by microarrays. Besson *et al.* [76] proposed D-miner, a algorithm to mine concepts under constraints, and applied it to analyze expression properties of genes from microarray data. A novel algorithm that predicts co-regulated sets of genes with a lower false positive rate than the Apriori algorithm was presented by Ji *et al.* [101].

A promoter for a gene is a DNA sequence, usually located near the gene, that plays a role in governing expression of that gene. Huang *et al.* [102] mined promoter regions of genes sharing similar expression levels and extracted rules associating known sequence features related to gene expression (transcription factors) and repetitive sequences in promoter regions. From the same research group, Horng *et al.* [103] published a study describing results of association rule mining in repetitive DNA from humans. They created a Repeat Sequence Database containing repetitive DNA from sequenced genomes. Rules were then extracted representing combinations of transcription factor binding sites in repetitive DNA present in the database and screened using a Chi-square test. Morgan *et al.* [104] also verified the effectiveness of association rule mining as a data mining technique to find biologically relevant pairs of transcription factor binding sites in DNA. They found that pairs of transcription factor binding sites identified by association rules to be co-located consistently were also co-cited in PubMed abstracts and that the predicted associations had been confirmed experimentally.

Negative association rules have been utilized by Artamonova *et al.* [105, 106] to identify erroneous protein annotation in the publicly available PEDANT genome database [107]. A majority of the identified proteins have been characterized using unsupervised similarity based methods. This approach, due to propagation over time, has led to a high rate of errors in annotations produced by computational characterization. The authors mine for strong negative association rules denoting incorrectly characterized proteins in public repositories.

CARSVM is a hybrid classification framework proposed by Kianmehr *et al.* [108] using association rule mining and support vector machines. They report an increase in classification accuracy of gene expression data when features selected through association rule mining are used as input for the learning process for the support vector machine. Another application of association rule mining for classification called

NETCAR was implemented by Tamura *et al.* [109] for mapping genotype to phenotype in microbes. Phenotype is defined as an observable characteristic of an organism such as a developmental stage or behavior. Genotype is the genetic composition of an organism. Tamura *et al*. mined pairwise and multiple-to-one associations from gene to phenotype and demonstrated the advantages of association of groups to genes to a phenotype over simple pairwise rules.

2.2 Spatial data mining in biology

Spatial data mining has also been applied to biological data. There is evidence of a higher organization of genes [110] in eukaryotes. Co-regulated genes have been reported to be located together and this makes spatial mining of the genes a desirable option. Horimoto *et al.* [111] performed a comparative analysis of gene locations in 19 genomes and discovered statistically significant spatial correspondence of similar genes between related organisms.

One of the major challenges in structural biology is to understand the mechanism of protein folding. Yang *et al.* [112] proposed a spatio-temporal approach to mine three-dimensional (3D) structural data and summarize protein folding trajectories. They applied their technique to predict a consensus trajectory from multiple folding pathways. Rantanen *et al.* [113] have used the spatial locations of protein atoms to revise 24 pre-defined protein atom classes. The authors model the atoms using a Gaussian mixture model and estimate parameters with the expectation-maximization (EM) algorithm. A dissimilarity matrix is derived from the Gaussian mixture models and used to compare and reclassify the 24 classes.

Researchers have also applied GIS spatial mining techniques and software to structural information from biology. A three-dimensional (3D) morphological database was mined to extract topographical attributes using GIS techniques [114]. The attributes were then used as features for classification of the three-dimensional phenotypes. Dolan *et al.* [115] designed an interactive genome browser called Genome Spatial Information System (GenoSIS) for the mouse genome that uses the ArcGIS commercial GIS software and, unlike other genome browsers, supports spatial queries.

Hemert *et al.* [116] have applied association rule mining to derive two separate categories of association rules from a database containing spatio-temporal gene expression data from mouse including images showing gene expression patterns over time. They discovered regular association rules relating genes that are co-expressed and, in a novel application of spatial data mining, also extracted rules associating spatial regions where genes were co-expressed.

2.3 Graph mining in biology

The advent of high throughput biology methods has enabled scientists to discover different types of relational information about genes, proteins, gene-protein interaction, protein-protein interaction, transcription factors, phenotype and other properties of biological systems. Graphs have proven to be useful mechanisms for representing these relationships and graph mining algorithms are popular for extracting knowledge from the graph structures [17, 117-119]. Aittokallio and Schwikowski [117] provide a highly useful review of graph based methods for analyzing networks in cell biology . Below we discuss a few representative examples from this large body of literature.

Hu *et al.* [118] represented different microarray gene expression data sets as graphs and then mined for frequent subgraphs that occur in the graphs to find sets of genes that are commonly co-expressed. Their algorithm, CODENSE, discovers coherent subgraphs from a set of graphs and also extracts information about overlapping subgraphs, an area of interest in biology when a single gene is a part of multiple networks. Yan *et al.* [120] mined 105 human microarray datasets and to identify the frequent dense subgraphs, each representing a set of co-expressing genes or transcription modules.

Predicting and understanding protein-protein interaction networks or protein complexes is another important area of research in functional genomics. A large body of work exists based on the principle that the interacting proteins can be found as cliques in interaction networks representing pairwise protein interaction data. King *et al.* [121] proposed a technique to partition a interaction graph into clusters using a cost-based local search algorithm for predicting sets of protein complexes. Some clustering with overlap approaches [122, 123] are also based on the fact that a number of proteins have been reported to participate in multiple complexes. Qi *et al.* [124] propose a graph mining approach targeting protein complexes that may not be represented by cliques but other structures such as a “star” in an interaction graph. The authors used a supervised clustering technique by learning properties from known complexes and show an improved performance over clique-based approaches. Li *et al.* [125] put forward a modified version of the DPClus graph clustering algorithm [126] that uses topological characteristics of vertex distance and subgraph density to identify clusters with small diameter, a characteristic of protein complexes. The authors show that their algorithm, IPCA, when compared to other approaches, is able to identify most experimentally determined complexes with a low false positive rate.

RNAmine was proposed by Hamada *et al.* [127] to tackle the problem of identifying similar substrings from a set of RNA sequences. The authors use a directed labeled graph, called a stem graph, and constraint based graph mining to identify the stem pattern for an RNA family. The stem patterns can then be used for genome wide scans to identify members of the RNA family.

2.4 Computational analysis of repeat regions in DNA

Here we review the major algorithmic approaches currently employed for repeat identification in DNA. There are two main categories of repeat finding tools—tools that rely on a database of known repeats and *ab initio* tools that do not rely on prior knowledge of repeat sequences. While we provide an overview of signature based methods, the bulk of the review is focused on *ab inito* algorithms because our spatial mining algorithm is an *ab initio* repeat finder. *Ab initio* tools are becoming increasingly important due to the tremendous growth in the amount and diversity of sequences being generated by genome sequencing projects. For each *ab initio* tool we describe the sequence data utilized, the approach used for initial identification of repeats, and the method used to extract descriptions of repeat families. Sequence data can be a chromosomal length DNA sequences or sets of short DNA sequences. In the discussion below, repeat identification tools are introduced based upon the type of algorithm they use to identify and classify potential repeats.

2.4.1 Database- and signature-based identification techniques

Library-based systems identify repetitive sequences by comparing input sequence datasets against a database of known repeat sequences [45]. The predominant tool used in repeat identification, RepeatMasker [128], is database driven. A string matching engine such as BLAST [129, 130], WU-BLAST (http://blast.wustl.edu/), or Crossmatch (www.phrap.org) is utilized for comparison of an input sequence to known repeat sequences. The singular disadvantage of library-based systems is their inability to detect or analyze novel repeat families.

Signature-based repeat identification tools search the query sequence for certain short DNA strings that represent patterns that occur in known repeats. These tools may also search for spatial arrangements of DNA strings characteristic of a particular known repeat family. Unlike database-dependent tools, all signature-based tools employ heuristics based on *a priori* information of particular known repeat families. However, some signature-based tools also may use a database of repeats at some stage in the analysis process.

LTR_STRUC [131], FINDMITE [132] and RetroTector [133] are examples of signature based tools that look for DNA patterns characteristic of certain types of known repeat families. LTR_STRUC and FINDMITE search for presence of DNA strings arranged in a specific spatial layout. FINDMITE uses an adaptation of the Knuth-Morris-Pratt string matching algorithm [134]. RetroTector also provides a graphical interface and various other utilities to investigate the detected elements.

MAK [135] uses an input repeat family to seed a search for all similar repeat families and their elements. Inverted Repeats Finder [136] searches for palindromic DNA strings that are a hallmark of a number of known repeat families. The approach proposed by Andrieu *et al.*, TE-HMM [137], involves building different hidden Markov models using training sets of repeats and gene sequences. TE-HMM is then used to classify the query sequence or part of a query sequence into one of these categories based upon DNA composition of the sequence. TSDFinder [138] and SINEDR [139] use structural knowledge of known repeats to predict potential repeat regions.

Signature-based tools use spatial mining heuristics derived from known classes of repeats. However, these tools are also limited to discovering novel repeat families that are similar to known repeat regions. Our data mining approach is applied to identify diverged and fragmented repeat regions in DNA that may otherwise not be identified by available repeat finders. There are some computational techniques available to recognize and defragment co-linear elements. Given a database of repeat families, Transposon Cluster Finder [49] can identify fragmented and/or nested elements of a family. LTR_MINER [140] is another tool that clusters elements of a repeat family when the fragments exist within a range expected from the size of the repeat family consensus sequence. TEnest [141] and REannotate [142] also defragment elements of repeat families and construct a visualization of the nesting order among the repeat families. All these tools work with the Repbase library [45] or a database of known repeats to seed their search and the focus is on computing the chronological history of known repeats using defragmentation of co-linear elements and nesting order.

2.4.2 *Ab initio* identification techniques

Algorithms falling into this category identify repeat families and their elements without any prior knowledge of structure or organization of known repeat regions. The process of *de novo* identification enables these tools to discover previously unknown repeat families. These methods are therefore of critical importance when analyzing newly sequenced genomes about which little is known. Most of the tools produce a set of *consensus sequences*, each representing one repeat family, as the final output. We use one such tool, RepeatScout [44], for initial repeat identification in our data mining pipeline.

For the sake of clarity and organization, we have divided all the algorithms into two stages. The first stage deals with initial identification of elements of repeat families. The second stage, repeat family definition, is focused on identifying the boundaries of each element and extracting a consensus sequence for each family. We discuss major *ab initio* repeat finding algorithms/tools within the framework of these two stages. Note that some tools perform element identification without generating a consensus sequence for the repeat family.

2.4.2.1 Initial identification of elements

All *ab initio* discovery of repeat families begins with identification of substrings that are present in numbers above a certain threshold in the input DNA sequence. Four basic (but not entirely exclusive) groups of approaches have been utilized in initial identification of elements.

Two of the most widely used exact string matching and sequence alignment algorithms are BLAST and WU-BLAST. All words from the query and reference dataset with length equal to or more than a user specified minimum threshold are compared and the match is extended at both ends as long as significant similarity is found. These tools are the string matching engine for many of the available tools. Repeat Pattern Toolkit [143] and RECON [144] use the BLAST module (www.ncbi.nlm.nih.gov/blast/) and WU-BLAST respectively. Both tools then use a graph-based single link clustering algorithm to group elements of a repeat family. Single link clustering regards two elements as belonging to the same family if they share a sequence identity longer than a certain proportion of either one of the two. PALS is a banded searching local alignment algorithm that is employed by PILER [145]. Banded searching is local alignment of only those strings that are located within a certain distance of each other.

The rest of the repeat finding tools implement various exact string matching techniques to find elements of repeat families. These techniques are classified as *k*-mer approaches [14]. A *k*-mer or "word counting" approach views a repeat as a substring *w* of length *k* that occurs more than once in a sequence *S* of length *n*. A repetitive subsequence *w* that cannot be extended without introducing mismatches is called a *maximal repeat*.

Since there are 4^k possible words of length k (DNA has a 4-letter alphabet), these approaches usually require that k be at least $\log_4 (n)$ where n is the length of the sequence [44, 47],47,[146]. The value of k required for indexing assembled plant genomes is roughly 12 to 19 based on reported eukaryotic genome size estimates.

REPuter [147] was one of the first tools to implement a *k*-mer search algorithm for repeat finding. Its search engine component, REPfind, uses a suffix tree data structure developed by Weiner [148] for storing all repeated exact *k*-mers or string that have lengths greater than or equal to a user-specified size. Suffix trees can be used to search for strings in linear space and time with a complexity of $O(n + z)$ where z is the number of maximal repeats. The REPuter *k*-mer approach has also been effectively used by other tools. For example, RepeatFinder [149] and RepeatGluer [48] both use the REPuter engine to generate an initial list of maximal repeats. ReAS [47] and RepeatScout [44], the latter being a part of our data mining pipeline, use a similar approach where only fixed length *k*-mers are used to seed the search. The ReAS algorithm employs a randomly selected, high frequency *k*-mer as “bait” to retrieve strings of length 100 bp centered on the *k*-mer. RepeatScout also builds a library of high frequency fixed length *k*-mers and uses these as seeds for an efficient greedy search during the family definition stage. RepeatScout implements a modified version of the classical local alignment algorithm by incorporating a penalty-based scoring system for screening the *k*-mers [44].

Instead of searching for only exactly identical strings of length *k*, spaced seed algorithms search for strings allowing for a predefined number of mismatches. The first spaced seed tool, PatternHunter [150], allowed mismatches in fixed positions only.

Multiple spaced seed techniques [150-152] extend this idea by using several optimal spaced seed patterns in searches. “Indel seeds” proposed by Mak *et al*. [153] use a spaced seed strategy more suited to weakly conserved regions in DNA such as repeats. The mismatch positions in the string can tolerate not only single base mismatches but also insertions and deletions of short DNA strings [136]. “Indel seeds” potentially offer higher sensitivity as compared to exactly identical seeds and are of considerable utility when evaluating repeat regions where insertions and deletions are more likely to be present.

Repeat Analysis Program [146] employs a hybrid string matching approach by creating an index of spaced seeds according to its location on the genome. Word counters are created for each position in the sequence, and all potential words of size *k* beginning from each sequence position are enumerated using a multi-array data structure.

One of the earliest and simplest repeat finding techniques was based on the Dot matrix. The dot-plot [154] tool plotted a input DNA sequence against itself. Auto Dot PLOT or Adplot [155] is an adaptation of the dot-plot principle wherein similar *k*-mer elements located within a user-specified range are detected in the first step. A sliding window based filtering is applied to screen out repeat families whose sum of element lengths is below a threshold. The focus of this class of tools is visualization of the distribution of repetitive regions over the sequence.

Periodicity-based approaches are fundamentally different from the aforementioned techniques. The Spectral Repeat Finder from Sharma *et al*. [156] uses Fourier transforms to analyze DNA sequence in the frequency domain rather than the commonly used time domain. The power spectrum of DNA sequence generated from the

Fourier transforms is used to identify both short term and long term autocorrelations of the sequence with itself. High intensity peaks in the power spectrum of the sequence represent candidate repeat elements. These candidate repeats are used to seed a local alignment search to detect similar elements and to determine the consensus sequence for the family. The time complexity of the algorithm is $O(n^2)$.

2.4.2.2 Defining repeat families

The methods described in the preceding section are used by computational repeat finders to identify sets of similar elements whereas the following section discusses techniques used to extend and combine elements into repeat families, where possible, and to extract descriptions of the consensus sequence for each repeat family.

Some tools implement repeat family identification by further clustering to derive the final family definition. This process may be guided by biological heuristics. RepeatFinder [149] begins with the initial set of exact repeats identified using one of two suffix tree approaches (see stage 1) and then merges different exact repeat elements that are close together (merging using gaps) or that overlap (merging using overlap) to generate a set of "merged repeats." Multiple rounds of clustering follow and a single repeat element is selected from each cluster as the representative. Unlike other repeat finding tools, RepeatFinder does not construct a consensus sequence. PILER clusters similar repeat elements identified by PALS into "piles". The characteristics of elements clustered in a pile are matched against profiles of four types of known repeats. The

MUSCLE [157] alignment program is then used to generate the consensus sequence for each family detected.

Another technique used for family definition is graph mining. Repeat Pattern Toolkit [143] builds a repeat graph $G = (V,E)$ using only exact matches from the clustering step. Vertices V represent the repetitive sequences or elements. Weighted edges E represent the relationship among similar elements. Connected components from the graph are converted into minimum spanning trees using Kruskal's algorithm and Binsort [134] in $O(|E| + |V|\log|V|)$ time. Each minimum spanning tree represents a repeat family. Each tree is reduced to a single vertex to deduce the consensus sequence for the family. This vertex is the weighted midpoint of all the other vertices in the graph. A major limitation of this technique is its inability to address repeat families that have elements with insertions and deletions since only exact alignments are analyzed.

Bao and Eddy [144] extended and improved upon the work of Agarwal and States [143] with RECON. The algorithm refines the elements derived from the results of local alignment. This is required since RECON works with multiple DNA sequences instead of a single long DNA sequence. The final set of elements is represented as a repeat graph H where each element is a vertex and edges represent relationships among elements. Elements with an overlap ratio above a specified threshold are deemed to belong to the same repeat family while those with significant alignment but with overlap ratio below the threshold are considered to belong to different families. RECON only reports the elements for each repeat family.

Repeat elements are known to frequently contain smaller repetitive subsequences within them. RepeatGluer [48] models the complex structure of nested repeats into a weighted digraph using A-Bruijn graphs, an extension of the de Bruijn graphs [158]. The original de Bruijn graph can model only perfect repeats. The concept has been generalized by Pevzner *et al*. [48] to A-Bruijn graphs to enable approximate matches or imperfect repeats to be represented. The algorithm constructs an adjacency matrix that is used to construct a weighted A-Bruijn graph G where the weight of the edge between two vertices is the number of edges joining them. A number of biologically derived heuristics are used to simplify the graph. Finally, each set of connected components or "tangle" is resolved to a consensus sequence.

The most widely used method for family definition is string extension. Algorithms covered in stage 1 that cluster high frequency k-mers as a first step often employ string extension techniques for the second step of family definition. REPuter was one of the first repeat finders to use the string extension method [147]. The output of its suffix tree based search engine is processed further for finding degenerate repeats using either a Hamming distance model or an edit distance model [159]. The edit (or Levenshtein) distance approach has an overall time efficiency of $O(n+zk^3)$ where n is the size of the sequence and z is the number of k-mers extended. Of note, the REPuter package has been subsumed by Vmatch [160]. Vmatch uses suffix arrays [161] that have a reduced space requirement compared to suffix trees for indexing substrings.

RepeatScout [44] generates consensus sequences by first detecting a set of highly repetitive fixed length k-mers as described in stage 1. The algorithm extracts all instances of each *k*-mer repeat element with its surrounding region and then greedily extends the boundaries on both ends yielding a consensus sequence for the repeat family representing the *k*-mer.

The set of 100 bp sequences selected by ReAS [47] for each *k*-mer are processed by ClustalW [162] to generate an initial 100 bp consensus sequence centered on the *k*-mer. If another *k*-mer exists near either end of the initial consensus sequence, it is used to capture additional sequences from the input dataset. The newly retrieved sequences are then utilized to extend the initial consensus sequence up to five times.

CHAPTER 3

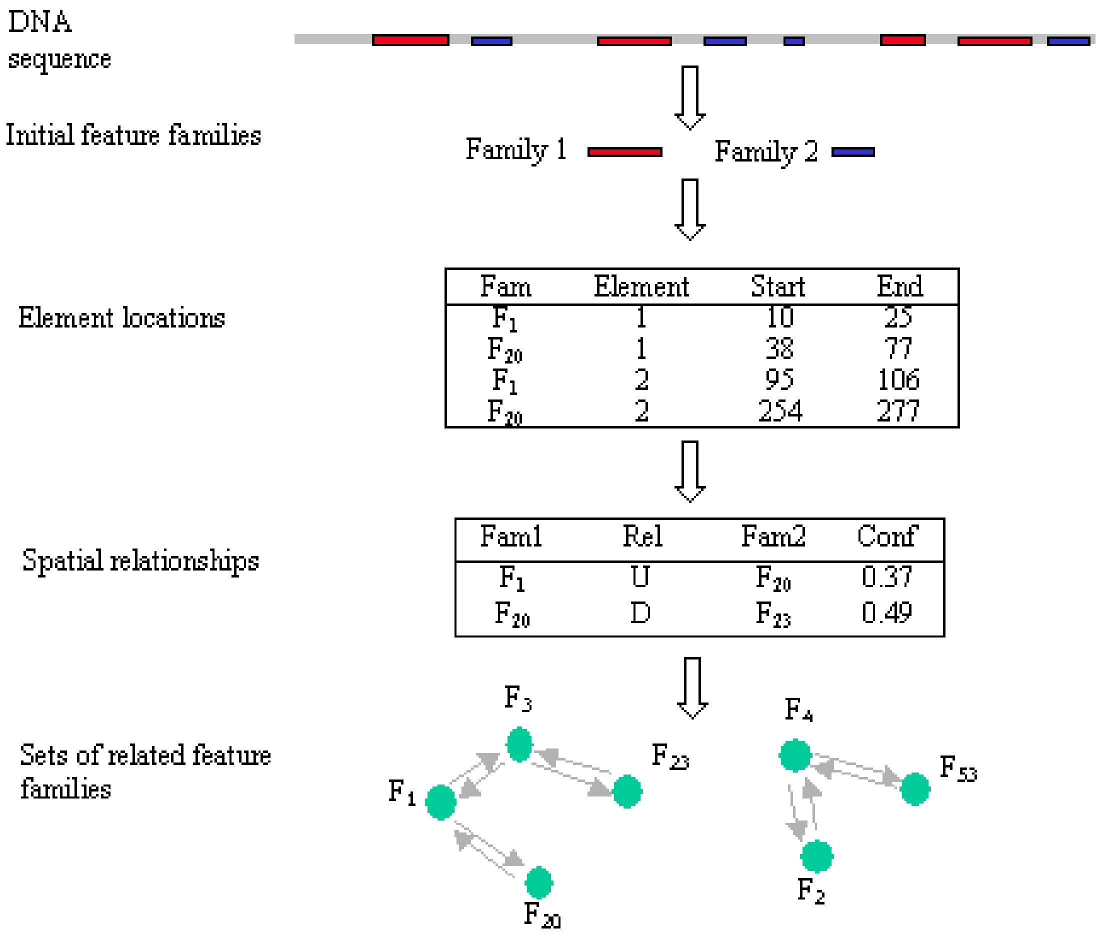
SPATIAL ASSOCIATION RULE MINING

We have developed a data mining algorithm utilizing concepts from association rule mining followed by graph mining to mine for spatial proximity relationships in one-dimensional data.

Hypothesis: A combination of association rule mining and graph mining can be used to discover new and interesting spatial patterns among features in one-dimensional space genomic data. Specifically, we demonstrate the utility of our knowledge discovery approach for identifying novel and diverged repeat families in genomes. Our results are assessed and validated by biologists for contribution to repeat discovery in genomes.

Concepts from association rule mining have been adapted to mine and quantify directed spatial proximity relationships on genomic sequence. These spatial proximity relationships are modeled into a directed and weighted graph. The connected components within this graph represent the features that co-occur on the genome.

In this chapter, we first describe our general approach for discovery of spatial proximity relationships in genomic data. Figure 3.1 illustrates how we apply spatial association rule mining to genomic sequence. We then describe the types of spatial relationships our algorithm identifies, the association rule metrics used to assess the interestingness of the discovered rules, and the subsequent use of graph mining to identify frequently co-occurring patterns. In order to demonstrate the utility of our spatial data mining approach, we describe its application to the problem of *ab initio* repeat finding in the next chapter.



Fam	Element	Start	End
F_1	1	10	25
F_{20}	1	38	77
F_1	2	95	106
F_{20}	2	254	277

Fam1	Rel	Fam2	Conf
F_1	U	F_{20}	0.37
F_{20}	D	F_{23}	0.49

Figure 3.1

Process flow of the spatial association rule mining algorithm

3.1 Spatial association rule mining

Association rules have been widely used for knowledge discovery in a variety of data types to discover co-occurring entities [16, 60]. The concept lends itself well to extracting and quantifying spatial proximity relationships from one-dimensional data like a genome.

Unlike classical association rule mining applications that mine for simple co-occurrence type relationships, we are interested in extracting directed spatial proximity relationships. We identify directed spatial proximity relationships based on co-occurrence of features within a range or distance measure. We then apply the confidence metric from association rule mining theory to calculate the strength of the identified spatial proximity association rules among features.

3.1.1 Spatial relationships

Here, we define the spatial proximity relationships that are the object of our mining algorithm. The genomic sequence, for the purposes of our algorithm, is a long string of characters. For simplicity, we define the process in terms of a single chromosome instead of the whole genome. However, the following definitions can be applied a whole genome without modification. Let the entire chromosome be $C = x_1,x_2,\ldots,x_n$ where n is the length of the chromosome and x_i represents the nucleotide at position i on the chromosome from the alphabet $S = \{A,T,G,C,N\}$ where A, T, G and C are one of the four nucleotides that make up DNA and N represents an unknown nucleotide. Genomic features are often classified into families. Our overall goal is to discover frequently occurring spatial relationships among families. Let F be the set of feature *families* under consideration (for example, repeat families or gene families). For each family F_i, the set of all instances of F_i on the genome is E_i. Figure 3.2 shows the layout of instances of families F_1, F_2 and F_3 where $E_1=\{elem_{11}, elem_{12}, elem_{13}\}$, $E_2=\{elem_{21}, elem_{22}\}$ and $E_3=\{elem_{31}\}$. We call each instance an *element*.

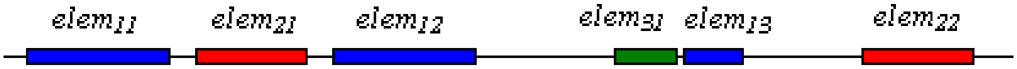


Figure 3.2

Illustration of instances of families F_1, F_2 and F_3

$F = \{F_1,F_2,\ldots,F_m\}$ where m is the total number of families. Each family represents a hypothetical ancestral sequence that can be denote by a graph, grammar or consensus sequence.

$E = \{E_1, \ldots. E_m\}$ where E_j is the set of all elements of family F_j and m is total number of families. Each set E_j consists of all instances of family F_j.

$E_j= \{elem_{j1}, elem_{j2},\ldots., elem_{jp}\}$ where j is the family and p is the number of instances of F_j on the chromosome C.

The location of each element on the chromosome is designated by specifying the starting and ending positions in the genomic sequence: $elem_{jk}(s,e)$ where s is the starting position on chromosome C and e is the ending position. Although genomic sequence is typically given as a single sequence of characters, this sequence actually represents the two complementary strands of DNA with opposite orientation as shown in Figure 3.3.

The sequence of one strand completely specifies the sequence of the other strand by complementary base pairing (A pairs with T and C pairs with G). Each strand of the DNA also has a direction. Genomic features can occur on either strand. By convention, the sequence that is given is called the “positive” strand. The implied complementary strand has the opposite orientation and is called the “negative” strand. Each position on the chromosome C is numbered according to the positive strand, i.e., in left to right orientation. If the element is on the negative strand, the starting position will be greater than the ending position. The length L of $elem_{jk}(s,e)$ = $(s\text{-}e)$+1 for elements on the negative strand while the length of elements on the positive strand is $(e\text{-}s)$+1. This orientation property affects our directed spatial proximity relationships since the strand of the element will determine how spatial relations are calculated.

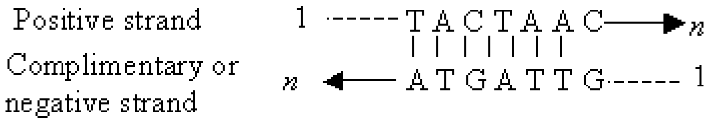


Figure 3.3

Double stranded nature of DNA

If the starting and ending positions of an element are not specified, we denote the strand using a superscript as shown in Figure 3.4. The strand of an element must be taken into account by the data mining algorithm for identifying directed spatial proximity relationships.

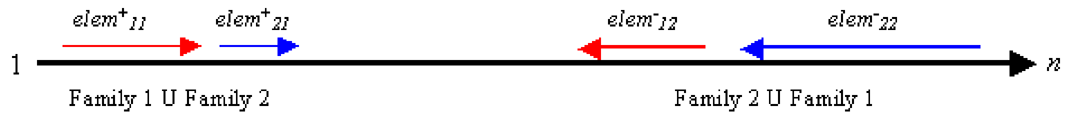


Figure 3.4

Illustration of elements from two families on different DNA strands

We mine for spatial relationships among the elements of families on the chromosome. If we are considering two instances $elem_{xk}$ and $elem_{yl}$ of families F_x and F_y then the spatial relationship discovery process will identify Upstream/Downstream, Overlap, and Within relations as illustrated in Figure 3.5 and Figure 3.6 and explained below. Let the maximum distance for a spatial relationship be a user-supplied parameter d_{max}. The definitions for these relations are stated in terms of the positive strand. The definitions for the negative strand follow in a straightforward way. We consider relationships among all elements irrespective of their family, and therefore, we can discover both intra-family and inter-family relationships.

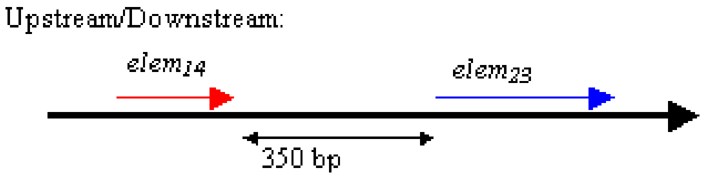


Figure 3.5

Depiction of Upstream and Downstream spatial relationships

An Upstream (U) relationship occurs between two elements if the end of the first element is prior to the start of the second element and the distance from the end of the first to start of second is within the range [1..d$_{max}$]. More formally, an instance *elem*$_{xk}$(s_1,e_1) is related to instance *elem*$_{yl}$(s_2,e_2) by relation U if $e_1 < s_2$ and ($s_2 - e_1$+1) is in the interval [1..d$_{max}$]. The length of this interval is the distance between the two elements and will be positive. Downstream relationships (D) can also be defined similarly, but it is not necessary to consider both Upstream and Downstream because the first implies the second. Figure 3.5 shows an instance of Family 1 U Family 2 spatial relationship.

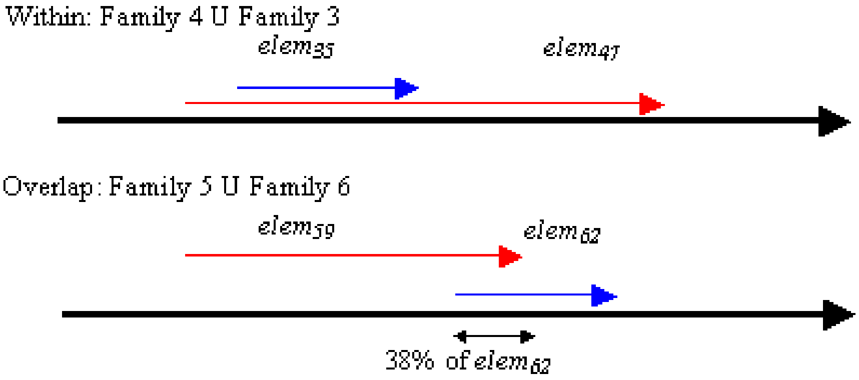


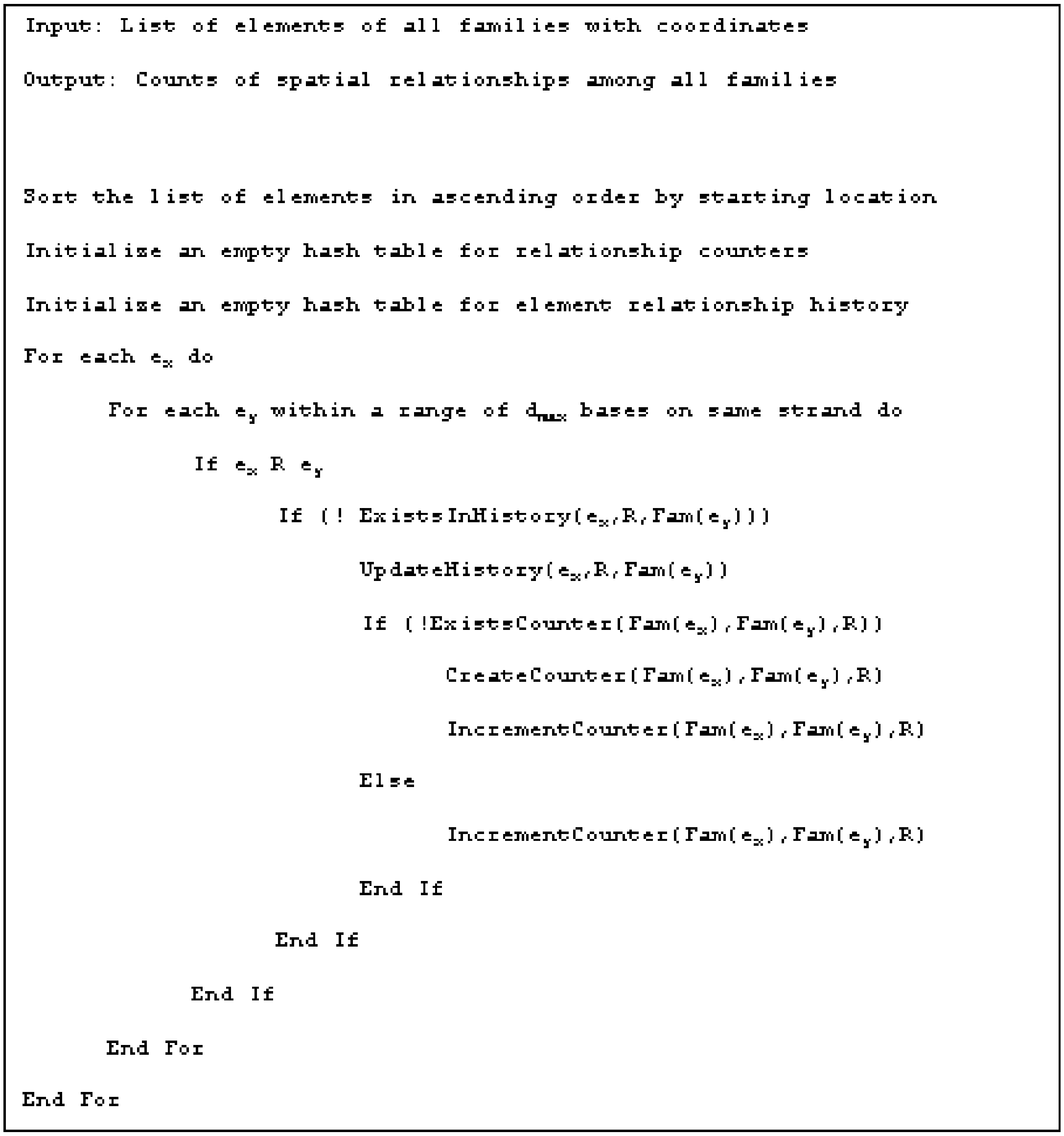
Figure 3.6

Depiction of within and overlap special case spatial relationships

We resolve cases of overlap of two elements and cases where one element exists entirely within another to special cases of Upstream relationships as shown in Figure 3.6. In the case where one element is entirely within another element, the spatial relationship is Upstream if $s_2 \geq s_1$ and $e_2 \leq e_1$. The distance between the two elements is defined as for Upstream above, but in this case, the value is negative. Similarly overlapping elements are defined as Upstream if $s_2 > s_1$ and $s_2 <= e_1$ and $e_2 > e_1$. The distance between the two elements is defined as for Upstream and is again negative. Figure 3.6 shows special cases of Overlap and Within spatial relationships that are converted into Upstream and Downstream spatial relationships. The Within relationship is represented as an Upstream relationship (Family 4 U Family 3). The Overlap relationship is represented as Upstream relationship (Family 5 U Family 6).

3.1.2 Directed spatial association rule mining algorithm

We provide an overview of the directed spatial association rule mining procedure in this section. This algorithm is initialized with the coordinates of elements for all families and a distance parameter d_{max}. Neighboring elements within a range d_{max} are detected for each element and the appropriate Upstream or Downstream relationship is recorded in a hash table (Figure 3.7). In order to prevent redundancy while counting occurrences of a relationship between two families, we allow each element to have only one relationship with any other element of a particular family.



```
Input: List of elements of all families with coordinates
Output: Counts of spatial relationships among all families

Sort the list of elements in ascending order by starting location
Initialize an empty hash table for relationship counters
Initialize an empty hash table for element relationship history
For each $e_x$ do
    For each $e_y$ within a range of $d_{max}$ bases on same strand do
        If $e_x$ R $e_y$
            If (! ExistsInHistory($e_x$,R,Fam($e_y$)))
                UpdateHistory($e_x$,R,Fam($e_y$))
                If (!ExistsCounter(Fam($e_x$),Fam($e_y$),R))
                    CreateCounter(Fam($e_x$),Fam($e_y$),R)
                    IncrementCounter(Fam($e_x$),Fam($e_y$),R)
                Else
                    IncrementCounter(Fam($e_x$),Fam($e_y$),R)
                End If
            End If
        End If
    End For
End For
```

Figure 3.7

Algorithm for association rule mining to discover spatial relationships on the genome

3.1.3 Spatial association rule metrics

We use the *confidence* metric for computing “interestingness” of the spatial association rules we discover using our algorithm. The number of occurrences that relate elements of two feature families by a certain relationship must be viewed in light of the number of elements of both families present in the genome. We calculate the confidence of a rule with respect to the participating families as follows

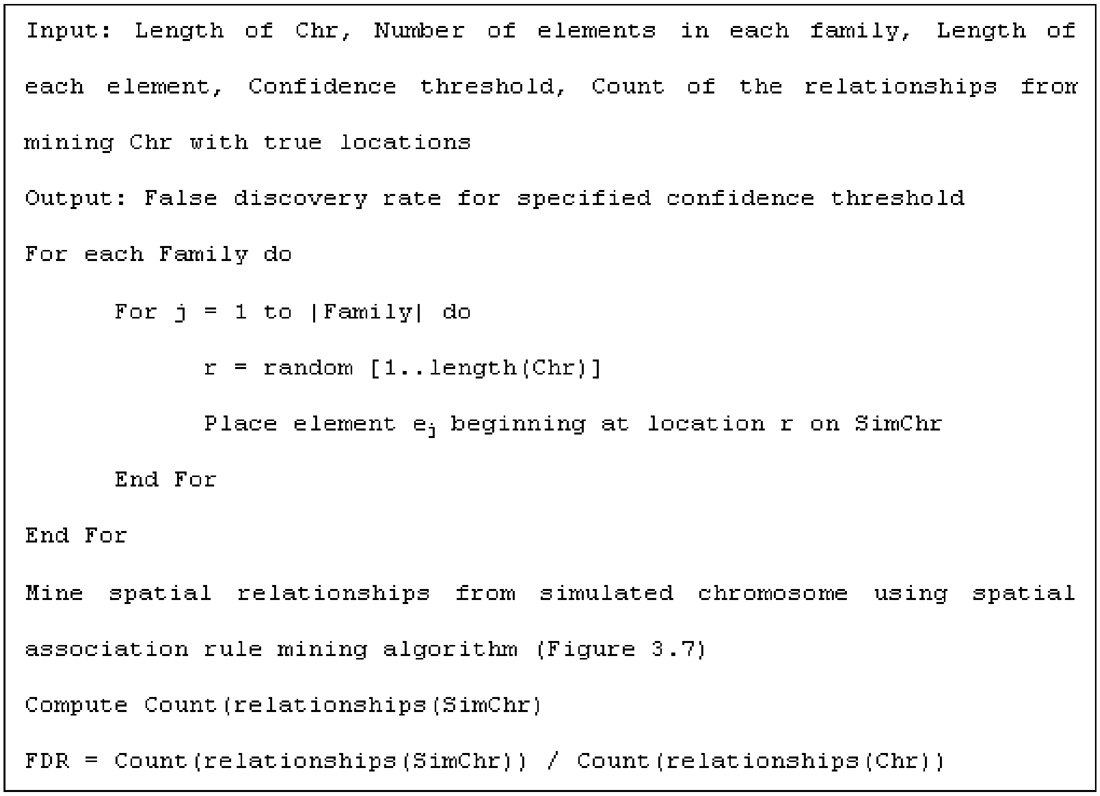
$$Confidence = \frac{\left| F_x \sigma F_y \right|}{Min\left(\left| F_x \right|, \left| F_y \right| \right)} \tag{3.1}$$

where σ is a spatial relation such as Upstream, | F_x σ F_y | is the number of times this relationship is observed among elements of families F_x and F_y. $|F_x|$ and $|F_y|$ is the number of elements of the respective family on the genome.

The support metric is typically applied to transactional data where items present in large number of transactions are of greater interest from a financial point of view. However, the focus of our algorithm is to discover spatial association rules that occur more frequently than would be expected by chance. For example, in the domain of repeat finding, repeats that occur very frequently in the genome have been studied for many years and the most “interesting” new relationships that remain to be discovered may be among repeats that do not occur in large numbers in the genome [15]. Therefore, we do not use a support metric.

3.1.4 Monte Carlo analysis

Any methodology such as ours that involves making inferences from distribution profiles is subject to influence by counts of items. There is a possibility of discovering relationships that are a result of random association due to very high number of elements in a family. In order to avoid this problem, we derive a false discovery rate (FDR) for spatial relationships [163] using Monte Carlo simulation.



```
Input: Length of Chr, Number of elements in each family, Length of
each element, Confidence threshold, Count of the relationships from
mining Chr with true locations
Output: False discovery rate for specified confidence threshold
For each Family do
    For j = 1 to |Family| do
        r = random [1..length(Chr)]
        Place element $e_j$ beginning at location r on SimChr
    End For
End For
Mine spatial relationships from simulated chromosome using spatial
association rule mining algorithm (Figure 3.7)
Compute Count(relationships(SimChr)
FDR = Count(relationships(SimChr)) / Count(relationships(Chr))
```

Figure 3.8

Algorithm to construct simulated chromosomal locations and compute false discovery rate for spatial relationships

For each simulated chromosome (SimChr), we use a uniform distribution to select a random starting location of each element and retain the same length as the original element (See Figure 3.8). Note that the SimChr has no sequence—we are only dealing with location data. Spatial relationships discovered with randomly located elements are used to estimate the number of false positive identifications. The FDR is estimated as follows:

$$\text{FDR} = \#\text{ false positives }/\#\text{ positives} \tag{3.1}$$

where the # false positives is the number of spatial relationships discovered from the random locations and # positives is the number discovered with true locations. The FDR is estimated with varying confidence thresholds and the confidence threshold that gives the desired FDR is selected. The FDR provides a confidence threshold used by the overall data mining process. We have estimated the FDR from a single simulated chromosome and as an average of the value obtained with multiple simulated chromosomes and find little difference in the results. We conclude that a single simulated chromosome is sufficient for estimating the FDR. However, because the number of elements per family varies widely, we also compute the statistical significance of each discovered relationship using the Chi square (χ^2) statistical measure. We use an example to describe how Chi square (χ^2) has been applied to our spatial association rule mining algorithm.

Suppose we are considering two families F_1 and F_2 with 15 and 19 elements respectively where ten elements of F_1 are spatially related to ten elements of F_2 according to Upstream/Downstream relationships defined previously. We want to determine if the locations of elements of F_1 and F_2 are independent. Our null hypothesis states that the locations of all elements are independent. We then map the relationship details into a 2x2 contingency table for the observed frequencies as shown in Table 3.1.

Table 3.1

Example contingency table for observed values

Observed	F_1	F_2	
Within range R	10	10	20
!(Within range R)	5	9	14
	15	19	34

We compute the expected frequencies from the simulated chromosomal locations used for estimating the false discovery rate (Figure 3.8). Using this approach with the values above, we find an average of 3 pairs of elements of F_1 and F_2 exist within the distance range d_{max}. We map this information into a 2x2 contingency table (Table 3.2).

Table 3.2

Example contingency table for expected values

Expected	F_1	F_2	
Within range R	3	3	6
!(Within range R)	12	16	28
	15	19	34

The Chi-square value is calculated using the formula

$\chi^2 = \Sigma$ [(Observed frequency – Expected frequency)^2 / Expected frequency]

Therefore the χ^2 for the above contingency table is

$\chi^2 = (10\text{-}3)^2/3 + (10\text{-}3)^2/3 + (5\text{-}12)^2/12 + (9\text{-}16)^2/16 = 39.81$

The degree of freedom for a 2x2 contingency table is 1. The critical value for 1 *df* is 3.84 for a significance level (α) of 0.05. Hence our obtained value for χ^2 of 39.81 refutes the null hypothesis that the locations of elements of F_1 and F_2 are independent. The observed relationship between F_1 and F_2 is statistically significant at the p=0.05 level.

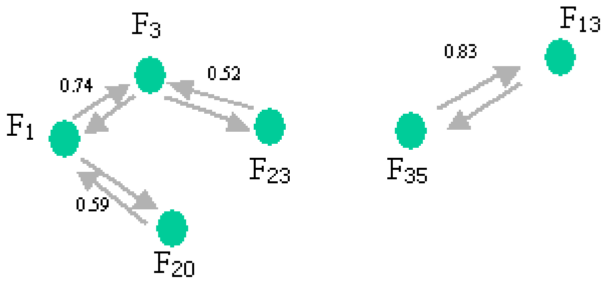


Figure 3.9

Example of a directed graph created from qualifying spatial relationships.

3.2 Graph mining

All relationships with confidence above the confidence threshold derived using the false discovery rate and that are statistically significant at a specified p value based on the Chi square test are modeled as a directed graph (See Figure 3.9). Each vertex in the graph represents a family. The weight of an edge is the confidence of the relationship between families or vertices the edge connects. The direction of the edge denotes orientation of the relationship. For a relation F_1 U F_{20}, we have edge $F_1 \rightarrow F_{20}$ and edge $F_1 \leftarrow F_{20}$ for the reciprocal Downstream relationship. In order to discover the set of families that share spatial proximity relationships, we extract all connected components from the graph. Figure 3.9 shows a directed graph with two connected components.

In this work, we are reporting connected components as groups of repeat families that we find to be co-located consistently in terms of pairwise spatial relationships. Therefore we are assuming that all families of elements reported in a connected component are related transitively. A more typical approach in association rule mining is to identify sets of items (often more than two) that co-occur. We have not taken this approach because we want to identify related families of elements where there may be substantial diversity that has evolved over time. For example, in the case of repeats, it is not unusual for sections to be missing in a substantial number of the elements of a repeat family. Mining of relationships among pairs and then inferring further relationships based on transitivity within the graph will enable us to discover such relationships. This approach may cause some features to be incorrectly grouped, but these cases can be identified during the analysis of results. By not constraining a family in a connected component to have spatial relationships with every other family in the connected component, we can identify weakly conserved patterns among the families.

CHAPTER 4

EXPERIMENTS AND RESULTS

We have implemented the directed spatial association rule mining algorithm described in Chapter 3 in a tool called ProxMiner and applied it to the problem of *ab initio* identification of repeat regions in the rice genome. This chapter first outlines the experiments designed to test the effectiveness of ProxMiner for discovery of repeat families in the rice genome. We then describe the results for two test datasets — one consisting of only rice Chromosome 12 and the other consisting of the entire rice genome. We explore examples of new repeat families discovered by our tool and we compare the capabilities of ProxMiner to other algorithms addressing similar problems.

4.1 Problem definition

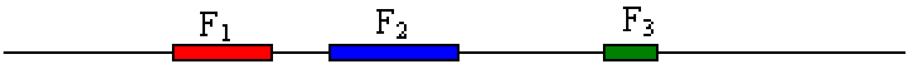


Figure 4.1

An illustration of a single diverged repeat composed of three fragments identified as separate *ab initio* repeat families (F_1, F_2 and F_3).

We have applied ProxMiner to the problem of *ab initio* identification of diverged repetitive regions in the genome to demonstrate the effectiveness of our hybrid spatial mining approach.

A DNA sequence is the one-dimensional substrate for these experiments. We will refer to all instances or *elements* of a repetitive sequence in DNA as a *repeat family*. The discovery of repeat families in newly sequenced genomes remains a challenging research problem despite the availability of a variety of computational repeat finders [44, 47, 48, 147]. Computational techniques are adept at recognizing elements where the sequence is identical or highly similar among all elements for a repeat family but are ineffective in capturing diverged elements. This is commonly the case with repeat regions [14, 15, 22]. Figure 4.1 shows a diverged repeat region for which three sub-sequences are recognized by an *ab initio* repeat finder as three separate repeat families instead of three pieces belonging to the same repeat element. The problem of determining how to reconstruct the longer repeat family from the small subsequences typically identified by computational repeat finders is called repeat defragmentation [49]. Our algorithm utilizes frequent spatial relationships among pairs of repeat families discovered by an *ab initio* repeat finder to mine for larger diverged repeat regions. The fact that we are mining the graphs for groups of transitively related *ab initio* repeat families makes our approach sensitive to weakly conserved repeat regions in the genome. In computer science terms, we can think of each repeat family as a class and the elements of this repeat found in the genome as instances of the class. Our algorithm finds frequently occurring relationships between instances of the classes and uses these relationships to construct a graph where connected components represent defragmented repeat families. In this case study, we used our ProxMiner algorithm to group repeat families identified by an *ab initio* repeat finder (i.e., RepeatScout [44]) and to discover fragmented repeat families where multiple repeat families reported by an *ab initio* repeat finder actually belong to a single large repeat family. For clarity, we will refer to the families identified by RepeatScout as *RepeatScout families*, and the longer families discovered by ProxMiner as *ProxMiner families*.

4.2 Experimental design and methods

We have used chromosomal DNA sequences from the *japonica* sub species of rice (*Oryza sativa*) to test our algorithm. Chromosome 12 from the rice genome [164] was used as an initial test dataset and the algorithm was subsequently tested with the complete rice genome (12 chromosomes). Chromosome 12 was selected for initial analysis because it has the highest percentage of repetitive DNA of rice chromosomes [43]. Genomic sequence was obtained from The Institute of Genomic Research website [164].

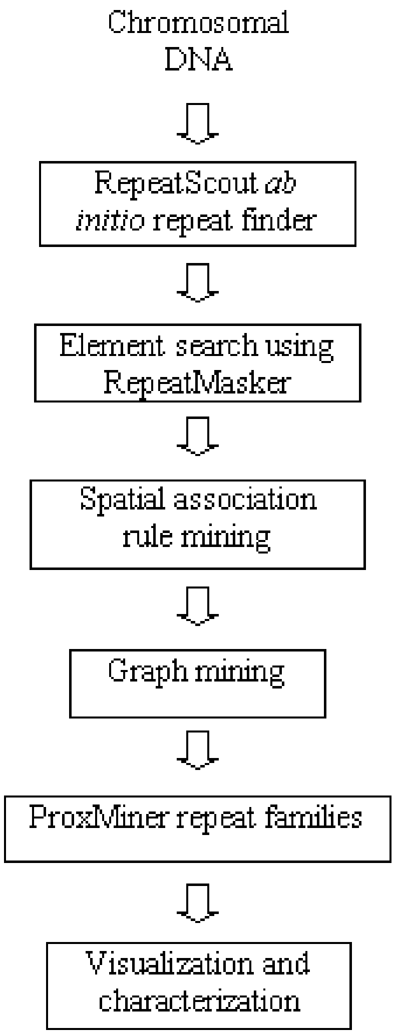

Chromosomal DNA
RepeatScout *ab initio* repeat finder
Element search using RepeatMasker
Spatial association rule mining
Graph mining
ProxMiner repeat families
Visualization and characterization

Figure 4.2

ProxMiner pipeline for repeat identification

The ProxMiner pipeline utilizes the *ab initio* computational repeat finder, RepeatScout [44] and a string matching engine, RepeatMasker [128] (Figure 4.2). The repeat families derived by RepeatScout from the DNA sequence are used as input in the spatial data mining process. Each repeat family found by RepeatScout is represented by a *consensus sequence*. We have conducted an empirical evaluation of *ab initio* repeat finders and have shown that these RepeatScout "repeat families" are often small fragments of larger repeat regions [15].

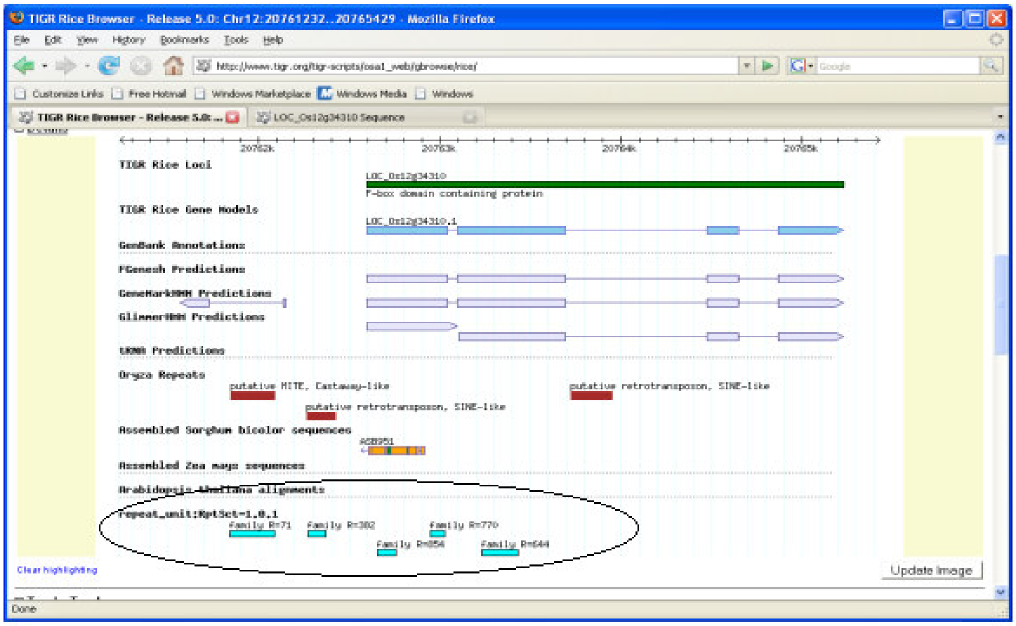

TIGR Rice Browser - Release 5.0: Chr12:20761232..20765429 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.tigr.org/tigr-scripts/osa1_web/gbrowse/rice/
Google
Customize Links
Free Hotmail
Windows Marketplace
Windows Media
Windows
TIGR Rice Browser - Release 5.0: ...
LOC_Os12g34310 Sequence
20762k
20763k
20764k
20765k
TIGR Rice Loci
LOC_Os12g34310
F-box domain containing protein
TIGR Rice Gene Models
LOC_Os12g34310.1
GenBank Annotations
FGenesh Predictions
GeneMarkHMM Predictions
GlimmerHMM Predictions
tRNA Predictions
Oryza Repeats
putative MITE, Castaway-like
putative retrotransposon, SINE-like
putative retrotransposon, SINE-like
Assembled Sorghum bicolor sequences
AGB951
Assembled Zea mays sequences
Arabidopsis thaliana alignments
repeat_unit:RptSct-1.0.1
Family R=71
Family R=302
Family R=770
Family R=854
Family R=644
Clear highlighting
Update Image
Done

Figure 4.3

A snapshot from the rice gbrowse genome browser showing elements of RepeatScout repeat families (in blue)

ProxMiner uses the RepeatMasker tool to locate the coordinates of all elements for each RepeatScout repeat family (Figure 4.3) on the DNA sequence. These coordinates are used as input for the spatial association rule miner that identifies upstream and downstream spatial relationships between RepeatScout families. A Monte Carlo procedure is then used to identify the confidence threshold that will give a specified false discovery rate (FDR). The significance of each spatial relationship meeting the confidence cutoff is further evaluated using the Chi square test. The spatial relationships with a confidence value sufficiently large to yield the specified FDR and significance level are then used to construct a directed weighted graph where vertices represent RepeatScout repeat families and the weights of edges represent the confidence of the association between the families. The connected components from this graph are extracted to provide sets of spatially related repeat families. Each set of related RepeatScout families is classified as a ProxMiner repeat family and represents a larger, diverged and fragmented repeat family.

4.2.1 Rice chromosomal DNA datasets

We report results of applying ProxMiner to Chromosome 12 and to the entire rice genome. Rice (*Oryza sativa*) is the most important food crop in the world with twelve chromosomes and a genome size of 389 Mb. The complete genome sequence for *Oryza sativa* (subspecies *japonica* cultivar Nipponbare) was reported in 2005 [43]. Rice is a model plant for the cereal species and also has a high repetitive content at 35% of the genome. Chromosome 12 has a length of 27.5 Mb and the highest repeat fraction at 38.3% [43].

4.2.2 Implementation

All experiments were performed on computers running Linux. The experiments for analyzing rice chromosome 12 were performed on a computer with dual 3.2 Ghz Xeon processors and 4 GB of memory while the experiments for rice whole genome were performed on a computer with four quad core 2.66 Ghz processors and 16 GB of memory. ProxMiner is implemented in Perl and utilizes the Graph module developed by Jarkko Hietaniemi [165] for graph mining procedures.

The various tools we utilize in our pipeline and for analysis of ProxMiner repeat families are listed below along with the parameters used:

- RepeatScout [44]: This *k*-mer based tool is used as the *ab initio* repeat finder for initial identification of repeat families. The *k*-mer lengths used for rice chromosome 12 and the whole genome were 13 and 16 respectively. They were calculated according to the formula log4(g) [44] where g is the length of the genome. The *k*-mer frequency threshold was 3 and the threshold for minimum number of elements in a repeat family was 10. All parameters were selected based on the recommendations of Price *et al.* [44].
- RepeatMasker [128]: We use RepeatMasker for identifying elements of all RepeatScout repeat families. The parameters used were search type as “sensitive” and species name as “Oryza sativa”. We use the –gccalc option to force RepeatMasker to compute the percentage of G’s and C’s in the input sequence in order to select the most appropriate scoring matrix for scoring alignments.
- ClustalW [162]: We use ClustalW with default parameters for all multiple sequence alignment (MSA) experiments.
- Jalview [166]: Multiple sequence alignments were visualized and analyzed with the Jalview alignment editor.

We annotate known repeats in a sequence using RepeatMasker with the Repbase [45] library of known repeats. When used with Repbase, RepeatMasker produces an output that lists the types and numbers of known repeats found.

4.3 Experimental Results

The first step in our pipeline (See Figure 4.2) applies the *ab initio* repeat finder, RepeatScout. We set the parameter for the minimum number of occurrences of a family to ten as recommended by Price *et al.* [44]. Table 4.1 shows that RepeatScout identifies 653 families in chromosome 12 and 6554 families in the entire genome.

Table 4.1

Summary of RepeatScout families for the two rice datasets

	Number of RepeatScout families	Average consensus sequence length (bp)	Number of elements of all RepeatScout families	Average length of elements for all RepeatScout families (bp)
Chromosome 12	653	865	31,536	409
Whole genome	6,554	717	593,946	302

RepeatMasker [128] was used to identify the locations of elements of the RepeatScout families on the respective datasets. We then applied our spatial data mining algorithm to identify the families whose elements are found to co-occur in significant numbers. We identified 107,244 spatial relationships among the 653 RepeatScout families on chromosome 12 and 3,407,936 spatial relationships among the 6,554 RepeatScout families for the entire genome.

Table 4.2

FDR values for rice chromosome 12 and whole genome derived using the algorithm described in Figure 3.8

Confidence threshold	Chromosome 12			Whole genome		
	Number of relationships found on Chr	Number of relationships found on SimChr	FDR	Number of relationships found on Chr	Number of relationships found on SimChr	FDR
0.05	57520	74255	129.09%	801767	912773	113.85%
0.1	16124	20503	127.16%	155343	152972	98.47%
0.15	5647	6301	111.58%	41835	26547	63.46%
0.2	2884	2394	83.01%	21180	8390	39.61%
0.25	1656	782	47.22%	12494	2118	16.95%
0.3	1197	288	24.06%	8950	686	7.66%
0.35	915	88	9.62%	6719	140	2.08%
0.4	757	41	5.42%	5567	60	1.08%
0.45	628	9	1.43%	*4487*	*10*	*0.22%*
0.5	*550*	*4*	*0.73%*	3919	6	0.15%
0.55	457	0	0.00%	3103	3	0.10%
0.6	398	0	0.00%	2694	3	0.11%
0.65	305	0	0.00%	2124	0	0.00%
0.7	261	0	0.00%	1794	0	0.00%
0.75	206	0	0.00%	1456	0	0.00%
0.8	159	0	0.00%	1156	0	0.00%
0.85	121	0	0.00%	817	0	0.00%
0.9	81	0	0.00%	655	0	0.00%
0.95	36	0	0.00%	342	0	0.00%
1	28	0	0.00%	282	0	0.00%

As described in Chapter 3, we used Monte Carlo simulation to compute false discovery rates (FDR) at varying confidence thresholds. The FDR is the ratio of the number of associations discovered from the simulated dataset to the number of rules found from the actual rice dataset (See Table 4.2). The results in Table 4.2 for both the chromosome 12 and whole genome show similar behavior although the FDR for the entire genome approaches zero at a lower confidence threshold. As the amount of sequence considered increases, the likelihood of random associations decreases. Both datasets show an unexpected behavior for very low threshold values. At low confidence threshold values, the number of spatial relationships identified in the randomly placed repeats is higher than for the real locations. We speculate that this is caused by the uniform distribution of elements on the simulated dataset as compared to the concentration of elements in some repeat rich regions in the actual dataset. The uniform distribution of elements on the simulated set results in a higher number of random spatial relationships when the confidence threshold is low. As the confidence threshold is raised, these relationships rapidly disappear while a substantial number of those in the actual data remain.

We set our target FDR to 1% and thus selected a confidence threshold of 0.5 or 50% for chromosome 12 (Table 4.2) and 0.45 for the whole rice genome. We chose confidence thresholds with estimated FDR below 1% in both cases in order to focus on the more conserved fragmented repeat families in the rice genome for our initial screening.

Spatial relations between RepeatScout families with confidence above the confidence threshold and statistically significant at p=0.05 were used to construct a directed weighted graph. The vertices in this graph represent RepeatScout families and the edges correspond to the relationship between a pair of RepeatScout families. The connected components from this graph are extracted to provide sets of spatially related RepeatScout families, each of which is grouped into a ProxMiner family.

Table 4.3

Comparison of the directed graphs constructed from the rice datasets

	Confidence threshold	Vertices	Edges
Chromosome 12	0.5	415	951
Whole genome	0.45	3592	7857

A directed graph was constructed using relationships from chromosome 12 with confidence above the 0.5 confidence threshold and with a p-value cutoff of 0.05 while, for the whole genome dataset, relationships with confidence above the 0.45 confidence threshold were used (See Table 4.3). The proportion of edges to vertices was similar in both the directed graphs.

Table 4.4

The size profile of ProxMiner families from the two rice datasets including singletons

RepeatScout families in a ProxMiner family	Chromosome 12	Whole genome
1	250	3067
2	60	463
3	21	191
4	11	99
5	8	57
6	2	27
7	2	14
8	3	15
9	0	12
10 or more	4	52

We extracted 112 connected components (ProxMiner families) of two or more RepeatScout families from the directed graph for chromosome 12 where each ProxMiner family consists of 3.59 components on average. Similarly, we extracted 930 connected components of two or more RepeatScout families representing ProxMiner families from the graph for the whole genome and found them to contain 3.74 components on average (See Table 4.4). Table 4.4 shows that the majority of ProxMiner families discovered on chromosome 12 (53%) and the rice whole genome (50%) consist of two RepeatScout families.

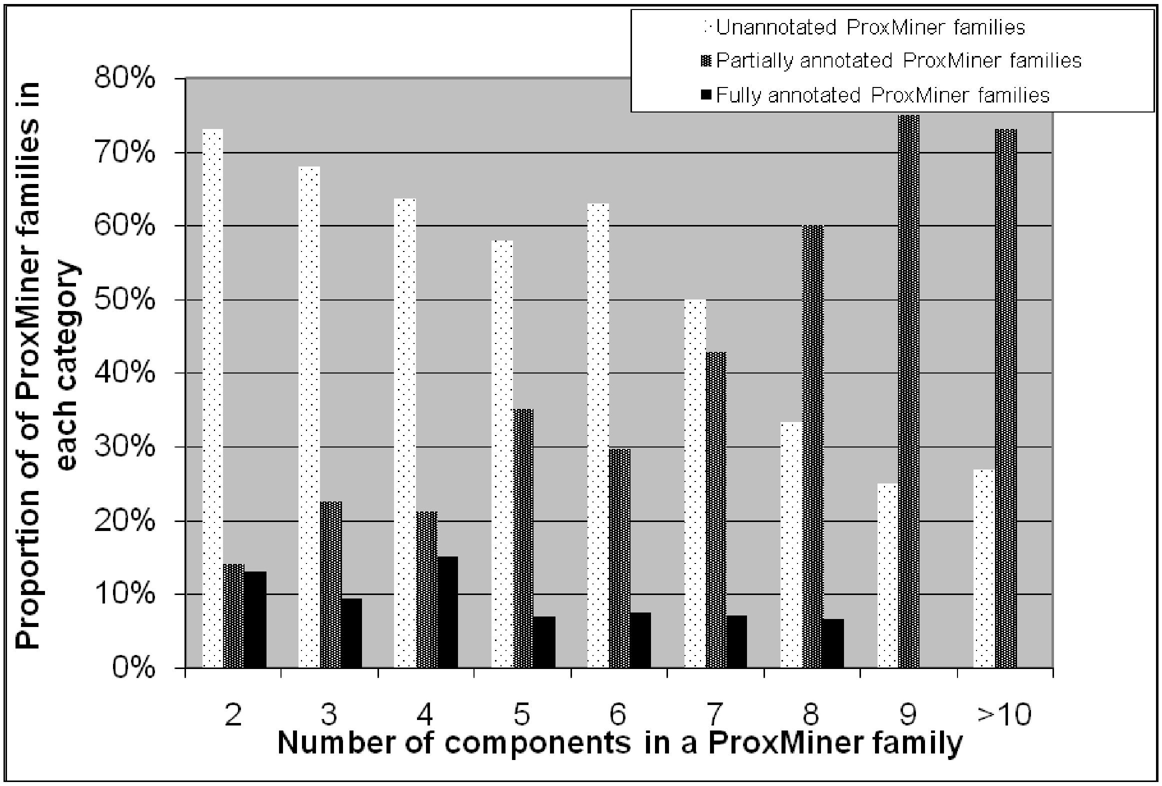


Figure 4.4

A comparison of the number of components in a ProxMiner family with its similarity to known repeat regions

Figure 4.4 shows the similarity of ProxMiner families to known repeats [45, 128] as a function of family size. Because the focus of our research is identifying fragments of families that can be combined, we do not include singletons (Table 4.4 and Table 4.5) representing RepeatScout families that were not associated with other families in our list of ProxMiner families. We divided the ProxMiner families into three categories: known, extended and novel repeat regions based on the number of components in a connected component that shared similarity with known repeats (See Table 4.5 and Appendix A). The proportion of ProxMiner families with no similarity to known repeats decreases as the number of components per ProxMiner family increases. These observations emphasize the effectiveness of ProxMiner for discovering short novel repeat families. The results in Table 4.5 show that ProxMiner condensed the 653 components from chromosome 12 to 362 ProxMiner families (112 connected components and 250 singletons) and the 6554 components from the entire genome to 3997 ProxMiner families giving reductions of 44% and 39% respectively. This provides a much more reasonable number of families for subsequent characterization and analysis by biologists. If we disregard singleton components shorter than 100 bp in length, the number of ProxMiner families for chromosome 12 is reduced to 333, a reduction of 49%. Therefore, we have shown that ProxMiner successfully merged fragmented repeat regions on chromosome 12 and the entire genome.

Table 4.5

Overview of results from the graph mining procedure of ProxMiner

	ProxMiner families	RepeatScout families that remained singletons	Novel ProxMiner families	Extended ProxMiner families	Known ProxMiner families
Chromosome 12	112	221	87	22	224
Whole genome	930	3067	2302	219	1046

We compared the initial families found by RepeatScout and the ProxMiner families identified by our algorithm against known repeats [45, 128] to study the defragmentation achieved for different classes of known repeats. Table 4.6 shows that the majority of matches found in both datasets were in two classes of known repeats (retrotransposons and DNA transposons).

Table 4.6

Presence of known repeats in the initial RepeatScout families, ProxMiner connected components of size 2 or greater and in the remaining singletons from rice chromosome 12 (Chr 12) and whole genome (WG)

Repeat Class	Initial RepeatScout families		ProxMiner families		Number of singleton RepeatScout families	
	Chr 12	WG	Chr 12	WG	Chr 12	WG
Retrotransposons						
SINE	17	76	3	4	2	5
LINE	9	132	1	18	3	50
Ty1/Copia LTR	47	207	12	27	11	71
Gypsy LTR	180	601	32	72	44	241
LTR	13	70	3	6	4	30
DNA Transposons						
En-Spm	55	280	6	33	10	79
MuDR	33	295	9	36	13	87
Tourist/Harbinger	29	300	5	27	17	134
TcMar-Stowaway	28	171	3	15	15	45
hAT-Ac	5	50	1	6	0	7
hAT	15	107	3	16	6	34
DNA	86	535	10	49	30	147
Others						
Helitron	0	42	0	6	0	6
Unclassified	3	26	1	3	1	2
Satellite	0	11	0	2	0	1
No known annotation						
Novel repeats	198	4672	23	610	64	1692

4.4 Evaluation of ProxMiner

Discovery and characterization of repetitive regions in DNA is a rapidly developing area of research [167-169]. We have demonstrated that the ProxMiner algorithm can be used to identify potentially novel diverged and fragmented repeat regions in the genome and that it is particularly useful for identifying smaller repetitive elements. In this section, we further demonstrate the utility of ProxMiner by providing a detailed analysis of selected repeat families identified by ProxMiner in the rice genome, compare the capabilities of ProxMiner to the capabilities of other repeat analysis tools that address similar problems, and an provide an evaluation by biologists who work in the area of repeat analysis.

4.4.1 Validation of results

First we demonstrated ProxMiner's effectiveness as a discovery algorithm through a detailed analysis conducted in collaboration with Dr. Zenaida Magbanua and Dr. Daniel Peterson—both biologists with expertise in the area of repeat analysis and discovery [14, 15, 170, 171]. We have selected some of the ProxMiner families discovered in the mining of chromosome 12 for further analysis in context of the entire rice genome. The connected components used for examples include two novel repeat regions (DR1 and DR2) and one extension of a known repeat region (DR3). We use DR as an abbreviation of Diverged Repeat. We worked with biologists to explore these ProxMiner repeats using a genome browser [164], the annotation information provided by ProxMiner, multiple sequence alignment, and molecular biology experiments (See Appendix C for experimental protocol).

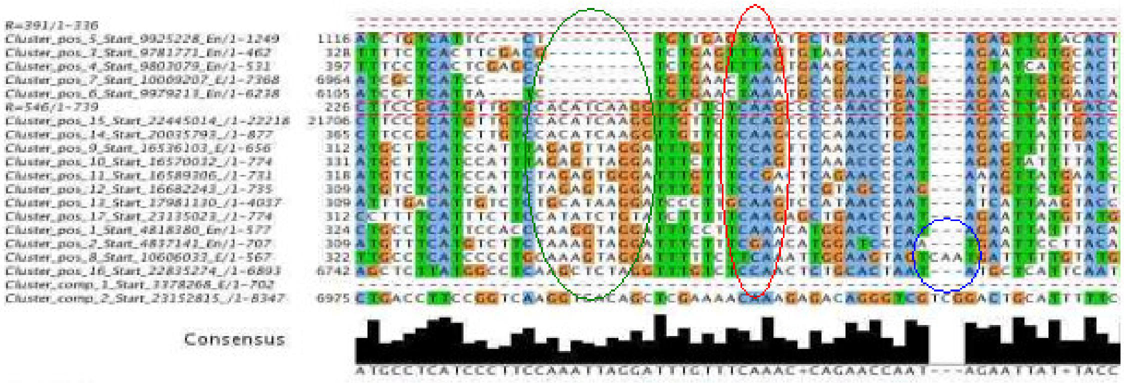

R=391/1-336
Cluster_pos_5_Start_9925228_En/1-1249 1116
Cluster_pos_3_Start_9781771_En/1-462 328
Cluster_pos_4_Start_9803079_En/1-531 397
Cluster_pos_7_Start_10009207_E/1-7368 6964
Cluster_pos_6_Start_9979213_En/1-6238 6105
R=546/1-739 226
Cluster_pos_15_Start_22445014_/1-22218 21706
Cluster_pos_14_Start_20035793_/1-877 365
Cluster_pos_9_Start_16536103_E/1-656 312
Cluster_pos_10_Start_16570032_/1-774 331
Cluster_pos_11_Start_16589306_/1-731 318
Cluster_pos_12_Start_16682243_/1-735 309
Cluster_pos_13_Start_17981130_/1-4037 309
Cluster_pos_17_Start_23135023_/1-774 312
Cluster_pos_1_Start_4818380_En/1-577 324
Cluster_pos_2_Start_4837141_En/1-707 309
Cluster_pos_8_Start_10606033_E/1-567 322
Cluster_pos_16_Start_22835274_/1-6893 6742
Cluster_comp_1_Start_3378268_E/1-702
Cluster_comp_2_Start_23152815_/1-8347 6975
Consensus
ATGCCTCATCCCTTCCAAATTAGGATTTGTTTCAAAC-CAGAACCAAT---AGAATTAT-TACC

Figure 4.5

A view of multiple sequence alignment of a set of DNA sequences

A genome browser [164] is a web based visualization of a genome and associated annotations. Figure 4.3 shows a screen shot of a portion of the rice genome displayed in the GBrowse browser. GBrowse [172] is a popular open source genome browser developed by the Generic Model Organism Database project [173-176] to enable biologists to compile, share and visualize features on a genome. We have deployed a local GBrowse for rice that displays the rice DNA sequence along with the locations of different features that have been identified on the genome sequence by the rice research community including known genes, gene predictions, experimental data, genetic markers, DNA alignments, protein alignments, etc. Researchers can also upload new annotations in standard formats that are displayed as tracks for viewing on the browser. We stored

ProxMiner repeats in GFF format [177] and imported custom tracks into GBrowse to enable biologists to explore the genomic context of the ProxMiner repeats.

Multiple sequence alignment is a commonly used method for finding regions of similarity among a set of DNA or protein sequences. We used the ClustalW [162] tool to align a set of DNA sequences representing instances of ProxMiner repeat families on the rice genome. We then visualized the alignment with Jalview [166] to analyze the DNA sequences for regions where identity is conserved (e.g., Figure 4.5). In the figure, a highly conserved section is demarcated by a red circle and a poorly conserved section is enclosed within a green circle. An insertion of CAA in one sequence is shown by a blue circle.

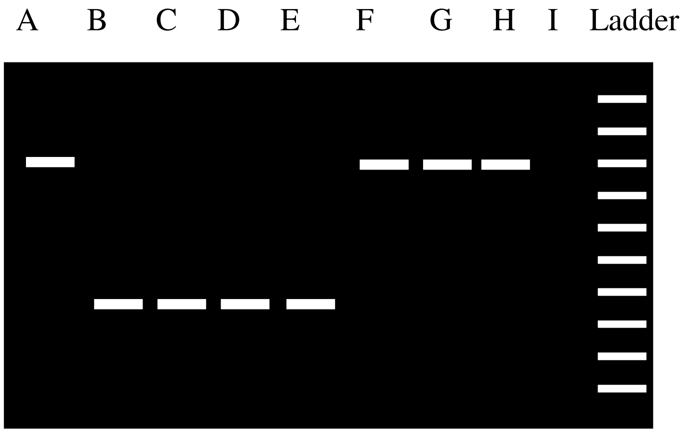


Figure 4.6

Schematic of results of a PCR experiment

We have also validated the predictions produced by our algorithm using molecular biology experiments. Researchers have developed a number of methods to analyze repeat regions in DNA in the laboratory [178-180]. One of the most widely used methods is the polymerase chain reaction (PCR) that allows one to determine the abundance of a DNA sequence in the genome based on the presence of short conserved regions of the sequence and also provides information about the relative size of the DNA region found. See Figure 4.6 for an example of an output of a PCR experiment. The presence of a band in a column confirms the presence of the target sequence in the DNA sample loaded in that column. The position of the band is used to estimate the sample size by comparing the bands produced by a sample of DNA to segments of known size called a ladder (last column in Figure 4.6). A ladder is used to calibrate the lengths of DNA sequences produced by PCR. A more complete description of PCR and its use for studying repeats is given in Appendix C [181]. One of our collaborators, Dr. David Ray has used PCR to describe DNA transposon activity in bats [182, 183] and to analyze a novel repeat region in crocodiles [184]. Another of our collaborators, Dr. Cedric Feschotte has also utilized PCR experiments to examine the presence or absence of SPIN transposons in mammals [185] and in a study of enzymes related to repeats in flowering plants [186]. The regions of DNA produced using PCR are often subsequently extracted and sequenced [183] for comparison with other sequenced genomes.

We used PCR to verify the presence of the ProxMiner families in DNA extracted from several rice species. We used the genome sequence from *Oryza sativa* (Nipponbare, japonica cultivar group) for our experimental dataset [164]. We refer to this species as the *reference species*. We selected a set of related rice species to explore for the presence of the families we have discovered. Dr. Susan McCouch at Cornell University graciously supplied the DNA for these experiments. These species are *Oryza sativa* (indica), *Oryza glaberrima*, *Oryza nivara*, *Oryza rufipogon* and *Oryza officinalis*. *Oryza sativa* (Nipponbare, japonica cultivar group) and *Oryza sativa* (indica) are the domesticated species of rice grown in Asia as a food source. *Oryza rufipogon* and *Oryza nivara* are wild relatives of rice from which the domesticated *Oryza sativa* is thought to have evolved [187]. *Oryza glaberrima* is a cultivated variety from Africa [188]. *Oryza officinalis* is a wild variety of rice from China with genes of economic importance such as insect resistance and high water stress tolerance [189]. We have selected a diverse set of species from the *Oryza* genus to estimate the conservation of our ProxMiner families across the whole genus. The goal was to confirm that the repeat families identified by ProxMiner exist in various rice species, to find the size of the repeat families in those species and to deduce their evolutionary history, if possible.

Table 4.7

Details about component families from rice chromosome 12 (Chr 12) and the whole genome (WG) for DR1, DR2 and DR3 ProxMiner families

ProxMiner family	Component	Number of elements		Average length (bp)		Consensus sequence length (bp)
		Chr 12	WG	Chr 12	WG	
DR1	R546	28	158	522	494	739
	R391	33	221	275	269	336
DR2	R398	47	372	73	74	82
	R956	44	528	86	81	114
	R418	32	338	56	55	60
DR3	R970	30	410	109	140	128 (91.4% annotated as SETARIA1)
	R293	25	442	169	109	244

Table 4.8

Details about spatial relationships from rice chromosome 12 (Chr 12) and the whole genome (WG) for DR1, DR2 and DR3 ProxMiner families

ProxMiner family	Relationship	Confidence		Average distance (bp)		Standard deviation in distance (bp)	
		Chr 12	WG	Chr 12	WG	Chr 12	WG
DR1	R391 upstream R546	0.89	0.91	128	-54	1290.9	1027.53
DR2	R956 upstream R398	0.86	0.88	-20	18	55.46	583.72
	R418 upstream R398	0.81	0.73	168	239	113.82	973.65
	R418 upstream R956	0.75	0.80	109	159	103.46	905.26
DR3	R970 upstream R293	0.76	0.61	688	1183	442.11	1769.45

4.4.1.1 DR1 repeat family

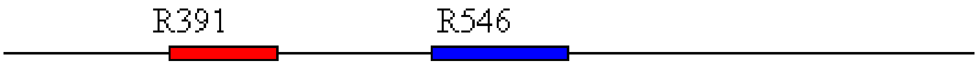


Figure 4.7

Illustration of DR1 composed of two components (R391 and R546)

The DR1 novel repeat family (Figure 4.7) consists of two conserved sections represented by two RepeatScout repeats (R391, R546). The lengths of consensus sequences for R391 and R546 are 336 bp and 749 bp respectively. On rice chromosome

12, the average distance between the elements of the two families is 128 bp as compared to -54 bp for the whole genome analysis (See Table 4.7 and Table 4.8). The negative distance indicates an overlap spatial relationship. The overlap and high confidence (0.91) of the relationship shows that elements of the two RepeatScout families are consistently co-located in close proximity.

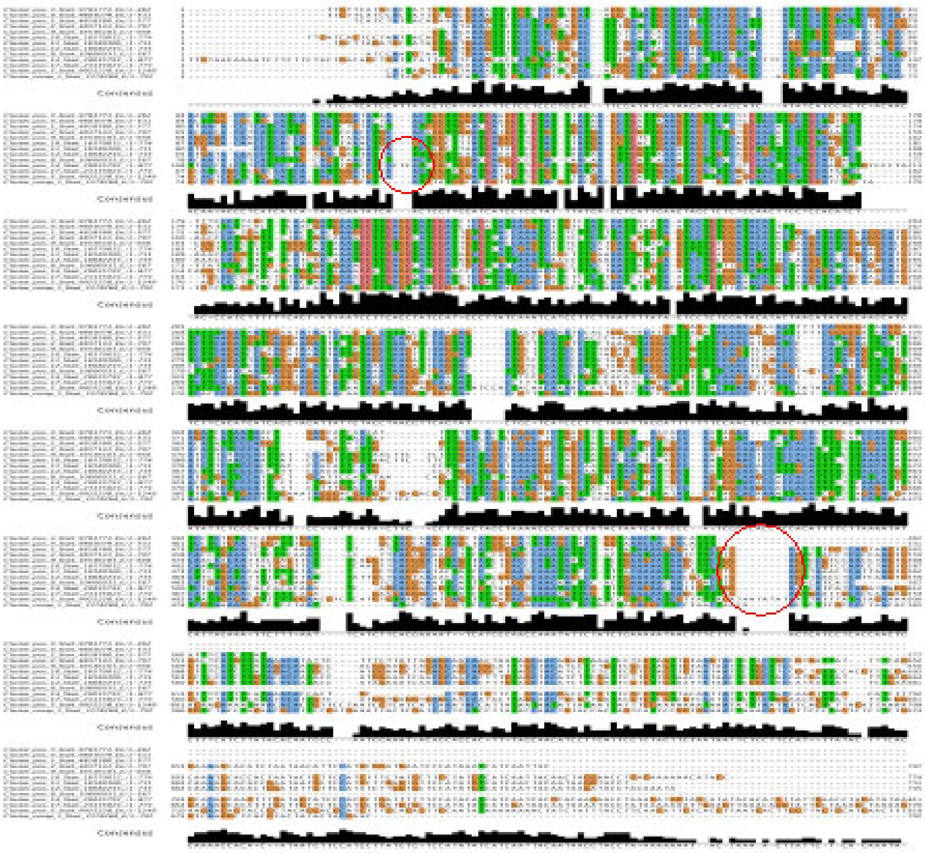


Figure 4.8

Multiple sequence alignment of instances of DR1 repeat family on rice chromosome 12. Insertions/deletions are in red circles

Both the repeats have no similarity with known repeat regions [45, 128]. The multiple sequence alignment of instances of DR1 on rice chromosome 12 shows good overall conservation with clear insertions/deletions in some regions (Figure 4.8). The multiple sequence alignment confirms that we have discovered a fragmented repeat family for which two separate repeats were reported by the ab initio repeat finder RepeatScout. Exploration of instances of DR1 on the rice genome browser (Figure 4.9) reveals that it is found in disease resistance genes and NBS-LRR proteins [190] for 15 of the 19 instances on rice chromosome 12. The GBrowse snapshot (Figure 4.9) reports the disease resistance gene (in green) as expressed, which implies that experimental evidence has been found for a protein corresponding to this gene. A number of genes near or overlapping with the ProxMiner family are expressed. The presence of a protein for a gene denotes that the gene is active in the organism and this makes our discovery more interesting for biologists.

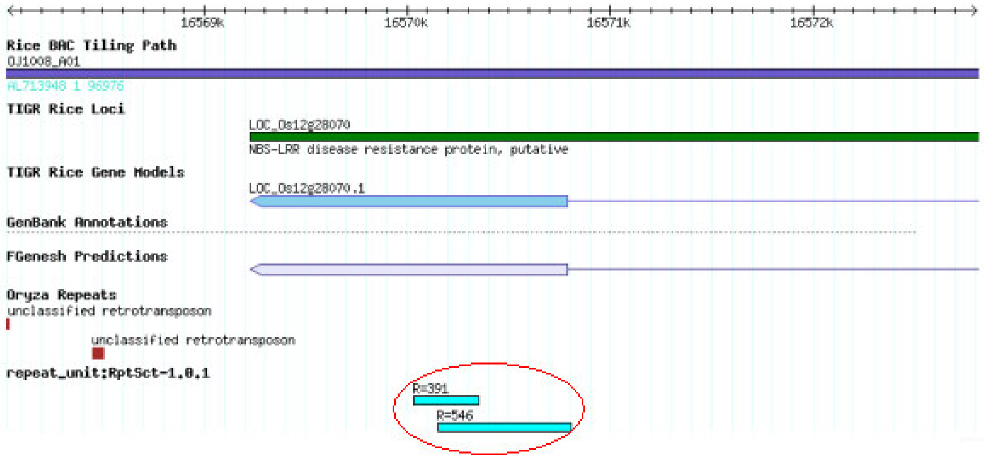


Figure 4.9

An instance of DR1 on rice chromosome 12 in the rice genome browser

We used PCR to study the instances of DR1 repeat family in a number of species in the rice genus. The objective here was to verify the presence of DR1 in rice DNA and to identify the lengths of DR1 in the different species.

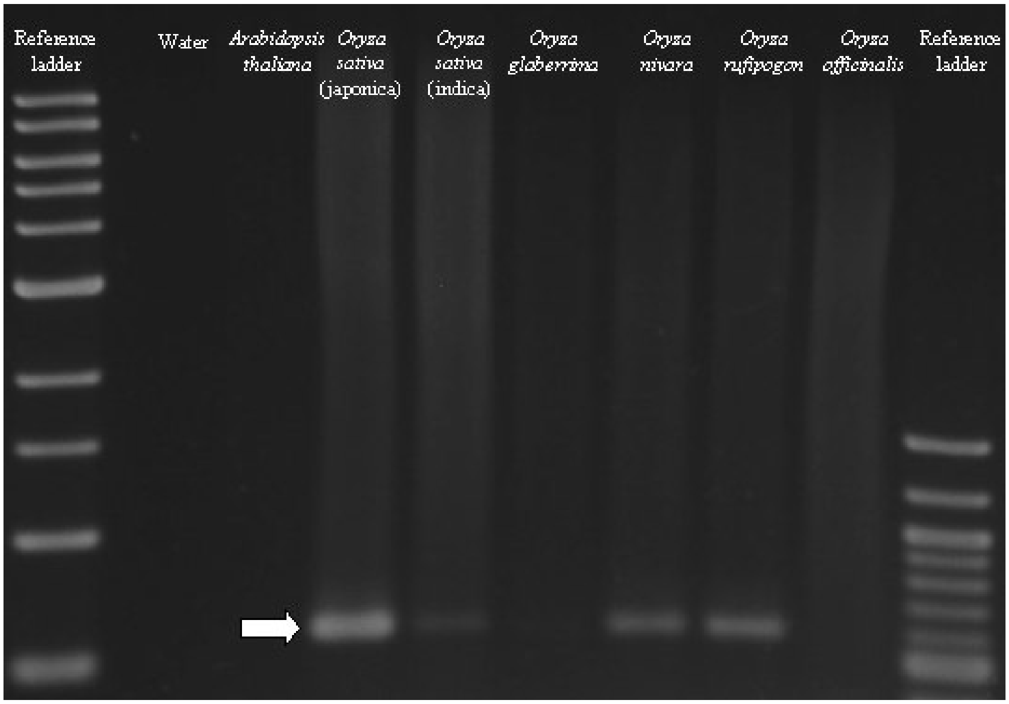


Figure 4.10

PCR results for DR1 repeat family in various rice species

The PCR results show that DR1 repeat family was found in the reference genome *Oryza sativa* (japonica) and also in *Oryza sativa* (indica), *Oryza nivara* and *Oryza rufipogon*. It is absent in *Oryza glaberrima*, a cultivated species from Africa and in *Oryza officinalis,* a wild species from China. The product from the experiment is of the expected length at 600 bp. Moreover, the length of the product was the same in all the species where it was found indicating conservation of this repeat family in *Oryza* genera. The

PCR results (Figure 4.10) show that the family is conserved in the wild relatives (*Oryza nivara* and *Oryza rufipogon*) as well as domesticated species of rice (*Oryza sativa* (japonica) and *Oryza sativa* (indica)) from Asia.

The DR1 repeat family is found in nucleotide binding site leucine rich repeat (NBS-LRR) [191] disease resistance genes. The NBS-LRR proteins mediate interaction among various proteins and also have protein activation roles [191]. Disease resistance genes have been reported to exist in multiple inexact copies in plant genomes [192]. This diversity is maintained so that the plant can tap new sources of resistance to respond to pathogens that have developed tolerance to current resistance genes. In conclusion, we hypothesize that we have discovered a novel repeat family that may be active and play a role in the disease resistance mechanism of rice [193, 194].

4.4.1.2 DR2 repeat family

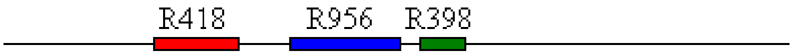


Figure 4.11

Illustration of DR2 composed of three components (R418, R956 and R398)

The DR2 novel repeat family (Figure 4.11) consists of three repeats (R418, R956 and R398) reported by RepeatScout and the lengths of consensus sequences for all components are relatively short at 60 bp, 114 bp and 82 bp respectively. The DR2 repeat family is an example of a novel fragmented repeat comprised of very short conserved sections. This ProxMiner family was constructed from three spatial relationships (see Table 4.7 and Table 4.8). The average distance between the three is less than 168 bp on chromosome 12 and less than 239 bp in the whole genome, denoting that elements are found very close to one another. For R956 and R398, we also observe an overlap spatial relationship on chromosome 12.

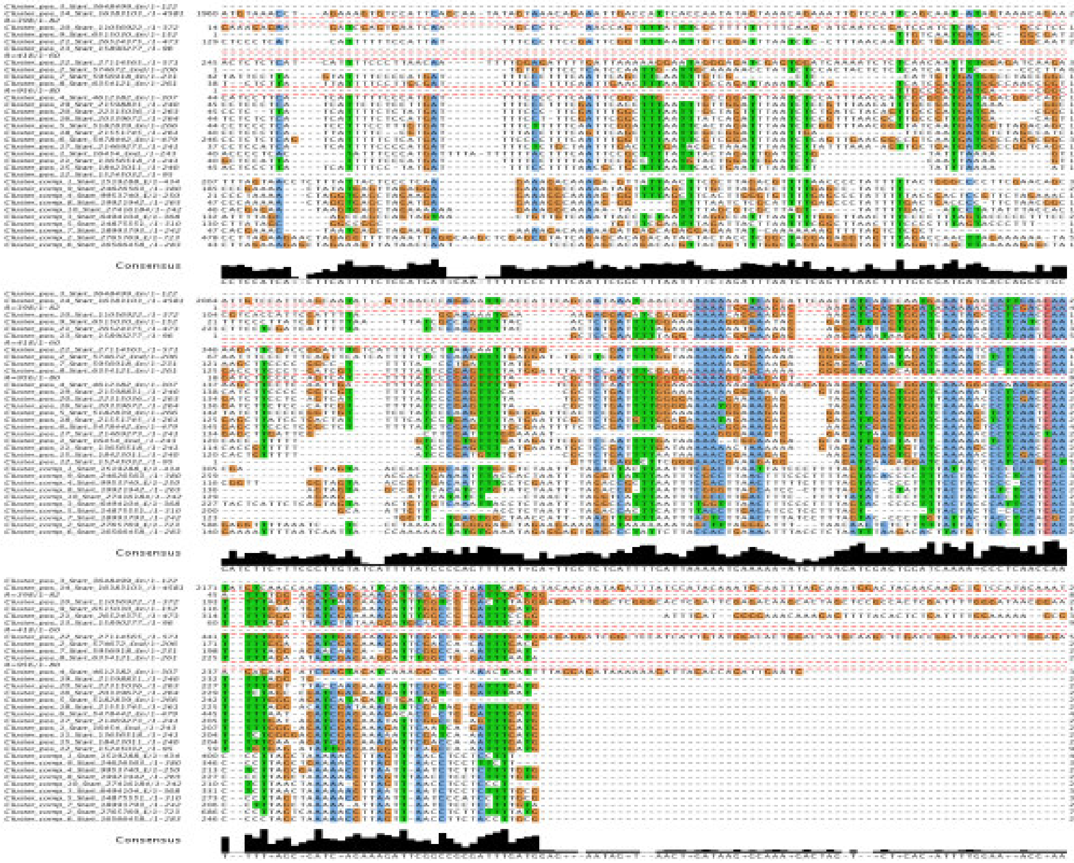


Figure 4.12

Multiple sequence alignment of instances of DR2 repeat family on rice chromosome 12

The multiple sequence alignment of instances of DR2 on rice chromosome 12 shows weak conservation overall (Figure 4.12). The lengths of the instances vary over a wide range from 123 bp to 4582 bp. We found a majority of instances of DR2 to exist in the intron regions of predicted genes using the rice genome browser (Figure 4.13). The environment around the instances was enriched in repeats. A number of instances were also located upstream of genes, some of which are expressed and are involved in responding to external stimuli [195] or disease resistance [164].

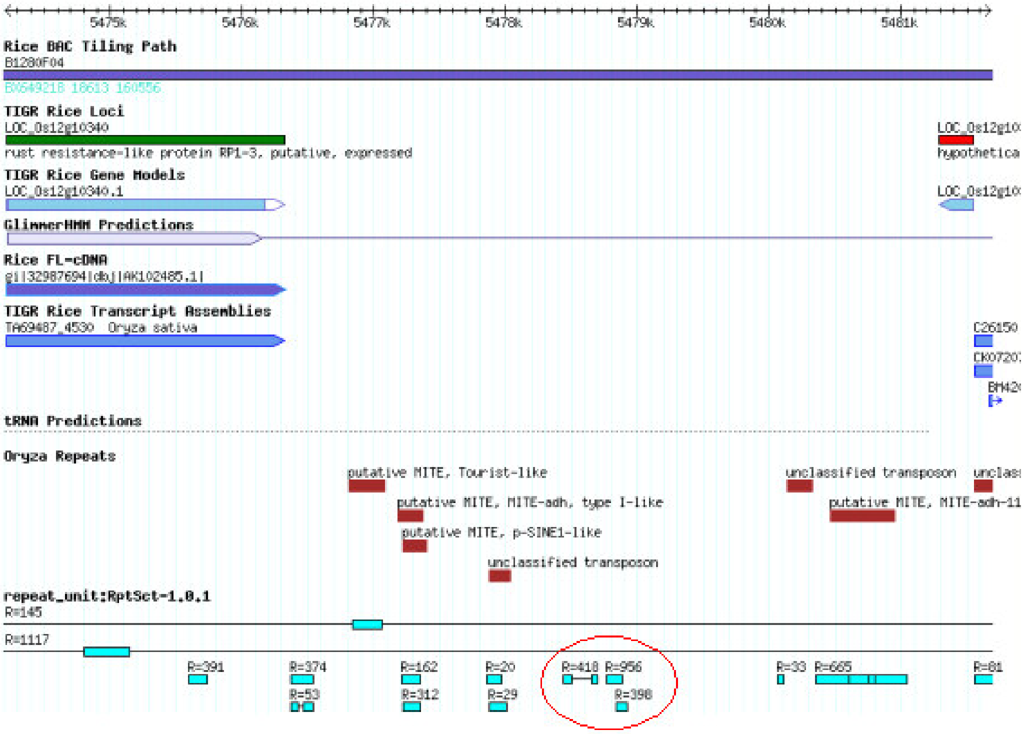


Figure 4.13

An instance of DR2 on rice chromosome 12 located within an intron and near a disease resistance gene

The PCR results are similar to those for the DR1 repeat family. We find that the DR2 repeat family exists in the reference genome *Oryza sativa* (japonica) and in *Oryza sativa* (indica), *Oryza nivara* and *Oryza rufipogon*. The products from the experiment are of expected lengths (1000 bp). The length of elements is also found to be conserved across all the species where the family is observed.

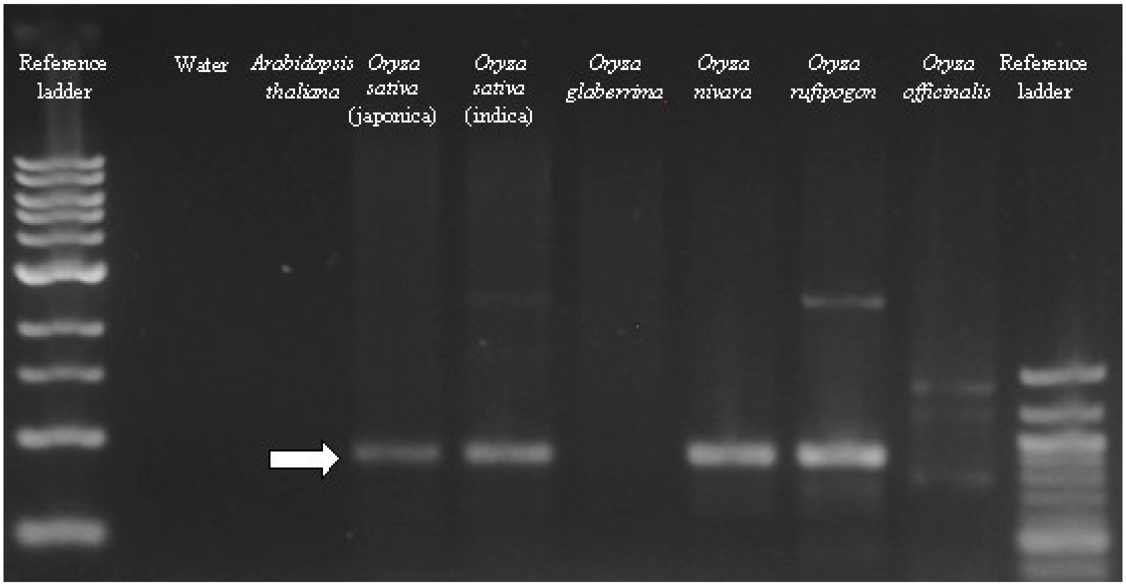


Figure 4.14

PCR results for DR2 repeat family in various rice species

The DR2 ProxMiner repeat family with its short conserved sections represents a previously unknown ancient repeat region in the rice genome. The PCR results show the conservation of this repeat family in wild and domesticated species of rice from Asia. Its conservation across *Oryza* genera, presence in repeat rich areas and in introns of predicted genes leads us to hypothesize that DR2 may be an inactive novel repeat region in Asian rice species.

4.4.1.3 DR3 repeat family

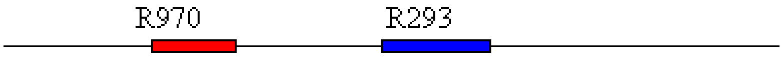


Figure 4.15

Illustration of DR3 composed of two components (R970 and R293). R970 is similar to the known repeat SETARIA1 in rice

We report the DR3 repeat family discovered by ProxMiner as an extension to a known repeat region. SETARIA1 [196] is a non-autonomous Mutator-like [197] DNA transposon found in rice. DR3 is made up of components R970 and R293 (See Table 4.7 and Table 4.8). The consensus sequence for R970 has a length of 128 bp and shares 91% similarity with SETARIA1. R293 with a consensus sequence of 169 bp is found downstream of R970 and does not have any similarity to SETARIA1 or any other known repeat. The SETARIA1 DNA transposon has a length of 1293 bp [196].

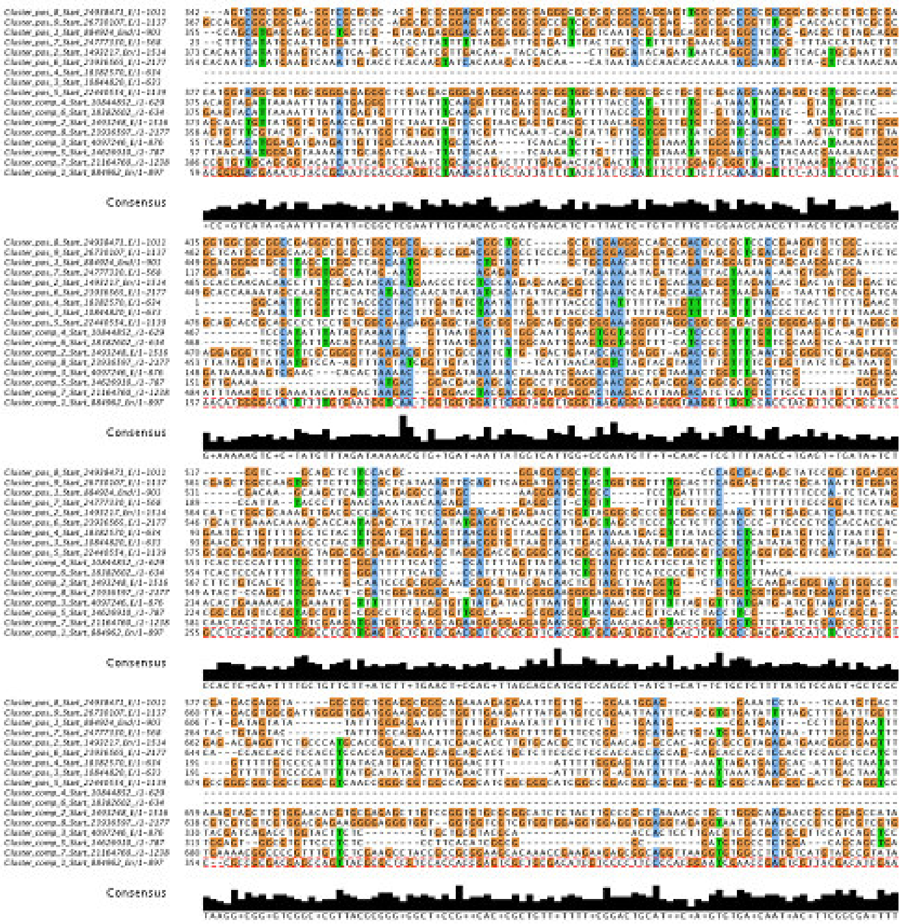


Figure 4.16

Multiple sequence alignment of instances of DR3 on chromosome 12 dataset

The multiple sequence alignment of instances of DR3 on rice chromosome 12 shows weak conservation (Figure 4.16). The lengths of the instances vary over a wide range from 569 bp to 2178 bp.

Exploration of instances on the rice genome browser reveals that a majority of the instances consist of consecutive overlapping elements of R970 and R293 (Figure 4.17).

We also find instances to exist in introns of predicted genes or in protein rich regions with proteins annotated for stress response such as MLA1 [164, 195].

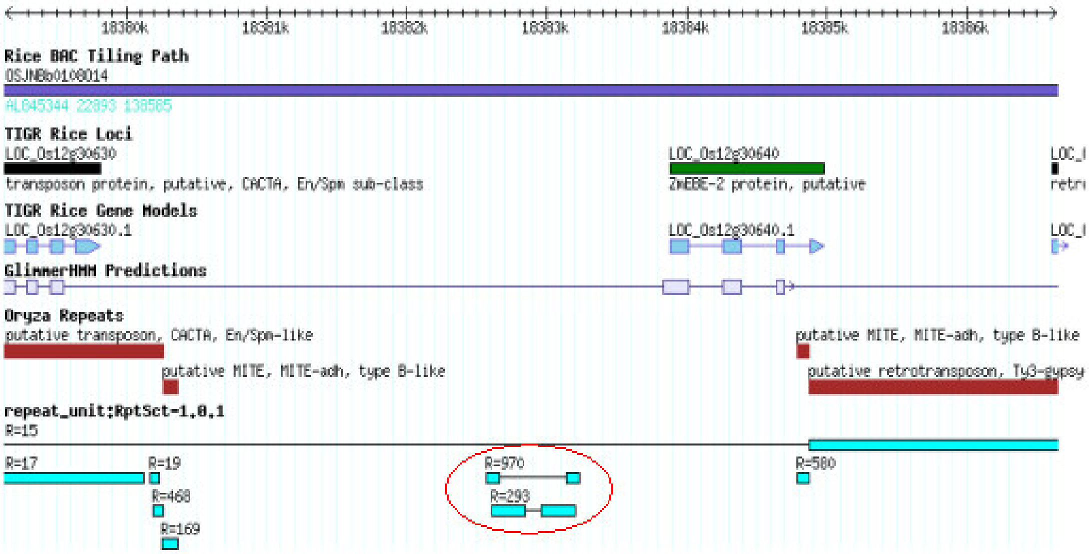


Figure 4.17

An instance of DR3 on rice chromosome 12 on the genome browser

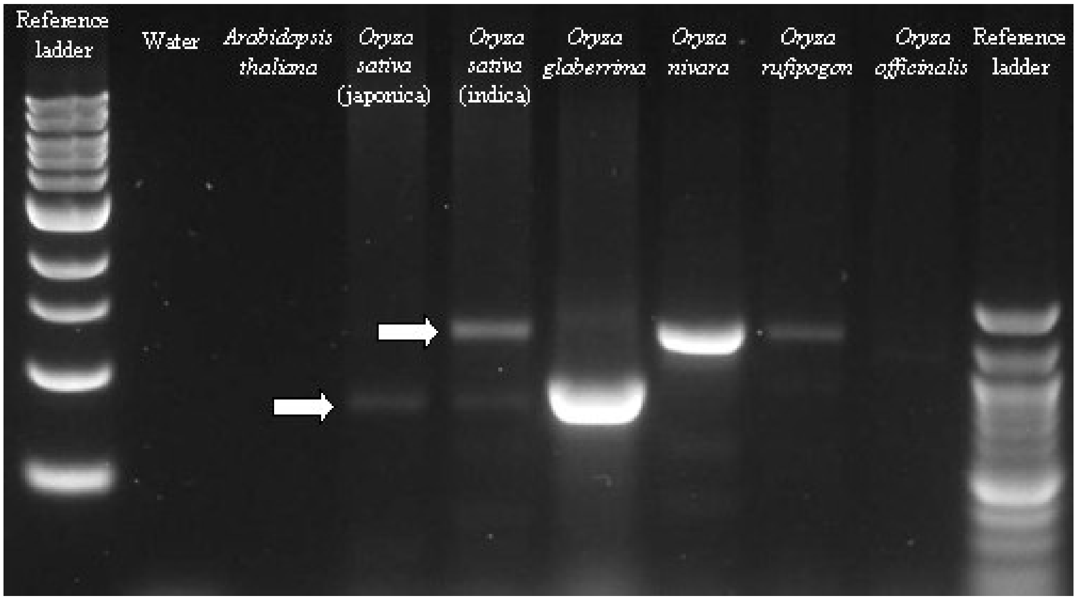


Figure 4.18

PCR results for DR3 repeat family in various rice species

The PCR results for DR3 repeat family suggests that although the repeat family exists in the reference genome *Oryza sativa* (japonica), it has two forms in *Oryza sativa* (indica). One of these forms is conserved in both *Oryza nivara* and *Oryza rufipogon*, both wild species from Asia, with comparatively more elements present in *Oryza nivara*, a wild species from Africa. Interestingly, the other form is present in large numbers in *Oryza glaberrima*, a cultivated variety from Africa. Both forms are absent from *Oryza officinalis*, a wild species from China. Our reference genome, *Oryza sativa* (japonica), contains the form whose size is consistent with the elements of this repeat family (900 bp). The discovery of DR3 repeat demonstrates how ProxMiner can extend knowledge of known repeat regions. The presence of multiple forms in the *Oryza* genus and low conservation in the multiple sequence alignment of instances suggest that the region is under low evolutionary pressure.

In conclusion, we hypothesize that the DR3 repeat family discovered by ProxMiner is a variant of the SETARIA1 DNA transposon [196] where R293 in an insertion in the original SETARIA1 repeat.

4.4.2 Comparison to defragmentation tools

In the literature review (Chapter 2), we described some algorithms that use spatial proximity knowledge to discover fragmented repeat regions in DNA. We now compare our ProxMiner algorithm with other available defragmentation algorithms in terms of their utility for defragmenting and discovering novel repeats (Table 4.9).

Table 4.9

Comparison of capabilities of available defragmentation algorithms with ProxMiner

	TCF [49]	LTR_Miner [140]	TEnest [141]	REannotate [142]	ProxMiner
Uses standard input format	+	+AR only	-	+	+
Can be used to analyze known repeats	+RT only	+AR only, RT only	+RT only	+RT only	+
Can be used to analyze *ab initio* repeat families	-	-	-	-	+
Visualization of discovered repeat families	+	-	+	+	+
Defragmentation of individual repeats	+	+AR only	+	+	+
Defragmentation of repeat families	-	-	-	-	+

(AR stands for Arabidopsis thaliana, RT stands for retrotransposons)

Transposon Cluster Finder (TCF) was designed by Giordano *et al.* [49] to defragment known repeats in DNA and compute their chronological order based on the occurrence of nesting of one repeat within another. TCF is designed to defragment various known repeats including LTR retrotransposons, DNA transposons, LINEs and SINEs [39, 169, 198, 199]. LTR_MINER by Pereira [140] was specifically designed to defragment LTR retrotransposons in the five chromosomes of *Arabidopsis thaliana*. The TEnest algorithm by Kronmiller *et al*. [141] and REannotate by Pereira [142] are both designed to defragment, compute the chronological ordering of LTR retrotransposons and visualize nested repeats given a database of repeats. The predominant focus for these tools is to understand a known repeat and identify its interactions with other known repeats from the arrangement of these repeats on the genome.

All aforementioned tools except TEnest accept repeat families as input in a standard and widely used input format. These defragmentation algorithms are designed to specifically work with known repeat regions such as LTR retrotransposons. They use spatial heuristics derived from sequence characteristics of known repeat regions to identify and attempt to defragment known repeat regions. All the tools describe their use only with known repeats but, with the exception of TEnest, they can be adapted to defragment repeat families discovered by an *ab initio* repeat finder such as RepeatScout. The input files required by TEnest are particular to the LTR retrotransposons class of known repeats. ProxMiner can be used to analyze both known repeats, *ab initio* repeat families and a combination of the two. Defragmenting known repeats along with *ab initio* repeat families is a useful method to extend the knowledge about known repeat regions as we have demonstrated with the DR3 repeat family. Please note that LTR_Miner can only be adapted to work with known and *ab initio* repeats for the *Arabidopsis thaliana* genome, unlike other tools that can be applied to any genome of interest. All the algorithms discussed here have a visualization component except LTR_Miner. Like ProxMiner, TCF and REannotate use genome browsers [164] to display the defragmented repeats. TEnest includes a visualization engine called svg_ltr that generates graphical displays for the repeats defragmented by TEnest. The visualization provided by TEnest does not include information about other features on the genome such as genes that are available on a genome browser. Table 4.10 shows the rich set of information produced by ProxMiner to assist the biologist in exploring and characterizing the ProxMiner repeats. These include files for sequence analysis and visualization on GBrowse[172], a genome browser. The other defragmentation algorithms are designed to address only known repeats and therefore do not provide any information to help characterize the defragmented repeats.

Table 4.10

Overview of information produced by ProxMiner about the repeat families it discovers

Information type	Description
ProxMiner families	Information about all connected components including member RepeatScout families and details of any similarity to known repeats. Information about spatial relationships and relevant properties like average distance is also included.
Statistics	Information about of connected component sizes. Summarization of similarities of ProxMiner families and remaining RepeatScout singletons to known repeats.
DNA sequences	Instances of the ProxMiner family on the genome including flanking sequences. The flanking sequences are useful for biologists analyzing ProxMiner results.
HTML genome browser file	An interface HTML file that links out to an online genome browser, GBrowse [172].

The available defragmentation algorithms have not been designed specifically for defragmentation of repeats but use defragmentation as a means to discover the nesting order and chronology of known repeat regions. The focus on known classes of repeats decreases their utility when modified to discover novel repeat regions. This drawback is exacerbated when they are applied for repeat discovery in newly sequenced genomes for which very few or no known repeats have been reported. When supplied *ab initio* repeat families, all available defragmentation algorithms presume that each input repeat family represents a true and complete repeat region and therefore, do not look for associations among the repeat families. Available methods analyze and defragment each given repeat instance in isolation instead of trying to discover a higher level grouping among repeat families. Using the classes and objects analogy, the available methods only find relationships between instances and not relationships between the classes themselves.

We used the 653 RepeatScout families from chromosome 12 as input to the available defragmentation tools to evaluate their effectiveness for *ab initio* repeat families and compared their results to the results from ProxMiner (when possible). The first tool, TCF, produces a list of regions on chromosome 12 where elements of a RepeatScout family exist within a 500 bp range. In essence, the output is a list of the repeat rich regions on the chromosome. The list of 470 clusters produced is not directly comparable to the output from ProxMiner since TCF discovers relationships among individual instances while ProxMiner discovers associations among component repeats that occur across the genome. This demonstrates that TCF is not effective in defragmenting *ab initio* repeat families. We were not able to conduct a direct comparison of ProxMiner with the second and third tools--LTR_MINER and TEnest. LTR_MINER cannot be applied to genomes other than *Arabidopsis*. TEnest requires input files specific to a single class of known repeats--LTR retrotransposons. Even when TEnest is given a database of LTR retrotransposons, TEnest only produces a list of coordinates of defragmented elements along with the nesting level but no information about which LTR retrotransposons are consistently co-located. Given the 653 repeat families generated from rice chromosome 12, the fourth tool, REannotate, produces a list of defragmented elements for each family. Like other defragmentation algorithms, it defragments each *ab initio* repeat family in isolation and therefore, is not effective for identifying associations among repeat families that ProxMiner is able to discover.

4.4.3 User feedback from biologists

We also worked with our end-users, molecular biologists in the Department of Plant and Soil Sciences at Mississippi State University, to evaluate the utility of our algorithm and to validate the predictions. The biologists were given a short introduction to ProxMiner and its output files and then asked to analyze and characterize a set of ProxMiner repeat families. Each was then asked to fill out a survey form and their feedback has been included as Appendix B in this dissertation and summarized below.

The primary method available to biologists for analysis of repeat families generated by *ab initio* repeat finders is to use BLAST [130] based systems [200] to find the elements of the families. The locations of these elements can then be used to discover families that co-occur for subsequent analysis. However, the large number of families makes this approach very time consuming and labor intensive. The results produced by our algorithm provide biologists with a condensed summary of the repeat families that co-occur frequently. We also include information about spatial distribution of the families, visualization of the repeat regions and any similarity to known repeats (Table 4.10). This enables biologists to analyze each data point and perform association studies for repeats, an important area of research that has not been addressed by any of the previously available defragmentation algorithms. Our algorithm will also enable spatial analysis of other features on DNA such as genes and regulatory elements.

A user guide and more user-friendly interface to all the information produced was requested in the feedback process (Appendix B) to enable a biologist to easily interpret results. We will be working with biologists to develop an improved interface before this work is submitted for publication.

4.5 Conclusions

In this chapter, we have demonstrated that our approach of using spatial data mining followed by graph mining can be successfully applied to identify weakly conserved patterns among features in genomic data. The application of pairwise spatial relationships increased the sensitivity of our algorithm while the use of a confidence threshold based on false discovery rate reduced the noise in our results.

Using a case study of *ab initio* repeat finding, we have shown that our algorithm is able to condense the number of repeat families discovered by an *ab initio* repeat finder for subsequent analysis by biologists by grouping 6554 RepeatScout families from the rice genome into 3997 ProxMiner families. Available defragmentation algorithms only defragment elements belonging to a single repeat family while ProxMiner discovers associations among repeat families to identify larger more complete repeat families. We have reported and analyzed two examples of novel repeat families and one example of a known repeat family that is extended by ProxMiner.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The central focus of this dissertation is the mining of one dimensional genomic sequence data for highly degenerate patterns using a combination of association rule mining and graph mining. We show that our approach can be utilized effectively to address the problem of defragmenting repetitive DNA for the discovery of novel diverged repeat regions.

Our interest in repetitive regions in DNA was the motivation for conducting a comprehensive survey of available repeat finding algorithms [14]. We also performed an empirical comparison of a subset of these algorithms [15]. The results of the empirical analysis of a set of *ab initio* repeat finders revealed that a majority of the novel repeats found were short repeat families that are actually fragments of larger families. There was a large variation in the sensitivity and specificity of the repeat finders we evaluated. Some of the weaknesses we discovered in *ab initio* repeat finders included underestimation or overestimation of repeat content, short consensus sequences and high memory requirement. One study reports a manual process for defragmentation of repeats in the pig genome [201], but there are no *ab initio* repeat finders that defragment the novel repeat families they discover. Defragmentation of *ab initio* repeat families, as we have shown in our results from ProxMiner in Chapter 4, can identify larger repeat families encompassing multiple *ab initio* repeat families. Moreover, defragmentation of repeats from newly sequenced genomes can help characterize novel repeats by associating them with known repeats in the genome.

In this dissertation, we have demonstrated that spatial association rule mining followed by graph mining can be used to discover weakly conserved patterns in genomic data. The case study demonstrates how our ProxMiner algorithm can be used to defragment repeat families in the genome. We describe the contributions of our work for data mining and repeat discovery research followed by limitations and future work in the subsequent sections of this chapter.

5.1 Contributions

This dissertation makes several contributions to the field of bioinformatics and data mining. The key contribution is the development of a novel spatial data mining algorithm. Our approach differs from other association rule mining approaches because we begin by discovery of significant binary associations among genomic features and use these associations to build a directed graph where the edge weights are the strength of the associations. This two step process allows us to detect more subtle associations among large groups of features than is possible when using association rules that require all features to be present in qualifying "transactions." We demonstrate that our new algorithm can be successfully applied to discover novel repeat families in the rice genome.

We have implemented a tool for repeat discovery called ProxMiner as a case study. ProxMiner uses spatial association rule mining and graph mining to identify fragments of a repeat family on the genome and assemble the fragments for analysis by biologists. Our technique provides some major improvements over available algorithms as described below.

1. Other defragmentation algorithms cluster only items with the same labels together while we find associations among items with both the same and with different labels. We use association rule mining to find repeat families that are co-located consistently. This approach enables ProxMiner to construct larger repeat families as compared to other *ab initio* repeat finders. Combining association rule mining with graph mining supports inference of transitively related components and thus provides ProxMiner with increased sensitivity for identification of weakly conserved parts of repeat families.
2. ProxMiner can work with both known repeats, novel repeats detected by an *ab initio* repeat finder or a combination of the two. Other available defragmentation algorithms are designed to work specifically with known repeats.
3. We produce detailed characterization reports for all repeats [45, 202-204] defragmented by ProxMiner (Table 12). We also provide means for interfacing with genome browsers to allow biologists to visually explore a repeat in its genomic context [172]. Other defragmentation algorithms focus only on visualization of defragmented repeats.

4. We have also developed a method for evaluating the confidence of spatial association rules derived from genomic data using Monte Carlo simulation to estimate false discovery rates for discovered binary relationships and the Chi-square (χ^2) statistic for measuring the significance of individual associations. Other repeat discovery and repeat defragmentation methods do not provide these capabilities.

We have demonstrated the effectiveness of ProxMiner for repeat discovery by analyzing the *Oryza sativa* (japonica) genome. We report two novel repeat families and one repeat family extension. The DR1 novel repeat family is an example of a repeat family involved in disease resistance and demonstrates the continuum between repeats and genes in a genome. The DR2 novel repeat family is a typical example of an ancient and diverged repeat family where only small sections are conserved. The DR3 repeat family is related to a known DNA transposon [196] and is an example that demonstrates how ProxMiner can be used to extend the knowledge of known repeat regions.

These examples encompass some of the different types of repeat families that can be discovered by our algorithm. We have also discovered many other potentially interesting novel repeat families that can be further studied by biologists. ProxMiner will increase the effectiveness of repeat discovery techniques for newly sequenced genomes where *ab initio* repeat finders are only able to identify partial repeat families. The use of matching engines like BLAST for repeat families detected by an *ab initio* repeat finder followed by tedious and time consuming manual analysis is the other option currently available to biologists trying to defragment novel repeats. ProxMiner can be used in conjunction with other repeat finders to identify repeats in a newly sequenced genome and as well as to locate novel repeats and extensions to known repeats in well studied genomes.

Our algorithm has implications for bioinformatics research that extend beyond repeat regions. ProxMiner is designed to mine genomic DNA sequence and identify reoccurring spatial patterns. Using ProxMiner, it is possible for biologists to find spatial relationships among other genomic features of interest such as genes, transcription factors, repeats etc. These relationships can be used to discover associations between features and study "migration patterns" of different genomic features as well as identify novel interactions among them.

5.2 Limitations

The ability of ProxMiner to find repeats is dependent on the effectiveness of the *ab initio* repeat finder used to detect the initial set of repeat families. We have used RepeatScout [44] in our experiments because we found it to have high true positive rates and report long consensus sequences in an empirical comparison of various *ab initio* repeat finders [15]. The performance of ProxMiner will obviously improve as *ab initio* repeat finding algorithms improve.

Our approach of using pairwise associations among components followed by graph construction and mining may result in inferring transitive relationships that do not exist. Such cases can be identified in subsequent analysis by biologists based on the reports created by ProxMiner and during the visual exploration step of the ProxMiner repeat families.

Our current Monte Carlo method for estimating the false discovery rate results in a false discovery rate that is greater than 100% for very low confidence values. We currently use a uniform distribution to generate locations for the simulated chromosome. A different method for generating the locations that results in a distribution of locations similar to the one found on the chromosome might eliminate this problem.

Like any other repeat finding approach, the effectiveness of ProxMiner cannot be quantified strictly in terms of true positives or true negatives. This is because the exact number and identity of all repeats in an organism's genome is unknown. The value of ProxMiner results can only be evaluated by biologists by analyzing the identified repeat families.

5.3 Future work

While developing an algorithm with a focus on addressing weaknesses in repeat discovery techniques, we have created a computational technique that can be applied to other spatial data mining problems in genomics. There are a number of extensions possible to the ProxMiner pipeline architecture, including the following:

- Visualization: Available defragmentation algorithms and ProxMiner utilize genome browsers [172, 205] to visualize their results. There is no GUI that is tailored for visualizing novel repeat regions on the genome to aid biologists in the analysis and characterization of these important genomic constituents.

ProxMiner can be extended to visualize the defragmented repeats along with the available annotation information to help biologists to explore these regions with greater ease. Current browsers only allow viewing a single region of the genome at a time. A browser that allows exploration of multiple locations simultaneously would be particularly useful for comparative analysis of repeats.

- Post discovery processing: ProxMiner currently reports instances of each repeat family that is present on the input genome. ProxMiner can be extended by building sequence analysis pipelines like SRCP [200] to:
  - Search for similarity to known repeats [45, 128] in order to classify a ProxMiner repeat family into known repeat classes or designate it as an extension to a known repeat region.
  - Use gene prediction algorithms [206, 207] to identify coding regions in the sequences that can explain their functional role in the genome. The proteins for the coding region can then be compared [130] to proteins in rice and related species.
  - Identify sequence features characteristic of repeat regions in DNA such as target site duplications (TSD) and terminal inverted repeats (TIR). We can search [138, 139] for the presence of such features in the instances and in the flanking regions to characterize a ProxMiner repeat family.

In the future, as datasets are scaled up to include multiple genomes and a larger number of features, we can consider parallelizing our spatial data mining algorithm. The association rule mining step is the computational bottleneck and this step can be easily parallelized by dividing the data among different processors, mining rules from each data subsequence, and then combining the counts. Assigning each chromosome to a different processor avoids boundary problems, but would result in load imbalance due to the different sizes of the chromosomes.

Another promising extension to this research lies in further biological analysis of ProxMiner results. In this dissertation, we have discovered two novel repeat families and one repeat family that is an extension of a known repeat. A typical next step after discovery is to sequence the repeat region [183, 208] in other rice species for comparative genomics. We can sequence instances of DR1, DR2 and DR3 ProxMiner repeat families in other genomes for comparative studies and computational characterization. The instances of ProxMiner families can be identified in DNA using high throughput primer design [209] and PCR.

Although our case study has focused on repeat discovery, the algorithm can also be applied to mine the spatial arrangement of other features in the genome. The presence of gene neighborhoods in eukaryotes has been reported in literature [110, 210-212] and gene islands are a known phenomena in prokaryotes [213-216]. Repeats have also been shown to play functional roles in the genome such as generation of new genes [20, 37], increasing genetic diversity [38, 40] and influencing expression of nearby genes [217-219]. We can use ProxMiner to identify spatial relationships between genes, repeats and other regulatory regions in rice etc. This has the potential to discover unreported interactions among these features.

Our data mining approach can also be applied to one dimensional spaces other than a genome. Text mining and temporal data mining are possible application areas.

5.4 Related publications

A list of publications related to this work is presented below:

- Saha S, Bridges S, Magbanua Z, Peterson DG (2008) Discovering relationships among dispersed repeats using spatial association rule mining. Highlights from the Fourth ISCB Student Council Symposium. *BMC Bioinformatics*, 9(Suppl 10):04.
- Saha S, Bridges S, Magbanua Z, Peterson DG (2008) Computational approaches and tools used in identification of dispersed repetitive DNA sequences. *Tropical Plant Biology*, 1: 85-96.
- Saha S, Bridges S, Magbanua Z, Peterson DG (2008) Empirical comparison of ab initio repeat finding programs for identification of repetitive DNA sequences. *Nucleic Acids Research*, 36: 2284-2294.

REFERENCES

[1] J. Xing, D. J. Witherspoon, D. A. Ray, M. A. Batzer, and L. B. Jorde, "Mobile DNA elements in primate and human evolution," *American Journal of Physical Anthropology,* vol. 134, pp. 2-19, 2007.

[2] E. R. Mardis, "The impact of next-generation sequencing technology on genetics," *Trends in Genetics,* vol. 24, pp. 133-141, 2008.

[3] "Prepare for the deluge," in *Nature Biotechnology*. vol. 26: Nature Publishing Group, 2008, pp. 1099-1099.

[4] M. Pop and S. L. Salzberg, "Bioinformatics challenges of new sequencing technology," *Trends in Genetics,* vol. 24, pp. 142-149, 2008.

[5] O. Morozova and M. A. Marra, "Applications of next-generation sequencing technologies in functional genomics," *Genomics,* vol. 92, pp. 255-264, 2008.

[6] T. F. Smith and X. Zhang, "The challenges of genome sequence annotation or "The devil is in the details"," *Nature Biotechnology,* vol. 15, pp. 1222-1223, 1997.

[7] J. F. D. Dean, "Tagging all genes," *Nature Biotechnology,* vol. 22, pp. 961-962, 2004.

[8] M.-F. S. Catherine Mathé, Thomas Schiex, Pierre Rouzé, "Current methods of gene prediction, their strengths and weaknesses," *Nucleic Acids Research,* vol. 30, pp. 4103-4117, 2002.

[9] G. D. Stormo, "Gene-Finding Approaches for Eukaryotes," *Genome Research,* vol. 10, pp. 394-397, April 1, 2000 2000.

[10] E. Jabri, "Small, but in control," in *Nature Reviews Molecular Cell Biology*. vol. 6, 2005, pp. 361-361.

[11] W. W. Wasserman and A. Sandelin, "Applied bioinformatics for the identification of regulatory elements," *Nature Reviews Genetics,* vol. 5, pp. 276-287, 2004.

[12] J. Nardone, D. U. Lee, K. M. Ansel, and A. Rao, "Bioinformatics for the 'bench biologist': how to find regulatory regions in genomic DNA," *Nature Immunology,* vol. 5, pp. 768-774, 2004.

[13] S. J. M. Jones, "Prediction of Genomic Functional Elements," *Annual Review of Genomics and Human Genetics,* vol. 7, pp. 315-338, 2006.

[14] S. Saha, S. Bridges, Z. Magbanua, and D. Peterson, "Computational Approaches and Tools Used in Identification of Dispersed Repetitive DNA Sequences," *Tropical Plant Biology,* vol. 1, pp. 85-96, 2008.

[15] S. Saha, S. Bridges, Z. V. Magbanua, and D. G. Peterson, "Empirical comparison of ab initio repeat finding programs," *Nucleic Acids Research,* vol. 36, pp. 2284-2294, April 1, 2008 2008.

[16] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Conference*, Washington D.C., 1993, pp. 207-216.

[17] G. Chin, G. C. Nakamura, D. G. Chavarria, and H. J. Sofia, "Graph Mining of Networks from Genome Biology," in *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, 2007, pp. 1265-1269.

[18] C. Biemont and C. Vieira, "Genetics: Junk DNA as an evolutionary force," *Nature,* vol. 443, pp. 521-524, 2006.

[19] J. Brosius, "How significant is 98.5% 'junk' in mammalian genomes," *Bioinformatics,* vol. 19, p. II35, 2003.

[20] N. Jiang, Z. Bao, X. Zhang, S. R. Eddy, and S. R. Wessler, "Pack-MULE transposable elements mediate gene evolution in plants," *Nature,* vol. 431, pp. 569-573, 2004.

[21] M. Morgante, S. Brunner, G. Pea, K. Fengler, A. Zuccolo, and A. Rafalski, "Gene duplication and exon shuffling by helitron-like transposons generate intraspecies diversity in maize," *Nature Genetics,* vol. 37, pp. 997-1002, 2005.

[22] C. M. Bergman and H. Quesneville, "Discovering and detecting transposable elements in genome sequences," *Briefings in Bioinformatics,* vol. 8, p. bbm048, October 10, 2007 2007.

[23] J. Luo and S. M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," *International Journal of Intelligent Systems,* vol. 15, pp. 687-703, 2000.

[24] S. Jacques, L. Veronique, and U. Stephane, "Modeling of information system correlated events time dependencies," in *Proceedings of the 8th international conference on New technologies in distributed systems* Lyon, France: ACM, 2008.

[25] L. Wu, D. Hai-Xin, R. Ping, L. Xing, and W. Jian-Ping, "Wavelet based data mining and querying in network security databases," in *Machine Learning and Cybernetics, 2003 International Conference on*, 2003, pp. 178-182 Vol.1.

[26] L. Xiangyang and Y. Nong, "A supervised clustering and classification algorithm for mining data with mixed variables," *Systems, Man and Cybernetics, Part A, IEEE Transactions on,* vol. 36, pp. 396-406, 2006.

[27] A. Maedche and S. Staab, "Ontology learning for the Semantic Web," *Intelligent Systems, IEEE,* vol. 16, pp. 72-79, 2001.

[28] H. Chien-Jen, T. Ping-Heng, H. Chia-Ling, and P. Reuay-Ching, "Exploring Cognitive Difference in instructional outcomes using Text mining technology," in *IEEE International Conference on Systems, Man and Cybernetics, 2006 (SMC '06).* 2006, pp. 2116-2120.

[29] G. Leroy, M. Fiszman, and T. C. Rindflesch, "The Impact of Directionality in Predications on Text Mining," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008, pp. 228-228.

[30] C. C. Latiri and S. Ben Yahia, "Generating implicit association rules from textual data," in *ACS/IEEE International Conference on Computer Systems and Applications 2001*, 2001, pp. 137-143.

[31] B. Jun-Peng, S. Jun-Yi, L. Xiao-Dong, and S. Qin-Bao, "A new text feature extraction model and its application in document copy detection," in *International Conference on Machine Learning and Cybernetics, 2003* 2003, pp. 82-87 Vol.1.

[32] N. D. Trinklein, U. Karaoz, J. Wu, A. Halees, S. Force Aldred, P. J. Collins, D. Zheng, Z. D. Zhang, M. B. Gerstein, M. Snyder, R. M. Myers, and Z. Weng, "Integrated analysis of experimental data sets reveals many novel promoters in 1% of the human genome," *Genome Research,* vol. 17, pp. 720-731, June 1, 2007 2007.

[33] F. Denoeud, P. Kapranov, C. Ucla, A. Frankish, R. Castelo, J. Drenkow, J. Lagarde, T. Alioto, C. Manzano, J. Chrast, S. Dike, C. Wyss, C. N. Henrichsen, N. Holroyd, M. C. Dickson, R. Taylor, Z. Hance, S. Foissac, R. M. Myers, J. Rogers, T. Hubbard, J. Harrow, R. Guigo, T. R. Gingeras, S. E. Antonarakis, and A. Reymond, "Prominent use of distal 5' transcription start sites and discovery of a large number of additional exons in ENCODE regions," *Genome Research,* vol. 17, pp. 746-759, June 1, 2007 2007.

[34] Z. D. Zhang, A. Paccanaro, Y. Fu, S. Weissman, Z. Weng, J. Chang, M. Snyder, and M. B. Gerstein, "Statistical analysis of the genomic distribution and correlation of regulatory elements in the ENCODE regions," *Genome Research,* vol. 17, pp. 787-797, June 1, 2007 2007.

[35] V. V. Kapitonov and J. Jurka, "Rolling-circle transposons in eukaryotes," *Proceedings of National Academy of Sciences U.S.A.,* vol. 98, pp. 8714-8719, July 17, 2001 2001.

[36] H. K. Dooner and C. F. Weil, "Give-and-take: interactions between DNA transposons and their host plant genomes," *Current Opinion in Genetics & Development,* vol. 17, pp. 486-492, 2007.

[37] J. Lai, Y. Li, J. Messing, and H. K. Dooner, "Gene movement by Helitron transposons contributes to the haplotype variability of maize," *Proceedings of National Academy of Sciencs U. S. A,* vol. 102, pp. 9068-9073, June 21, 2005 2005.

[38] J. Bennetzen, "Transposable element contributions to plant gene and genome evolution," *Plant Molecular Biology,* vol. 42, pp. 251-269, 2000.

[39] B. McClintock, "The significance of responses of the genome to challenge," *Science,* vol. 226, pp. 792-801, 16 Nov 1984.

[40] N. Jiang, Z. Bao, X. Zhang, H. Hirochika, S. R. Eddy, S. R. McCouch, and S. R. Wessler, "An active DNA transposon family in rice," *Nature,* vol. 421, pp. 163-167, 2003.

[41] J. Jurka, "Repeats in genomic DNA: mining and meaning," *Current Opinion in Structural Biology,* vol. 8, pp. 333 - 337, 1998.

[42] C. Chicken Genome Sequencing, "Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution," *Nature,* vol. 432, pp. 695 - 716, 2004.

[43] I. R. G. S. Project, "The map-based sequence of the rice genome," *Nature,* vol. 436, pp. 793-800, 2005.

[44] A. L. Price, N. C. Jones, and P. A. Pevzner, "De novo identification of repeat families in large genomes," *Bioinformatics,* vol. 21, pp. i351–i358, 2005.

[45] J. Jurka, Kapitonov, V.V., Pavlicek, A., Klonowski, P., Kohany, O., Walichiewicz, "Repbase Update, a database of eukaryotic repetitive elements," *Cytogentic and Genome Research,* vol. 110, pp. 462-467, 2005.

[46] J. Jurka, P. Klonowski, V. Dagman, and P. Pelton, "Censor--a program for identification and elimination of repetitive elements from DNA sequences," *Computers & Chemistry,* vol. 20, pp. 119-121, 1996.

[47] R. Li, J. Ye, S. Li, J. Wang, Y. Han, C. Ye, J. Wang, H. Yang, J. Yu, G. K. Wong, and J. Wang, "ReAS: Recovery of Ancestral Sequences for Transposable Elements from the Unassembled Reads of a Whole Genome Shotgun," *PLOS Computational Biology,* vol. 1, p. e43, September 2005.

[48] P. A. Pevzner, H. Tang, and G. Tesler, "De Novo Repeat Classification and Fragment Assembly," *Genome Research,* vol. 14, pp. 1786-1796, September 1, 2004 2004.

[49] J. Giordano, Y. Ge, Y. Gelfand, G. r. AbrusÃ¡n, G. Benson, and P. E. Warburton, "Evolutionary History of Mammalian Transposons Determined by Genome-Wide Defragmentation," *PLoS Computational Biology,* vol. 3, p. e137, 2007.

[50] D. Y. Lin, "An efficient Monte Carlo approach to assessing statistical significance in genomic studies," *Bioinformatics,* vol. 21, pp. 781-787, March 15, 2005 2005.

[51] J. M. Hammersley and D. C. Handscomb, *Monte Carlo methods*: Methuen, 1964.

[52] C. Zhang and S. Zhang, *Association rule mining: models and algorithms* vol. 2307: Springer Verlag New York, Inc., 2002.

[53] K. Koperski and J. Han, "Discovery of spatial association rules in geographic information databases," in *Advances in Spatial Databases*. vol. 951/1995 Berlin / Heidelberg: Springer, 1995, pp. 47-66.

[54] D. Malerba, F. Esposito, F. A. Lisi, and A. Appice, "Mining spatial association rules in census data," in *Joint conference on "New techniques and technologies for Statistics" and "Exchange of technology and know-how"*, 2001, pp. 19-44.

[55] B. Berendt, A. Hotho, and G. Stumme, "Towards Semantic Web Mining," in *The Semantic Web — ISWC 2002* Berlin/Heidelberg: Springer, 2002, pp. 264-278.

[56] J. Roddick, K. Hornsby, and M. Spiliopoulou, "An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research," in *Temporal, Spatial, and Spatio-Temporal Data Mining* Berlin/Heidelberg: Springer, 2001, pp. 147-163.

[57] M. J. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*: John Wiley & Sons, Inc. New York, NY, USA, 1997.

[58] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: generalizing association rules to correlations," in *ACM SIGMOD international conference on Management of data* Tucson, AZ, 1997, pp. 265-276.

[59] V. Dhar and A. Tuzhulin, "Abstract-driven pattern discovery in databases," *IEEE Transactions on Knowledge and Data Engineering,* vol. 5, pp. 926-938, 1993.

[60] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *Int'l Conf. Very Large Databases (VLDB '94)*, 1994, p. 12.

[61] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," in *Proceedings of 21st International Conference on VLDB*, Zurich, Switzerland, 1995, pp. 432-444.

[62] J. S. Park, M. S. Chen, and P. S. Yu, "An effective hash-based algorithm for mining association rules," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, San Jose, CA, 1995, pp. 175-186.

[63] O. R. Zaïane and M.-L. Antonie, "On Pruning and Tuning Rules for Associative Classifiers," in *Knowledge-Based Intelligent Information and Engineering Systems*. vol. 3683/2005 Berlin/Heidelberg: Springer, 2005, pp. 966-973.

[64] K. Shyamala and S. P. Rajagopalan, "Mining Essential and Interesting Rules for Efficient Prediction," *Asian Journal of Information Technology,* vol. 6, pp. 1192-1195, 2007.

[65] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM SIGMOD Intl. Conference on Management of Data*, Dallas, TX, 2000, pp. 1–12.

[66] J. Han and J. Pei, "Mining frequent patterns by pattern-growth: methodology and implications," in *ACM SIGKDD Explorations Newsletter*. vol. 2: ACM New York, NY, USA, 2000, pp. 14-20.

[67] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery,* vol. 8, pp. 53-87, 2004.

[68] G. Psaila and P. L. Lanzi, "Hierarchy-based mining of association rules in data warehouses," in *Proceedings of the 2000 ACM symposium on Applied computing - Volume 1* Como, Italy: ACM, 2000.

[69] J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 11, pp. 798-805, 1999.

[70] J. Han, "Mining knowledge at multiple concept levels," in *Proceedings of the fourth international conference on Information and knowledge management* Baltimore, Maryland, United States: ACM, 1995.

[71] B. Lent, A. Swami, and J. Widom, "Clustering association rules," in *Data Engineering, 1997. Proceedings. 13th International Conference on*, 1997, pp. 220-231.

[72] H. Lu, L. Feng, and J. Han, "Beyond intratransaction association analysis: mining multidimensional intertransaction association rules," *ACM Transactions on Information Systems,* vol. 18, pp. 423-454, 2000.

[73] W. Wang, J. Yang, and P. S. Yu, "Efficient mining of weighted association rules (WAR)," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* Boston, Massachusetts, United States: ACM, 2000.

[74] U. Yun, "A new framework for detecting weighted sequential patterns in large sequence databases," *Knowledge-Based Systems,* vol. 21, pp. 110-122, 2008.

[75] C. Chiu and P. L. Hsu, "A constraint-based genetic algorithm approach for mining classification rules," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 35, pp. 205-220, 2005.

[76] J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome, "Constraint-based concept mining and its application to microarray data analysis," *Intelligent Data Analysis,* vol. 9, pp. 59-82, 2005.

[77] J. Pei and J. Han, "Constraint frequent pattern mining: A pattern-growth view," in *SIGKDD Explorations*. vol. 4: ACM, 2002, pp. 31-39.

[78] A. J. T. Lee, W. C. Lin, and C. S. Wang, "Mining association rules with multi-dimensional constraints," *Journal of Systems and Software,* vol. 79, pp. 79-92, 2006.

[79] Y. Fu and J. Han, "Meta-rule-guided mining of association rules in relational databases," in *Knowledge Discovery in Databases with Deductive and Object-Oriented Databases (KDOOD) and Temporal Reasoning in Deductive and Object-Oriented Databases (TDOOD)*, Singapore, 1995, pp. 39–46.

[80] M. K. J. Han and J. Y. Chiang, "Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes," in *Proceedings ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, 1997, pp. 207-210.

[81] X. Yin and J. Han, "CPAR: Classification based on Predictive Association Rules," in *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, 2003.

[82] F. A. Thabtah and P. I. Cowling, "A greedy classification algorithm based on association rule," *Applied Soft Computing,* vol. 7, pp. 1102-1111, 2007.

[83] M. J. Zaki, "Parallel and distributed association mining: a survey," *IEEE Concurrency,* vol. 7, pp. 14-25, 1999.

[84] R. H. Güting, "An introduction to spatial database systems," *The VLDB Journal — The International Journal on Very Large Data Bases,* vol. 3, pp. 357-399, 1994.

[85] K. Koperski and J. Han, "Data mining methods for the analysis of large geographic databases," in *10th Annual Conference on GIS for Natural Resources, Environment and Land Information Management*, Vancouver, Canada, 1996, pp. B4-1.

[86] K. Koperski, J. Han, and N. Stefanovic, "An Efficient Two-Step Method for Classification of Spatial Data," in *Symposium on Spatial Data Handling (SDH 98)*, Vancouver, Canada, 1998, pp. 45-54.

[87] J. Han, J. Y. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B. B. Xia, O. R. Zaiane, S. Zhang, and H. Zhu, "DBMiner: a system for data mining in relational databases and data warehouses," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research* Toronto, Ontario, Canada: IBM Press, 1997.

[88] J. Han, K. Koperski, and N. Stefanovic, "GeoMiner: a system prototype for spatial data mining," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data* Tucson, Arizona, United States: ACM, 1997.

[89] M. J. Egenhofer, "Spatial SQL: a query and presentation language," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 6, pp. 86-95, 1994.

[90] M. Ester, A. Frommelt, H.-P. Kriegel, and J. Sander, "Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support," *Data Mining and Knowledge Discovery,* vol. 4, pp. 193-216, 2000.

[91] S. Rinzivillo and F. Turini, "Extracting spatial association rules from spatial transactions," in *Proceedings of the 13th annual ACM international workshop on Geographic information systems* Bremen, Germany: ACM, 2005.

[92] E. M. Knorr and R. T. Ng, "Finding aggregate proximity relationships and commonalities in spatial data mining," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 8, pp. 884-897, 1996.

[93] B. Braunmuller, M. Ester, H. P. Kriegel, and J. Sander, "Multiple similarity queries: a basic DBMS operation for mining in metric databases," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 13, pp. 79-95, 2001.

[94] S. Dai, L. Gao, Q. Zhu, and C. Zhu, "A Novel Genetic Algorithm Based on Image Databases for Mining Association Rules," in *6th IEEE/ACIS International Conference on Computer and Information Science*, Melbourne, Australia, 2007, pp. 977-980.

[95] K. Satou, G. Shibayama, T. Ono, Y. Yamamura, E. Furuichi, S. Kuhara, and T. Takagi, "Finding association rules on heterogeneous genome data," in *Pacific Symp Biocomputing* Hawaii, 1997, pp. 397-408.

[96] K. Satou, E. Furuichi, K. Takiguchi, S. Kuhara, and T. Takagi, "Application of a deductive database system PACADE toward discovery of clusters of similar structures in proteins," in *System Sciences, 1994. Vol.V: Biotechnology Computing, Proceedings of the Twenty-Seventh Hawaii International Conference on*, 1994, pp. 160-169.

[97] C. Becquet, S. Blachon, B. Jeudy, J. F. Boulicaut, and O. Gandrillon, "Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data," *Genome Biology,* vol. 3, pp. research0067.1 - research0067.16, 2002.

[98] M. J. Zaki, "Generating non-redundant association rules," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* Boston, Massachusetts, United States: ACM, 2000.

[99] E. Georgii, L. Richter, U. Ruckert, and S. Kramer, "Analyzing microarray data using quantitative association rules," *Bioinformatics,* vol. 21, pp. ii123-129, September 1, 2005 2005.

[100] U. Ruckert, L. Richter, and S. Kramer, "Quantitative association rules based on half-spaces: an optimization approach," in *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, Munchen, Germany, 2004, pp. 507-510.

[101] L. Ji and K. L. Tan, "Mining gene expression data for positive and negative co-regulated gene clusters," *Bioinformatics,* vol. 20, pp. 2711-2718, November 1, 2004 2004.

[102] H. D. Huang, H. L. Chang, T. S. Tsou, B. J. Liu, C. Y. Kao, and J. T. Horng, "A Data Mining Method to Predict Transcriptional Regulatory Sites Based on Differentially Expressed Genes in Human Genome," in *Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering*, Washington, DC, USA, 2003.

[103] H. Jorng-Tzong, F. M. Lin, J. H. Lin, H. D. Huang, and B. J. Liu, "Database of repetitive elements in complete genomes and data mining using transcription factor binding sites," *IEEE Transactions on Information Technology in Biomedicine,* vol. 7, pp. 93-100, 2003.

[104] X. Morgan, S. Ni, D. Miranker, and V. Iyer, "Predicting combinatorial binding of transcription factors to regulatory elements in the human genome by association rule mining," *BMC Bioinformatics,* vol. 8, p. 445, 2007.

[105] I. Artamonova, G. Frishman, and D. Frishman, "Applying negative rule mining to improve genome annotation," *BMC Bioinformatics,* vol. 8, p. 261, 2007.

[106] I. Artamonova, G. Frishman, M. S. Gelfand, and D. Frishman, "Mining sequence annotation databanks for association patterns," *Bioinformatics,* vol. 21, pp. iii49-57, November 1, 2005 2005.

[107] M. L. Riley, T. Schmidt, I. I. Artamonova, C. Wagner, A. Volz, K. Heumann, H.-W. Mewes, and D. Frishman, "PEDANT genome database: 10 years online," *Nucleic Acids Research,* vol. 35, pp. D354-357, January 12, 2007 2007.

[108] K. Kianmehr and R. Alhajj, "CARSVM: A class association rule-based classification framework and its application to gene expression data," *Artificial Intelligence in Medicine,* vol. 44, pp. 7-25, 2008.

[109] M. Tamura and P. D'Haeseleer, "Microbial genotype-phenotype mapping by class association rule mining," *Bioinformatics,* vol. 24, pp. 1523-1529, July 1, 2008 2008.

[110] B. Oliver and T. Misteli, "A non-random walk through the genome," *Genome Biology,* vol. 6, 2005.

[111] K. Horimoto, S. Fukuchi, and K. Mori, "Comprehensive comparison between locations of orthologous genes on archaeal and bacterial genomes," *Bioinformatics,* vol. 17, pp. 791-802, September 1, 2001 2001.

[112] H. Yang, S. Parthasarathy, and D. Ucar, "A spatio-temporal mining approach towards summarizing and analyzing protein folding trajectories," *Algorithms for Molecular Biology,* vol. 2, p. 3, 2007.

[113] V.-V. Rantanen, M. Gyllenberg, T. Koski, and M. S. Johnson, "A dissimilarity matrix between protein atom classes based on Gaussian mixtures," *Bioinformatics,* vol. 18, pp. 1257-1263, August 1, 2002 2002.

[114] I. Plyusnin, A. R. Evans, A. Karme, A. Gionis, and J. Jernvall, "Automated 3D Phenotype Analysis Using Data Mining," *PLoS ONE,* vol. 3, p. e1742, 2008.

[115] M. Dolan, C. Holden, M. K. Beard, and C. Bult, "Genomes as geography: using GIS technology to build interactive genome feature maps," *BMC Bioinformatics,* vol. 7, p. 416, 2006.

[116] J. van Hemert and R. Baldock, "Mining Spatial Gene Expression Data for Association Rules," in *Bioinformatics Research and Development*. vol. 4414 Berlin: Springer, 2007, p. 66.

[117] T. Aittokallio and B. Schwikowski, "Graph-based methods for analysing networks in cell biology," *Briefings in Bioinformatics,* vol. 7, pp. 243-255, September 1, 2006 2006.

[118] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou, "Mining coherent dense subgraphs across massive biological networks for functional discovery," *Bioinformatics,* vol. 21, pp. i213-221, June 1, 2005 2005.

[119] L. c. Royer, M. Reimann, B. Andreopoulos, and M. Schroeder, "Unraveling Protein Networks with Power Graph Analysis," *PLoS Computational Biology,* vol. 4, p. e1000108, 2008.

[120] X. Yan, M. R. Mehan, Y. Huang, M. S. Waterman, P. S. Yu, and X. J. Zhou, "A graph-based approach to systematically reconstruct human transcriptional regulatory modules," *Bioinformatics,* vol. 23, pp. i577-586, July 1, 2007 2007.

[121] A. D. King, N. Przulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics,* vol. 20, pp. 3013-3020, November 22, 2004 2004.

[122] G. Bader and C. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics,* vol. 4, p. 2, 2003.

[123] B. Adamcsek, G. Palla, I. J. Farkas, I. Derenyi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," *Bioinformatics,* vol. 22, pp. 1021-1023, April 15, 2006 2006.

[124] Y. Qi, F. Balem, C. Faloutsos, J. Klein-Seetharaman, and Z. Bar-Joseph, "Protein complex identification by supervised graph local clustering," *Bioinformatics,* vol. 24, pp. i250-268, July 1, 2008 2008.

[125] M. Li, J. Chen, J. Wang, B. Hu, and G. Chen, "Modifying the DPClus algorithm for identifying protein complexes based on new topological structures," *BMC Bioinformatics,* vol. 9, 2008.

[126] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks," *BMC Bioinformatics,* vol. 7, p. 207, 2006.

[127] M. Hamada, K. Tsuda, T. Kudo, T. Kin, and K. Asai, "Mining frequent stem patterns from unaligned RNA sequences," *Bioinformatics,* vol. 22, pp. 2480-2487, October 15, 2006 2006.

[128] A. F. A. Smit, R. Hubley, and P. Green, "RepeatMasker Open-3.0," 3.0 ed, 1996-2004.

[129] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." vol. 215, 1990, pp. 403 - 410.

[130] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research,* vol. 25, p. 14, September 1, 1997 1997.

[131] E. M. McCarthy and J. F. McDonald, "LTR_STRUC: a novel search and identification program for LTR retrotransposons," *Bioinformatics,* vol. 19, pp. 362-367, 2003.

[132] Z. Tu, "Eight novel families of miniature inverted repeat transposable elements in the African malaria mosquito, Anopheles gambiae," *PNAS,* vol. 98, pp. 1699-1704, Feb 13 2001.

[133] G. O. Sperber, T. Airola, P. Jern, and J. Blomberg, "Automated recognition of retroviral sequences in genomic data RetroTector(C)," *Nucleic Acids Research,* vol. 35, p. 12, July 17, 2007 2007.

[134] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition ed.: MIT Press and McGraw-Hill, 2001.

[135] G. Yang and T. C. Hall, "MAK, a computational tool kit for automated MITE analysis," *Nucleic Acids Research,* vol. 31, pp. 3659-3665, 2003.

[136] P. E. Warburton, J. Giordano, F. Cheung, Y. Gelfand, and G. Benson, "Inverted Repeat Structure of the Human Genome: The X-Chromosome Contains a Preponderance of Large, Highly Homologous Inverted Repeats That Contain Testes Genes," *Genome Research,* vol. 14, p. 8, October 1, 2004 2004.

[137] O. Andrieu, A.-S. Fiston, D. Anxolabehere, and H. Quesneville, "Detection of transposable elements by their compositional bias," *BMC Bioinformatics,* vol. 5, p. 94, 2004.

[138] S. Szak, O. Pickeral, W. Makalowski, M. Boguski, D. Landsman, and J. Boeke, "Molecular archeology of L1 insertions in the human genome," *Genome Biology,* vol. 3, pp. research0052.1 - research0052.18, 2002.

[139] Z. Tu, S. Li, and C. Mao, "The Changing Tails of a Novel Short Interspersed Element in Aedes aegypti: Genomic Evidence for Slippage Retrotransposition and the Relationship Between 3' Tandem Repeats and the poly(dA) Tail," *Genetics,* vol. 168, pp. 2037-2047, December 1, 2004 2004.

[140] V. Pereira, "Insertion bias and purifying selection of retrotransposons in the Arabidopsis thaliana genome," *Genome Biology,* vol. 5, p. R79, 2004.

[141] B. A. Kronmiller and R. P. Wise, "TEnest: Automated Chronological Annotation and Visualization of Nested Plant Transposable Elements," *Plant Physiology,* vol. 146, pp. 45-59, January 1, 2008 2008.

[142] V. Pereira, "Automated Paleontology of Repetitive DNA with REannotate," *BMC Genomics,* vol. 9, p. 614, 2008.

[143] P. Agarwal and D. J. States, "The Repeat Pattern Toolkit (RPT): analyzing the structure and evolution of the C. elegans genome," in *International Conference on Intelligent Systems for Molecular Biology* Stanford, CA: AAAI, 1994, pp. 1-9.

[144] Z. Bao and S. R. Eddy, "Automated De Novo Identification of Repeat Sequence Families in Sequenced Genomes," *Genome Research,* vol. 12, pp. 1269-1276, 2002.

[145] R. C. Edgar and E. W. Myers, "PILER: identification and classification of genomic repeats," *Bioinformatics,* vol. 21, pp. i152-i158, 2005.

[146] 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,* vol. 21, pp. 582-588, 2005.

[147] S. Kurtz, J. V. Choudhuri, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich, "REPuter: the manifold applications of repeat analysis on a genomic scale," *Nucleic Acids Research,* vol. 29, pp. 4633-4642, November 15, 2001 2001.

[148] P. Weiner, "Linear pattern matching algorithm," in *14th Annual IEEE Symposium on Switching and Automata Theory*, 1973, pp. 1-11.

[149] N. Volfovsky, B. Haas, and S. Salzberg, "A clustering method for repeat analysis in DNA sequences," *Genome Biology,* vol. 2, pp. 1–11, 2001.

[150] B. Ma, J. Tromp, and M. Li, "PatternHunter: faster and more sensitive homology search," *Bioinformatics,* vol. 18, pp. 440-445, March 1, 2002 2002.

[151] M. Li, B. Ma, D. Kisman, and J. Tromp, "PatternHunter II: Highly Sensitive and Fast Homology Search," *Journal of Bioinformatics and Computational Biology,* vol. 2, pp. 417-440, 2004.

[152] L. Ilie and S. Ilie, "Multiple spaced seeds for homology search," *Bioinformatics,* vol. 23, pp. 2969-2977, November 15, 2007 2007.

[153] D. Mak, Y. Gelfand, and G. Benson, "Indel seeds for homology search," *Bioinformatics,* vol. 22, pp. e341-349, July 15, 2006 2006.

[154] E. L. L. Sonnhammer and R. Durbin, "A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis," *Gene,* vol. 167, pp. GC1-10, 1995.

[155] A. Taneda, "Adplot: detection and visualization of repetitive patterns in complete genomes," *Bioinformatics,* vol. 20, pp. 701-708, March 22, 2004 2004.

[156] D. Sharma, B. Issac, G. P. S. Raghava, and R. Ramaswamy, "Spectral Repeat Finder (SRF): identification of repetitive sequences using Fourier transformation," *Bioinformatics,* vol. 20, pp. 1405-1412, June 12, 2004 2004.

[157] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research,* vol. 32, pp. 1792-1797, March 19, 2004 2004.

[158] N. de Bruijn, "A combinatorial problem," in *Proc. Koninklijke Nederlandse Akademie van Wetenschappen*. vol. A49 Germany, 1946, pp. 758-764.

[159] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. USA: Cambridge University Press, 1999.

[160] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch, "Replacing Suffix Trees with Enhanced Suffix Arrays," *Journal of Discrete Algorithms,* vol. 2, pp. 53-86, 2004.

[161] U. Manber and G. Myers, "Suffix arrays: a new method for on-line string searches," *SIAM Journal on Computing,* vol. 22, p. 13, 1991.

[162] R. Chenna, H. Sugawara, T. Koike, R. Lopez, T. J. Gibson, D. G. Higgins, and J. D. Thompson, "Multiple sequence alignment with the Clustal series of programs," *Nucleic Acids Research,* vol. 31, pp. 3497-3500, July 1, 2003 2003.

[163] L. Francisco, B. Armando, G. Fernando, C. Carlos, and M. Antonio, "Fuzzy association rules for biological data analysis: A case study on yeast," *BMC Bioinformatics,* vol. 9, p. 107, 2008.

[164] S. Ouyang, W. Zhu, J. Hamilton, H. Lin, M. Campbell, K. Childs, F. Thibaud-Nissen, R. L. Malek, Y. Lee, L. Zheng, J. Orvis, B. Haas, J. Wortman, and C. R. Buell, "The TIGR Rice Genome Annotation Resource: improvements and new features," *Nucleic Acids Research,* vol. 35, pp. D883-887, January 12, 2007 2007.

[165] J. Hietaniemi, "Graph-0.91," 2009.

[166] M. Clamp, J. Cuff, S. M. Searle, and G. J. Barton, "The Jalview Java alignment editor," *Bioinformatics,* vol. 20, pp. 426-427, 2004.

[167] C. Feschotte and S. R. Wessler, "Treasures in the attic: Rolling circle transposons discovered in eukaryotic genomes," *Proceedings of National Academy of Sciences U. S. A,* vol. 98, pp. 8923-8924, July 31, 2001 2001.

[168] C. Feschotte, "Transposable elements and the evolution of regulatory networks," *Nat Rev Genet,* vol. 9, pp. 397-405, 2008.

[169] E. Havecker, X. Gao, and D. Voytas, "The diversity of LTR retrotransposons," *Genome Biology,* vol. 5, p. 225, 2004.

[170] R. J. Wicker T, SCHULZE SR, FELTUS FA, MAGRINI V, MORRISON JA, MARDIS ER, WILSON RK, Peterson DG, Paterson AH, IVARIE R, "The repetitive landscape of the chicken genome," *Genome Research,* vol. 15, pp. 126-136, 2005.

[171] D. G. Peterson, S. R. Schulze, E. B. Sciara, S. A. Lee, J. E. Bowers, A. Nagel, N. Jiang, D. C. Tibbitts, S. R. Wessler, and A. H. Paterson, "Integration of Cot Analysis, DNA Cloning, and High-Throughput Sequencing Facilitates Genome Characterization and Gene Discovery," *Genome Research,* vol. 12, p. 12, 2002.

[172] L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis, "The Generic Genome Browser: A Building Block for a Model Organism System Database," *Genome Research,* vol. 12, pp. 1599-1610, 2002.

[173] M. A. Crosby, J. L. Goodman, V. B. Strelets, P. Zhang, W. M. Gelbart, and C. The FlyBase, "FlyBase: genomes by the dozen," *Nucleic Acids Research,* p. gkl827, 2006.

[174] E. M. Schwarz, I. Antoshechkin, C. Bastiani, T. Bieri, D. Blasiar, P. Canaran, J. Chan, N. Chen, W. J. Chen, P. Davis, T. J. Fiedler, L. Girard, T. W. Harris, E. E. Kenny, R. Kishore, D. Lawson, R. Lee, H.-M. Muller, C. Nakamura, P. Ozersky, A. Petcherski, A. Rogers, W. Spooner, M. A. Tuli, K. Van Auken, D. Wang, R. Durbin, J. Spieth, L. D. Stein, and P. W. Sternberg, "WormBase: better software, richer content," *Nucleic Acids Research,* vol. 34, pp. D475-478, 2006.

[175] D. Swarbreck, C. Wilks, P. Lamesch, T. Z. Berardini, M. Garcia-Hernandez, H. Foerster, D. Li, T. Meyer, R. Muller, L. Ploetz, A. Radenbaugh, S. Singh, V. Swing, C. Tissier, P. Zhang, and E. Huala, "The Arabidopsis Information Resource (TAIR): gene structure and function annotation," *Nucleic Acids Research,* p. gkm965, 2007.

[176] H. Ohyanagi, T. Tanaka, H. Sakai, Y. Shigemoto, K. Yamaguchi, T. Habara, Y. Fujii, B. A. Antonio, Y. Nagamura, T. Imanishi, K. Ikeo, T. Itoh, T. Gojobori, and T. Sasaki, "The Rice Annotation Project Database (RAP-DB): hub for Oryza sativa ssp. japonica genome information," *Nucleic Acids Research,* vol. 34, pp. D741-744, 2006.

[177] R. Durbin and D. Haussler, "General Feature Format," 2007.

[178] A. Bouck, R. Peeler, M. L. Arnold, and S. R. Wessler, "Genetic Mapping of Species Boundaries in Louisiana Irises Using IRRE Retrotransposon Display Markers," *Genetics,* vol. 171, pp. 1289-1303, 2005.

[179] P. Warburton, D. Hasson, F. Guillem, C. Lescale, X. Jin, and G. Abrusan, "Analysis of the largest tandemly repeated DNA families in the human genome," *BMC Genomics,* vol. 9, 2008.

[180] J. Macas, P. Neumann, and A. Navratilova, "Repetitive DNA in the pea (Pisum sativum L.) genome: comprehensive characterization using 454 sequencing and comparison to soybean and Medicago truncatula," *BMC Genomics,* vol. 8, 2007.

[181] K. B. Mullis and F. A. Faloona, "Specific synthesis of DNA *in vitro* via a polymerase-catalyzed chain reaction," *Methods Enzymology,* vol. 155, p. 15, 1987.

[182] D. A. Ray, C. Feschotte, H. J. T. Pagan, J. D. Smith, E. J. Pritham, P. Arensburger, P. W. Atkinson, and N. L. Craig, "Multiple waves of recent DNA transposon activity in the bat, Myotis lucifugus," *Genome Research,* vol. 18, pp. 717-728, May 2008.

[183] D. A. Ray, H. J. T. Pagan, M. L. Thompson, and R. D. Stevens, "Bats with hATs: Evidence for Recent DNA Transposon Activity in Genus Myotis," *Molecular Biology and Evolution,* vol. 24, pp. 632-639, 2007.

[184] D. A. Ray, D. J. Hedges, S. W. Herke, J. D. Fowlkes, E. W. Barnes, D. K. LaVie, L. M. Goodwin, L. D. Densmore, and M. A. Batzer, "Chompy: An infestation of MITE-like repetitive elements in the crocodilian genome," *Gene,* vol. 362, pp. 1-10, 2005.

[185] J. K. Pace, C. Gilbert, M. S. Clark, and C. Feschotte, "Repeated horizontal transfer of a DNA transposon in mammals and other tetrapods," *PNAS,* vol. 105, pp. 17023-17028, 2008.

[186] C. Feschotte and S. R. Wessler, "Mariner-like transposases are widespread and diverse in flowering plants," *PNAS,* vol. 99, pp. 280-285, 2002.

[187] D. A. Vaughan, H. Morishima, and K. Kadowaki, "Diversity in the Oryza genus," *Current Opinion in Plant Biology,* vol. 6, pp. 139-146, 2003.

[188] O. F. Linares, "African rice (*Oryza glaberrima*): History and future potential," *PNAS,* vol. 99, pp. 16360-16365, 2002.

[189] L.-Z. Gao, "Microsatellite variation within and among populations of *Oryza officinalis* (Poaceae), an endangered wild rice from China," *Molecular Ecology,* vol. 14, pp. 4287-4297, 2005.

[190] T. Zhou, Y. Wang, J. Q. Chen, H. Araki, Z. Jing, K. Jiang, J. Shen, and D. Tian, "Genome-wide identification of NBS genes in japonica rice reveals significant expansion of divergent non-TIR NBS-LRR genes," *Molecular Genetics and Genomics,* vol. 271, pp. 402-415, 2004.

[191] Y. Belkhadir, R. Subramaniam, and J. L. Dangl, "Plant disease resistance protein signaling: NBS-LRR proteins and their partners," *Current Opinion in Plant Biology,* vol. 7, pp. 391-399, 2004.

[192] J. Ellis, P. Dodds, and T. Pryor, "Structure, function and evolution of plant disease resistance genes," *Current Opinion in Plant Biology,* vol. 3, pp. 278-284, 2000.

[193] D. Leister, J. Kurth, D. A. Laurie, M. Yano, T. Sasaki, K. Devos, A. Graner, and P. Schulze-Lefert, "Rapid reorganization of resistance gene homologues in cereal genomes," *PNAS,* vol. 95, pp. 370-375, 1998.

[194] A. R. Friedman and B. J. Baker, "The evolution of resistance genes in multi-protein plant resistance systems," *Current Opinion in Genetics & Development,* vol. 17, pp. 493-499, 2007.

[195] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology," *Nat Genet,* vol. 25, pp. 25-29, 2000.

[196] N. Jiang and S. R. Wessler, "SETARIA : a Mutator-like non-autonomous DNA transposon from Oryza," *Repbase Reports,* vol. 2, pp. 42-42, 2002.

[197] N. Juretic, T. E. Bureau, and R. M. Bruskiewich, "Transposable element annotation of the rice genome," *Bioinformatics,* vol. 20, pp. 155-160, 2003.

[198] T. Wicker, F. Sabot, A. Hua-Van, J. L. Bennetzen, P. Capy, B. Chalhoub, A. Flavell, P. Leroy, M. Morgante, O. Panaud, E. Paux, P. SanMiguel, and A. H. Schulman, "A unified classification system for eukaryotic transposable elements," *Nat Rev Genet,* vol. 8, pp. 973-982, 2007.

[199] T. H. Eickbush and H. S. Malik, "Origins and evolution of retrotransposons," in *Mobile DNA II*, N. L. Craig, R. Craigie, M. Gellert, and A. L. Lambowitz, Eds.: Washington, DC: ASM Press, 2002, pp. 1111 - 1144.

[200] P. Chouvarine, S. Saha, and D. G. Peterson, "An automated, high-throughput sequence read classification pipeline for preliminary genome characterization," *Analytical Biochemistry,* vol. 373, p. 9, 2008.

[201] R. Wiedmann, D. Nonneman, and J. Keele, "Novel porcine repetitive elements." vol. 7, 2006, p. 304.

[202] S. Ouyang and C. R. Buell, "The TIGR Plant Repeat Databases: a collective resource for the identification of repetitive sequences in plants," *Nucleic Acids Research,* vol. 32, pp. D360-363, January 1, 2004 2004.

[203] C. Chaparro, R. Guyot, A. Zuccolo, B. Piegu, and O. Panaud, "RetrOryza: a database of the rice LTR-retrotransposons," *Nucleic Acids Research,* p. gkl780, October 28, 2006 2006.

[204] M. Spannagl, O. Noubibou, D. Haase, L. Yang, H. Gundlach, T. Hindemitt, K. Klee, G. Haberer, H. Schoof, and K. F. X. Mayer, "MIPSPlantsDB--plant database resource for integrative and comparative plant genome research," *Nucleic Acids Research,* vol. 35, pp. D834-840, January 12, 2007 2007.

[205] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and a. D. Haussler, "The Human Genome Browser at UCSC," *Genome Research,* vol. 12, pp. 996-1006, 2002.

[206] A. A. Salamov and V. V. Solovyev, "Ab initio Gene Finding in Drosophila Genomic DNA," *Genome Research,* vol. 10, pp. 516-522, 2000.

[207] W. H. Majoros, M. Pertea, and S. L. Salzberg, "TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders," *Bioinformatics,* vol. 20, pp. 2878-2879, 2004.

[208] J. Xing, H. Wang, K. Han, D. A. Ray, C. H. Huang, L. G. Chemnick, C.-B. Stewart, T. R. Disotell, O. A. Ryder, and M. A. Batzer, "A mobile element based phylogeny of Old World monkeys," in *Molecular Phylogenetics and Evolution*. vol. 37, 2005, pp. 872-880.

[209] F. You, N. Huo, Y. Gu, M.-c. Luo, Y. Ma, D. Hane, G. Lazo, J. Dvorak, and O. Anderson, "BatchPrimer3: A high throughput web application for PCR and sequencing primer design," *BMC Bioinformatics,* vol. 9, p. 253, 2008.

[210] H. Sun, G. Skogerbo, X. Zheng, W. Liu, and Y. Li, "Genomic regions with distinct genomic distance conservation in vertebrate genomes," *BMC Genomics,* vol. 10, p. 133, 2009.

[211] R. Panstruga, R. Buschges, P. Piffanelli, and P. Schulze-Lefert, "A contiguous 60 kb genomic stretch from barley reveals molecular evidence for gene islands in a monocot genome," *Nucleic Acids Res.,* vol. 26, p. 1056, 1998.

[212] R. A. Martienssen, P. D. Rabinowicz, A. O'Shaughnessy, and W. R. McCombie, "Sequencing the maize genome," *Current Opinion in Plant Biology,* vol. 7, pp. 102-107, 2004.

[213] M. G. Kalyuzhnaya, N. Korotkova, G. Crowther, C. J. Marx, M. E. Lidstrom, and L. Chistoserdova, "Analysis of Gene Islands Involved in Methanopterin-Linked C1 Transfer Reactions Reveals New Functions and Provides Evolutionary Insights," *Journal of Bacteriology,* vol. 187, pp. 4607-4614, 2005.

[214] W. C. Lima, A. C. M. Paquola, A. M. Varani, M.-A. Van Sluys, and C. F. M. Menck, "Laterally transferred genomic islands in *Xanthomonadales* related to pathogenicity and primary metabolism," *FEMS Microbiology Letters,* vol. 281, pp. 87-97, 2008.

[215] W. C. Lima, M.-A. V. Sluys, and C. F. M. Menck, "Non-Gamma-Proteobacteria Gene Islands Contribute to the Xanthomonas Genome," *Omics,* vol. 9, pp. 160-172, 2005.

[216] U. Dobrindt, B. Hochhut, U. Hentschel, and J. Hacker, "Genomic islands in pathogenic and environmental microorganisms," *Nat Rev Micro,* vol. 2, pp. 414-424, 2004.

[217] H. J. Müller, "Types of viable variations induced by X-rays in Drosophila," *Genetics,* vol. 22, pp. 299-337, 1930.

[218] F. F. Assaad, K. L. Tucker, and E. R. Signer, "Epigenetic repeat-induced gene silencing (RIGS) in Arabidopsis," *Plant Molecular Biology,* vol. 22, pp. 1067-1085, 1993.

[219] Z. Lippman, A. V. Gendrel, M. Black, M. W. Vaughn, N. Dedhia, W. R. McCombie, K. Lavine, V. Mittal, B. May, and K. D. Kasschau, "Role of transposable elements in heterochromatin and epigenetic control," *Nature,* vol. 430, pp. 471-476, 2004.

[220] J. Bunge, P. Chouvarine, and D. G. Peterson, "CotQuest: Improved algorithm and software for nonlinear regression analysis of DNA reassociation kinetics data," *Analytical Biochemistry,* vol. 388, pp. 322-330, 2009.

APPENDIX A

RICE CHROMOSOME 12 RESULTS

This appendix contains a list of all the connected components identified by ProxMiner on rice chromosome 12. Each connected component represents a ProxMiner family. The connected components are classified [45, 128] into three categories: Confirms known repeat, Novel repetitive region and Extends known repeat. RepeatScout [44] has been used to discover the member families in each connected component. We have used RepeatMasker [128] with the Repbase [45] library of known repeats for identifying similarity to known repeats.

Table A.1

List of all ProxMiner families from rice chromosome 12

Component	Families	Category	Notes
0	R=683, R=499	Confirms known repeat	OSTE27#DNA annotations for both families
1	R=473, R=498	Novel repetitive region	No annotations
2	R=155, R=131	Confirms known repeat	Corresponds to putative retrotransposon Ty1-copia like
3	R=1296, R=177	Confirms known repeat	MuDR DNA transposon
4	R=549, R=1083	Confirms known repeat	RPO_OS SINE retrotransposon annotation for both families
5	R=551, R=326	Novel repetitive region	No annotations
6	R=431, R=819	Confirms known repeat	SZ-31_LTR#LTR/Gypsy annotations for both families.
7	R=706, R=272, R=706, R=149	Confirms known repeat	SZ-38_int# LTR/Gypsy annotations for all families.
8	R=290, R=364	Confirms known repeat	Strong Os6_10_2L#LTR from the gypsy super family annotations for both families
9	R=896, R=504	Confirms known repeat	R=896 as putative Ty3-gypsy RTRP. 504 is low copy and short.

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

10	R=707, R=393, R=1108, R=707	Confirms known repeat	Comp 48 with strong Os8_06_2L #LINE annotations for both families.
11	R=674, R=200	Novel repetitive region	Families found in this order: R=674, R=713, R=200, R=278. Usually downstream of unclassified transposon
12	R=430, R=438, R=725 , R=438	Confirms known repeat	OSTONOR1_LTR #LTR/Copia annotations for all families
13	R=6, R=134	Confirms known repeat	LTR annotations for both families
14	R=770, R=644	Novel repetitive region	Corresponds to hypothetical protein. Upstream of retrotransposon. Concentrated in one area of Chr12.
15	R=223, R=844	Confirms known repeat	Exist in a repeat rich region
16	R=890, R=639	Novel repetitive region	No annotations
17	R=324, R=42	Confirms known repeat	DNA transposon annotations
18	R=718, R=410	Confirms known repeat	Copia LTR annotations for both families
19	R=1010, R=455	Novel repetitive region	No annotations
20	R=970, R=293	Extends known repeat	MuDR Mutator transposase annotation for 970
21	R=995, R=1005	Extends known repeat	MuDR DNA transposon
22	R=782, R=140	Confirms known repeat	Strong MuDR Mutator transposase annotation for both families
23	R=810, R=394	Confirms known repeat	3 families. Fam 505 and 28 with 81 are in a different cluster
24	R=54, R=1017	Confirms known repeat	Strong LTR annotations for both families
25	R=162, R=105	Confirms known repeat	SINE
26	R=741, R=605	Novel repetitive region	No annotations
27	R=32, R=1039	Confirms known repeat	Explorer DNA transposon

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

28	R=120, R=758	Confirms known repeat	Both families 100% annotated as retrotransposons
29	R=100, R=534	Confirms known repeat	Both families annotated as retrotransposons
30	R=147, R=160	Novel repetitive region	No annotations
31	R=825 R=851	Novel repetitive region	Always downstream of transposon mariner and upstream of a putative protein. Mariners are not supposed to have 2 ORFs
32	R=86, R=166	Confirms known repeat	Gypsy-like LTR element, aligns with a putative centromere sequence
33	R=23, R=61	Confirms known repeat	Gypsy-like LTR, aligns with a putative retrotransposon
34	R=391, R=546	Novel repetitive region	No annotations
35	R=154, R=319, R=153	Confirms known repeat	Gypsy LTR
36	R=669, R=902	Extends known repeat	DNA transposon annotations
37	R=246, R=368, R=381, R=535, R=45, R=465, R=637	Confirms known repeat	Corresponds to unclassified retrotransposon
38	R=252 U(0.71) R=216	Extends known repeat	Gypsy LTR
39	R=596, R=938	Novel repetitive region	No annotations
40	R=336, R=41, R=168	Confirms known repeat	Gypsy LTR annotations
41	R=287, R=585, R=829	Confirms known repeat	hAT DNA transposon annotations
42	R=911, R=783	Novel repetitive region	No annotations
43	R=169, R=481	Confirms known repeat	Stowaway DNA transposon annotations
44	R=394 R=810	Extends known repeat	In repeat rich regions with a variety of annotations. 467 almost always found also. These are very short families

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

45	R=360, R=812, R=126, R=159	Confirms known repeat	MuDR DNA transposon
46	R=818, R=775	Confirms known repeat	Copia LTR annotations for both families
47	R=570, R=1103, R=264	Novel repetitive region	No annotations
49	R=471, R=17, R=215, R=854	Confirms known repeat	Connects a putative DNA transposon
50	R=270, R=439, R=903, R=503	Novel repetitive region	No annotations
51	R=375, R=124, R=406, R=288	Confirms known repeat	Connects an unclassified retrotransposon
52	R=156, R=249, R=340	Confirms known repeat	TEMPINDAS #DNA/hAT DNA transposon annotation
53	R=680, R=271, R=1082, R=373	Confirms known repeat	Corresponds to known retrotransposon. 387 is Gypsy type. 645 is SZ-43_LTR#LTR/Gypsy type.
54	R=4, R=444	Confirms known repeat	Tourist DNA transposon annotations
55	R=713, R=278, R=1080	Extends known repeat	1080 has DNA transposon annotation
56	R=358, R=889, R=589, R=587, R=645, R=387, R=1017, R=1147	Confirms known repeat	8 families. Maps to well annotated Gypsy region
57	R=437, R=165	Confirms known repeat	437 has OSLINE1-4# LINE/L1annot. 165 has ORMOSIA #SINE annotation
58	R=488, R=383	Confirms known repeat	Strong F569#DNA /Tourist annotations for both families.
59	R=688, R=401	Confirms known repeat	Strong ENSPM7_OS#DNA/ En-Spm annotations for both families.
60	R=337, R=219	Confirms known repeat	Lots of annotations to SC-3_LTR#LTR/Copia for both families
61	R=344, R=276	Confirms known repeat	MU_OS#DNA/MuDR annotations for both families.

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

62	R=69, R=744, R=695, R=179, R=847, R=883, R=367, R=629	Confirms known repeat	EnSPM_OS#DNA/En-Spm annotation for most families. EnSpm is a DNA transposon with 4 complete copies in rice.
63	R=505, R=28, R=81, R=98	Confirms known repeat	LTR Gypsy annotation for all families.
64	R=298, R=541, R=82, R=407, R=922, R=682	Confirms known repeat	hAT DNA transposon. Aligns with an Ac/Ds-like putative transposon
65	R=404, R=1095, R=462, R=449	Confirms known repeat	Corresponds to unclassified retrotransposon. There are nearby genes to almost all copies, but they are not well conserved.
66	R=925, R=649	Confirms known repeat	Conf 1.0. These 2 are TEOS DNA transposons.
67	R=658, R=396, R=397	Confirms known repeat	Comp 41 with COPI1_I#LTR/Copia annotation for all families.
68	R=414, R=361, R=421	Confirms known repeat	SZ-36_LTR#LTR annotations for all families
69	R=513, R=737, R=513, R=386	Confirms known repeat	LTR annotations for all families. No single annotation that is consistent.
70	R=466, R=181, R=1161, R=779, R=181	Confirms known repeat	OSTE33#DNA/MuDR annotations for all families. families are short.
71	R=170, R=94	Confirms known repeat	Full SZ-37_LTR#LTR/Copia annotations for both families
72	R=313, R=1144	Confirms known repeat	Full CRR1_CH1-1-CRR1-retrotransposon,-partial-sequence annotation for both families
73	R=137, R=218	Confirms known repeat	Full RETROSAT2_LTR#LTR/Gypsy annotations for both families
74	R=777, R=411, R=529	Confirms known repeat	Strong CACTA-F#DNA/En-Spm annotations for all families

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

75	R=1128, R=205, R=323, R=1019, R=536, R=321, R=303, R=483, R=620, R=564, R=370, R=118, R=873, R=423, R=893	Confirms known repeat	Corresponds to known retrotransposon
76	R=415, R=748, R=419, R=225	Novel repetitive region	Not conserved in other species. Sometimes next to other repetitive elements
77	R=308, R=953	Confirms known repeat	Full MUDRN3_OS#DNA/MuDR annotations for both families
78	R=759, R=657	Novel repetitive region	These two families found together alternating in 3 different areas. Other associated families are consistent. Lots of MITES in the area
79	R=563, R=720, R=283, R=339, R=981	Confirms known repeat	Full ATLANTYS-I_OS#LTR/Gypsy(RB) for all families besides others
80	R=330, R=301, R=1026, R=918	Confirms known repeat	Full SZ-24_LTR#LTR(MIPS) annotations for most families.
81	R=206, R=487	Confirms known repeat	Full LTR RTRP annotation for 487.
82	R=787, R=338, R=238, R=888, R=609, R=19	Confirms known repeat	CACTA-I#DNA/En-Spm or SPMLIKE#DNA/En-Spm annotations for all families
83	R=385, R=102, R=15, R=347, R=248, R=139, R=441, R=187	Confirms known repeat	Gypsy LTR annotations
84	R=418, R=956, R=398	Novel repetitive region	No annotations
85	R=547, R=697	Confirms known repeat	Full SZ-36_LTR#LTR annotations for both families
86	R=615, R=48, R=282, R=163, R=359	Confirms known repeat	Full Os7_09_1L#Retroelement, Gypsy-B_int#LTR/Gypsy and GYPSY1-I_OS#LTR/Gypsy annotations for all families

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

87	R=739, R=240, R=739, R=189	Confirms known repeat	Full Gypsy-B_int#LTR/Gypsy and GYPSY1-LTR_OS#LTR/Gypsy annotations for all families
88	R=1059, R=409, R=29, R=625, R=652	Confirms known repeat	Aligns with several MULEs and a number of putative expressed genes
89	R=940, R=987	Confirms known repeat	Aligns with DNA transposons
90	R=490, R=642, R=1112, R=822, R=924	Confirms known repeat	DNA transposon annotation
91	R=672, R=227, R=167, R=941, R=1107	Confirms known repeat	Gypsy LTR annotations. Some unannotated families.
92	R=457, R=51, R=1252, R=309, R=604, R=241, R=790, R=671, R=477, R=856, R=726, R=580, R=312, R=325, R=88	Confirms known repeat	DNA transposon annotations. Some unannotated families
93	R=116, R=353	Confirms known repeat	hAT DNA transposon
94	R=446, R=52, R=484, R=327, R=197	Confirms known repeat	Copia LTR annotations
95	R=12, R=558	Confirms known repeat	Gypsy LTR annotations
96	R=963, R=785, R=107, R=661	Confirms known repeat	Copia LTR annotations
97	R=33, R=43, R=328, R=573, R=260, R=305, R=557, R=520, R=243, R=590, R=691, R=628, R=1034, R=191, R=132	Confirms known repeat	Gypsy LTR annotations
98	R=799, R=871	Novel repetitive region	No annotations
99	R=184, R=769, R=999	Confirms known repeat	Gypsy LTR annotations

Table A.1 (contd.)

List of all ProxMiner families from rice chromosome 12

100	R=192, R=901, R=592, R=515, R=794, R=659, R=614, R=824, R=633, R=663, R=608, R=255, R=334, R=348, R=501,R=151, R=752, R=576, R=250, R=650, R=1132, R=468, R=730, R=426	Confirms known repeat	DNA En/Spm transposon
101	R=236, R=714	Confirms known repeat	STOWAWAY36_OS#DNA annotation for both families.
102	R=333, R=79	Confirms known repeat	Gypsy LTR annotations
103	R=351, R=14, R=70, R=390, R=320	Confirms known repeat	Gypsy LTR annotations
104	R=104, R=662, R=865	Confirms known repeat	Gypsy LTR annotations
105	R=366, R=500	Novel repetitive region	No annotations
106	R=150, R=1122, R=58, R=584	Confirms known repeat	RIRE3A_LTR#LTR/Gypsy and Os6_13_1L#LTR/Gypsy annotations for families. An LTR region
107	R=1198, R=105, R=1020	Confirms known repeat	SINE
108	R=30, R=75, R=89	Confirms known repeat	Copia LTR annotations
109	R=1071, R=852	Confirms known repeat	DNA transposon
110	R=235, R=1069, R=198, R=525, R=1027, R=299, R=228, R=417, R=99, R=959, R=489, R=194, R=307, R=1154, R=395, R=442, R=416	Confirms known repeat	CACTA, En/Spm sub-class
111	R=269, R=1143, R=508	Confirms known repeat	Gypsy LTR with Pack MULE with ULP1 protease domain

APPENDIX B

FEEDBACK FROM BIOLOGISTS

This appendix contains the feedback received from molecular biologists in the Department of Plant and Soil Sciences at Mississippi State University. The evaluators were provided with a short introduction to ProxMiner and then given a set of repeat families from ProxMiner results to analyze. They were then asked to complete a user survey. Their answers to the survey are given below.

Evaluator 1

Zenaida V. Magbanua, Ph.D.
Department of Plant and Soil Sciences
Mississippi State University
Mississippi State, MS 39762
Phone: 662-325-2455
www.mgel.msstate.edu

1. What computational tools do you use to discover and annotate repeats? What output do those tools produce?

   I mainly use BLAST, which gives an alignment of the sequences that aligned with my query sequence and the alignment score.

2. In absence of ProxMiner

   a. How could you have otherwise discovered the repeats found by ProxMiner on Chr 12?

      BLAST

b. How much time would it have taken?

   Several weeks to a few months.

c. Compare the information you would obtain using some other method to the information provided by ProxMiner.

   The information from ProxMiner gives a good detail and summary of the results. It also allows the user to go back at the actual browser and visualize each data point locally and globally, to a certain extent. These are two important aspects of data analysis.

3. Do you find the output of ProxMiner useful? Why?

   Yes. It's easy to understand.

4. How much time did it take for you to understand and use ProxMiners output?

   A few sessions.

5. What additional information would you like ProxMiner to give you?

   I can't think of any additional information at this time.

6. How would you describe the novelty of ProxMiner?

   It's the first program that deals with association studies in repeats. This is an area of genomics research that is not yet explored and having this tool will definitely advance it.

7. What did you like best about ProxMiner? What did you like the least?

It's user-friendly and smart. We had a problem uploading the repeats file to the rice browser and though it's not a problem with the tool, I hope it can be addressed when it is up and running.

Evaluator 2

Daniel G. Peterson, Ph.D.

Associate Professor

Department of Plant and Soil Sciences

117 Dorman Hall, Box 9555

Mississippi State University

Mississippi State, MS 39762

phone: (662) 325-2747

fax: (662) 325-8742

website: www.mgel.msstate.edu

1. What computational tools do you use to discover and annotate repeats? What output do those tools produce?

   I utilize RepeatMasker and the various BLAST algorithms available through NCBI to identify and annotate repeats. My lab also uses a "sequence read classification pipeline" (SCRP) customized repeat identification pipeline [200] which includes analysis via RepeatScout.

2. In absence of ProxMiner

   a. How could you have otherwise discovered the repeats found by ProxMiner on Chr 12?

      RepeatMasker would likely be used to identify repeats with similarity to known repeat classes. *Ab initio* repeat identification of repeats could potentially be performed using the SCRP or RepeatScout. ProxMiner identified some repeats not identified by RepeatMasker and RepeatScout. These repeats may have been identified by the SCRP, but the SCRP uses sequence reads as data and thus it doesn't really tell one much about repeat spatial organization.

b. How much time would it have taken?

Identification of the repeat content of a whole chromosome would probably take close to three weeks. Annotation and characterization of the repeats would probably be a 6 month process.

c. Compare the information you would obtain using some other method to the information provided by ProxMiner.

No, thank you.

3. Do you find the output of ProxMiner useful? Why?

The ProxMiner output can be very useful. Of note, I am unaware of any DNA sequence analysis program that identifies and describes spatial relationships between repeat families.

4. How much time did it take for you to understand and use ProxMiners output?

Mr. Saha did a good job of walking me through the output data so that I could understand what the various output values meant. However, I could not have made head or tails of the data if I didn't have his help.

5. What additional information would you like ProxMiner to give you?

ProxMiner is an extremely powerful tool. However, it lacks a user's guide. It is my understanding that a guide will be written in the near future. The guide will need a clearly defined vocabulary list. What is a family? What is a connected component? What are the various statistical values produced and what do they mean? These things need to be defined clearly (and when possible, concisely).

I think data from an analysis needs to be distilled down into an HTML results page with links to more detailed results, graphs, etc. I recommend something similar to Bunge, Chouvarine, and Peterson [220].

6. How would you describe the novelty of ProxMiner?

   To my knowledge, the proximity relationship mining performed by ProxMiner is novel among sequence analysis tools.

7. What did you like best about ProxMiner? What did you like the least?

   ProxMiner potentially opens doorways to new lines of research on the "migration patterns" of different genomic elements. This has tremendous implications in comparative genomics. I guess I like this potential of ProxMiner best. I think the program needs lots of work on the "user friendliness" front if it is to actually be used by other research groups.

APPENDIX C

POLYMERASE CHAIN REACTION

Polymerase chain reaction (PCR) [181] is a molecular biology technique to generate duplicates of a DNA substring present in a given DNA sample using a biochemical reaction. The DNA substring is targeted using short DNA strings called primers to initiate the duplication process. The duplication process is called amplification. PCR essentially displays the presence or absence of the targeted DNA substring in the given DNA sample along with the size of the region, if present. PCR is one of the most widely used techniques in a molecular biology laboratory and is a part of many experimental protocols. Xing *et al.* [1, 208] have explained the utility of PCR for investigating repeat regions and used PCR to study mobile element insertions in DNA of primates and humans.

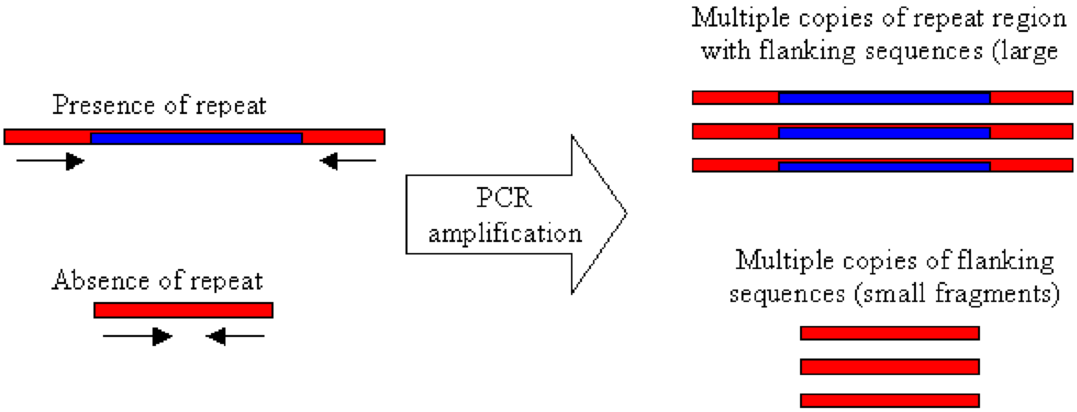

Presence of repeat
Absence of repeat
PCR amplification
Multiple copies of repeat region with flanking sequences (large
Multiple copies of flanking sequences (small fragments)

Figure C.1

Description of PCR technique to generate copies of a target repeat region adapted from Xing *et al.* [1]

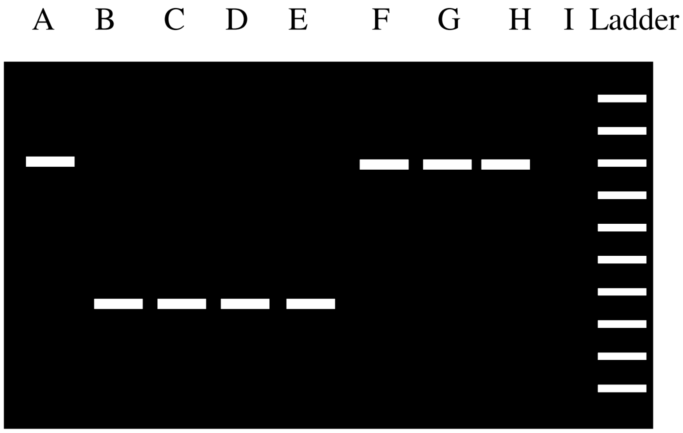


Figure C.2

Schematic of results of a PCR experiment

The PCR experiment (Figure C.1) starts with DNA from multiple organisms and sets of primers. Each set of primers targets a specific region of DNA and includes a forward primer and a reverse primer. The primers initiate the copying mechanism that copies or amplifies the target DNA region. The PCR product contains many copies of the DNA fragment with the target region if the region is present. The product contains only copies of the flanking DNA sequences if the target DNA region is absent. A schematic of a PCR experiment is shown in Figure C.2 Columns or lanes A-H contain DNA from different species and each bar is an amplified DNA fragment. The last column is a ladder, which is a set of DNA fragments of known size that acts as a scale for calculating the size of the amplified fragments produced by the different DNA samples. The higher bars (A, F, G and H) are the longer fragments that show the presence of the target DNA region. The lower bars (B, C, D and E) represent shorter DNA fragments that denote absence of the target DNA region. Lane I contains DNA from a negative control. A negative control in a PCR experiment is DNA where we do not expect our target DNA region.

We have used the genome sequence from *Oryza sativa* (Nipponbare, japonica cultivar group) for our experimental dataset [164]. We have selected a set of related rice species to explore for the presence of the repeat families we have discovered. These species are *Oryza sativa* (indica), *Oryza glaberrima* , *Oryza nivara*, *Oryza rufipogon* and *Oryza officinalis*. We are using DNA from *Arabidopsis thaliana* and water as the negative controls. Table C.1 lists the DNA sequence for the primer pairs we have used in our experiments. We have designed 2 sets of primers for each repeat family, when possible. The PCR experimental protocol followed is presented in Table C.2 and the chemicals used for the reaction in Table C.3.

Table C.1

Primers used for amplifying DR1, DR2 and DR3 in PCR experiments

Primer	Comment	Sequence (5'-3')
DR1clf	DR1 cluster, forward	TTGATGGGTTGA[AG]ATCATGC
DR1clr	DR1 cluster, reverse	AGATGTGCT[TC]TGCCAGAAAAT
DR1flkf	DR1 flanking, forward	ATTCTAAATCAAATACACGTAGA
DR1flkr	DR1 flanking, reverse	TCAAATTGTTATAAACTATCCGTTAAG
DR2clf	DR2 cluster, forward	GCGATCTTGTAGTAGCCGTTG
DR2clr	DR2 cluster, reverse	TCAAATCCGGTCGAATCTTT
DR3clf	DR1 cluster, forward	GAAACAACCGTTTTCCCGTA
DR3clr	DR2 cluster, reverse	GCATT[TA]TGCCCCAATTTTT
DR3flkf	DR3 flanking, forward	AAAAATGGGGATGAAAGTATAGG
DR3flkr	DR3 flanking, reverse	TTAGGGGCAATTGTGTTTTTG

Table C.2

Protocol used for PCR experiments

Step	Temp (^{0}C)	Time (min)
Step 1	94	5
Step 2	94	1
(30 cycles)	Annealing temp	0.5
	72	1
Step 3	72	5

Table C.3

Reactants used for a PCR experiment

Reactant	Amount or final concentration
Genomic DNA	200 ng
NEB Taq buffer	1X
dNTPs	0.25 µM each
Forward primer	0.3 µM
Reverse primer	0.3 µM
Taq polymerase	2.5 units
Double-distilled water	To make 50 mL