IDENTIFICATION OF TANDEM REPEATS: SIMPLE AND COMPLEX PATTERN STRUCTURES IN DNA SEQUENCES

by

Amy M. Hauth

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Computational Molecular Biology)

at the

UNIVERSITY OF WISCONSIN-MADISON

2002

ACKNOWLEDGMENTS

I want to express my appreciation to Deb Joseph, my research advisor, my fellow graduate students and to the support staff of the Computer Sciences Department for facilitating my research through laughter, fresh perspectives and the occasional prod. In addition, many people in the Madison community have been a source of encouragement. In particular, I thank my friends at Faith Community Bible Church for their support, their Faith and their encouragement. Finally from beginning to end, I thank my parents and siblings for their love and support. Thanks to all for keeping me pointed in the right direction.

I dedicate this work to my husband, Karl, and my children, Amelia and Daniel. Thank you for your *laughter* and for *listening* and especially for your *love*.

Now to *him* who is able to do immeasurably more than all
we ask or imagine, according to *his* power that is at work
within us, to *him* be glory in the church and in *Christ Jesus*
throughout all generations, forever and ever! Amen.
Ephesians 3: 20, 21.

ABSTRACT

Sequence duplication is one process that enables DNA to flexibly adapt and evolve in a changing environment. Duplication creates sequence repetition that over time may mutate to form unique sequence. Sequence repetition present in DNA is interesting biologically in the context of its role in evolution, its association with human congenital diseases and cancer, its occurrence both within genes and to contain genes and its regulatory function. Despite the importance of repetitive DNA, locating and characterizing repetitive patterns within anonymous DNA sequences remains a challenge. In part, the difficulty is due to imperfect pattern conservation and complex pattern structures. This dissertation describes and identifies complex pattern structures associated with tandem repeats and locates non-contiguous regions of similarity associated with interspersed repeats, gene clusters and other dispersed, related sequences.

The difficulty with locating and characterizing tandem repeat regions can be attributed, in part, to the formation of complex pattern structures and imperfect pattern conservation. This research defines a class of regular tandem repeats (RegTRs), as well as, two important subclasses; variable length tandem repeats (VLTRs) and multi-periodic tandem repeats (MPTRs). A tandem repeat identification algorithm locates and characterizes regions having both simple pattern structures and complex pattern structures associated with VLTRs and MPTRs without prior knowledge of the nature of the tandem repetition. Furthermore, the algorithm identifies degenerate MPTRs, VLTRs and regions with simple pattern structures; imperfectly conserved repeats containing substitutions, insertions and deletions.

An extension to the tandem repeat identification algorithm locates similarity between two non-contiguous regions. A proof of concept algorithm locates Alu sequences, long terminal

repeats, related tandem repeat regions, distant yet similar genes, gene clusters and other similar features in DNA sequences.

Access to these algorithms is available through a collection of HTML-based webpages generated by a companion program. A webpage interface enables a researcher to submit a DNA sequence for analysis. Once analysis is complete, the program generates a webpage that displays tandem repeat regions and regions of similarity in the sequence in several forms; as graphic images, as alignments of copies within a tandem repeat region and as tables containing region specific information.

TABLE OF CONTENTS (BRIEF)

TABLE OF CONTENTS (EXPANDED)

LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
tRNA	Transfer RNA
SNP	Single nucleotide polymorphism
TR	Tandem repeat
SSR	Simple sequence repeat
VNTR	Variable number tandem repeat
SINE	Short interspersed element
LINE	Long interspersed element
LTR	Long terminal repeat
RegTR	Regular tandem repeat
MPTR	Multi-periodic tandem repeat
VLTR	Variable length tandem repeat
DP	Dynamic programming
WDP	Wraparound dynamic programming
RegWDP	WDP using regular expressions
ROS	Regions of similarity

LIST OF DEFINITIONS

Repeat Recurrence of a pattern

Direct repeat The pattern recurs on the same strand in the same nucleotide order

Indirect repeat The pattern recurs on the same strand but, the nucleotide order is reversed

Complement The pattern recurs on the same strand in the same nucleotide order but, the nucleotides are complemented such that A and T replace one another and C and G replace one another

Reverse complement The pattern recurs on the same strand but, the nucleotides are complemented and the nucleotide order is reversed

Forward repeat See direct repeat

Reverse repeat See indirect repeat

Inverse repeat See indirect repeat

Palindrome A sequence that reads the same in the forward and reverse direction

Genetic palindrome A sequence that reads the same in the forward and reverse complement direction

Tandem repetition Consecutive head-to-tail recurrence of a pattern

Interspersed repetition Recurrence of a pattern that may or may not be proximal

Tandem repeat Repetitive element exhibiting direct, tandem repetition

Interspersed repeat Repetitive element exhibiting interspersed repetition, direct or otherwise

Regions of similarity Two sequence similar regions that do not exhibit tandem repetition

Period The length of the pattern associated with a tandem repeat region

Periodicity Tandem repeat region descriptor captures how often a subsequence recurs within the region.

LIST OF FIGURES AND TABLES

1. Introduction

Scenario: *A medical condition occurs widely within an extended family. Individuals with the condition require substantial medical attention throughout their short life. Doctors believe it is a genetic condition passed from parent to child. Researchers determine the region on a DNA chromosome causing the "abnormality." The analysis reveals a nucleotide triplet that exhibits high tandem repetition.*

Several familial diseases are caused by a change in the number of copies (or copy number) associated with a tandem repeat region contained within a gene. Fragile X Syndrome is a mental impairment disease resulting from high tandem repetition of a CGG nucleotide triplet (See Section 2.1.3.3). Occurrence in most of the population is between seven and sixty tandem copies of the triplet with thirty copies being the most prevalent version. Individuals with Fragile X Syndrome have over two hundred CGG copies. This region becomes highly susceptible to methylation of the cytosine (C) nucleotides, perhaps enabling strong nucleosome binding or forming alternative DNA structures and thereby, blocking transcription of the gene and eliminating the presence of the associated protein (for more details see Section 2.1.3.2 and http://www.yourgenesyourhealth.org/).

Scenario: *An investigator on a murder scene finds a single hair. The hair is carefully picked up, placed in a bag and sent to a laboratory for analysis. The hope is that it will lead to the murderer. In actuality, it could belong to the family cat. Analysis of the hair reveals the species and provides a "fingerprint" of an individual. The investigator has a similar analysis performed on the blood or hair of a suspect. A match provides the investigator with one more piece of* *evidence placing the suspect at the scene of the crime. Forensic analysis of hair and other items containing DNA is known as DNA fingerprinting.*

Forensic science uses DNA fingerprinting extensively to support criminal investigations. In 1985, Jeffreys *et al.* (1985a, 1985b) presented techniques for fingerprinting human DNA. In England in 1987, Colin Pitchfork was convicted of rape and murder of two teenage girls based solely on his DNA (Alder & McCormick 1998). Investigators took blood samples of every man between 13 and 30 in three nearby villages and performed DNA fingerprinting before identifying him as a suspect. In May 1993, Mark Bogan also was convicted of murder (Erickson 1993). In his truck, police officers found seed pods from a *palo verde* tree that were later linked to trees at the crime scene.

<u>Scenario</u>: *A leading producer of seed corn markets varieties having the highest production yield levels per acre. As a result, the producer dominates the seed corn market. Recently, one competitor has steadily increased its production yield levels and is garnishing a larger market share. Surprised and concerned, the producer orders a full analysis of the competitor's varieties using various techniques including DNA fingerprinting. The tests identify a striking similarity between the producer's patented varieties and the competitor's varieties. The producer sues for patent infringement.*

Companies and individuals that breed plants and animals routinely use DNA fingerprints for pedigree identification. In 1987, Jeffreys and Morton presented techniques for fingerprinting dog and cat DNA. In 1991, Pioneer Hi-Bred International Inc. won a decade long case alleging Holden Foundation Seeds Inc. stole Pioneer's proprietary corn breeding materials. Key evidence included a comparison of DNA fingerprints of Holden's seed corn and Pioneer's corn germplasm in which the odds of developing seed corn so similar to Pioneer's was one in a trillion (a 1994 appeals court agreed, Jordon 1994).

DNA fingerprinting has a wide base of applications including forensics, plant and animal parentage analysis and even anthrax and salmonella strain identification. Original DNA fingerprinting techniques (Jeffreys *et al.* 1985a, 1985b) used variability in the copy number of tandem repeats to distinguish between individuals. Since then, other types of DNA fingerprinting have been developed although most analyze genomic differences of repetitive regions in DNA.

A rapid growth in interest of repetitive DNA has occurred during the past decade. Most is spurred by the development of new technologies and the push to understand the human genome. Medical research has researched inheritable diseases, isolating several as differences in copy number of a tandem repeat. Geneticists are isolating more transposable elements, are unlocking the role of tandem repeats in telomeric and centromeric regions and, in general, are determining the structural and regulatory function of repetitive DNA.

1.1. Defining repetition

A *repeat* is a recurrence of a pattern. DNA contains repetition of many features. Genes for transfer RNAs (tRNAs) are abundant in genomes and retain similar sequences. Gene clusters that likely evolved via sequence duplication are groups of genes proximally located having similar sequence and often similar, though different, function. Promoters for polymerase I, II and III are abundant. Stem loop structures recur throughout genomes. All are repetitions. None are known as a ***repetitive element*** or as ***repetitive DNA***.

A repeat is the recurrence of a *pattern*. Patterns occur in most spheres of knowledge. Music, dance, art, landscaping, architecture, political structures, chemistry, biology and genetics are a few disciplines where patterns are an integral aspect of the discipline. In music, musicians play jovial tunes using one *pattern of notes*, or a key, while they play sad tunes using a different key. When building a home, a carpenter places studs sixteen inches apart. In chemistry, a recurring *pattern of atoms* forms a uniform crystal. In proteins, formation of an α-helix involves a *pattern of hydrogen bonding* between the amino acids. In DNA, patterns recur as well.

A repeat is a *recurrence* of a pattern. A DNA pattern recurs in four ways; ***direct***, ***indirect***, ***complement*** or ***reverse complement***. A ***direct*** or ***forward repeat*** is the same pattern recurring on the same strand in the same nucleotide order, e.g. **ACCG** recurs as **ACCG**. An ***indirect***, ***inverse*** or ***reverse repeat*** recurs on the same strand but, the order of the nucleotides is reversed, e.g. the indirect recurrence of **ACCG** is **GCCA**. A ***complement repeat*** recurs on the same strand with the order preserved but, the nucleotides are complemented such that **A** and **T** replace one another and **C** and **G** replace one another, e.g. the complement of **ACCG** is **TGGC**. A ***reverse complement repeat*** recurs on the same strand but, the nucleotides are complemented and the order of the nucleotides is reversed, e.g. the reverse complement of **ACCG** is **CGGT**. In DNA, most repetitions occur as forward or reverse complement repeats and rarely as reverse or complement repeats (Grumbach & Tahi 1994).

A palindrome is a special recurrence combining two consecutive occurrences in opposite orientations. In language, a ***palindrome*** is a word, or phrase, whose letters form the same word or phrase when read from left to right or from right to left, e.g. madam. A ***genetic palindrome*** (i.e. a palindrome in DNA) is a sequence that reads the same in the forward and reverse complement directions, e.g. **`ACCATGGT`**. Typically, genetic palindromes contain a non-repeated spacer region between the pattern and its reverse complement. These allow stem loop structures to form by pairing the pattern with its reverse complement.

Recurrence can be characterized as either ***tandem*** or ***interspersed***. ***Tandem*** repetition is the consecutive head-to-tail recurrence of a pattern and is the result of local duplication. ***Interspersed*** repetition is the recurrence of a pattern that may or may not be proximal. Interspersed repeats are formed by either non-local duplication or multiple introductions of the same or similar extraneous DNA segment. Repetitive DNA elements are characterized as either tandem or interspersed. A ***tandem repeat*** designates a repetitive element exhibiting direct, tandem repetition. An ***interspersed repeat*** designates a repetitive element exhibiting interspersed repetition, direct or otherwise.

A repeat is a recurrence of a pattern. This dissertation presents algorithms that identify recurring patterns exhibiting both tandem and interspersed repetition. Tandem repetitive regions recur in both simple and complex pattern structures. Non-contiguous ***regions of similarity*** indicate dispersed sequence similarities. The aim of this research is to locate and characterize repetitive features present in DNA.

1.2. Contributions of this dissertation

This dissertation presents an algorithm for locating and characterizing repetitive elements in anonymous DNA sequences. The research focuses on tandem repeat identification and is the first work to present algorithms for identifying tandem repeats having complex pattern structures. In addition, non-contiguous regions of similarity link significant sequence similarities such as those associated with interspersed repeats.

The primary goal of this work is to develop computational techniques for identifying naturally occurring regions of DNA repetition. To determine the blueprint for real DNA repeats considerable effort was expended to find and evaluate tandem repeats in GenBank sequence data. The data provided the basis and motivation for the remainder of this work. Four steps describe the ensuing process:

1. Creation of a data model to represent observed tandem repeat regions
2. Selection of a collection of tandem repeat regions in GenBank sequence representing both easy and challenging regions
3. Development of an identification algorithm
4. Analysis of sequences in the GenBank collection

The complex pattern structure model represents significant, structural tandem repetitive components observed in the extensive analysis of sequence data. The sequence collection is representative of repeats in a variety of species, of regions with simple and complex pattern structures having both easily identifiable and challenging characteristics. The algorithm was developed to capture the intricacies represented by the sequence collection and was cross-checked against other sequence data and the success of other identification algorithms to ensure broad identification of repeats.

The final algorithm is a collection of techniques that identifies tandem repeat regions having both simple and complex pattern structures without prior knowledge of either the pattern or the pattern length, i.e. the ***period***. An extensive period analysis facilitates identification of regions with poor pattern conservation and complex pattern structures. The approach identifies regions with short patterns and patterns several hundred nucleotides in length and places no artificial limit on the pattern length. The performance of this approach is comparable to other identification algorithms for simple pattern structures but this approach also identifies regions having complex pattern structures.

In addition, an extension to the algorithm explores identification of other types of repetition. In particular, it identifies non-contiguous, regions of similarity such as interspersed repeats, gene clusters and other dispersed, related sequences. Results are preliminary.

1.3. Bibliography

Alder, J. and McCormick, J. (1998) The DNA detectives. Newsweek 16 November:66-70.

Erickson, J. (1993) Tree genes: UA professor's DNA work helps convict killer. Arizona Daily Star, 28 May: 1B.

Grumbach, S. and Tahi, F. (1994) A new challenge for compression algorithms - genetic sequences. Information Processing & Management **30**(6):875-886.

Jeffreys, A.J., Wilson, V. and Thein, S.L. (1985a) Hypervariable 'minisatellite' regions in human DNA. Nature **314**(6006):67-73.

Jeffreys, A.J., Wilson, V. and Thein, S.L. (1985b) Individual-specific 'fingerprints' of human DNA. Nature **316**(6023):76-79.

Jeffreys, A.J. and Morton, D.B. (1987) DNA fingerprints of dogs and cats. Animal Genetics **18**(1):1-15.

Jordon, S. (1994) Seed corn firm to pay Pioneer over disputed genetic secrets. Omaha World – Herald, Omaha, Nebraska, 13 July:22.

(This page left intentionally blank)

2. Background

Identification and analysis of repetitive DNA is an active area of biological and computational research. Biological researchers study the biology of repetitive elements while characterizing heterochromatic regions that contain high concentrations of tandem repeats, by identifying "jumping genes" within transposable elements and when linking genetic diseases and DNA fingerprints to hypervariable tandem repeat regions. On the other hand, computational researchers develop computational algorithms to locate repetitive elements. Early algorithms found interspersed repeats by sequence similarity to a consensus sequence and located tandem repeats by tandem occurrence of perfect short sequences. Recent algorithms locate tandem recurrences of degenerate sequences and locate non-contiguous recurrences of similar sequences associated with tandem and interspersed repeats and features with multiple copies within a genome.

This section describes the current knowledge concerning repetitive elements in DNA sequence. Section 2.1 describes biological recurrences that occur in DNA sequences. Section 2.2 describes various techniques borrowed from computer science to locate recurrences. Finally, Section 2.3 describes the current algorithms for locating tandem and interspersed repeats within DNA sequences.

2.1. Biology/genetics: recurrences in DNA sequences

This section describes a wide range of topics regarding DNA sequence data. Section 2.1.1 discusses biological classification of tandem and interspersed repeats. Section 2.1.2 describes the structure of other recurring genomic features. In addition, Section 2.1.3 discusses structural and functional roles, association with degenerative diseases and application to DNA fingerprints.

2.1.12. Biological classes of repetitive DNA

The biological literature contains many repetitive DNA classification schemas. Each schema classifies repetitive DNA characteristics as measured by different techniques. A consolidation of these systems defines four broad classes: satellites, microsatellites and minisatellites, retroposons and proretroviral transposons. The description of each class will focus upon a structural definition of the elements in the class. Other genomic features recurring in DNA are discussed in Section 2.1.2.

The classification scheme makes a distinction between repetitive regions exhibiting tandem repetition and interspersed repetition (defined in Section 1.1). Classification of elements as tandem or interspersed is not precise since each class retains characteristics of both. Nevertheless, tandem and interspersed repetition are prominent in biological descriptions of repetitive classes and are important structurally for computational identification.

The number of occurrences of a pattern is referred to as the *copy number*. The *region copy number* refers to the number of copies in a particular tandem repetitive region. The *genome copy number* refers to the number of copies of tandem or interspersed repeats in the entire genome.

2.1.12.1. Tandem repeats

Tandem repeat regions in DNA exhibit *periodic* recurrence of the same sequence of nucleotides. The pattern structure underlying this periodicity is either a simple sequence of nucleotides or a complex pattern structure having multiple periodicities. The pattern structures vary from several nucleotides to thousands of nucleotides in length and vary from several copies of the pattern to many pattern copies. Regions occur in specific locations in a genome, e.g. in telomeric regions, but also are scattered throughout the genome. Furthermore, some regions have structural or functional roles in the genome (for more details, see Section 2.1.3.2) and other regions appear to have no role.

Satellite DNA is a common term that can refer to all or a subset of tandem repeats. DNA density studies led to the discovery of tandem repeats having large copy numbers, termed satellites. As tandem repeats were discovered in different locations and exhibiting different copy numbers, new terms arose such as minisatellite and microsatellite. Today, the distinction between satellite, microsatellite and minisatellite regions is less prominent. Some researchers refer to all types of satellites as tandem repeats and describe a specific tandem repeat region according to its location within the genome, its periodicity, its pattern structure and its copy number.

Satellite DNA. These repeats were first identified on a CsCl buoyant density gradient as peaks separate from the primary DNA peak. The separate or satellite peaks were composed of large arrays of highly conserved tandem repeats. Hybridization studies localized the large tandem arrays to heterochromatic regions of chromosomes, areas such as centromeres. Later studies identified individual regions to contain over 5,000 copies of a pattern and individual peaks in the gradient representing a single pattern throughout the genome to contain over 100,000 copies.

Centromeres are composed predominately of tandem repeats (Schueler *et al.* 2001). The structure of a tandem repeat region revolves around a well-conserved pattern. The pattern varies in size from less than 20 basepairs (bp) to several thousand basepairs. In humans, the basic pattern is 171 bp long.

Some tandem repeats in centromeric regions have *structurally complex, pattern structures*. For example (Pech *et al.* 1979), a calf satellite region is defined by a 2,200 bp pattern, a 23 bp pattern and an 11/12 bp pattern. Specifically, the 2,200 bp pattern is composed of four subunits such that two subunits, one a 1,200 bp pattern and the other a 650 bp pattern, are composed primarily of a 23 bp pattern. The 23 bp pattern is composed of two similar subunits, one an 11 bp pattern and the other a 12 bp pattern. Regions such as this exhibit a collection of periodicities, one for each level of recurrence within the pattern structure. These multiple periodicities cause the region to be difficult to analyze and difficult to characterize with a pattern or pattern structure.

Microsatellite and minisatellite DNA. These are tandem repeats that are found throughout the genome though predominately outside of heterochromatic regions (Vogt 1990) and rarely within genes. Most regions contain short to moderate region copy number (Jeffreys 1985a). Minisatellites are characterized by moderate length patterns, usually less than 50 bp (Jeffreys 1985a).

Microsatellites, also known as *simple sequence repeats* (***SSRs***) or *simple tandem repeats* (***STRs***), have a short, well-conserved pattern length arbitrarily set at 2 to 6 bp and typically have a region copy number of 10 to 40 pattern copies. Microsatellite regions have been identified in all non-centromeric regions with many located near genes and some, within genes.

Hypervariable regions, also known as *variable number of tandem repeats* (***VNTRs***), are regions that contain a variable copy number. Two types of variability are observed. One displays copy number variation with each replication event. The other displays distinct alleles within a population such that different alleles contain different copy numbers. DNA fingerprinting capitalizes on the differences between alleles at specific VNTR loci in the genome. Various human diseases are attributed to high copy numbers associated with a specific VNTR locus (see Section 2.1.3.3).

Automatic identification and characterization of tandem repeats is crucial as genome projects generate an ever-increasing quantity of sequence data. Tandem repeats increase the complexity of most full sequence analysis algorithms. For instance, the process of generating full chromosome sequences often utilizes the sequence assembly procedure; a procedure that stitches short, similar fragments together to reconstruct a larger sequence. The consecutive recurrence of a pattern associated with tandem repeats confuses this process. Some commercially available algorithms avoid assembling tandem repeat regions. Others often assemble moderate-sized tandem repeat regions improperly. At present, algorithms are being developed that offer a process for handling tandem repeat regions.

2.1.12.2. Interspersed repeats

Interspersed repeats disperse throughout the genome and have no restriction on the relative positions of identical occurrences occurring in tandem and in non-contiguous locations. Research indicates that interspersed repeats are inserts since they resemble either processed RNAs, i.e. *retroposons* (see Section 2.1.1.2.1), or viruses, i.e. *proretroviral transposons* (see Section 2.1.1.2.2). Also, a suspected target sequence for insertion occurs at both ends of these

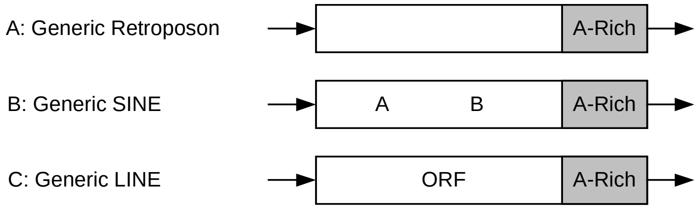


Figure 2.1.1.2.1.A: A pictorial representation of retroposons. The small filled arrows are direct repeats flanking the retroposon. SINEs contain one or more RNA polymerase III promoter sites (A and B). LINEs contain one or more genes or open reading frames (ORFs). All structures have an **A**-rich tail and are flanked by small, direct repeats.

repeats as expected for a circular DNA, crossover insertion. Furthermore, some repeats actively move within the genome, such as Barbara McClintock's "jumping genes" in maize.

2.1.12.2.1. Retroposons

Retroposons resemble processed RNAs and transpose passively via an RNA intermediate (Weiner 1986). Each element is composed of an **A**-rich tail at the 3' end and short target site duplications (direct repeats of 5-21 bp) flanking the repeat (Rogers 1985, Figure 2.1.1.2.1.A). Two main subclasses dominate this class: short interspersed elements (SINEs) and long interspersed elements (LINEs).

SINEs are often present in over 100,000 copies per genome (Weiner 1986). Elements are short (30-300 bp) and well distributed throughout the non-centromeric regions of the genome. A SINE contains one or more RNA polymerase III promoter sites and an **A**-rich region (see Figure 2.1.1.2.1.A). One subfamily is composed of a single promoter site and **A**-rich region pair and the other subfamily is a head-to-tail catenation of two promoter site, **A**-rich region pairs (Weiner 1986). Both subfamilies are flanked by short direct repeats of 5 to 21 bp.

The Alu sequence is a SINE with two promoter sites, a dimer, and is specific to primates. The genome contains 300,000 to 500,000 copies and, on average, occurs once every 5 to 9 kbp (Rogers 1985). The uniqueness of Alu sequences to primates provides a wonderful tool for separating primate DNA from DNA for other species. Alus present challenges to sequence assembly due to their high genome copy number and their dimer structure.

LINEs (Hutchison 1989) are present in 20,000 to 50,000 copies per genome and exclusively in mammalian genomes (Weiner 1986). The basic LINE structure (see Figure 2.1.1.2.1.A) is composed of genes or open reading frames (ORFs) followed by a 3' **A**-rich region. The element is flanked by direct repeats of 6-15 bp.

The L1 family, the primary LINE family, is 6 to 7 kbp long (Weiner 1986). The consensus structure of the family is well defined but not well conserved. A L1 element can deviate significantly from the structure such that entire structural components are deleted or duplicated (Weiner 1986).

2.1.12.2.2. Proretroviral transposons

Proretroviral transposons are mobile elements that transpose via an RNA intermediate (Varmus 1989). Their structure and content resembles integrated viruses and often contain genes encoding viral products, e.g. protease, reverse transcriptase and integrase (Boeke 1989). The genes are flanked by long terminal direct repeats (LTRs) that are several hundred bases long. The LTRs contain transcriptional signals for initiating and terminating transcripts, a promoter, an

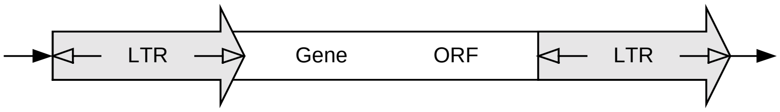


Figure 2.1.1.2.2.A: A pictorial representation of proretroviral transposons. The small filled arrows are direct repeats that flank the repeat. The large filled arrows are long terminal repeats (LTRs). Each LTR has an inverse repeat at each end, denoted by small open arrows.

enhancer and a polyadenylation signal (Temin 1985, Schmid 1990). Inverse repeats exist at the ends of each LTR and always begin with the bases, **TG**, and end with **CA** (Temin 1985). The two LTRs and the genes are flanked by 4 to 6 bp direct repeats, the target site duplications (see Figure 2.1.1.2.2.A).

2.1.13. Other recurring genetic features

DNA contains many recurring features that are not classified as a tandem or interspersed repeat. A gene cluster is a group of proximal genes having similar sequence and often, similar structure but, different function. Genes have multiple, non-proximal instances for high throughput requirements or for creating entities with similar structures but slightly different function. For example, transfer RNAs (tRNAs) require multiple, related genes, one for each amino acid required during translation from a messenger RNA (mRNA) to a protein. Also, some genomes require multiple instances of a tRNA for a specific amino acid to ensure that a sufficient quantity of tRNA is present. Promoters and other regulatory regions are associated with many genes. None of these features are repetitive DNA but these and other features do recur within a genome.

2.1.14. Additional issues

Repetitive DNA was classified initially as 'junk DNA.' Now, structural and functional roles are known for repetitive DNA. This section discusses various roles involving repeats, primarily tandem repeats. Tandem repeats are known to form alternative DNA structures (see Section 2.1.3.1). Those structures may be critical to its structural and functional role within the genome (see Section 2.1.3.2). The tandem repetitive structure in telomeres plays a role in cancer and hyper-variable, tri-nucleotide tandem repeats that are linked to over a dozen, major degenerative diseases (see Section 2.1.3.3). DNA fingerprints use hyper-variable and other repeats to distinguish between individuals (see Section 2.1.3.4).

2.1.14.1. Alternative DNA structures

Alternative DNA structural models suggest alternative pairing of nucleotides in tandem repeats. These models form duplex, triplex and quadraplex DNA structures. DNA containing regions capable of forming these structures becomes both more stable and more instable. Particular interest has been paid to these structures with respect to tri-nucleotide tandem repeats and their association with degenerative diseases.

Duplex, triplex and quadraplex structures use Watson-Crick hydrogen bonds, Hoogsteen hydrogen bonds or both (reviewed in Pearson & Sinden 1998). Hairpins and cruciforms, both duplex structures, involve one strand folding and binding to itself using Watson-Crick hydrogen bonds between complementary nucleotides. Triplex structures involve three strands derived from one or more strands and form using both Watson-Crick and Hoogsteen hydrogen bonds. Quadraplex structures form nucleotide planes in which either like nucleotides use Hoogsteen hydrogen bonds or alternating complementary nucleotides use Watson-Crick hydrogen bonds. *In* *vitro* studies under physiological conditions render these structures stable (Catasti *et al*. 1999). *In vivo* studies are actively being pursued.

The formation of alternative structures is both advantageous and disadvantageous for DNA (Calasti *et al*. 1999). Region length is more variable where alternative structures are formed. During replication, all or a portion of these alternative structures can be easily excised or duplicated potentially creating two strands of different length. The excision and duplication coupled with the effect of differing length strands decreases the stability of this region of the DNA molecule. On the other hand, formation of these structures decreases supercoiling, thereby increasing the ability to transcribe nearby regions. Thus for tandem repeats, the region copy number is critical as it represents a balance between the advantage gained during transcription and the disadvantage encountered during replication.

2.1.14.2. Structural and functional roles

Tandem repeats play significant structural and functional roles in DNA. They occur in abundance in structural areas such as telomeres, centromeres and histone binding regions. They play a regulatory role near genes and perhaps even within genes. Alternative DNA structural models provide one explanation of how tandem repeats may serve in these roles.

Telomere. This portion of a chromosome provides a capping mechanism for the chromosome protecting it from chromosomal fusion, instability and degradation. Blackburn (2001) states that the telomere "prevents or strictly controls at least four distinct biochemical processes …: (1) fusion with another telomere or DNA end; (2) replication of telomeric sequences by telomerase; (3) degradation of telomeric DNA; and (4) recombination at and near a telomere" (McClintock

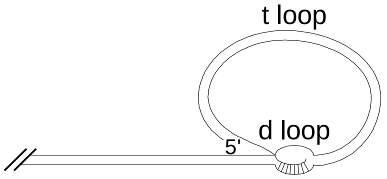


Figure 2.1.3.2.A (from Greider 1999): The d-loop-t-loop model for the structure of the telomere. The d-loop forms when the 3′ overhang invades the duplex structure to form triplex DNA. The overhang sequence represents one or more copies of a tandem repeat region and has a sequence that is similar to the sequence in the d-loop. The formation of the triplex DNA results in the formation of the t-loop.

1942, Muller 1938, Gottschling *et al.* 1990, Gilson *et al.* 1993, van Steensel *et al.* 1998, Evans & Lundblad 2000, Kass-Eisler & Greider 2000). These factors require that DNA and protein act together to maintain a functional telomere. Even with maintenance, telomeres shorten by 50-200 bp with each cell division and eventually, cell growth ceases though in a few cases, telomerase activation leads to cancerous cells (See Section 2.1.3.3). In this section, I discuss the DNA sequence of telomeres.

Telomeric DNA consists of the 3′ overhang and adjacent tandem repeat at the chromosomal terminus. A recent model suggests that this region does a *d-loop-t-loop* by having the 3′ overhang invade the tandem repeat (Griffith *et al.* 1999, see Figure 2.1.3.2.A). This invasion forms a triplex DNA structure, *d-loop*, and encloses a large segment of duplex DNA in a terminal loop or *t-loop*. The size of the loops is species dependent with each species maintaining specific telomere length equilibrium (Shore 2001). It is not clear how universal this structure is across species though some small telomeres likely are unable to form a t-loop.

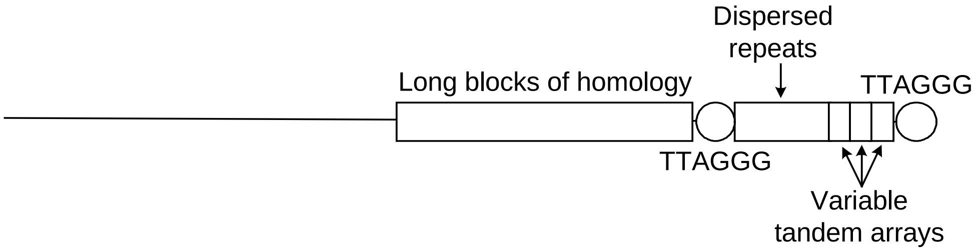


Figure 2.1.3.2.B (from Pryde *et al.* 1997): The human subtelomeric structure. The telomeric tandem repeat with a **TTAGGG** pattern recurs partway through the subtelomeric region. Between the **TTAGGG** repeats, regions of dispersed repeats and several tandem repeats occur. Proximal to the subtelomeric **TTAGGG** are long blocks of homology.

The telomere is a **G**-rich region with wide variation in length across species. The 3′ overhang is similar or identical to the tandem repeat pattern and usually represents one or several copies of the tandem repeat (Hemann & Greider 1999), but also, can be several hundred bases long (reviewed in Blackburn 1998). For example, humans and mouse, *Mus musculus*, both have a **TTAGGG** pattern but, mouse repeats are ~50 kbp while human repeats are only 5 to15 kbp. Furthermore, yeast, *Saccharomyces cerevisiae*, has an irregular pattern of $\mathbf{TG}_{1\text{-}3}$ and a repeat length of only ~300 bp.

The subtelomeric region is adjacent to the telomere and for most species contains many repetitive sequences (review in Pryde *et al.* 1997). In humans, the structure of four subtelomeres, 4p, 4q, 16p and 22q is identical (see Figure 2.1.3.2.B). A comparison to yeast (*Saccharomyces cerevisiae*) shows a remarkably similar structure. Furthermore, Flint *et al.* (1997) explain how the internal, degenerate **TTAGGG** repeats enable alignment with other subtelomeric regions allowing sequence exchange between subtelomeres. Of course, the advent of the d-loop-t-loop structure may affect these findings. Finally in his review, Pryde shows that not all species have the same structure but all have structures containing tandem repeats, interspersed repeats or both.

The centromere structure is gradually being defined (review in Sullivan *et al.* 2001). Most centromeres contain a significant satellite tandem repeat with most patterns being in the 150 to 200 bp range (Henikoff *et al.* 2001). The function of the satellite is debated though evidence suggests it is the primary functional component of active centromeres. For example, Schueler *et al.* (2001) studied the centromeric structure of the human X chromosome using physical mapping, genetic and functional approaches and sequencing of selected regions. The centromeric core is an alpha satellite about three megabasepairs (mbp) long having a 171 bp pattern. The core region has higher order periodicities in which groups of similar 171 bp patterns recur in a tandem fashion. Schueler demonstrates that this core region is sufficient for centromeric function by deletion analysis and insertion into an artificial human chromosome.

In addition, Schueler's work describes the region from the euchromatin on the Xp arm to the core alpha satellite region of the centromere (see Figure 2.1.3.2.C). The nearby euchromatin contains a high density of known repeats, most of which are L1 interspersed repeats. A 450 kbp long transitional region lies between the alpha satellite core and the euchromatin Xp region. It contains primarily alpha satellites but also, includes a 40 to 50 kbp gamma satellite region (Lee *et al.* 2000), interspersed LINEs and a tandem repeat having a 35 bp pattern.

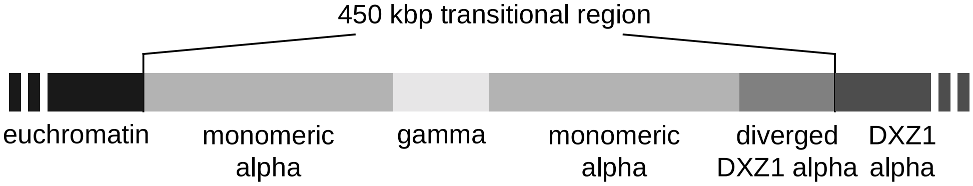


Figure 2.1.3.2.C (taken from Schueler *et al.* 2001): The structure of the human X chromosome centromere extending into the Xp euchromatin. The DXZ1 alpha satellite region represents the functional centromere. The function of the 450 kbp transitional region is unclear. It contains several alpha satellite variants, a gamma satellite region, interspersed repeats and a tandem repeat with a 35 bp pattern.

Histones. These proteins bind to DNA in complexes known as nucleosomes (review in Kornberg & Lorch 1999, Widom 1998). Nucleosomes represent a piece of *in vivo* DNA compaction machinery composed of eight histones bound to a core region of DNA ~147 bp long. DNA linker regions of 10 to 100 nucleotides connect nucleosomes. Thus, the fundamental repeating unit is a minimum of ~157 bp long (Godde & Widom 1992, Makarov *et al.* 1993). The helical twist of the DNA in this region is on average 10.2 bp per turn of the helix with variation along the DNA (Widom 1992).

The search for the primary DNA binding sequence led to studies on pattern periodicity, di-nucleotide frequencies and sequence binding affinities to name a few (Widlund *et al.* 1997). Results indicate ~10 bp periodicities for **A**-runs, **TATA**-tetrads and several di-nucleotides. For example, a particularly strong tandem repeat with a 10 bp pattern of 5′–**TATAA(A/C)CG(T/C)C**–3′ contains many of the individual properties. In addition, tandem repeats increase and decrease nucleosome stability. For example, a tandem repeat having a **CAG**(=**CTG**) pattern located close to a nucleosome increases its stability (Wang *et al.* 1994, Wang & Griffith 1995, Godde & Wolffe 1996). On the other hand, a tandem repeat having a **CGG**(=**CCG**) pattern has no impact unless it is methylated. Methylated **CGG** (=**CCG**) pattern with a limited copy number increase the nucleosome stability while those with large copy numbers decrease nucleosome stability (Godde *et al.* 1996, Wang & Griffith 1996). Furthermore, nucleosomes appear to influence transcription.

DNA transcription to RNA. Transcription by RNA polymerases is regulated extensively. The precise role of tandem repeats in transcription regulation is not known. It is known that tandem repeats affect nucleosome positioning and stability, change the degree and direction of DNA supercoiling and form alternative DNA structures (Ohyama 2001). In turn, all these are known to influence transcription initiation and elongation.

Nucleosomes can repress or enhance both transcription initiation and elongation though most often via repression (Hartzog & Winston 1997, Kornberg & Lorch 1999). Tandem repeats affect nucleosome stability (see above) such that increased stability forms tighter bonds between the histone complex and the DNA. Tighter bonds restrict access to the DNA in the nucleosome both for proteins that repress or enhance transcription and for the RNA polymerase. Also, tighter bonds limit transcription elongation by not releasing the DNA for transcription. DNA access is critical for transcription and nucleosomes play a critical role in allowing or disallowing DNA exposure.

Supercoiling affects transcription both directly and indirectly (Dai & Rothman-Denes 1999). Formation of alternative DNA structures such as Z-DNA and triplex DNA structures is an indirect form of regulation. Tandem repeats having an alternating purine (**R**=**A** or **G**)-

pyrimidine (**Y**=**C** or **U/T**) pattern forms Z-DNA (Yang *et al.* 1996). Tandem repeats with a **RRY** or a **YRY** pattern form triplex DNA structures (Grabcyzk & Usdin 2000). Other structures such as cruciforms and hairpins form from tandem repeats (Shlyakntenko *et al.* 1998). More research is required to determine whether tandem repeats form these structures near or within transcription regulatory regions.

A tandem repeat having a triplet pattern impedes transcription elongation. Grabcyzk and Usdin (2000) studied the Friedreich's ataxia gene containing a tandem repeat with a **GAA**(=**TTC**)-pattern (see Section 2.1.3.3 for more on degenerative diseases). They observed T7 RNA polymerase advance into the repeat and pause at the distal end of the repeat, a phenomenon consistent with a **RRY** triplex DNA structure. When the copy number of the repeat is increased, elongation impedance is stronger.

Tandem repeats in genes. Several occurrences are known to improve function and to be the cause of several degenerative diseases. At present, clear cut cases are associated with repeats having high copy numbers. In fish, an antifreeze gene provides better protection against frigid temperatures when a tandem repeat increases its copy number. On the other hand, Fragile X syndrome causes mental retardation when high copy numbers are present. Changes in function and tandem repeat hypervariability enables identification of both the antifreeze gene and degenerative diseases. The effect of tandem repeats having stable copy numbers is not known.

2.1.14.3. Degenerative diseases and cancer

Both degenerative diseases and cancer correlate to regions containing tandem repeats. Over a dozen major human degenerative diseases are associated with hypervariability of tandem repeats having trinucleotide patterns. Telomerase maintenance of telomeres correlates to several cell types including cancerous cells.

Major human degenerative diseases. These diseases result from the expansion of tandem repeat loci composed of trinucleotide patterns (see Table 2.1.3.3.A). Typically, the disease affects individuals having more than fifty copies of the pattern, while individuals having less than thirty copies are unaffected (Ashley & Warren 1995, Mitas 1997). The exact mechanism causing the disease state varies. **CAG** repeats are translated into polyglutamine tracts and result in a gain-of-function, possibly a toxin (Perutz *et al*. 1994, Baldi *et al*. 1999). **GAA** repeats form triplex DNA structures that can act as effective blocks to DNA replication and can cause reduced replication or transcription (Sinden 1999). **CGG**, **AGG** and **TGG** repeats form quadraplex DNA structures which may act as efficient blocks to DNA replication or transcription (Sinden 1999). In addition, **CGG** repeats destabilize nucleosomes (Sinden 1999) due to **C*p*G** hypermethylation thereby leading to promoter repression and lack of gene expression (Nelson 1995, Baldi *et al*. 1999). On the other hand, ***CTG*** repeats stabilize nucleosomes and block replication forks in *E. coli* (Sinden 1999). Hypervariable repeats with negative impacts are likely more widespread than just these diseases.

Human cancer. Cancer cells have telomerase activity in 85-90% of the cells (Kim *et al.* 1994). Telomerase maintains telomeres by increasing the copy number of the tandem repeat. In a normal human cell, replication occurs a finite number of times, the cell ceases to grow and telomere shortening leads to cellular senescence (Elenitoba-Johnson 2001). On the other hand, cancerous cells do not die but continue to live as if immortalized. Telomerase activity is

Disease	Pattern	Copy Number (Sinden 1999)			(Copy Number (Baldi 1999)		
		Normal	Unstable	Affected	Normal	Unstable	Affected
Spinobulbar muscular atrophy (Kennedy's Disease)	`CAG`	14-32		40-55	9-36	> 47	38-62
Huntington's Disease	`CAG`	10-34	36-39	40-121	6-35	> 35	36-121
Spinocerebellar ataxia 1	`CAG`	6-39		40-81	6-35		40-81
Spinocerebellar ataxia 2	`CAG`	14-31		34-59	14-32		33-77
Spinocerebellar ataxia 3 (Machado Joseph disease)	`CAG`	13-44		60-84	12-40		67-82
Spinocerebellar ataxia 6	`CAG`	4-18		21-28	4-17		20-30
Spinocerebellar ataxia 7	`CAG`	7-17		38-130	7-17		38-130
Dentatorubropallidoluysian atrophy (Haw River syndrome)	`CAG`				3-36		49-88
Autosomal dominant pure spastic paraplegia	`CAG`						
FRA16A	`CCG`	16-49		1000-1900			
Jacobsen syndrome	`CGG`	11	80	100-1000			
Myotonic dystrophy	`CTG`	5-37	50-80	80-3000	5-30	36-50	50 to > 700
Fragile X-A syndrome	`CGG`	6-52	59-230	230-2000	5-52		200 to > 1000
Fragile X-E syndrome	`CCG`	4-39	(31-61)?	200-900	6-25	>60	130 to > 700
Fragile X-F syndrome	`CGG`	7-40		306-1008			
Friedreich's ataxia	`GAA`	6-29	>34-40	200-900	7-22	34-65	200 to >900

Table 2.1.3.3.A: Degenerate human diseases associated with high tandem repetition of three nucleotide pattern tandem repeats (Composite assembled from Pearson & Sinden 1998 and Baldi *et al*. 1999).

correlated with cancerous cells such that most normal cells lack telomerase and most cancerous cells contain active telomerase. Current cancer research is looking for a mechanism to inhibit telomerase activity in cancerous cells.

2.1.14.4. DNA fingerprints

In 1985, Jeffreys *et al.* presented a technique for fingerprinting DNA that used DNA to distinguish individual people (Jeffreys *et al.* 1985a, 1985b). By year's end, the technique was applied to four-year-old forensics data to positively identify a male suspect and to immigration data to verify a familial relationship (Gill *et al.* 1985, Jeffreys *et al.* 1985c). Today, DNA fingerprinting use is extensive, not only for human paternity and forensics testing but also, for plant and animal pedigree determination (Jeffreys & Morton 1987), salmonella strain identification (Wampler 2001), anthrax genetic variation (*Bacillus anthracis*, Jackson *et al.* 1998, Keim *et al.* 1999, Enserink 2001) and even medically for differentiation between cancerous and constitutional DNA (Thein *et al.* 1987) and aging studies (Vynnycky *et al.* 2001, Dolle *et al.* 2002). In addition, fingerprints map chromosomes (Bellanne-Chantelot *et al.* 1991) including single nucleotide polymorphisms, SNPs (Brookes 1999, Chicurel 2001).

Jeffreys' initial DNA fingerprints measured the length of hypervariable minisatellites (VNTRs). Today, DNA fingerprints measure microsatellites (Amos & Pemberton 1992, Gill *et al.* 1995), Alu/SINEs (Nelson *et al.* 1989), LINEs and other interspersed repeats (Ledbetter *et al.* 1990), and even arbitrary DNA (Williams *et al.* 1990) and deletion (Schwartz *et al.* 1993) or duplication (Lupski *et al.* 1993) of DNA.

2.2. Informatics: algorithms and concepts for locating recurrences in strings

A DNA sequence, $\boldsymbol{S}$, is regarded computationally as a linear string of $\boldsymbol{n}$ characters over the alphabet $\Sigma = \{\mathtt{A,C,G,T}\}$. Algorithms for processing strings abound in computer science. Most involve exact or inexact matching of substrings of $\boldsymbol{S}$. A substring of $\boldsymbol{S}$ is denoted $\boldsymbol{S}[\boldsymbol{i}, \boldsymbol{j}]$ and represents the characters that start at position $\boldsymbol{i}$ and end at position $\boldsymbol{j}$ of $\boldsymbol{S}$ for $1 \le \boldsymbol{i} \le \boldsymbol{j} \le \boldsymbol{n}$.

This section discusses computational concepts for identification of repeats in strings. First, Section 2.2.1 discusses analyzing a DNA sequence using a window: the difference between a target region and word content. Section 2.2.2 discusses exact and inexact repetition and various methods for measuring similarity. Section 2.2.3 defines and explains regular expressions. Section 2.2.4 discusses alignment between two sequences using dynamic programming. Section 2.2.5 discusses several classic string matching algorithms. Section 2.2.6 describes string searches for exact and inexact repeats. Section 2.2.7 describes the suffix tree data structure. Last, Section 2.2.8 briefly describes string compression algorithms.

2.2.1. Windows: target regions versus word content

Researchers use fixed length windows for two primary purposes: to restrict analysis to the substring contained in the window, termed the ***target region***, and to measure content of the substring contained in the window, termed ***window*** or ***word content***. A target region restricts analysis to the consecutive positions demarked by two positions in the string. The window content is a measure of the occurrences within the window such as the word content composed of all consecutive positions in the window or the single, double or triple character content (e.g.

mono-, di-, or tri-nucleotide content), the occurrence of single characters or multiple character consecutive combinations.

Window length affects an algorithm's efficiency and accuracy. Often, a tradeoff exists between time efficiency and identification accuracy. For instance, large target region windows may increase the time to process a single window but on the other hand, they may reduce the number of windows needed to process the string for non-overlapping or slightly overlapping windows. For large content windows, the occurrence probabilities usually are spread across more words than for small content windows resulting in fewer identical but larger words in a string or target region. Fewer recurring identical words decrease the number of items to process. Of course, additional identical words exist and could be found with a smaller content window. On the flip side, a small content window may generate a massive number of identical words, overwhelming the identification system and magnifying the time required to process the string. Also, small content windows do not always lead to a more accurate identification scheme. All in all, the selection of a window size impacts most algorithms significantly.

2.2.2. Exact and inexact repeats: distance measures between recurrences

A repeat is a substring that occurs at least twice in the string. Definitions of repeats vary from exact or identical recurrence of a substring to inexact or similar recurrence of a substring (see Figure 2.2.2.A). Now, it is agreed that two identical substrings represent a repeat, but inexact recurrence requires a measure of similarity. The two most common measures are the ***Hamming distance*** and the ***Levenshtein*** or ***edit distance*** (Levenshtein 1966).

A distance measure compares two strings by selecting a set of operations that transforms one string into the other string. The Hamming distance allows mismatch operations while the

Pattern	**ACCGTGA**
Identical (exact) match	ACCGTGA
Similar (inexact) match with 3 mismatches (Hamming distance of $\boldsymbol{k} = 3$)	AC**G**G**A**G**G**
Similar match with 1 deletion, 1 insertion and 1 mismatch (edit distance of $\boldsymbol{k} = 3$)	A**A**▨GTG**G**A

Figure 2.2.2.A: Exact and similar matches to a pattern. Three ways to match a string to a pattern are presented. Mismatches between the pattern and the string are depicted by boldfaced nucleotides in the matches. A gray background represents an insertion or deletion in the string relative to the pattern with the inserted nucleotide in the foreground.

edit distance allows mismatch, deletion and insertion operations. A single mismatch operation changes a single character into another character in Σ. A single deletion operation removes a character from a string. A single insertion operation adds a character in Σ to a string. Each operation has an associated cost, e.g. a cost of one per operation. The final distance is the set of operations that transforms one string into the other string with minimum cost.

Most computational algorithms apply a threshold to denote the maximum number of operations or maximum cost allowed between similar strings. For instance, in the ***k***-mismatch problem, one finds all occurrences in a string of a pattern having at most ***k*** mismatches, i.e. the Hamming distance with a threshold of ***k*** where mismatches have a cost of one. On the other hand, the ***k***-difference problem accepts ***k*** mismatches, insertions or deletions, i.e. the edit distance with a threshold of ***k*** where each operation has a cost of one.

2.2.3. Regular expressions

A regular expression is a common computational method for expressing a pattern as a set of possible strings. Three basic operations are concatenation, union and closure. Concatenation is a way of representing something following something else. Union is a way of choosing between several substrings. Closure is a way of allowing a substring to occur zero or more times.

Definition (modified from Gusfield 1997): A **regular expression** is a set defined recursively over the alphabet Σ such that Σ does not contain the symbols $\in$, |, (,) and $*$. The set is specified by the following rules:

1. *Empty String*: The symbol $\in$ is a regular expression.
2. A single character from Σ is a regular expression.
3. *Concatenation*: A regular expression followed by another regular expression is a regular expression.
4. *Union*: Two regular expressions separated by a "|" form a regular expression.
5. A regular expression enclosed in parentheses is a regular expression.
6. *Closure*: A regular expression enclosed in parentheses and followed by "$*$" is a regular expression. The symbol $*$ is called the Kleene closure.

Example: **TG** is a simple concatenation of two characters. In my research, this represents the pattern for a simple microsatellite.

Example: (**TG**)* is a regular expression expressed as a closure expression. In my research, this represents a tandem repeat region having a **TG** pattern.

Example: (**CAGTA**(**TG**)*)* is a nested closure expression that contains a closure expression. In my research, this represents a tandem repeat region known as a variable length tandem repeat (for more details, see Section 3.2).

Example: **CAGTA**|**CAGCA**|**CAATA**|**CAGCA** is a union of four substrings. In my research, this represents a pattern associated with a multi-period tandem repeat (for more details, see Section 3.2). The tandem repeat region is expressed as (**CAGTA**|**CAGCA**|**CAATA**|**CAGCA**)*.

2.2.4. Alignment using dynamic programming

Dynamic programming (DP) is a technique for aligning two or more strings to one another in order to determine the optimal edit distance between strings. In 1955, Bellman began a systematic study of dynamic programming providing a solid mathematical basis for the area (Bellman 1957). In biology, two major works were presented to solve first the global and then local alignment. Needleman and Wunsch (1970) presented a solution for the global alignment problem of two strings. Smith and Waterman (1981) solved the local alignment problem. The algorithms are described for local or global alignment but both algorithms solve local and global alignment. In 1988, Myers and Miller (1988) presented a wrapping form of dynamic programming that was largely unknown until Fischetti *et al.* (1992) presented wraparound dynamic programming (WDP) in 1992.

The DP description given below borrows heavily from Gusfield (1997) with many of the definitions extracted directly from this work. Here, only alignment between two strings is addressed since this application performs an alignment between a pattern and a DNA sequence. First, a global alignment is defined between two strings and local alignment between substrings of two strings. Next, the general problem of computing the optimal edit distance is defined. Then, an efficient computation of the optimal edit distance using dynamic programming is shown for all substrings of two strings by constructing a two-dimensional matrix such that a path through the matrix specifies the matches, mismatches and gaps associated with an alignment. Finally, wraparound dynamic programming is described. This is all a precursor to the algorithm used for identifying tandem repeats (see Section 4.4.8).

Definition (modified from Gusfield 1997): A global alignment of two strings S_1 and S_2 is obtained by inserting spaces (representing gaps), either into or at the ends of S_1 and S_2, and then placing the two resulting strings one above the other so that every character or space in either string is opposite a unique character or a unique space in the other string. Spaces are gaps and represent a deletion in the string containing the space or an insertion in the other string opposite the space. Each space is treated as an individual gap (other definitions assign a single gap to each run of spaces).

Example: Consider the global alignment of the strings **ACGCTCTA** and **ACCTATGA**.

```
ACGCTCT A
AC CTATGA
```

In this alignment, gaps occur opposite both ***G***s and a mismatch occurs between the boldfaced **C** and **A**. All other positions represent matches between the two strings.

Definition: A local alignment of two strings S_1 and S_2 is obtained by aligning the substring s_1 in S_1 and s_2 in S_2 by inserting gaps (represented as spaces) into s_1 and s_2 and then placing the two resulting substrings one above the other so that every character or space in either string is opposite a unique character or a unique space in the other string.

Example: Consider the local alignment of substrings in **ACGCTCTA** and **ACCTATGA**.

*CT**C**T*

*CT**A**T*

In this alignment, there is a mismatch between the boldfaced **C** and **A.** All other positions represent matches between the two strings.

2.2.4.1. Edit distance

The definitions for local and global alignment allow the same operations as those associated with the edit distance; matches, mismatches and gaps between two strings. The edit distance is the number of operations between two fixed strings. Dynamic programming provides a technique for computing the edit distance for all substrings of the two strings.

Definition (modified from Gusfield 1997): For two strings $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j})$ is defined to be the edit distance of $\boldsymbol{S}_1[1..\boldsymbol{i}]$ and $\boldsymbol{S}_2[1..\boldsymbol{j}]$ where $\boldsymbol{S}_1$ contains ***n*** characters, $\boldsymbol{S}_2$ contains ***m*** characters and $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j})$ denotes the minimum number of edit operations required to transform the first ***i*** characters of $\boldsymbol{S}_1$ into the first ***j*** characters of $\boldsymbol{S}_2$. The edit distance, $\boldsymbol{T}(\boldsymbol{n},\boldsymbol{m})$ corresponds to a global alignment of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ and its computation solves the more general problem of computing $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j})$ for all combinations of ***i*** and ***j***, where $0 \leq \boldsymbol{i} \leq \boldsymbol{n}$ and $0 \leq \boldsymbol{j} \leq \boldsymbol{m}$.

2.2.4.2. Dynamic programming

This technique locates alignments using three major steps: the recurrence relation, the matrix computation and the traceback path. The recurrence relation describes all acceptable edit operations. The matrix, $\boldsymbol{T}$, is constructed such that each dimension represents a string to be aligned and each position in the matrix, $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$, represents an edit distance. The traceback path specifies the order of the edit operations for aligning the two strings.

The recurrence relation is composed of base and recursive conditions and describes how the score is computed for each $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$ where $0 \le \boldsymbol{i} \le \boldsymbol{n}$ and $0 \le \boldsymbol{j} \le \boldsymbol{m}$. Since gaps placed at the beginning or end of a string or substring are of no consequence, the base conditions are

$$\boldsymbol{T}(\boldsymbol{i}, 0) = 0$$

$$\boldsymbol{T}(0, \boldsymbol{j}) = 0.$$

for all $0 \le \boldsymbol{i} \le \boldsymbol{n}$ and $0 \le \boldsymbol{j} \le \boldsymbol{m}$. A single $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$ is computed using elements in the matrix previously computed and transitioning from those elements to $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$ using a match, mismatch or gap. Since an individual gap is assigned to each space in an alignment, the recursive condition for computing $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$ is the best of

$$\boldsymbol{T}(\boldsymbol{i}-1, \boldsymbol{j}-1) + \boldsymbol{m}-\boldsymbol{test}(\boldsymbol{i}, \boldsymbol{j})$$

$$\boldsymbol{T}(\boldsymbol{i}-1, \boldsymbol{j}) + \boldsymbol{gap}$$

$$\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j}-1) + \boldsymbol{gap}$$

for all $1 \le \boldsymbol{i} \le \boldsymbol{n}$ and $1 \le \boldsymbol{j} \le \boldsymbol{m}$ and where $\boldsymbol{m}-\boldsymbol{test}(\boldsymbol{i}, \boldsymbol{j})$ is a ***match*** if $\boldsymbol{S}_1[\boldsymbol{i}] = \boldsymbol{S}_2[\boldsymbol{j}]$ and a ***mismatch*** if $\boldsymbol{S}_1[\boldsymbol{i}] \ne \boldsymbol{S}_2[\boldsymbol{j}]$. The traditional edit distance assigns a cost of one to each of ***match***,

mismatch and ***gap*** but use of different costs reflects alternative optimizations. For clarity, the cost of each is

match = cost of placing identical characters in Σ opposite one another

mismatch = cost of placing two different characters in Σ opposite one another

gap = cost of placing a space opposite a character in Σ.

The matrix is constructed such that all positions above and to the left are computed prior to computation of $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$. A simple approach for filling the matrix is a nested for loop using the above definitions.

```
for each row $i = 1..n$
    for each column $j = 1..m$
        Compute $T(i, j)$ using recurrence relations
```

The traceback path through the matrix defines the edit operations associated with an alignment. Typically, each $\boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j})$ stores additional information indicating how the score was computed such that the edit operation and the element originating the transition can be reconstructed. The traceback path specifies which positions in the two strings align opposite one another and when a position is opposite a gap.

2.2.4.3. Wraparound dynamic programming

This technique aligns tandem occurrences in a string to a single pattern and is invaluable for the identification of tandem repeats in DNA. The wraparound feature extends the standard dynamic programming algorithm by allowing transitions from $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{m})$ to $\boldsymbol{T}(\boldsymbol{i},1)$ and from $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{m})$ to $\boldsymbol{T}(\boldsymbol{i}+1,1)$. This is implemented during construction of the matrix by a second pass through each row.

```
for each row $i = 1..n$
    Pass 1: Compute $T(i, j)$
        for each column $j = 1..m$
            Compute $T(i, j)$ using recurrence relations
    Copy $T(i, m)$ to $T(i, 0)$
    Pass 2: Update $T(i, j)$
        while $T(i, j)$ is a transition from $T(i, j-1)$
            Re-compute $T(i, j)$ using recurrence relations
```

The traceback procedure is modified to allow transitions to the ***m***th element in the ***i***th row.

2.2.5. String matching

The classic string matching problem is to find all occurrences of a pattern, ***P***, of length ***m*** in a string, ***S***, of length ***n***. The naïve approach is to compare ***P*** to every position in ***S*** by sliding ***P*** along ***S***. This requires ***m*** comparisons at each position in ***S***. Algorithms were proposed by

Rabin-Karp (Karp & Rabin 1981), Knuth-Morris-Pratt (Knuth, Morris & Pratt 1977) and Boyer-Moore (Boyer & Moore 1977) that perform better in practice than this naïve algorithm These algorithms do not slide along the sequence but, skip along the sequence. The character at position ***i*** on ***S*** is cleverly evaluated as part of its first comparison operation such that additional pattern comparisons can be avoided when it is known that failure will occur.

These algorithms are difficult to use for general tandem repeat identification in DNA for two reasons. First, the pattern needs to be known *a priori*. One approach handles this by testing all possible patterns (see Section 2.3.3). This approach is inefficient and is limited since a minimum and maximum pattern length must be chosen. In addition, complex pattern structures differ in pattern length (see Section 3.2). Second, inexact matching of ***P*** to ***S*** causes degradation of algorithmic efficiency. Extreme degradation kicks in at similarity levels where the biology begins to get interesting. Thus, application of string matching to identify all tandem repeats is inappropriate.

2.2.6. String searches for exact and inexact repeats

A large body of work exists for finding repetitions in a string. Given a string of length ***n*** and a substring, ***u***, a tandem repetition can be written in the form $\boldsymbol{u^k u'}$ where ***k*** consecutive occurrences of ***u*** are followed by a prefix of ***u***, $\boldsymbol{u'}$. Several notions of repetition are studied. Most computational algorithms for finding repeats are based on one or more of these notions.

Definition: ***Integer power***, ***k-power*** or ***tandem array*** refer to exactly ***k*** copies of ***u*** with no partial copies.

Definition: A ***square***, sometimes called a tandem repeat, refers to two consecutive occurrences of $\boldsymbol{u}$, i.e. $\boldsymbol{k} = 2$.

Definition: ***Maximal periodicity*** or ***maximal repetition*** refers to the largest possible substring, $\boldsymbol{u}^{\boldsymbol{k}}\boldsymbol{u}'$, such that the substring cannot be extended either direction without incurring a mismatch. Right-maximal repetitions cannot be extended to the right while left-maximal repetitions cannot be extended to the left.

Definition: A ***super-maximal repetition*** is a maximal repetition that is not contained in a maximal repetition.

The classic repetition problem is to find all repetitions in a string of length $\boldsymbol{n}$. In the early 1980’s, three very different, $\mathrm{O}(\boldsymbol{n} \log \boldsymbol{n})$ time algorithms solved the problem by detecting all squares in a string (Crochemore 1981, Apostolico & Preparata 1983, Main & Lorentz 1984). Crochemore looked for $\boldsymbol{k}$-powers, Apostolico and Preparata used a suffix tree technique to find all right-maximal repetitions (for more details on suffix trees, see Section 2.2.7) and Main and Lorentz found all maximal repetitions. In addition, linear time algorithms were developed for detecting strings containing no squares, i.e. squarefree strings (Main & Lorentz 1985, Rabin 1985, Crochemore 1986) and to find all leftmost maximal repetitions (Main 1989). Main and Lorentz (1984) showed that $\mathrm{O}(\boldsymbol{n} \log \boldsymbol{n})$ time was optimal for finding all maximal repetitions. Yet in the late 1990’s, two linear time algorithms used suffix trees to find all maximal repetitions in a string (Gusfield & Stoye 1998 and Kolpakov & Kucherov 1998, 1999, Giraud & Kucherov 2000). The bounds differ due to differing assumptions; the $\mathrm{O}(\boldsymbol{n} \log \boldsymbol{n})$ bound assumes an algorithm is based only on symbol comparison.

During the 1990s, approximate repetitions in a string were addressed. The number of mismatches or edit differences required to convert from one substring to another measured similarity between substrings. Algorithms were developed for finding substring recurrences having at most ***k***-mismatches (Hamming distance) or ***k***-edit differences (Levenshtein distance). Landau and Schmidt (1993, 2001) extended Main and Lorentz's algorithms (1984, 1985) to approximate repetitions and presented a $\mathrm{O}\left(\boldsymbol{nk}\log\left(\frac{\boldsymbol{n}}{\boldsymbol{k}}\right)\right)$ algorithm for ***k***-mismatch tandem repeats and a $\mathrm{O}\left(\boldsymbol{nk}\log\boldsymbol{k}\log\left(\frac{\boldsymbol{n}}{\boldsymbol{k}}\right)\right)$ algorithm for ***k***-edit difference tandem repeats. Kannan and Myers (1993, 1996) frame the ***k***-edit difference problem as finding paths through a weighted edit graph and describe a recursive, dynamic programming search algorithm for identifying all highest scoring non-overlapping alignments in time $\mathrm{O}\left(\boldsymbol{n}^2\log^2\boldsymbol{n}\right)$ and space $\mathrm{O}\left(\boldsymbol{n}^2\log\boldsymbol{n}\right)$. Benson (1994, 1995) and Kannan and Myers (1996) reduce the space requirement to $\mathrm{O}\left(\boldsymbol{n}^2\right)$. Schmidt (1998) reduces the time requirement to $\mathrm{O}\left(\boldsymbol{n}^2\log\boldsymbol{n}\right)$ for locally optimal approximate tandem or non-tandem repeats.

2.2.7. Suffix trees

A complete suffix tree represents all suffixes in a string (see Figure 2.2.7.A). A suffix is a substring that extends from a position in the string to the end of the string. Suffixes are inserted into a tree such that two suffixes that begin with an identical series of characters traverse the same path through the suffix tree. The path begins at the root node and proceeds down the same path until a difference between the suffixes occurs at which point they each proceed along separate paths. Conceptually, each node in the tree has one branch entering the node such that

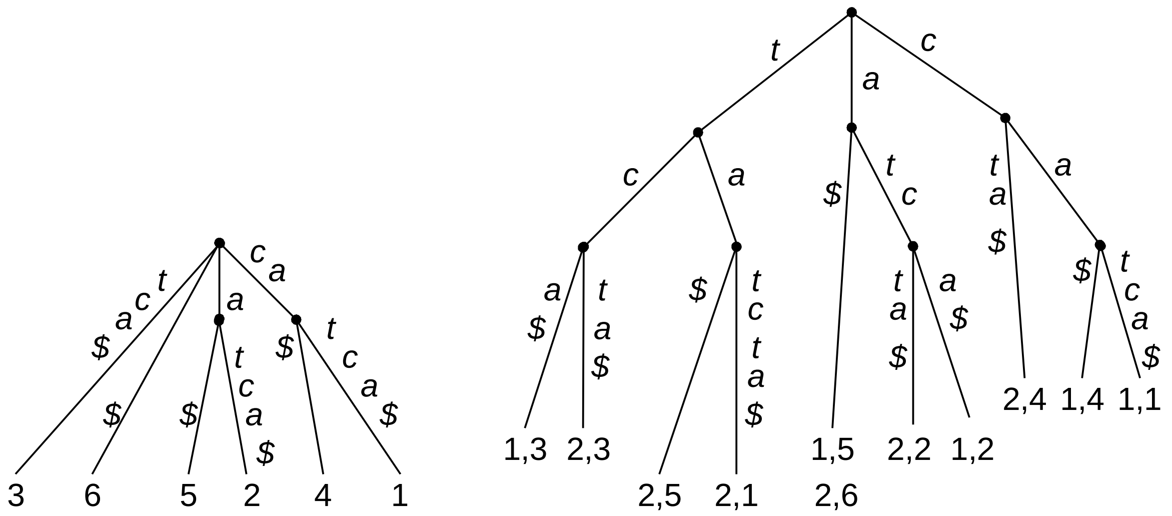


Figure 2.2.7.A (Gusfield 1997, modified): Suffix trees for one string (leftmost tree) and a generalized suffix tree for two strings (rightmost tree). A suffix is a substring from a position in the string to the end of the string. The leftmost tree contains every suffix in the string **catca**. The rightmost tree contains every suffix for the strings S_1=**catca** and S_2=**tatcta**. A suffix is read from the topmost, root node to a leaf. In the left tree, the leaves are composed of a number indicating the starting position of the suffix in the string. The leaves in the right tree are composed of two numbers; the first number indicates the string and the second number indicates the starting position of the suffix in that string. The black dots are nodes in the tree and represent points where the paths of two or more suffixes divide. Each line segment between nodes represents one or more characters in the suffix. The '$' symbol represents the end of a string.

---

all suffixes that enter the node have an identical series of characters and each node has one or more separate branches leaving the node such that suffixes that continue to be identical traverse the same branch and suffixes which differ traverse via different branches. Actual implementations vary considerably using trees, directed acyclic graphs (dags), directed acyclic weighted graphs (dawgs), lists, arrays and hash tables.

Weiner 1973 is credited with the original algorithm for suffix tree construction. Today, there are many competing algorithms both for general suffix tree construction and specialized suffix trees but most algorithms are modifications or extensions of McCreight's or Ukkonen's algorithm (McCreight 1976, Ukkonen 1995 also, see Gusfield 1997 for a good description of suffix tree construction). Most algorithms construct suffix trees in linear time and space, though usually the space requirements are huge. Numerous recent papers present techniques for minimizing the space requirement of the algorithms (e.g. Delcher *et al.* 1999, Kurtz 1999).

From a computational perspective, suffix trees offer a viable solution for processing string-based information. Suffix tree and compression based algorithms for identifying tandem repeats in DNA sequences are presented in Section 2.3.1 and Section 2.3.2.

2.2.8. String compression

String compression algorithms reduce the space required for a string by targeting recurrences of a substring for compression. Compression schemes are characterized by whether (1) the entire sequence is processed prior to encoding or the sequence processing (i.e. off-line processing) and encoding occur together in a stepwise fashion (i.e. on-line processing) and (2) the coding scheme is static with identical encoding for a substring throughout the sequence or the coding scheme adapts with different encoding for a substring at different locations in the sequence. Many different compression schemes exist including Huffman coding (Huffman 1952), arithmetic coding (Witten *et al.* 1987) and Lempel-Ziv coding (Ziv & Lempel 1977, Lempel & Ziv 1978).

Lempel-Ziv coding is one of the best in practice, enjoys widespread use (e.g. gzip, compress and pkzip). It is an adaptive, on-line compression that uses suffix trees to achieve an efficient implementation. Algorithms in this class replace a substring by the position of an earlier occurrence in the sequence followed by the length of the substring. Here are two examples.

Example (modified from Gusfield 1997): ***S***=ACAGACAXACZ is represented as AC(1,1)G(1,3)X(1,2)Z. This encoding of ***S*** does not compress the string but causes an increase in the number of symbols required from 11 to 20.

Example (modified from Gusfield 1997): ***S***=ACACACACACACACACACACACACACACACAC is represented as AC(1,2)(1,4)(1,8)(1,16). ***S*** is compressed from 32 symbols down to 24 symbols. The original Ziv-Lempel method represents ***S*** as AC(1,2)A(2,4)C(1,10)A(2,12)C, a slightly smaller compression rate.

String compression algorithms often do not have as great of a compression rate in DNA compared to other text strings. Furthermore, most compression algorithms assign a byte per character but the basic DNA alphabet, $\Sigma = \{\mathtt{A,C,G,T}\}$, is small and requires only two bits to encode a character. Thus, compression algorithms are developed specific to DNA sequences. In Section 2.3.2, several compression algorithms are presented which provide specialized compression of tandem repeats.

2.3. Bioinformatics: algorithms for locating recurrences in DNA sequences

This section presents existing algorithms for identification of tandem repeats and non-contiguous regions of similarity. Current algorithms identify forward and reverse complement repeats in DNA including simple recurrences of a substring, tandem repeats, interspersed repeats and genetic palindromes. Researchers developed exact and heuristic methods that locate either local similarities restricted to a small sequence region or global similarities of substrings throughout an entire sequence. Some approaches require *a priori* knowledge while others acquire all information solely from DNA sequence analysis.

This review focuses on algorithms for tandem repeat identification. Two algorithmic groups emerge; those that extend existing computational, string-based algorithms to identify DNA repeats and special purpose algorithms whose sole purpose is DNA repeat identification. Computational approaches are suffix tree based algorithms (Section 2.3.1) and compression based algorithms (Section 2.3.2). Special purpose algorithms for identifying tandem repeats are designed for DNA analysis but rely heavily on combinatorics and statistics. Special purpose algorithms are divided into *a priori* period- and pattern-based algorithms (Section 2.3.3) and unknown period- and pattern-based algorithms (Section 2.3.4). Alignment using dynamic programming often is used by both approaches to characterize repetitive regions. While tandem repeat identification is the metric for inclusion in this discussion, other repeats identified by algorithms are discussed as well.

2.3.1. Suffix tree based algorithms

A suffix tree is a well known, computational data structure used to store and sort substrings of a string. Recurring substrings are easy to identify in suffix trees; substrings that begin with the same characters have identical paths from a tree's root node. Algorithms for suffix tree construction exist for time and space requirements linear to the string length (for computational details, see Section 2.2.7). Suffix trees have numerous applications in DNA sequence analysis including DNA sequence compression (see Section 2.2.8) and identification of tandem repeats and palindromes.

In 1983, Martinez identified direct and inverse repeats in RNA and DNA sequences using *position trees*, a structure similar to suffix trees. Lefévre and Ikeda (1993) extended Martinez's work and described a more space-efficient approach called the *position end-set tree*. During the 1990s, DNA sequence analysis using suffix trees emerged as a technique to identify forward and reverse complement recurrences in small and large DNA sequences and even complete genomes. Data compression algorithms began using suffix trees to identify short identical substrings including tandem repeats having short patterns (for more details, see later in this section; Grumbach & Tahi 1993, 1994, Rivals *et al.* 1995, 1996, 1997). As interest in sequence comparison, sequence duplication, phylogeny and repeat identification grew; suffix tree construction algorithms became more efficient.

In the late 1990s, several groups presented suffix tree algorithms for identifying recurrences in DNA sequences. Kurtz and Schleiermacher presented a two-pass algorithm using a very efficient suffix tree implementation (Kurtz 1999) for identifying all forward and reverse complement *maximal exact repeats* in a complete genome (Kurtz & Schleiermacher 1999). Two linear algorithms locate all exact maximal repetitions (Kolpakov & Kucherov 1998, 1999, Giraud & Kucherov 2000 and Gusfield & Stoye 1998). Also, Stoye and Gusfield present an algorithm for identifying all exact tandem repeats (Stoye & Gusfield 2001).

These groups and others extended identification to degenerate repeats. One approach is to extend exact maximal repetitions, called seeds, to the surrounding sequence by allowing ***k***-mismatches (based on Hamming distance) or ***k***-differences (based on the edit distance) between recurrences (Kurtz *et al.* 2000, 2001, Adebeiyi *et al.* 2001, Volfovsky *et al.* 2001, Kolpakov & Kucherov 2001). The other approach looked for substrings separated by a fixed gap, essentially looking for recurrences of substrings at a fixed distance or period (Kolpakov & Kucherov 2000). In addition, an algorithm for maximal approximate palindromes was proposed (Porto & Barbosa 2002).

Algorithms based on maximal repeats ensure exact (non-heuristic) results but, several parameters play a crucial role in the meaning of the results; ***r***, the minimum length of exact recurrences, ***k***, the number of mismatches or differences and ***l***, the minimum length of a repeat containing ***k*** errors. In practice, Kurtz *et al.* 2001 sets $10 \le \boldsymbol{r} \le 13$ and $\boldsymbol{l} \ge 20$. The value of ***k*** is not directly stated but examples given in the paper include 10 errors in a 700 bp region and 2 errors in a 20 bp region. So, are these values reasonable? That depends on the goal of the analysis. My goal of identifying all biologically interesting tandem repeats combined with actual regions I've analyzed suggest that ***r*** is too large and ***k*** is too small relative to the size of ***l***. In fact, yeast (*Saccharomyces cerevisiae*) chromosome I (Bussey *et al.*1995, GenBank locus: NC_001133) has a tandem repeat region with a 15 bp pattern length that is difficult to identify with exact recurrences of six nucleotides!

Suffix trees are a compact, efficient data structure for storing information useful to data mining in DNA either on a local or genomic level. Identification of forward and reverse complement recurrences leads and will continue to lead to identifying many common features in DNA. The raw data stored in the tree is sufficient to identify some features while for other features additional effort is required to not only locate but also characterize the feature. Tandem repeats represent a feature that requires additional characterization beyond the raw data present in the tree.

2.3.2. Compression algorithms

The development of compression algorithms is a classic research area in both mathematics and computer science (for computational details, see Section 2.2.8). The goal of data compression is to decrease the space required to store data. Today, universal compression algorithms exhibit high compression rates for most English, text-based strings. Unfortunately, these universal algorithms do not compress DNA sequences well, primarily due to a small DNA alphabet composed of only four nucleotides.

An alphabet with fewer characters encodes a single character using fewer bits. The four-nucleotide, DNA alphabet requires only two bits to encode any DNA character, i.e. **A**=00, **C**=01, **G**=10 and **T**=11. The 26-character, English alphabet requires at least five bits but English text-based encodings typically use eight bits to compensate for uppercase and lowercase letters, numbers and punctuation. Many universal compression algorithms perform compression using an eight-bit-per-character encoding. In fact, “compression” of some DNA sequences using universal compression algorithms actually results in an *increase* of space rather than a decrease.

An alphabet with fewer characters decreases the number of unique ***k***-length substrings. As a result, the occurrence probabilities are spread across fewer unique ***k***-length substrings. Many universal compression algorithms are based on English, text-based occurrence probabilities. Significant deviation of alphabetic and substring occurrences from English, text-based data affects the compression rate.

During the past decade, DNA specific, compression algorithms were developed. Extensions to Lempel-Ziv coding (see Section 2.2.8) are the most popular approach (Milosavljevic & Jurka 1993, Grumbach & Tahi 1993, 1994, Rivals *et al.* 1995, 1996, 1997, Chen *et al.* 1999) though not the only approach (Matsumoto *et al.* 2000). At minimum, most schemes encode nucleotides using a two-bit-per-character encoding. In addition, many schemes test to ensure that local compression does not increase the space required relative to the two-bit encoding. Additional compression targets tandem repetition and imperfect, substring recurrence. Furthermore, techniques are available for on-line and off-line compression using either static or adaptive encoding. Here, I discuss DNA specific compression algorithms.

Milosavljevic & Jurka pioneered work to algorithmically detect repetitive sequences in DNA. Their work for detection of simple sequence repeats (SSRs) intersects their development of a compression scheme, the ***algorithmic significance method*** (Milosavljevic 1993, Milosavljevic & Jurka 1993). This scheme compresses SSRs by locating recurring substrings within small target windows of 128 nucleotides and replacing the recurrences by pointers to the original substring. The substrings are selected statistically. Non-SSR regions are encoded using two-bit-per-character encoding. The minimal length-encoding algorithm (Storer 1988) is used to compress the sequence using dynamic programming and directed acyclic word graphs (Blumer *et al.* 1985).

Grumbach and Tahi developed two compression schemes for DNA sequences, ***Biocompress*** and ***Biocompress-2*** (Grumbach & Tahi 1993, 1994). Both schemes target forward and reverse complement substring recurrences. Biocompress uses a scheme similar to Lempel-Ziv modified to use a simple suffix tree, a 4-ary complete tree of height $\boldsymbol{h}$, where for their tests, $\boldsymbol{h} = 8$. Non-recurring regions are encoded using two-bit-per-character encoding. Biocompress-2 extends the method allowing a region to be encoded by either the above-modified Lempel-Ziv scheme or two-bit-per-character encoding; whichever requires the fewest bits.

Rivals *et al.* are the first to address approximate tandem repeats in a compression algorithm (***Cfact***; Rivals *et al.* 1995, 1996, 1997). The two-pass algorithm constructs a static suffix tree of the entire sequence on the first pass and on the second pass encodes the sequence. All mono-nucleotide to penta-nucleotide substrings are considered within non-overlapping, 500 bp target windows. The substring with the most occurrences within the target window is selected as the encoding pattern. A successful encoding treats a region as a perfect tandem repeat of the selected pattern and supplements with substitutions (S), insertions (I) and deletions (D). For example, the region **CGCGCCGCGCGTGCG** becomes (R,7,**CG**)(I,5,**C**)(S,12,**T**); a tandem repeat composed of 7 copies of the pattern **CG** with an insertion at position 5 of a **C** and a substitution at position 12 of a **T**. In addition, the same authors present another method, ***TurboOptLift***, which uses wraparound dynamic programming to analyze any compression based scheme in order to locate edges of approximate tandem repeats (Delgrange *et al.* 1999).

Chen *et al.* (1999) compresses both approximate tandem repeats and approximate palindromes using an on-line, one-pass, conditional, sequence compression algorithm, ***GenCompress***. The *on-line*, *one-pass* component of the algorithm enables sequence data to be encoded before all the data is known. The *conditional* component allows selection of the best encoding from four alternative encoding schemes: the two-bit-per-character encoding method, an exact matching method, an approximate matching method using Hamming distance and an approximate matching method using edit distance. The exact and approximate matching schemes have access to any substring already processed by the compression algorithm. For example (Chen *et al.* 1999), a substring **GACCGTCA** has been encoded and a second substring, **GACCTTCA** is ready to be encoded. The two-bit encoding method converts an **A** into 00, a **C** into 01, a **G** into 10 and a **T** into 11, i.e. the 16 bits 10 00 01 01 10 11 01 00. The exact matching method encodes the substring as {(0,4),**G**,(5,3)}; i.e. the 17 bits 0 000 100 1 10 0 101 011. The approximate matching method using Hamming distance encodes the substring as a single nucleotide replacement of a perfect recurrence, {(0,8),(R,4,**G**)}; i.e., the 15 bits 0 000 111 1 00 100 10. Finally, the approximate matching method using edit distance encodes the substring as {(0,8),(I,4,**G**),(D,6)}, the 21 bits 0 000 11. The encoding using the fewest bits represents the second substring. In this example, the 15 bit encoding associated with the approximate matching method using Hamming distance.

Matsumoto *et al.* (2000) presents a hybrid compression scheme involving both Lempel-Ziv encoding and Context Tree Weighting (CTW, Willems *et al.* 1995) that compresses both approximate forward and reverse complement recurrences. The algorithm uses dynamic programming to determine the edit operations for a region and then chooses the best scheme to encode a region. In general, long repeats use Lempel-Ziv encoding while short repeats are encoded using CTW.

Compression algorithms target redundant data and in DNA, tandem repeats are a strategic, redundant feature. Unfortunately, the approach has limitations with regard to locating and characterizing tandem repeats. These algorithms locate tandem repeat regions but the regions are not fully characterized, i.e. a good pattern structure is not determined. Also, a single tandem repeat region may fracture and be represented as multiple regions. Furthermore, only moderate and well-conserved regions exhibit sufficient redundancies to induce compression. The rest are not located but are encoded using two-bit encoding. While add-on techniques can overcome the region fracturization and characterization, those tandem repeats not compressed are not identified.

2.3.3. *A priori* period- and pattern-based algorithms

Algorithms based upon *a priori* periods or patterns use straightforward techniques for locating tandem repeat regions. Interspersed repeats are located with *a priori* consensus patterns formed using known repeat occurrences. Tandem repeats are located with *a priori* patterns by iterative testing of all possible substrings of a fixed length. Also, tandem repeats are identified using *a priori* periods by finding identical substring occurrences, a fixed distance, or period, apart. In general, the period and the pattern are central and critical aspects of repeat analysis.

Since the late 1980s, compilation and classification of known interspersed repeats resulted in the creation of repetitive databases (***Repbase***, Jurka 2000; http://www.girinst.org). For instance, Jurka and colleagues created a collection of Alu consensus sequences, each representing distinct subfamilies of Alu occurrences (Jurka *et al.* 1992, Jurka 1995, Batzer *et al.*

1996, Kapitonov & Jurka 1996). In addition, algorithms were developed to find homologies between database consensus sequences and a DNA sequence of interest (CENSOR, Jurka *et al.* 1996).

A similar approach is to create a sequence database of the DNA sequence and perform a homology search of the sequence itself. A number of tools exist for this purpose: ***FASTA*** (Pearson & Lipman 1988), ***BLAST*** (Altschul *et al.* 1990), ***FLASH*** (Califano & Rigoutsos 1993) and ***SENSEI*** (States & Agarwal 1996). Several algorithms find repeats using the results from these tools. ***RPT*** identifies families of repeats (Agarwal & States 1994). ***MaskerAid*** masks repeats from sequence data (Bedell *et al.* 2000).

A simple, brute force method for tandem repeat identification is to iteratively test all possible ***k***-length patterns, for small values of ***k***, using standard algorithms for perfect and imperfect string matching (***Tandyman***, http://www.stdgen.lanl.gov/tandyman; ***Cfact***, Rivals *et al.* 1995, 1996, 1997). This approach identifies perfect and degenerate tandem repeats quite well. Unfortunately, linear increases of ***k*** and the pattern length, generate exponential increases in the number of patterns, $4^{\boldsymbol{k}}$ patterns, such that iteration over all patterns becomes prohibitive, i.e. the entire sequence is processed for each pattern. Thus, pattern length is limited and patterns over the prohibitive length are not identified. Furthermore, this approach is not linear to the length of the sequence but rather is exponential relative to ***k***, $\mathrm{O}\left(\boldsymbol{n}\,|\,\Sigma^{\boldsymbol{k}}\,|\right)$.

A better approach is to allow the pattern to vary. One method is to look for identical occurrences of ***k***-length substrings, *words*, located ***p*** positions apart (Benson and Waterman 1994). Here, the period, ***p***, is fixed but, the pattern varies since any two identical words are allowed to match. Identification using identical ***k***-length words works well given that ***k*** is small.

Furthermore, degenerate repeats are identified when $\boldsymbol{k} < \boldsymbol{p}$. Another method is a phase alignment technique (Coward & Drablos 1998). Here, a sequence is split into consecutive subsequences of length, ***p***, and iterative phase adjustments are made to compensate for insertions and deletions. This approach appears to handle mismatches, insertions and deletions well. Unfortunately, identification of all tandem repeats for both of these methods requires analysis of each period. The maximum length of a pattern is limited by the upperbound of the iteration.

2.3.4. Unknown period- and pattern-based algorithms

These algorithms locate tandem repeat regions without prior knowledge of the period or the pattern. Rather, the analysis is designed to find interesting periods and patterns. There are two basic algorithms. The first exhaustively tests increasing longer constructed patterns by locating substrings within ***e***-differences of the pattern and linking them together into trains (Sagot & Myers 1998). The second looks for ***k***-length substrings a common distance apart and uses statistical analysis to set thresholds for selecting distances that indicate a potential tandem repeat region (***Tandem Repeat Finder***, Benson & Su 1998, Benson 1999; Klaerr-Blanchard *et al.* 2000).

Sagot and Myers (1998) present an algorithm in which the core algorithm is an exhaustive search of all possible patterns coupled with an optional filter that limits the search space to interesting patterns. This algorithm finds occurrences within ***e***-differences of a pattern and links them together into a train. The pattern is iteratively increased in length and re-tested. The result of the algorithm is a list of valid trains with the strongest train listed first.

The core of Sagot and Myers' algorithm is a smart re-packaging of the brute force approach of testing all possible patterns. The optional filter narrows the search space considerably and targets regions suspected to be tandem repetitive. The prefix-extension approach allows trains of weak models and the ability to construct strong models that reflect the actual sequence. Furthermore, the algorithm allows jumps between wagons, thereby enabling poorly conserved regions to be jumped. In addition, many other useful insights are incorporated into the algorithm.

Unfortunately, the algorithm does have a limit; the exhaustive search is space and time intensive. Therefore, the authors limit the search space by restricting the periodicity of the tandem repeats. In particular, the authors state that the algorithm is designed to concentrate on periods in the 30 to 40 nucleotide range. Their examples depict parameters in which the period is at most 50 nucleotides and the range of acceptable periods is at most 10 nucleotides, e.g. a range of 40 to 50 nucleotides. Thus, the primary restriction of the algorithm is the narrow periodicity range and the limited maximum periodicity.

The second set of algorithms looks for common periods separating identical substrings (***Tandem Repeat Finder***, Benson & Su 1998, Benson 1999; Klaerr-Blanchard *et al.* 2000). The basic analysis uses a fixed, ***k***-length window and looks for recurring distances (or *periods*), ***d***, between occurrences of ***k***-length substrings, termed *words*. The number of occurrences for ***d*** in the last ***d*** positions is compared to the expected frequency to determine whether a distance indicates a potential tandem repeat periodicity (Benson & Su 1998). A pattern is selected from the region and aligned to the region using wraparound dynamic programming.

The window length, ***k,*** determines the sensitivity of the algorithm. Short periods are identified best with short windows, e.g. $\boldsymbol{k} = 3$. Longer periods are identified better using longer windows, e.g. $\boldsymbol{k} = 7$. Often, algorithms analyze a sequence using multiple values of ***k***; each ***k*** may look for all periods or for a specific set of periods (Benson 1999, Klaerr-Blanchard *et al.* 2000).

This dissertation uses a similar sliding ***k***-length window technique to analyze DNA sequences. Selection of potential tandem repeat periodicities differs from Benson's approach; a threshold-based approach looks for clusters of identical distances (see Section 4.1 and 4.4.1). In addition, this approach identifies regions having complex pattern structures that correlate to multiple periodicities within a tandem repeat region.

2.4. Bibliography

Adebiyi, E.F., Jiang, T. and Kaufmann, M. (2001) An efficient algorithm for finding short approximate non-tandem repeats. Bioinformatics **S1**:S5-S12.

Agarwal, P. and States, D.J. (1994) The Repeat Pattern Toolkit (RPT): Analyzing the structure and evolution of the *C. elegans* genome. Proceedings of the Second International Conference for Intelligent Systems for Molecular Biology **2**:1-9.

Altschul, S., Gish, W., Miller, W., Myers, E. and Lipman, D. (1990) Basic local alignment search tool. J. Mol. Biol., **215**:403-410.

Amos, B. and Pemberton, J. (1992) DNA fingerprinting in non-human populations. Current Opinion in Genetics and Development **2**(6):857-860.

Apostolico, A. and Preparata, F.P. (1983) Optimal off-line detection of repetitions in a string. Theoretical Computer Science **22**:297-315.

Ashley, C.T. Jr. and Warren, S.T. (1995) Trinucleotide repeat expansion and human disease. Annual Review of Genetics **29**:703-728.

Baldi, P., Brunak, S., Chauvin, Y. and Pedersen, A.G. (1999) Structural basis for triplet repeat disorders: a computational analysis. Bioinformatics **15**(11):918-929.

Batzer, M. A., Deininger, P. L., Hellmann-Blumberg, U., Jurka, J., Labuda, D., Rubin, C. M., Schmid, C. W., Zietkiewicz, E. and Zuckerkandl, E. (1996) Standardized nomenclature for Alu repeats. Journal of Molecular Evolution **42**:3-6.

Bedell, J.A., Kork, I. and Gish, W. (2000) MaskerAid: a performance enhancement to RepeatMasker. Bioinformatics **16**(11):1040-1041.

Bellanne-Chantelot C, Barillot E, Lacroix B, Le Paslier D, Cohen D. (1991) A test case for physical mapping of human genome by repetitive sequence fingerprints: construction of a physical map of a 420 kb YAC subcloned into cosmids. Nucleic Acids Research **19**(3):505-510.

Bellman, R. (1957) Dynamic Programming. Princeton University Press.

Benson, G. (1994) A space efficient algorithm for finding the best nonoverlapping alignment score. In Crochemore, M. et al. (eds), *Proc. of the 5th Annual Symposium on Combinatorial Pattern Matching, CPM 94. Asilomar, CA, USA*, volume 807 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin. pp. 1-14.

Benson, G. (1995) A space efficient algorithm for finding the best nonoverlapping alignment score. Theoretical Computer Science **145:**357-369.

Benson, G. and Su, X. (1998) On the distribution of k-tuple matches for sequence homology: a constant time exact calculation of the variance. Journal of Computational Biology **5**:86-100.

Benson, G. (1999) Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Research **27**(2):573-580.

Benson, G. and Waterman, M. (1994) A method for fast database search for all k-nucleotide repeats. Nucleic Acids Research **22**:4828-4836.

Blackburn, E.H. (1998) Telomerase RNA structure and function. In RNA Structure and Function, R.W. Simons and M. Grunberg-Manago, eds. (Cold Spring Harbor Laboratory Press), pp. 669-694.

Blackburn, E.H. (2001) Switching and signaling at the telomere. Cell **106**(6):661-673.

Blumer, A., Blumer, J., Haussler, D., Ehrenfeucht, A., Chen, M.T. and Seiferas, J. (1985) The smallest automaton recognizing the subwords of a text. Theoretical Computer Science **40**:31-55.

Boefe, J. D. and Corces, V. G. (1989) Transcription and reverse transcription of retroposons. Annual Review of Microbiology **43**:403-434.

Boyer, R.S. and Moore, J.S. (1977) A fast string-searching algorithm. Communications of the ACM **20**(10):762-772.

Brookes, A.J. (1999) The essence of SNPs. Gene **234**(2):177-186.

Bussey, H., Kaback, D.B., Zhong, D.T. Vo, Clark, M.W., Fortin, N., Hall, J., Ouellette, BFF., Keng, T., Barton, A.B., Su, Y., Davies, C.J. and Storms, R.K. (1995) The nucleotide sequence of chromosome I from *Saccaromyces cerevisiae*. Proceedings of the National Academy of Science, U.S.A **92**(9):3809-3813.

Califano, A. and Rigoutsos, I. (1993) FLASH: a fast look-up algorithm for string homology. Proceedings of the First International Conference for Intelligent Systems for Molecular Biology 1:56-64.

Catasti, P., Chen, X., Mariappan, S.V., Bradbury, E.M. and Gupta, G. (1999) DNA repeats in the human genome. Genetica **106**(1-2):15-36.

Chen, X., Kwong, S. and Li, M. (1999) A compression algorithm for DNA sequences and its applications to genome comparison. The Tenth Workshop on Genome Informatics. Genome Informatics **10**:51-61.

Chircurel, M. (2001) Faster, better, cheaper genotyping. Nature **412**(6847):580-582.

Coward, E. and Drablos, F. (1998) Detecting periodic patterns in biological sequences. Bioinformatics **14**(6):498-507.

Crochemore, M. (1981) An optimal algorithm for computing all the repetitions in a word. Information Processing Letters **12**:244-250.

Crochemore, M. (1986) Transducers and repetitions. Theoretical Computer Science **45**:63-86.

Dai, X., Rothman-Dened, L.B. (1999) DNA structure and transcription. Current Opinion in Microbiology **2**(2):126-130.

Delcher, A.L., Kasif, S., Fleischmann, R.D., Peterson, J., White, O. and Salzberg, S.L. (1999) Alignment of whole genomes. Nucleic Acids Research **27**(11):2369-2376.

Delgrange, O., Dauchet, M. and Rivals, E. (1999) Location of repetitive regions in sequences by optimizing a compression method. Pacific Symposium on Biocomputing 254-265.

Dolle, M.E., Snyder, W.K., Dunson, D.B. and Vijg, J. (2002) Mutational fingerprints of aging. Nucleic Acids Research **30**(2):545-549.

Elenitoba-Johnson, K.S. (2001) Complex regulation of telomerase activity:implications for cancer therapy. American Journal of Pathology **159**(2):405-410.

Enserink, M. (2001) Taking anthrax's genetic fingerprints. Science **294**(5548):1810-1812.

Evans, S.K. and Lundblad, V. (2000) Positive and negative regulation of telomerase access to the telomere. Journal of Cell Science **113**:3357-3364.

Flint, J., Bates, G.P., Clark, K., Dorman, A., Willingham, D., Roe, B.A., Micklem, G., Higgs, D.R. and Louis, E.J. (1997) Sequence comparison of human and yeast telomeres identifies structurally distinct subtelomeric domains. Human Molecular Genetics **6**:1305-1314.

Fischetti, V., Landau, G., Schmidt, J. and Sellers, P. (1992) Identifying Periodic Occurrences of a Template with Applications to Protein Structure. In Apostolico, A., Crochemore, M., et al. (eds). *Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Computer Science.* Springer-Verlag, Berlin, **644**, 111-120.

Gill, P., Jeffreys, A.J. and Werrett, D.J. (1985) Forensic application of DNA 'fingerprints.' Nature **318**(6046):577-579.

Gill, P., Kimpton, C.P., Urquhart, A., Oldroyd, N., Millican, E.S., Watson, S.K. and Downs, T.J. (1995) Automated short tandem repeat (STR) analysis in forensic casework—a strategy for the future. Electrophoresis **16**(9):1543-1552.

Gilson, E., Laroche, T. and Gasser, S.M. (1993) Telomeres and the functional architecture of the nucleus. Trends in Cell Biology **3**:128-134.

Giraud, G. and Kucherov, G. (2000) Maximal repetitions and application to DNA sequences. Journées Ouverles: Biologie, Informatique et Mathématiques, Montpellier, pp 165-172.

Godde, J.S. and Widom, J. (1992) Chromatin structure of *Schizosaccaromyces pombe*. A nucleosome repeat length that is shorter than the chromatosomal DNA length. Journal of Molecular Biology **226**(4):1009-1025.

Godde, J.S. and Wolffe, A.P. (1996) Nucleosome assembly on CTG triplet repeats. Journal of Biological Chemistry **271**(25):15222-15229.

Godde, J.S., Kass, S.U., Hirst, M.C. and Wolffe, A.P. (1996) Nucleosome assembly on methylated CGG triplet repeats in the fragile X mental retardation gene 1 promoter. Journal of Biological Chemistry **271**(40):24325-24328.

Gottschling, D.E., Aparicio, O.M., Billington, B.L. and Zakian, V.A. (1990) Position effect of *S. cerevisiae* telomeres: reversible repression of Pol II transcription. Cell **63**:751-762.

Grabczyk, E. and Usdin, K. (2000) The GAA•TTC triplet repeat expanded in Friedreich's ataxia impedes transcription elongation by T7 RNA polymerase in a length and supercoil dependent manner. Nucleic Acids Research **28**(14):2815-2822.

Greider, C. W. (1999) Telomeres do D-loop-T-loop. Cell **97**(4):419-422.

Griffith, J.D., Comeau, L., Rosenfield, S., Stansel, R.M., Bianchi, A., Moss, H. and de Lange, T. (1999) Mammalian telomeres end in a large duplex loop. Cell **97**(4):503-514.

Grumbach, S. and Tahi, F. (1993) Compression of DNA sequences. Proceedings of the IEEE Symposium on Data Compression, pp 340-350.

Grumbach, S. and Tahi, F. (1994) A new challenge for compression algorithms - genetic sequences. Information Processing & Management **30**(6):875-886.

Gusfield, D. (1997) Algorithms on strings, trees, and sequences. New York: Cambridge University Press, pp 117.

Gusfield, D. and Stoye, J. (1998) Linear time algorithms for finding and representing all tandem repeats in a string. Technical Report CSE-98-4, Department of Computer Science, University of California at Davis.

Hartzog, G.A. and Winston, F. (1997) Nucleosomes and transcription: recent lessons from genetics. Current Opinion in Genetics and Development **7**:192-198.

Hemann, M.T. and Greider, C.W. (1999) G-strand overhangs on telomeres in telomerase-deficient mouse cells. Nucleic Acids Research **27**:3964-3969.

Henikoff, S. Ahmad, K. and Malik, H.S. (2001) The centromere paradox: stable inheritance with rapidly evolving DNA. Science **293**(5532):1098-1102.

Huffman, D. (1952) A method for the construction of minimum-redundancy codes. Proceedings of the Institute of Radio Engineers **40**(9):1090-1101.

Hutchison, C. A. III, Hardies, S. C., Loeb, D. D., Shehee, W. R. and Edgell, M. H. (1989) LINEs and related retroposons: Long interspersed repeated sequences in the eukaryotic genome. In Berg, D. E. and Howe, M. M. (eds), Mobile DNA, American Society for Microbiology, Washington, D. C. pp. 593-617.

Ohyama, T. (2001) Intrinsic DNA bends: an organizer of local chromatin structure for transcription. Bioessays **23**(8):708-715.

Jackson, P.J., Hugh-Jones, M.E., Adair, D.M., Green, G., Hill, K.K., Kuske, C.R., Grinberg, L.M., Abramova, F.A. and Keim, P. (1998) PCR analysis of tissue samples from the 1979 Sverdlovsk anthrax victims: the presence of multiple *Bacillus anthracis* strains in different victims. Proceeding of the National Academy of Science, USA **95**(3):1224-1229.

Jeffreys, A.J., Wilson, V. and Thein, S.L. (1985a) Hypervariable 'minisatellite' regions in human DNA. Nature **314**(6006):67-73.

Jeffreys, A.J., Wilson, V. and Thein, S.L. (1985b) Individual-specific 'fingerprints' of human DNA. Nature **316**(6023):76-79.

Jeffreys, A.J., Brookfield, J.F. and Semeonoff, R. (1985c) Positive identification of an immigration test-case using human DNA fingerprints. Nature **317**(6040):818-819.

Jeffreys, A.J. and Morton, D.B. (1987) DNA fingerprints of dogs and cats. Animal Genetics **18**(1):1-15.

Jurka, J. (1995) Origin and evolution of Alu repetitive elements. In Maraia, R. (eds), *The Impact of Short Interspersed Elements (SINEs) of the Host Genome*. R.G. Landes Co. pp 25-41.

Jurka, J. (2000) Repbase update: a database and an electronic journal of repetitive elements. Trends in Genetics **16**(9):418-420.

Jurka, J., Klonowski, P., Dagman, V. and Pelton, P. (1996) CENSOR – a program for identification and elimination of repetitive elements from DNA sequences. Computers and Chemistry **20**(1):119-121.

Jurka, J., Walichiewicz, J. and Milosavljevic, A. (1992) Prototypic sequences for human repetitive DNA. Journal of Molecular Evolution **35**:286-291.

Kannan, S. K. and Myers, E.W. (1993) An algorithm for locating non-overlapping regions of maximum alignment score. In Apostolico A., Crochemore, M., Galil, Z. and Manber, U.

(eds) Proc. Of the 4th Annual Symposium on Combinatorial Pattern matching (CPM 93), Padova, Italy, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, **684**:74-86.

Kannan, S.K. and Myers, E.W. (1996) An algorithm for locating nonoverlapping regions of maximum alignment score. SIAM Journal of Computing, **25**(3), 648-662.

Kapitonov, V. and Jurka, J. (1996) The age of Alu subfamilies. Journal of Molecular Evolution **42**:59-65.

Kass-Eisler, A. and Greider, C.W. (2000) Recombination in telomere-length maintenance. Trends in Biochemical Sciences **25**:200-206.

Karp, R.M. and Rabin, M.O. (1981) Efficient randomized pattern matching algorithms. Technical Report TR-31-81, Aiken Computation Laboratory, Harvard University.

Keim, P., Klevytska, A.M., Price, L.B., Schupp, J.M., Zinser, G., Smith, K.L., High-Jones, M.E., Okinaka, R., Hill, K.K. and Jackson, P.J. (1999) Molecular diversity in *Bacillus anthracis*. Journal of Applied Microbiology **87**(2):215-217.

Kim, N.W., Piatyszek, M.A., Prowse, K.R., Harley, C.B. West, M.D., Ho, P.L., Coviello, G.M., Wright, W.E., Weinrich, S.L. and Shay, J.W. (1994) Specific association of human telomerase activity with immortal cells and cancer. Science **266**(5193):2011-2015.

Klaerr-Blanchard, M., Chiapello, H. and Coward, E. (2000) Detecting localized repeats in genomic sequences: a new strategy and its application to *Bacillus subtilis* and *Arabidopsis thaliana* sequences. Computers and Chemistry **24**(1):57-70.

Knuth, D.E., Morris, J.H. Jr. and Pratt, V.R. (1977) Fast pattern matching in strings. SIAM Journal on Computing, **6**(2):323-350.

Kolpakov, R. and Kucherov, G. (1998) Maximal repetitions in words or how to find all squares in linear time. Technical Report 98-R-227, LORIA.

Kolpakov, R. and Kucherov, G. (1999) Finding maximal repetitions in a word in linear time. In Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, USA, pp. 596-604.

Kolapov, R. and Kucherov, G. (2000) Finding repeats with fixed gap. In Proceeding of the 7th International Symposium on String Processing and Information Retrieval, Coruña, Spain, IEEE, pp 162-168.

Kolapov, R. and Kucherov, G. (2001) Finding approximate repetitions under Hamming distance. Rapport du recherche, INRIA, RR-41630.

Kornberg, R.D. and Lorch, Y. (1999) Twenty-five years of the nucleosome fundamental particle of the eukaryote chromosome. Cell **98**:285-294.

Kurtz, S. (1999) Reducing the space requirement of suffix trees. Software-Practice and Experience **29**(13):1149-1171.

Kurtz, S. and Schleiermacher, C. (1999) REPuter: fast computation of maximal repeats in complete genomes. Bioinformatics **15**(5):426-427.

Kurtz, S., Ohlebusch, E., Schleiermacher, C., Stoye, J. and Giegerich, R. (2000) Computation and visualization of degenerate repeats in complete genome. Proceedings of the International Conference for Intelligent Systems for Molecular Biology **8**:228-238.

Kurtz, S., Choudhuri, J.V., Ohlebusch, E., Schleiermacher, C., Stoye, J. and Giegerich, R. (2001) REPuter: the manifold applications of repeat analysis on a genomic scale. Nucleic Acids Research **29**(22):4633-4642.

Landau, G.M. and Schmidt, J.P. (1993) An algorithm for approximate tandem repeats. In Apostolico A., Crochemore, M., Galil, Z. and Manber, U. (eds) Proc. Of the 4th Annual Symposium on Combinatorial Pattern Matching (CPM 93), Padova, Italy, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, **684**:120-133.

Landau, G.M., Schmidt, J.P. and Sokol, D. (2001) An algorithm for approximate tandem repeats. Journal of Computational Biology **8**(1):1-18.

Ledbetter, S.A., Nelson, D.L., Warren, S.T. and Ledbetter, D.H. (1990) Rapid isolation of DNA probes within specific chromosome regions by interspersed repetitive sequence polymerase chain reaction. Genomics **6**:475-481.

Lee, C. Critcher, R. Zhang, J.G., Mills, W. and Farr, C.J. (2000) Distribution of gamma satellite DNA on the human X and Y chromosomes suggests that it is not required for mitotic centromere function. Chromosoma **109**(6):381-389.

Lefévre, C. and Ikeda, J.E. (1993) The position end-set tree: a small automaton for word recognition in biological sequences. Computer Applications in the Biosciences **9**(30):343-348.

Lempel, A. and Ziv, J. (1978) Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory **IT-24**:530-536.

Levenshtein, V.I. (1966) Binary codes capable of correcting insertions and reversals. Soviet Physics Dokl. **10**:707-710.

Lupski, J.R., Wise, C.A., Kuwano, A., Pentao, L., Parke, J.T., Glaze, D.G., Ledbetter, D.H., Greenberg, F. and Patel, P.I. (1993) Gene dosage is a mechanism for Charcot-Marie-Tooth disease type 1A. Nature Genetics **1**(1):29-33.

Main, M.G. and Lorentz, R.J. (1984) An O(n log n) algorithm for finding all repetitions in a string. Journal of Algorithms **5**:422-432.

Main, M.G. and Lorentz, R.J. (1985) Linear time recognition of square free strings. In A. Apostolico and Z. Galil (eds), *Combinatorial Algorithms on Words*, a NATO ASI Series, Series F: Computer and System Sciences, Springer-Verlag, Berlin, **F12**:272-278.

Main, M.G. (1989) Detecting leftmost maximal periodicities. Discrete Applied Mathematics, **25**:145-153.

Makarov, V.L., Lejnine, S., Bedoyan, J. and Langmore, J.P. (1993) Nucleosomal organization of telomere-specific chromatin in rat. Cell 73(4):775-787.

Martinez, H.M. (1983) An efficient method for finding repeats in molecular sequences. Nucleic Acids Research **11**(13):4629-34.

Matsumoto, T., Sadakane, K. and Imai, H. (2000) Biological sequence compression algorithms. The Eleventh Workshop on Genome Informatics. Genome Informatics **11**:43-52.

McClintock, B. (1942). The fusion of broken ends of chromosomes following nuclear fission. Proceedings of the National Academy of Science, USA **28**:458-463.

McCreight, E.M. (1976) A space-economical suffix tree construction algorithm. Journal of the ACM **23**:262-272.

Milosavljevic, A. (1993) Discovering sequence similarity by the algorithmic significance method. Proceedings of the International Conference for Intelligent Systems for Molecular Biology **1**:284-291.

Milosavljevic, A. and Jurka, J. (1993) Discovering simple DNA sequences by the algorithmic significance method. Computer Applications in the Biosciences **9:**407-411.

Mitas, M. (1997) Trinucleotide repeats associated with human disease. Nucleic Acids Research **25**(12):2245-2254.

Muller, H.J. (1938) The remaking of chromosomes. Collecting Net **13**:181-198.

Myers, E.W. and Miller, W. (1988) Optimal alignments in linear space. Computer Applications in the Biosciences **4**:11-17.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48**:443-453.

Nelson, D.L. (1995) The fragile X syndrome. Seminars in Cell Biology **6**(1):5-11.

Nelson, D.L., Ledbetter, S.A., Corbo, L., Victoria, M.F., Ramirez-Solis, R., Webster, T.D., Ledbetter, D.H. and Caskey, C.T. (1989) Alu polymerase chain reaction: a method for rapid isolation of human-specific sequences from complex DNA sources. Proceedings of the National Academy of Science, USA **86**:6686-6690.

Pearson, C.E. and Sinden, R.R. (1998) Trinucleotide repeat DNA structures: dynamic mutations from dynamic DNA. Current Opinion in Structural Biology **8**(3):321-30.

Pearson, W. and Lipman, D. (1988) Improved tools for biological sequence comparison. Proceedings of the National Academy of Science, USA **85**:2444-2448.

Pech, M., Streeck, R.E. and Zachau, H.G. (1979) Patchwork structure of a bovine satellite DNA. Cell **18**(3):883-893.

Perutz, M.F., Johnson, T., Suzuki, M. and Finch, J.T. (1994) Glutamine repeats as polar zippers: their possible role in inherited neurodegenerative diseases. Proceedings of the National Academy of Science, USA **91**:5355-5358.

Porto, A. H. L. and Barbosa, V.C. (2002) Finding approximate palindromes in strings. Pattern Recognition **35**(11):2581-2591.

Pryde, F.E., Gorham, H.C. and Louis, E.J. (1997) Chromosome ends: all the same under their caps. Current Opinion in Genetics and Development **7**(6):822-828.

Rabin, M. (1985) Discovering repetitions in strings. In A. Apostolico and Z. Galil (eds), *Combinatorial Algorithms on Words*, a NATO ASI Series, Series F: Computer and System Sciences, Springer-Verlag, Berlin, **F12**:272-278.

Rivals, E., Delgrange, O., Delahaye, J-P. and Dauchet, M. (1995) A first step towards chromosome analysis by compression algorithms. First International IEEE Symposium on Intelligence in Neural and Biological Systems, 233-239.

Rivals, E., Dauchet, M., Delahaye, J-P. and Delgrange, O. (1996) Compression and genetic sequence analysis. Biochimie **78**(5):315-322.

Rivals, E., Delgrange, O., Delahaye, J-P., Dauchet, M., Delorme, M.O., Henaut, A. and Ollivier, E. (1997) Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. Computer Applications in the Biosciences **13**(2):131-136.

Sagot, M.F. and Myers, E.W. (1998) Identifying satellites and periodic repetitions in biological sequences. Journal of Computational Biology **5**(3):539-553.

Schmid, C. W., Deka, N. and Matera, A. G. (1990) Repetitive human DNA: The shape of things to come. In K. W. Adolph (ed.), Chromosome: Eukaryotic, prokaryotic and viral. CRC Press, Boca Raton, Florida, U. S. A. Volume I pp. 3-29.

Schmidt, J.P. (1998) All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings. SIAM Journal on Computing **27**(4):972-992.

Schueler, M.G., Higgins, A.W., Rudd, M.K., Gustashaw, K. and Willard H.F. (2001) Genomic and genetic definition of a functional human centromere. Science **294**(5540):109-115.

Schwartz, L.S., Tarleton, J., Popovich, B., Seltzer, W.K. and Hoffman, E.P. (1993) Fluorescent multiplex linkage analysis and carrier detection for Duchenne/Becker muscular dystrophy. American Journal of Human Genetics **51**:721-729.

Shore, D. (2001) Telomeric chromatin: replicating and wrapping up chromosome ends. Current Opinion on Genetics and Development 11(2):189-198.

Shlyakhtenko, L.S, Potaman, V.N., Sinden, R.R. and Lyubchenko, Y.L. (1998) Structure and dynamics of supercoil-stabilized DNA cruciforms. Journal of Molecular Biology **280**(1):61-72.

Sinden, R.R. (1999) HUMAN GENETICS '99: TRINUCLEOTIDE REPEATS Biological implications of the DNA structures associated with disease-causing triplet repeats. American Journal of Human Genetics **64**:346-353.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. Journal of Molecular Biology **147**:195-197.

States, D.J. and Agarwal, P. (1996) Compact encoding strategies for DNA sequence similarity search. Proceedings of the International Conference for Intelligent Systems in Molecular Biology **4**:211-217.

Storer, J.A. (1988) Data Compression: Methods and Theory. Computer Science Press, Rockville, MD.

Stoye, J. and Gusfield, D. (2001) Simple and flexible detection of contiguous repeats using a suffix tree. Theoretical ComputerScience 270(1-2):843-856.

Sullivan, B.A., Blower, M.D. and Karpen, G.H. (2001) Determining centromere identity: cyclical stories and forking paths. Nature Reviews Genetics **2**(8):584-596.

Temin, H. M. (1985) Reverse transcription in the eukaryotic genome: Retroviruses, pararetroviruses, retrotransposons and retrotranscripts. Molecular Biology and Evolution **2**(6):455-468.

Thein, S.L., Jeffreys, A.J., Gooi, H.C., Cotter, F., Flint, J., O'Connor, N.T., Weatherall, D.J. and Wainscoat, J.S. (1987) Detection of somatic changes in human cancer DNA by fingerprint analysis. British Journal of Cancer **55**(4):353-356.

Ukkonen, E. (1995) On-line construction of suffix-trees. Algorithmica **14**:249-260.

van Steensel, B. Smogorzewska, A. and de Lange, T. (1998) TRF2 protects human telomeres from end-to-end fusions. Cell **92**:401-413.

Vogt, P. (1990) Potential genetic functions of tandem repeated DNA sequence blocks in the human genome are based on a highly conserved "chromatin folding code." Human Genetics **84**(4):301-336.

Volfovsky, N., Haas, B.J. and Salberg, S.L. (2001) A clustering method for repeat analysis in DNA sequences. Genome Biology **2**(8):research0027.

Vynnycky, E., Nagelkerke, N., Borgdorff, M.W., van Soolingen, D., van Embden, J.D. and Fine, P.E. (2001) The effect of age and study duration on the relationship between 'clustering' of DNA fingerprint patterns and the proportion of tuberculosis disease attributable to recent transmission. Epidemiology and Infection **126**(1):43-62.

Wampler, S. (2001) Lab scientists develop DNA signatures for salmonella strain that are expected to drastically speed detection. Lawrence Livermore National Laboratory, News Release NR-01-10-08, 8 Oct.

Wang, Y.H., Amirhaeri, S., Kang, S., Wells, R.D. and Griffith, J.D. (1994) Preferential nucleosome assembly at DNA triplet repeats from the myotonic dystrophy gene. Science **265**(5172):669-671.

Wang, Y.H. and Griffith, J. (1995) Expanded CTG triplet blocks from the myotonic dystrophy gene create the strongest known natural nucleosome positioning elements. Genomics **25**(2):570-573.

Wang, Y.H. and Griffith, J. (1996) Methylation of expanded CCG triplet repeat DNA from fragile X syndrome patients enhances nucleosome exclusion. Journal of Biological Chemistry **271**(38):22937-22940.

Weiner, P. (1973) Linear pattern matching algorithms. Proc. Of the 14th IEEE Symp. On Switching and Automata Theory, pp. 1-11.

Weiner, A. M., Deininger, P. L. and Efstratiadis, A. (1986) Nonviral retroposons: genes, pseudogenes, and transposable elements generated by the reverse flow of genetic information. Annual Review of Biochemistry **55**:631-661.

Widlund, H.R., Cao, H., Simonsson, S., Magnusson, E., Simonsson, T., Nielsen, P.E., Kahn, J.D., Crothers, D.M. and Kubistas, M. (1997) Identification and characterization of genomic nucleosome-positioning sequences. Journal of Molecular Biology **267**(4):807-817.

Widom, J. (1992) A relationship between the helical twist of DNA and the ordered positioning of nucleosomes in all eukaryotic cells. Proceedings of the National Academy of Science, USA **89**(3):1095-1099.

Widom, J. (1998) Structure, dynamics and function of chromatin in vitro. Annual Review of Biophysics and Biomolecular Structure **27**:285-327.

Willems, F.M.J., Shtarkov, Y.M. and Tjalkens, T.J. (1995) The context tree weighting method: basic properties. IEEE Transactions on Information Theory **IT-41**(3):653-664.

Williams, J.G., Kubelik, A.R., Livak, K.J., Rafalski, J.A. and Tingey S.V. (1990) DNA polymorphisms amplified by arbitrary primers are useful as genetic markers. Nucleic Acids Research **18**(22):6531-6535.

Witten, I., Neal, R. and Cleary, J. (1987) Arithmetic coding for data compression. Communications of the ACM, **30**(6):520-541.

Yang, C.F., Kim, J.M., Molinari, E. and DasSarma, S. (1996) Genetic and topological analysis of the *bop* promoter of *Halobacterium halobium*: stimulation by DNA supercoiling and non-B DNA structure. Journal of Bacteriology **178**:840-845.

Ziv, J. and Lempel, A. (1977) A universal algorithm for sequential data compression. IEEE Transactions on Information Theory **IT-23**(3):337-343.

3. DNA sequence data and data model

The computational definition of repetition in character strings differs from the biological definition of repetition in DNA sequences. This research focuses on the biological structure and composition of tandem repeats in DNA sequences. A review of the biological literature and visual analysis of tandem repeat regions form the basis for defining the structure of tandem repeats. Structural definitions lend themselves to computational identification of tandem repeats. The goal is to develop a model that captures the essence of biological tandem repeats and that enables consistent identification of biologically relevant tandem repeat regions in DNA sequences.

This section describes a structure-based tandem repeat model: the basis for the model, the formal definition of the model and a collection of DNA sequences representing the model. The basis for the model includes a review of the biological literature (see Section 2.1) and a visual analysis of repetitive elements in DNA sequences. Section 3.1 describes a visual analysis of tandem repeat regions in DNA sequences that reveals a range of pattern conservation levels, a range of pattern lengths and various types of pattern structure complexities. Section 3.2 presents the formal definition of a model for pattern structures in tandem repeat regions based on the biological literature review and the visual analysis of tandem repeat regions. Finally, Section 3.3 describes a collection of tandem repeat regions in DNA sequences that are both easy and challenging to locate and characterize. The model and the collection of tandem repeat regions enable the development of a computational approach for identifying tandem repeat regions in DNA sequences (Chapter 4). Chapter 5 presents an analysis of the collection of sequences using the identification algorithm.

3.1. Sequence data visual analysis

A critical step in the development of any computational algorithm involves analyzing the data. Here, a visual analysis of DNA sequences from the GenBank database was performed for sequences with annotations having repetition related words or phrases. In total, 1700 sequences containing tandem repeat regions were analyzed; 250 sequences in detail.

Initial visual analysis for a tandem repeat region records a region pattern, the copy number of the pattern in the region and the pattern conservation level. In general, a single pattern captures the recurrence within the region but for some regions, several patterns are necessary or a pattern containing ambiguous nucleotide characters is necessary in order to represent the variation within the region. The copy number for a region's pattern includes both full and partial consecutive copies of the pattern, e.g. a region may have 7.5 copies of a pattern. Finally, most regions contain sequence differences due to nucleotide substitution, insertion or deletion. This analysis assigns a subjective rating of extreme, moderate, mild or no loss of pattern conservation and for moderate and mild cases, lists the differences individually.

The analysis determines that tandem repeat regions cover a wide range of pattern sizes, copy numbers and pattern conservation levels and exhibit complex pattern structures. Patterns range in length from several nucleotides to hundreds of nucleotides. Copy numbers range from hundreds of copies down to two occurrences, the minimum necessary. Also, although some regions do exhibit perfect conservation, most regions have imperfect pattern conservation. Pattern imperfections are differences in the sequence, i.e. nucleotide substitution, insertion and deletion such that substitutions are the most common difference. Regions having insertions and deletions almost always have substitutions as well. Finally, patterns exhibit both simple nucleotide sequence structures and complex pattern structures that involve nesting of one pattern within another.

Tandem repeat regions tend to reflect specific combinations of pattern size, copy number, pattern conservation level and pattern structure. Long patterns tend to have several occurrences while short patterns span the entire spectrum from two to many occurrences. Long patterns tend not to exhibit perfect conservation while regions with short patterns do exhibit perfect conservation, even for regions having a large number of pattern occurrences. Furthermore, initial visual observations of a region may yield a substantial number of nucleotide differences between occurrences that upon further analysis reveals more complex pattern structures recurring across multiple pattern copies.

The visual sequence analysis reveals several classes of tandem repeat regions: regions having a short, moderate or long, simple sequence pattern, regions having complex pattern structures with a SSR nested in the pattern or with multiple periodicities. Regions with short, simple sequence patterns tend to be well conserved and have greater than ten copies. Increasing the length of the pattern tends to decrease the copy number and the pattern conservation in the region. Other regions have complex pattern structures with nested SSRs in their pattern such that the nested SSRs have either a fixed copy number throughout the region or a variable copy number throughout the region. Variable copy number SSRs nested in a region’s pattern creates a variable periodicity for the overall region. Other regions have complex pattern structures defined by multiple periodicities reflecting multiple patterns. In some of these regions, the periodicities are multiples of a base periodicity and represents patterns across multiple copies of a base pattern. These classes form the basis for the structure-based tandem repeat region data model.

3.2. Data model.

A ***tandem repeat*** is a consecutive recurrence of a subsequence. Each recurrence is termed a ***copy*** and the number of copies in a tandem repeat region is termed the ***copy number***. Furthermore, a ***pattern*** describes the copies that occur in a tandem repeat region. Historically, the pattern is a sequence of nucleotides but this research allows a pattern to be defined by a complex expression of nucleotides. In particular, this research uses notation borrowed from regular expressions to define complex pattern structures.

This section presents a definition of tandem repeats by describing regular patterns, regular sets of pattern instances and regular sets of tandem repeats. A regular pattern is a shorthand notation for a regular set of pattern instances. Elements in this set represent individual copies that can occur in a tandem repeat, or more specifically, the set represents all copies that can possibly be defined for a regular pattern. Furthermore, a regular set of pattern instances has a corresponding regular set of tandem repeats that represents all possible tandem repeats or more specifically, all possible combinations of pattern instances that form tandem repeats. A formal definition follows.

Let Σ be the alphabet consisting of the nucleotide symbols: **A**, **C**, **G**, **T**, i.e. $\Sigma = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$. Let Σ^* be all words of finite length over Σ. Σ^* contains the empty word, ε. Let $\Sigma^+ = \Sigma^* - \{\varepsilon\}$.

A regular pattern, P, over Σ^+ is a simple or complex expression that denotes a regular set of pattern instances, I_P, that is over Σ^*. The regular set of tandem repeats, S_P, is over Σ^+

and represents regions composed of two or more non-empty elements of I_{P}. P, I_{P} and S_{P} are defined recursively as follows.

1) For any word $\boldsymbol{w}$ in Σ^{+}, the regular pattern $\mathsf{P} = \boldsymbol{w}$ corresponds to the regular set of pattern instances

$$\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{(\boldsymbol{w})} = \{\boldsymbol{w}\} \text{ .}$$

That is, in this base case, I_{P} simply contains the single, non-empty word, $\boldsymbol{w}$. The regular set of tandem repeats for this base case corresponds to

$$\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{(\boldsymbol{w})} = \{\boldsymbol{x} \in \Sigma^{+} : \boldsymbol{x} = \boldsymbol{u}_1 \circ \boldsymbol{u}_2 \circ \ldots \circ \boldsymbol{u}_{\boldsymbol{j}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{j}} \in \mathsf{I}_{(\boldsymbol{w})}, \boldsymbol{j} \geq 2\} .$$

That is, S_{P} is the collection of all words made up of two or more copies of $\boldsymbol{w}$. In subsequent definitions, $\boldsymbol{u}_1 \circ \boldsymbol{u}_2 \circ \ldots \circ \boldsymbol{u}_{\boldsymbol{j}}$ will be written as $\boldsymbol{u}_1 \boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{j}}$.

Example 1.1: The regular pattern, $\mathsf{P} = \mathbf{AC}$, denotes the sets $\mathsf{I}_{(\mathbf{AC})} = \{\mathbf{AC}\}$ and

$$\mathsf{S}_{(\mathbf{AC})} = \{\mathbf{ACAC}, \mathbf{ACACAC}, \ldots\} = \{(\mathbf{AC})_2, (\mathbf{AC})_3, \ldots\} .$$

Example 1.2: The regular pattern, $\mathsf{P} = \mathbf{CGG}$, denotes the sets $\mathsf{I}_{(\mathbf{CGG})} = \{\mathbf{CGG}\}$ and

$$\mathsf{S}_{(\mathbf{CGG})} = \{\mathbf{CGGCGG}, \mathbf{CGGCGGCGG}, \ldots\} = \{(\mathbf{CGG})_{\mathbf{2}}, (\mathbf{CGG})_{\mathbf{3}}, \ldots\} .$$

2) If R is a regular pattern and denotes the regular set of pattern instances I_{R} and the regular set of tandem repeats S_{R}, then the ***fixed repetition*** of R, $(\mathsf{R})_{\boldsymbol{k}}$, the ***variable repetition*** of R, $(\mathsf{R})_{\boldsymbol{k-l}}$, and the ***closure*** of R, $(\mathsf{R})_{*}$, denote the regular sets of pattern instances, $\mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})}$, $\mathsf{I}_{((\mathsf{R})_{\boldsymbol{k-l}})}$, and $\mathsf{I}_{((\mathsf{R})_{*})}$, respectively, and the regular sets of tandem repeats, $\mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})}$, $\mathsf{S}_{((\mathsf{R})_{\boldsymbol{k-l}})}$, and $\mathsf{S}_{((\mathsf{R})_{*})}$, respectively.

A. (Fixed Repetition): The regular pattern $\mathsf{P} = (\mathsf{R})_{\boldsymbol{k}}$, $\boldsymbol{k} \geq 1$ corresponds to the regular set of pattern instances

$$\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})} = \{\boldsymbol{x} \in \Sigma^{*} : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{k}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{k}} \in \mathsf{I}_{(\mathsf{R})}\}.$$

That is, the concatenation of ***k*** non-unique elements in I_{P}.

The regular set of tandem repeat regions for fixed repetition corresponds to

$$\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})} = \left\{ \begin{array}{r} \boldsymbol{x} \in \Sigma^{+} : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{j}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{j}} \in \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})}, \boldsymbol{j} \geq 2 \\ \text{and } \boldsymbol{u}_{\boldsymbol{m}}, \boldsymbol{u}_{\boldsymbol{n}} \neq \varepsilon, 1 \leq \boldsymbol{m}, \boldsymbol{n} \leq \boldsymbol{j}, \boldsymbol{m} \neq \boldsymbol{n} \end{array} \right\}.$$

That is, S_{P} is the collection of all words made up of two or more non-empty pattern instances in I_{P}.

Example 2.A.1: $\mathsf{P} = (\mathbf{TG})_3$ uses $\mathsf{R} = \mathbf{TG}$, $\boldsymbol{k} = 3$ and corresponds to

$$\mathsf{I}_{((\mathbf{TG})_3)} = \{\mathbf{TGTGTG}\} = \{(\mathbf{TG})_3\} \text{ and}$$

$$\begin{aligned} \mathsf{S}_{((\mathbf{TG})_3)} &= \{\mathbf{TGTGTGTGTGTG}, \mathbf{TGTGTGTGTGTGTGTGTG}, \ldots\} \\ &= \{(\mathbf{TG})_6, (\mathbf{TG})_9, (\mathbf{TG})_{12}, \ldots\} \end{aligned}.$$

Example 2.A.2: From our definitions, $\mathsf{I}_{(((\mathsf{R})_{\boldsymbol{c}})_{\boldsymbol{d}})} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{c}*\boldsymbol{d}})}$. Therefore, $\mathsf{P} = ((\mathsf{R})_{\boldsymbol{c}})_{\boldsymbol{d}}$ will denote the same set of tandem repeats as $\mathsf{P} = (\mathsf{R})_{\boldsymbol{c}*\boldsymbol{d}}$. So, if $\mathsf{R} = \mathbf{TG}$, $\boldsymbol{c} = 3$ and $\boldsymbol{d} = 2$ then $\mathsf{P} = ((\mathbf{TG})_3)_2$ which can be reduced to $\mathsf{P} = ((\mathbf{TG})_3)_2 = (\mathbf{TG})_{3*2} = (\mathbf{TG})_6$. This corresponds to

$\mathsf{I}_{((\mathbf{TG})_6)} = \{\mathbf{TGTGTGTGTGTG}\} = \{(\mathbf{TG})_6\}$ and the regular set of tandem

repeats $$\mathsf{S}_{((\mathbf{TG})_6)} = \left\{\begin{matrix}\mathbf{TGTGTGTGTGTGTGTGTGTGTGTG,} \\ \mathbf{TGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTG,}\ldots\end{matrix}\right\} . \\ = \{(\mathbf{TG})_{12}, (\mathbf{TG})_{18}, (\mathbf{TG})_{24}, \ldots\}$$

B. (Variable Repetition): The regular pattern $\mathsf{P} = (\mathsf{R})_{\boldsymbol{k-l}}$, $\boldsymbol{l} > \boldsymbol{k} \geq 0$ corresponds to the regular set of pattern instances

$$\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k-l}})} = \{\boldsymbol{x} \in \Sigma^* : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{i}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{i}} \in \mathsf{I}_{\mathsf{R}}, \boldsymbol{l} \geq \boldsymbol{i} \geq \boldsymbol{k}\} .$$

That is, the concatenation of at least ***l*** and at most ***k*** non-unique elements in I_{R}.

The regular set of tandem repeat regions for variable repetition corresponds to

$$\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{((\mathsf{R})_{\boldsymbol{k-l}})} = \left\{\begin{matrix}\boldsymbol{x} \in \Sigma^+ : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{j}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{j}} \in \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k-l}})}, \boldsymbol{j} \geq 2 \\ \text{and } \boldsymbol{u}_{\boldsymbol{m}}, \boldsymbol{u}_{\boldsymbol{n}} \neq \varepsilon, 1 \leq \boldsymbol{m}, \boldsymbol{n} \leq \boldsymbol{j}, \boldsymbol{m} \neq \boldsymbol{n}\end{matrix}\right\} .$$

That is, S_{P} is the collection of all words made up of two or more non-empty pattern instances in I_{P}.

Example 2.B.1: $\mathsf{P} = (\mathbf{TG})_{2-4}$ uses $\mathsf{R} = \mathbf{TG}$, $\boldsymbol{k} = 2$ and $\boldsymbol{l} = 4$ and corresponds to

$\mathsf{I}_{((\mathbf{TG})_{2-4})} = \{\mathbf{TGTG, TGTGTG, TGTGTGTG}\} = \{(\mathbf{TG})_2, (\mathbf{TG})_3, (\mathbf{TG})_4\}$ and

$$\mathsf{S}_{((\mathbf{TG})_{2-4})} = \{\mathbf{TGTGTGTG, TGTGTGTGTG, TGTGTGTGTGTG,} \ldots\} \\ = \{(\mathbf{TG})_4, (\mathbf{TG})_5, (\mathbf{TG})_6, \ldots\} .$$

Example 2.B.2: As in Example 2.A.2, $\mathsf{P} = ((\mathbf{TG})_{2-4})_{1-2}$ can be reduced to

$\mathsf{P} = ((\mathbf{TG})_{2-4})_{1-2} = (\mathbf{TG})_{(2*1)-(4*2)} = (\mathbf{TG})_{2-8}$. This corresponds to the sets

$$\mathsf{I}_{((\mathbf{TG})_{2-8})} = \left\{ \begin{array}{l} \mathtt{TGTG,TGTGTG,TGTGTGTG,TGTGTGTGTG,} \\ \mathtt{TGTGTGTGTGTG,TGTGTGTGTGTGTG,} \\ \mathtt{TGTGTGTGTGTGTGTG} \end{array} \right\} \text{ and}$$

$$\begin{aligned} \mathsf{S}_{((\mathbf{TG})_{2-8})} &= \{\mathtt{TGTGTGTG,TGTGTGTGTG,TGTGTGTGTGTG,}\ldots\} \\ &= \{(\mathbf{TG})_4, (\mathbf{TG})_5, (\mathbf{TG})_6, \ldots\} \end{aligned}.$$

C. (Open Closure): The regular pattern $\mathsf{P} = (\mathsf{R})_* = (\mathsf{R})_{0-*} = (\mathsf{R})_{0+}$ is a variable repetition of integers from zero to infinity and corresponds to the regular set of pattern instances

$$\mathsf{I}_\mathsf{P} = \mathsf{I}_{((\mathsf{R})_*)} = \{\boldsymbol{x} \in \Sigma^* : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{i}} \text{ for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{i}} \in \mathsf{I}_\mathsf{R} \text{, for each } \boldsymbol{i} \geq 0\}.$$

That is, the concatenation of zero or more non-unique elements in I_R including the empty word, ε. The regular set of tandem repeat regions corresponds to

$$\mathsf{S}_\mathsf{P} = \mathsf{S}_{((\mathsf{R})_*)} = \left\{ \begin{array}{l} \boldsymbol{x} \in \Sigma^+ : \boldsymbol{x} = \boldsymbol{u}_1\boldsymbol{u}_2 \ldots \boldsymbol{u}_{\boldsymbol{j}} \\ \qquad \text{for } \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{j}} \in \mathsf{I}_{((\mathsf{R})_*)}, \text{ for each } \boldsymbol{j} \geq 2 \\ \qquad \text{and } \boldsymbol{u}_{\boldsymbol{m}}, \boldsymbol{u}_{\boldsymbol{n}} \neq \varepsilon, 1 \leq \boldsymbol{m}, \boldsymbol{n} \leq \boldsymbol{j}, \boldsymbol{m} \neq \boldsymbol{n} \end{array} \right\}.$$

That is, S_P is the collection of all words made up of two or more non-empty pattern instances in I_P.

Example 2.C.1: $\mathsf{P}=(\mathbf{TG})_*$ uses $\mathsf{R}=\mathbf{TG}$ and corresponds to

$$\mathsf{I}_{((\mathbf{TG})_*)}=\{\varepsilon,\mathbf{TG},\mathbf{TGTG},\mathbf{TGTGTG},\ldots\}=\{\varepsilon,\mathbf{TG},(\mathbf{TG})_2,(\mathbf{TG})_3,\ldots\}\text{ and}$$

$$\begin{aligned}\mathsf{S}_{((\mathbf{TG})_*)}&=\{\mathbf{TGTG},\mathbf{TGTGTG},\mathbf{TGTGTGTG},\mathbf{TGTGTGTGTG},\ldots\}\\&=\{(\mathbf{TG})_2,(\mathbf{TG})_3,(\mathbf{TG})_4,(\mathbf{TG})_5,\ldots\}\end{aligned}.$$

Example 2.C.2: $\mathsf{P}=((\mathbf{TG})_4)_*$ uses $\mathsf{R}=(\mathbf{TG})_4$. As in Example 2.A.2, this can be reduced to $\mathsf{P}=((\mathbf{TG})_4)_*=((\mathbf{TG})_4)_{0-*}=(\mathbf{TG})_{(4*0),(4*1),\ldots}=(\mathbf{TG})_{0,4,8,\ldots}$ which corresponds to

$$\begin{aligned}\mathsf{I}_{((\mathbf{TG})_{4+})}&=\{\varepsilon,\mathbf{TGTGTGTG},\mathbf{TGTGTGTGTGTGTGTG},\ldots\}\\&=\{\varepsilon,(\mathbf{TG})_4,(\mathbf{TG})_8,\ldots\}\end{aligned}\text{ and}$$

$$\begin{aligned}\mathsf{S}_{((\mathbf{TG})_{4+})}&=\left\{\begin{matrix}\mathbf{TGTGTGTGTGTGTGTG},\mathbf{TGTGTGTGTGTGTGTGTGTGTGTG}\\\mathbf{TGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTG},\ldots\end{matrix}\right\}\\&=\{(\mathbf{TG})_8,(\mathbf{TG})_{12},(\mathbf{TG})_{16},(\mathbf{TG})_{20},\ldots\}\end{aligned}.$$

Example 2.C.3: $\mathsf{P}=((\mathbf{TG})_*)_4$ uses $\mathsf{R}=(\mathbf{TG})_*,\boldsymbol{k}=4$. As in Example 2.A.2, this can be reduced to $\mathsf{P}=((\mathbf{TG})_*)_4=((\mathbf{TG})_{0-*})_4=(\mathbf{TG})_{(0*4),(1*4),\ldots}=(\mathbf{TG})_{0,4,8,\ldots}$ which corresponds to $\mathsf{I}_{((\mathbf{TG})_{4+})}=\{\varepsilon,\mathbf{TGTGTGTG},\mathbf{TGTGTGTGTGTGTGTG},\ldots\}$ and

$$\begin{aligned}\mathsf{S}_{((\mathbf{TG})_{4+})}&=\left\{\begin{matrix}\mathbf{TGTGTGTGTGTGTGTG},\mathbf{TGTGTGTGTGTGTGTGTGTGTGTG},\\\mathbf{TGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTG},\ldots\end{matrix}\right\}\\&=\{(\mathbf{TG})_8,(\mathbf{TG})_{12},(\mathbf{TG})_{16},(\mathbf{TG})_{20},\ldots\}\end{aligned}.$$

Example 2.C.4: $\mathsf{P}=((\mathbf{TG})_*)_*$ uses $\mathsf{R}=(\mathbf{TG})_*$. As in Example 2.A.2, this can be reduced to $\mathsf{P}=((\mathbf{TG})_*)_*=((\mathbf{TG})_{0-*})_{0-*}=((\mathbf{TG})_{(0*0)-*})=(\mathbf{TG})_{0-*}=(\mathbf{TG})_*$

and corresponds to $\mathsf{I}_{((\mathbf{TG})_*)} = \{\varepsilon, \mathbf{TG}, \mathbf{TGTG}, \mathbf{TGTGTG}, \ldots\}$ and

$$\begin{aligned} \mathsf{S}_{((\mathbf{TG})_*)} &= \{\mathbf{TGTG}, \mathbf{TGTGTG}, \mathbf{TGTGTGTG}, \mathbf{TGTGTGTGTG}, \ldots\} \\ &= \{(\mathbf{TG})_2, (\mathbf{TG})_3, (\mathbf{TG})_4, (\mathbf{TG})_5, \ldots\} \end{aligned}.$$

3) If Q and R are regular patterns and denote the regular sets of pattern instances I_Q and I_R, respectively, and the regular sets of tandem repeats S_Q and S_R, respectively, then the ***concatenation*** of regular patterns, $\mathsf{P} = (\mathsf{QR})$, denotes the regular sets of pattern instances,

$$\mathsf{I}_\mathsf{P} = \mathsf{I}_{(\mathsf{QR})} = \{\boldsymbol{x} \in \Sigma^* : \boldsymbol{x} = \boldsymbol{uv} \text{ for } \boldsymbol{u} \in \mathsf{I}_\mathsf{Q}, \boldsymbol{v} \in \mathsf{I}_\mathsf{R}\}.$$

That is, the concatenation of an element from I_Q to an element from I_R. The regular sets of tandem repeats for the concatenation operation corresponds to

$$\mathsf{S}_\mathsf{P} = \mathsf{S}_{(\mathsf{QR})} = \left\{ \begin{aligned} \boldsymbol{x} \in \Sigma^+ : \boldsymbol{x} = \boldsymbol{w}_1\boldsymbol{w}_2\ldots\boldsymbol{w}_{\boldsymbol{j}} \text{ for } \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\boldsymbol{j}} \in \mathsf{I}_{(\mathsf{QR})}, \boldsymbol{j} \geq 2 \\ \text{and } \boldsymbol{u}_{\boldsymbol{m}}, \boldsymbol{u}_{\boldsymbol{n}} \neq \varepsilon, 1 \leq \boldsymbol{m}, \boldsymbol{n} \leq \boldsymbol{j}, \boldsymbol{m} \neq \boldsymbol{n} \end{aligned} \right\}.$$

That is, S_P is the collection of all words made up of two or more non-empty pattern instances in I_P.

Example 3.1: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = \mathbf{TG}$ corresponds to a regular pattern of

$\mathsf{P} = (\mathsf{QR}) = \mathbf{AC} \circ \mathbf{TG} = \mathbf{ACTG}$, a regular set of pattern instances

$\mathsf{I}_\mathsf{P} = \mathsf{I}_{(\mathsf{QR})} = \{\mathbf{ACTG}\}$ and the regular set of tandem repeats

$$\begin{aligned} \mathsf{S}_\mathsf{P} = \mathsf{S}_{(\mathsf{QR})} &= \{\mathbf{ACTGACTG}, \mathbf{ACTGACTGACTG}, \ldots\} \\ &= \{(\mathbf{ACTG})_2, (\mathbf{ACTG})_3, (\mathbf{ACTG})_4, \ldots\} \end{aligned}.$$

Example 3.2: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = (\mathbf{TG})_2$ corresponds to a regular pattern of

$\mathsf{P} = (\mathsf{QR}) = \mathbf{AC} \circ (\mathbf{TG})_2 = \mathbf{AC}(\mathbf{TG})_2$, a regular set of pattern instances

$\mathsf{I}_\mathsf{P} = \mathsf{I}_{(\mathsf{QR})} = \{\mathbf{ACTGTG}\}$ and the regular set of tandem repeats

$$\begin{aligned} \mathsf{S}_\mathsf{P} = \mathsf{S}_{(\mathsf{QR})} &= \{\mathbf{ACTGTGACTGTG}, \mathbf{ACTGTGACTGTGACTGTG}, \ldots\} \\ &= \{(\mathbf{ACTGTG})_2, (\mathbf{ACTGTG})_3, (\mathbf{ACTGTG})_4, \ldots\} \end{aligned}.$$

Example 3.3: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = (\mathbf{TG})_*$ corresponds a regular pattern of

$\mathsf{P} = (\mathsf{QR}) = \mathbf{AC} \circ (\mathbf{TG})_* = \mathbf{AC}(\mathbf{TG})_*$, a regular set of pattern instances

$$\begin{aligned} \mathsf{I}_\mathsf{P} = \mathsf{I}_{(\mathsf{QR})} &= \{\mathbf{AC}, \mathbf{ACTG}, \mathbf{ACTGTG}, \mathbf{ACTGTGTG}, \ldots\} \\ &= \{\mathbf{AC}, \mathbf{ACTG}, \mathbf{AC}(\mathbf{TG})_2, \mathbf{AC}(\mathbf{TG})_3, \ldots\} \end{aligned}$$

and the regular set of tandem repeats

$$\begin{aligned} \mathsf{S}_\mathsf{P} = \mathsf{S}_{(\mathsf{QR})} &= \{\mathbf{ACAC}, \mathbf{ACTGAC}, \mathbf{ACACTG}, \mathbf{ACTGTGAC}, \mathbf{ACTGTGACTG}, \ldots\} \\ &= \left\{ \begin{array}{l} (\mathbf{AC})_2, (\mathbf{AC})(\mathbf{ACTG}), (\mathbf{ACTG})(\mathbf{AC}), (\mathbf{AC})(\mathbf{AC}(\mathbf{TG})_2), \\ (\mathbf{AC}(\mathbf{TG})_2)(\mathbf{AC}), \ldots, (\mathbf{AC})_3, (\mathbf{AC})_2(\mathbf{ACTG}), (\mathbf{ACTG})(\mathbf{AC})_2, \ldots \end{array} \right\} \end{aligned}.$$

4) If Q and R are regular patterns and denote the regular sets of pattern instances I_Q and I_R, respectively, and the regular set of tandem repeats S_Q and S_R, then the ***union*** of regular patterns, $\mathsf{P} = (\mathsf{Q} \cup \mathsf{R})$, corresponds to the regular set of pattern instances

$$\mathsf{I}_\mathsf{P} = \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{x \in \Sigma^* : x \in \mathsf{I}_\mathsf{Q}, x \in \mathsf{I}_\mathsf{R}\}.$$

That is, the collection of pattern instances from both I_Q and I_R. The regular sets of tandem repeats for the union operation corresponds to

$$\mathsf{S}_\mathsf{P} = \mathsf{S}_{(\mathsf{Q} \cup \mathsf{R})} = \left\{ \begin{array}{r} x \in \Sigma : x = w_1 w_2 \ldots w_j \text{ for } w_1, \ldots, w_j \in \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})}, j \geq 2 \\ \text{and } u_m, u_n \neq \varepsilon, 1 \leq m, n \leq j, m \neq n \end{array} \right\}.$$

That is, $\mathsf{S_P}$ is the collection of all words made up of two or more non-empty pattern instances in $\mathsf{I_P}$.

Example 4.1: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = \mathbf{TG}$ corresponds to a regular pattern of $\mathsf{P} = (\mathsf{Q} \cup \mathsf{R}) = \mathbf{AC} \mid \mathbf{TG}$ where the symbol "|" denotes the "or" operator that designates alternative patterns. P corresponds to a regular set of pattern instances $\mathsf{I_P} = \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{\mathbf{AC,TG}\}$ and the regular set of tandem repeats

$$\begin{aligned}\mathsf{S_P} = \mathsf{S}_{(\mathsf{Q}\cup\mathsf{R})} &= \{\mathbf{ACAC,ACTG,TGAC,TGTG,ACACAC,ACACTG,}\ldots\} \\ &= \left\{\begin{matrix}(\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG}), (\mathbf{TG})(\mathbf{AC}), (\mathbf{TG})_2, \\ (\mathbf{AC})_3, (\mathbf{AC})_2(\mathbf{TG}), (\mathbf{AC})(\mathbf{TG})_2, \ldots\end{matrix}\right\}\end{aligned}.$$

Example 4.2: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = (\mathbf{TG})_2$ corresponds to a regular pattern of

$\mathsf{P} = (\mathsf{Q} \cup \mathsf{R}) = \mathbf{AC} \,|\, (\mathbf{TG})_2$, a regular set of pattern instances

$\mathsf{I_P} = \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{\mathbf{AC,TGTG}\}$ and the regular set of tandem repeats

$$\begin{aligned}\mathsf{S_P} = \mathsf{S}_{(\mathsf{Q}\cup\mathsf{R})} &= \left\{\begin{matrix}\mathbf{ACAC,ACTGTG,TGTGAC,TGTGTGTG,ACACAC,} \\ \mathbf{ACACTGTG,ACTGTGAC,ACTGTGTGTG,}\ldots\end{matrix}\right\} \\ &= \left\{\begin{matrix}(\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG})_2, (\mathbf{TG})_2(\mathbf{AC}), (\mathbf{TG})_4, \\ (\mathbf{AC})_3, (\mathbf{AC})_2(\mathbf{TG})_2, (\mathbf{AC})(\mathbf{TG})_2(\mathbf{AC}), (\mathbf{AC})(\mathbf{TG})_4, \ldots\end{matrix}\right\}\end{aligned}.$$

Example 4.3: $\mathsf{Q} = \mathbf{AC}$ and $\mathsf{R} = (\mathbf{TG})_*$ corresponds to a regular pattern of

$\mathsf{P} = (\mathsf{Q} \cup \mathsf{R}) = \mathbf{AC} \,|\, (\mathbf{TG})_*$, a regular set of pattern instances

$\mathsf{I_P} = \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{\varepsilon, \mathbf{AC,TG,TGTG,TGTGTG,}\ldots\}$ and the regular set of tandem repeats

$$\begin{aligned} \mathsf{S}_{\mathsf{P}} = \mathsf{S}_{(\mathsf{Q} \cup \mathsf{R})} &= \{\mathbf{ACAC,ACTG,TGAC,TGTG,ACACAC,ACACTG,}\ldots\} \\ &= \left\{ \begin{matrix} (\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG}), (\mathbf{TG})(\mathbf{AC}), (\mathbf{TG})_2, \ldots, \\ (\mathbf{AC})_3, (\mathbf{AC})_2(\mathbf{TG}), (\mathbf{TG})(\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG})_2, (\mathbf{TG})_2(\mathbf{AC}), \ldots \end{matrix} \right\}. \end{aligned}$$

Example 4.4: $\mathsf{Q} = (\mathbf{AC})_*$ and $\mathsf{R} = (\mathbf{TG})_*$ corresponds to a regular pattern of

$\mathsf{P} = (\mathsf{Q} \cup \mathsf{R}) = (\mathbf{AC})_* \,|\, (\mathbf{TG})_*$, a regular set of pattern instances

$\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{\varepsilon, \mathbf{AC,ACAC,ACACAC,}\ldots\mathbf{,TG,TGTG,TGTGTG,}\ldots\}$ and the

regular set of tandem repeats

$$\begin{aligned} \mathsf{S}_{\mathsf{P}} = \mathsf{S}_{(\mathsf{Q} \cup \mathsf{R})} &= \{\mathbf{ACAC,ACTG,TGAC,TGTG,ACACAC,ACACTG,}\ldots\} \\ &= \left\{ \begin{matrix} (\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG}), (\mathbf{TG})(\mathbf{AC}), (\mathbf{TG})_2, \ldots, \\ (\mathbf{AC})_3, (\mathbf{AC})_2(\mathbf{TG}), (\mathbf{TG})(\mathbf{AC})_2, (\mathbf{AC})(\mathbf{TG})_2, (\mathbf{TG})_2(\mathbf{AC}), \ldots \end{matrix} \right\}. \end{aligned}$$

Given that Q and R are regular patterns and denote the regular sets of pattern instances I_{Q} and I_{R}, respectively, and the regular set of tandem repeats S_{Q} and S_{R}, then the union, concatenation and closure operations hold as follows.

1. **Concatenation**

If $\mathsf{P} = (\mathsf{QR})$ then $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{(\mathsf{QR})} = \mathsf{I}_{\mathsf{Q}}\,\mathsf{I}_{\mathsf{R}}$ is guaranteed. By definition, each element in I_{Q} is concatenated to each element of I_{R} and the resulting elements are all represented in I_{P} and are the only elements of I_{P}.

If $\mathsf{P} = (\mathsf{QR})$ and $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{(\mathsf{QR})} = \mathsf{I}_{\mathsf{Q}}\,\mathsf{I}_{\mathsf{R}}$ then $\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{(\mathsf{QR})} = \mathsf{S}_{\mathsf{Q}}\mathsf{S}_{\mathsf{R}}$ is not guaranteed to be valid. By definition, construction of $\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{(\mathsf{QR})}$ only uses elements in $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{(\mathsf{QR})} = \mathsf{I}_{\mathsf{Q}}\,\mathsf{I}_{\mathsf{R}}$ composed of an element from I_{Q} and an element from I_{R}

but, $S_Q S_R$ concatenates elements in S_Q with elements in S_R where an element in S_Q is required to have at least two elements from I_Q. Thus, the $S_Q S_R$ set of sequences will not alternate between elements in $I_P = I_{(QR)} = I_Q I_R$.

Example: If $Q = \mathbf{AT}$ and $R = \mathbf{TG}$ then $I_Q = \{\mathbf{AT}\}$, $I_R = \{\mathbf{TG}\}$,

$S_Q = \{(\mathbf{AT})_2, (\mathbf{AT})_3, \ldots\}$ and $S_R = \{(\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$.

This implies that $I_Q I_R = \{\mathbf{ATTG}\}$ and

$S_Q S_R = \{(\mathbf{AT})_2(\mathbf{TG})_2, (\mathbf{AT})_2(\mathbf{TG})_3, \ldots, (\mathbf{AT})_3(\mathbf{TG})_2, (\mathbf{AT})_3(\mathbf{TG})_3, \ldots\}$.

Now if $P = (QR) = \mathbf{ATTG}$ then $I_{(QR)} = \{\mathbf{ATTG}\}$ and

$S_{(QR)} = \{(\mathbf{ATTG})_2, (\mathbf{ATTG})_3, \ldots\}$.

Thus, $I_{(QR)} = I_Q I_R$ and $S_{(QR)} \neq S_Q S_R$.

2. **Union**

If $P = (Q \cup R)$ then $I_P = I_{(Q \cup R)} = I_Q \cup I_R$ is guaranteed. By definition, all elements in I_Q and I_R are members of I_P and only elements from I_Q and I_R are members of I_P.

If $P = (Q \cup R)$ and $I_P = I_{(Q \cup R)} = I_Q \cup I_R$ then $S_P = S_{(Q \cup R)} = S_Q \cup S_R$ is not guaranteed to be valid. By definition, construction of $S_P = S_{(Q \cup R)}$ can use any element in I_P which means that elements from I_Q and I_R can be combined to form elements in $S_{(Q \cup R)}$ but, the sequences in $S_Q \cup S_R$ do not allow combinations between elements of I_Q and I_R.

Example: If $\mathsf{Q} = (\mathbf{AT})_*$ and $\mathsf{R} = (\mathbf{TG})_*$ then

$\mathsf{I}_{\mathsf{Q}} = \{\varepsilon, \mathbf{AT}, (\mathbf{AT})_2, (\mathbf{AT})_3, \ldots\}$, $\mathsf{I}_{\mathsf{R}} = \{\varepsilon, \mathbf{TG}, (\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$,

$\mathsf{S}_{\mathsf{Q}} = \{(\mathbf{AT})_2, (\mathbf{AT})_3, \ldots\}$ and $\mathsf{S}_{\mathsf{R}} = \{(\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$.

This implies that $\mathsf{I}_{\mathsf{Q}} \cup \mathsf{I}_{\mathsf{R}} = \{\varepsilon, \mathbf{AT}, (\mathbf{AT})_2, (\mathbf{AT})_3, \ldots, \mathbf{TG}, (\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$ and

$\mathsf{S}_{\mathsf{Q}} \cup \mathsf{S}_{\mathsf{R}} = \{(\mathbf{AT})_2, (\mathbf{AT})_3, \ldots, (\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$.

Now if $\mathsf{P} = (\mathsf{Q} \cup \mathsf{R}) = (\mathbf{AT})_* \mid (\mathbf{TG})_*$ then

$\mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \{\varepsilon, \mathbf{AT}, (\mathbf{AT})_2, (\mathbf{AT})_3, \ldots, \mathbf{TG}, (\mathbf{TG})_2, (\mathbf{TG})_3, \ldots\}$ and

$\mathsf{S}_{(\mathsf{Q} \cup \mathsf{R})} = \{(\mathbf{AT})_2, (\mathbf{AT})(\mathbf{TG}), (\mathbf{TG})(\mathbf{AT}), (\mathbf{TG})_2, (\mathbf{AT})_3, (\mathbf{AT})_2(\mathbf{TG}), \ldots\}$.

Thus, $\mathsf{I}_{(\mathsf{Q} \cup \mathsf{R})} = \mathsf{I}_{\mathsf{Q}} \cup \mathsf{I}_{\mathsf{R}}$ and $\mathsf{S}_{(\mathsf{Q} \cup \mathsf{R})} \neq \mathsf{S}_{\mathsf{Q}} \cup \mathsf{S}_{\mathsf{R}}$.

3. Closure

Fixed repetition, variable repetition and open closure have the same operation issues involving I_{P} and S_{P}. Here, only fixed repetition is discussed. The result holds for variable repetition and open closure.

- **Fixed repetition**

If $\mathsf{P} = (\mathsf{R})_{\boldsymbol{k}}$ then $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})} = (\mathsf{I}_{\mathsf{R}})^{\boldsymbol{k}}$ is ambiguous. It is not clear whether each element in I_{P} corresponds to the concatenation of *k* copies of each element in I_{R} or to the concatenation of ***k*** (non-unique) elements in I_{R}. This model utilizes the later definition using ***k***, non-unique elements in I_{R} to form each possible element in I_{P}.

If $\mathsf{P} = (\mathsf{R})_{\boldsymbol{k}}$ and $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})} = (\mathsf{I}_{\mathsf{R}})^{\boldsymbol{k}}$ as described above then,

$\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})} = (\mathsf{S}_{\mathsf{R}})^{\boldsymbol{k}}$ is not guaranteed to be valid. By definition for $(\mathsf{S}_{\mathsf{R}})^{\boldsymbol{k}}$, each element in S_{R} is transformed into a new element in $(\mathsf{S}_{\mathsf{R}})^{\boldsymbol{k}}$ having *k* consecutive occurrences of the element from S_{R}. On the other hand, elements in $\mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})}$ represent ***k*** non-unique elements in $\mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})}$.

Example: if $\mathsf{R} = \mathbf{AT}(\mathbf{TG})_{*}$ then $\mathsf{I}_{\mathsf{R}} = \{\varepsilon, \mathbf{AT}, \mathbf{ATTG}, \mathbf{AT}(\mathbf{TG})_{2}, \mathbf{AT}(\mathbf{TG})_{3}, \ldots\}$ and

$\mathsf{S}_{\mathsf{R}} = \{(\mathbf{AT})_{2}, (\mathbf{AT})_{2}\mathbf{TG}, (\mathbf{AT})_{2}(\mathbf{TG})_{2}, \ldots, (\mathbf{ATTG})(\mathbf{AT}), (\mathbf{AT})(\mathbf{TG})_{2}(\mathbf{AT}), \ldots\}$.

This implies that $(\mathsf{I}_{\mathsf{R}})^{\boldsymbol{k}} = \{(\mathbf{AT})_{\boldsymbol{k}} (\mathbf{ATTG})_{\boldsymbol{k}}, (\mathbf{AT}(\mathbf{TG})_{2})_{\boldsymbol{k}}, \ldots\}$ and

$$(\mathsf{S}_{\mathsf{R}})^{\boldsymbol{k}} = \begin{Bmatrix} (\mathbf{AT})_{2\boldsymbol{k}}, ((\mathbf{AT})_{2}\mathbf{TG})_{\boldsymbol{k}}, ((\mathbf{AT})_{2}(\mathbf{TG})_{2})_{\boldsymbol{k}}, \ldots, \\ ((\mathbf{ATTG})(\mathbf{AT}))_{\boldsymbol{k}}, ((\mathbf{AT})(\mathbf{TG})_{2}(\mathbf{AT}))_{\boldsymbol{k}}, \ldots \end{Bmatrix}.$$

Now if $\mathsf{P} = \mathsf{R}_{\boldsymbol{k}} = (\mathbf{AT}(\mathbf{TG})_{*})_{\boldsymbol{k}}$ then $\mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})} = \{(\mathbf{AT})_{\boldsymbol{k}}, (\mathbf{ATTG})_{\boldsymbol{k}}, (\mathbf{AT}(\mathbf{TG})_{2})_{\boldsymbol{k}}, \ldots\}$

and $\mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})} = \{(\mathbf{AT})_{2\boldsymbol{k}}, (\mathbf{AT})_{\boldsymbol{k}} (\mathbf{ATTG})_{\boldsymbol{k}}, (\mathbf{AT})_{\boldsymbol{k}} (\mathbf{AT}(\mathbf{TG})_{2})_{\boldsymbol{k}}, \ldots\}$.

Thus, $\mathsf{I}_{\mathsf{P}} = \mathsf{I}_{((\mathsf{R})_{\boldsymbol{k}})} = (\mathsf{I}_{\mathsf{R}})^{\boldsymbol{k}}$ and $\mathsf{S}_{\mathsf{P}} = \mathsf{S}_{((\mathsf{R})_{\boldsymbol{k}})} \neq (\mathsf{S}_{\mathsf{R}})^{\boldsymbol{k}}$.

In this dissertation, the set of all tandem repeats is split into two primary subsets; tandem repeats having a simple pattern structure and tandem repeats having a complex pattern structure. Here, several definitions are given that are utilized throughout the remainder of the dissertation.

Definition: A ***simple pattern structure*** is a regular pattern that is a sequence of nucleotides: $\mathsf{P} = x_{1}x_{2}\ldots x_{r}$ for a positive integer ***r*** and $\boldsymbol{x_i}$ in Σ. A tandem repeat region, $\boldsymbol{T}$, having a simple pattern structure is denoted $\boldsymbol{T} = \mathsf{P}_{\boldsymbol{c}}$ where the number of recurrences of P in the region is denoted by the copy number, a positive integer ***c***.

Definition: A ***simple nested tandem repeat*** has a regular pattern that is a complex pattern structure formed by the concatenation of two regular patterns: $\mathsf{P} = \mathsf{P}_1(\mathsf{P}_2)_{\boldsymbol{c}}$ where P_1 and P_2 are regular patterns and the copy number, ***c***, is a positive integer.

Definition: A ***variable length tandem repeat*** **(VLTR)** has a regular pattern that is a complex pattern structure formed by the concatenation of two regular patterns: $\mathsf{P} = \mathsf{P}_1(\mathsf{P}_2)_*$ where P_1 and P_2 are regular patterns and P_2 has a variable copy number.

Example. **`CATTAGCC TGTGTGTG CATTAGCC TGTG CATTAGCC TGTGTGTGTG`**

This VLTR sequence is composed of two repeated patterns, **`CATTAGCC`** and **`TG`** and can be written as $\mathtt{CATTAGCC(TG)_4CATTAGCC(TG)_2CATTAGCC(TG)_5}$, or as $\mathtt{((CATTAGCC)_1(TG)_*)_3}$. The later one clearly indicates that the copy number for the **`CATTAGCC`** pattern is fixed and equal to one, while the copy number for the **`TG`** pattern is variable, denoted with a $*$.

Definition: Two patterns with a simple pattern structure are said to be ***i****-similar* if one can be transformed to the other by at most ***i*** nucleotide substitutions, insertions or deletions. This definition actively extends to ***m*** patterns. A ***multi-periodic tandem repeat (MPTR)*** has a regular pattern that is a complex pattern structure formed by the concatenation of two or more ***i****-similar* regular patterns: $\mathsf{P} = \mathsf{P}_1\mathsf{P}_2\ldots\mathsf{P}_{\boldsymbol{m}}$ where the $\mathsf{P}_{\boldsymbol{j}}$, $1 \le \boldsymbol{j} \le \boldsymbol{m}$ are ***i****-similar*.

Example. **CAGTA CAGCA CAATA CAGCA CAGTA CAGCA CAATA CAGCA**

This sequence is almost a perfect tandem repeat of the form $(\mathbf{CAGTA})_8$. Rather, this MPTR is a simple nested concatenation of three 1-similar patterns, **CAGTA**, **CAGCA** and **CAATA**, arranged such that the **T** becomes a **C** in every other copy and the **G** becomes an **A** in every fourth copy. This multiple periodicity is written as follows; a first multiple (1X) of $(\mathbf{CA}^{\mathbf{G}}/_{\mathbf{A}}{}^{\mathbf{T}}/_{\mathbf{C}}\mathbf{A})_8$, a second multiple (2X) of $(\mathbf{CA}^{\mathbf{G}}/_{\mathbf{A}}\mathbf{TA}\ \mathbf{CA}^{\mathbf{G}}/_{\mathbf{A}}\mathbf{CA})_4$ and a fourth multiple (4X) of $(\mathbf{CAGTA}\ \mathbf{CAGCA}\ \mathbf{CAATA}\ \mathbf{CAGCA})_2$ where ${}^{X}/_{Y}$ is a position represented by both X and Y.

The focus of this dissertation is to develop identification algorithms for two subsets of tandem repeats having complex pattern structures. The first is a subset of VLTR where the region pattern is $\mathsf{P} = \mathsf{P}_1(\mathsf{P}_2)_*$ such that P_1 and P_2 have a simple pattern structure and P_2 has a short pattern, i.e. $(\mathsf{P}_2)_*$ represents simple sequence repeats (SSRs). The second is a subset of MPTR where the region pattern is $\mathsf{P} = \mathsf{P}_1\mathsf{P}_2\ldots\mathsf{P}_{\boldsymbol{m}}$ and $\mathsf{P}_{\boldsymbol{j}}$, $1 \le \boldsymbol{j} \le \boldsymbol{m}$ has a simple pattern structure.

3.3. Collection of GenBank sequences

A collection of GenBank sequences represents tandem repeat regions having simple and complex pattern structures. The regions include regions having long patterns and short patterns, i.e. simple sequence repeat (SSR), variable length tandem repeat (VLTR) regions and multi-periodic tandem repeat (MPTR) regions. Sequence lengths range from short sequences containing a single isolated tandem repeat region to medium sequences several kilobasepairs (kbp) long containing several tandem repeats to a long sequence several 100 kbp long representing a complete chromosome and containing many tandem repeat regions. Furthermore, the sequences span a range of species from bacteria and yeast to humans.

The purpose of the sequence collection is to provide a benchmark for validating the identification algorithm. These sequences provide the framework for common problems encountered in tandem repeat regions in DNA sequences and enable the development of techniques for addressing these common problems. Thus, the sequences are an intricate part of the process of developing a tandem repeat identification algorithm. Table 3.3.A, Table 3.3.B and Table 3.3.C describe the sequences, the tandem repeat regions within the sequences and the purpose for including the sequence in the collection.

Sequence (GenBank Locus)	Species (Common Name)	Sequence Length (bp)	Simple Pattern Structures			Complex Pattern Structures	
			General	SSR	Long	VLTR	MPTR
AF071126	Corn	723			•	•	
AF091342	Cow	2944		•			•
AMU73928	Honey Bee	283		•		•	
BOVTGN	Cow	725		•		•	
BTA132392	Cow	251	•	•			
BTU75906	Cow	364		•		•	
DMPUGDMG1	Fruit Fly	2468	•	•			
DMU41809	Fruit Fly	1424	•	•			
ECTRNYSU	Bacteria	1655	•		•	•	
HSVDJSAT	Human	1985					•
MM102B5	Mouse	704					•
MMMSAT5	Mouse	412		•			
NC_001133	Yeast	230203	•	•	•		•
SMOSATC	Salmon	662		•		•	
U00144	Cow	407		•			

Table 3.3.A: Collection of GenBank sequences. A bullet ("•") indicates a region type present in the sequence determined via sequence annotations and a visual analysis of the sequences.

GenBank Locus	Location	Sequence Annotation	Visual Observation
AF071126	1..723	Dimer of two TR-1 elements	1) Two copies each about 360 bp long
AF091342	various	KSPXX amino acid sequence repeat corresponds to a repeat having a 15 bp pattern.	1) At least 13 separate SSR regions which have an **AAG**, **AGG** or **AAGAGG** pattern 2) MPTR with 15 bp pattern
AMU73928	76..209	Minisatellite repeat	1) VLTR with 17 bp a non-SSR pattern and a SSR having a **T**-pattern
BOVTGN	311..703	BTGL1 minisatellite repeat having seven copies 46 to 82 bp long.	1) VLTR with a 23-28 bp non-SSR and a SSR having a **GT**-pattern
BTA132392	69..242	7 tandem occurrences of a prion protein gene having a 24 to 27 bp pattern	1) TR with 24 bp pattern

GenBank Locus	Location	Sequence Annotation	Visual Observation
BTU75906	1..364	BTGL1 minisatellite repeat having five copies 48 to 79 bp long.	1) VLTR with a 23-28 bp non-SSR and a SSR having a **GT**-pattern
DMPUGDMG1	2205..>2468	Eye pigmentation enzyme gene containing a tandem repeat having a **TCTCTCT** pattern	1) Large region having 25 exact copies of a **TCTCTCT** pattern, many similar copies. 2) The large region is flanked by two SSRs having a **CT** pattern.
DMU41809	1211..1259	Two direct overlapping 27 bp copies	1) Two 22 bp copies. 2) Several SSRs
ECTRNYSU	625..1158	*tyrT* operon composed of the gene for $\text{tRNA}_1^{\text{Tyr}}$ followed by three copies of a 178 bp pattern	1) Tandem repeat region having three copies of a 178 bp pattern 2) Nested SSRs having a **ACC** pattern

GenBank Locus	Location	Sequence Annotation	Visual Observation
HSVDJSAT	1200..1543	Minisatellite region having 36 copies of closely related 9 and 10 bp patterns that form an 11 copy pattern	1) Challenging MPTR having three patterns **CTGGGAGAGG**, **CTGGGAGAG** and **CTGGGATTG**
MM102B5	1..696	Gamma satellite repeat having a 234 bp pattern	1) MPTR with a 58 bp base pattern.
MMMSAT5	23..213	Microsatellite region having a mix of SSRs having **AC**, **AT** and **GT** patterns	1) SSR cluster having **AC**, **AT** and **GT** patterns
NC_001133	24,001..27,969 203,389..208,002	FLO1 and FLO9 genes containing an embedded repeat having a 135 bp pattern	1) Two MPTR regions composed of 13 or 18 copies of a 135 bp pattern
SMOSATC	1..662	Minisatellite region having 18 copies of pattern about 40 bp long	1) VLTR with 26 bp non-SSR pattern and a SSR having a **CAGT** pattern

GenBank Locus	Location	Sequence Annotation	Visual Observation
U00144	292..407	SSR cluster having **AG**, **GT** and **ACAG** patterns.	1) Several SSRs having **AT**, **GT**, **ACAG**, **AGGG** and **CCGGG** patterns

Table 3.3.B: Summary of tandem repeat region content for individual sequences in the GenBank collection. The tandem repeat regions of each sequence in Table 3.3.A is given. Regions are specified by sequence position, type of pattern structure and either pattern sequence or pattern length.

GenBank Locus: AF071126. This sequence represents two copies of a pattern about 360 bp long. It was selected for two features: (1) the exceeding length of the pattern and (2) the presence of only two copies, the minimal required for a true tandem repetition. In addition, during analysis of the sequence, it was determined that the pattern contains a nested tandem repeat having an **ATTTT** pattern that exhibits a variable copy number such that one SSR region is 13 bp long and the other copy is 18 bp long. The non-SSR pattern is 346 bp long. Thus, this sequence is also a VLTR.

GenBank Locus: AF091342. This sequence contains numerous tandem repeat regions. It was selected both for a collection of SSRs having similar patterns and for a borderline MPTR region composed of several related, yet distinct patterns. The SSR collection includes at least thirteen separate regions that have either an **AAG**, **AGG** or **AAGAGG** pattern. The challenge is twofold: to distinguish between closely related patterns and to locate each separate region, including those with an identical pattern. The borderline MPTR region is predominately composed of several related patterns: **AAGTCCCCAACAGCA**, **AAGTCCCCAGAGGCA**, **AAGTCCCCAACGGCC**, **AAGTCCCCAGTGGCC** and **AAGTCCCCGACGGCC**. Several times, a pattern recurs every six copies and similarities between the patterns indicate a possible recurrence every four copies. Neither of these periodicities is strong. Thus, this region provides a way to measure whether a weak periodicity is acquired and what type of analysis occurs for borderline situations.

GenBank Locus: AMU73928. This sequence contains a well-conserved VLTR having a pattern composed of a variable-copy **T**-pattern, nested SSR coupled with a 17 bp non-SSR pattern. This region is a good example of a VLTR containing a SSR that is a simple nucleotide run. The variable SSR is only slightly variable such that the region could be located using a simple pattern structure and inserting gaps to compensate for the variability.

GenBank Locus: BOVTGN. This sequence contains a well-conserved VLTR region having a pattern composed of a variable-copy, **GT**-pattern, nested SSR coupled with a 23-28 bp non-SSR pattern. This region is a good example of a VLTR in that there is a well-defined, nested SSR that is readily identifiable in each copy and the nested SSR has a highly variable copy number. Thus, the only way to identify the complete region is to be able to identify VLTRs.

In addition, the region contains two other features. First, one copy contains a run of seven **G** nucleotides at the junction between the non-SSR and the SSR and four copies contain a run of three **G** nucleotides in the same position. Second, the sequence near or within the SSR contains a consistent gap such that it is difficult to determine whether a **TGG** before and a **TG** after each SSR should be include in the SSR or be defined as part of the non-SSR pattern. Both the G nucleotide runs and the gap-difference based sequence near the SSR makes defining the non-SSR sequence difficult.

GenBank Locus: BTA132392. This sequence contains a tandem repeat region containing about seven copies of a 24 to 27 bp pattern. The region contains hints of both a nested tandem repeat and higher order periodicity across multiple copies. It was chosen as a case bordering between a simple and complex pattern structure.

GenBank Locus: BTU75906. This sequence contains a well-conserved VLTR region having a pattern composed of a variable-copy, **GT**-pattern, nested SSR coupled with a 23-28 bp non-SSR pattern. This region is a good example of a VLTR in that there is a well-defined, nested SSR. It was chosen for irregularities present in the SSRs that may cause a single SSR to be reported as two consecutive SSRs having identical patterns. Such an irregularity can make VLTR analysis more challenging.

GenBank Locus: DMPUGDMG1. This sequence contains a region with a simple **TCTCTCT** pattern. The pattern is perfectly conserved for 25 consecutive copies with additional well conserved, followed by poorly conserved copies at the right edge. It was chosen to distinguish between a **TCTCTCT** pattern, a **CT** pattern and a VLTR region containing a nested SSR having a **CT** pattern.

GenBank Locus: DMU41809. This sequence contains several tandem repeat regions. First, two consecutive copies of a 22 bp pattern occur. Second, the sequence ends with an **A** nucleotide run. Third, two perfect SSRs having an **AT**-pattern are flanked by similar sequence that almost forms two complete copies of a 15 bp pattern. This sequence was chosen for identification of regions with simple pattern structures.

GenBank Locus: ECTRNYSU. This sequence contains a tandem repeat region composed of three copies of a well-conserved 178 bp pattern. It was chosen for its long pattern. In addition, during algorithmic analysis, it was determined that the sequence contains a nested TR, a SSR having an **ACC**-pattern.

GenBank Locus: HSVDJSAT. This sequence contains a difficult and challenging MPTR region composed of three basic patterns: **CTGGGAGAGG**, **CTGGGAGAG** and **CTGGGATTG**. One pattern is 10 bp long while the other two patterns are 9 bp long. On a simple level, the 10-pattern pattern alternates with a 9 bp pattern such that every other copy is 9 bp in length. The 9 bp patterns alternate such that a specific 9 bp pattern occurs once every fourth copy. Occasionally, the 10 bp pattern hiccups and occurs twice. The pattern alterations and the consecutive 10 bp pattern hiccups create several higher order patterns including one across eleven copies.

This region was chosen for its many challenges. First, the base pattern is not 9 or 10 bp. It is both 9 and 10 bp in length. Second, the higher order periodicity is not exact and pretty but rather complex and difficult to analyze. Yet, at its root, the region is simply a complex intertwining of three base patterns; a sequence that is simple, yet exceeding complex.

GenBank Locus: MM102B5. This sequence represents a portion of a gamma satellite region having a 234 bp pattern. Visual analysis indicates a MPTR region having a base pattern of 58 bp. Additional periodicities combine two copies into a 116 bp pattern and four copies into a 231 bp pattern. The 231 bp pattern is well conserved and approximates the standard 234 bp gamma satellite pattern. It was chosen as a straightforward, well-conserved MPTR region.

GenBank Locus: MMMSAT5. This sequence contains a SSR cluster having **AC**, **AT** and **GT** patterns. Almost every position in the region can be accounted for based on these three patterns. It was chosen as a region composed of pattern mixtures; multiple occurrences of the same pattern in the same region not organized into a higher order periodicity.

GenBank Locus: NC_001133 – Yeast (*Saccharomyces cerevisiae*) Chromosome I. The L28920 sequence represents the flocculation gene (FLO1) containing a large tandem repeat region having 18 copies of a 135 bp pattern. The literature suggests that the region is a MPTR region though direct visual evidence did not confirm or deny this fact. This sequence is no longer available as an independent sequence. Thus, the complete chromosome I sequence is analyzed. This sequence contains two flocculation genes (FLO1 and FLO9) both containing the larger tandem repeat.

Analysis uses the complete chromosome for several reasons. First, it contains two flocculation genes both having the tandem repeat region with the long 135 bp pattern. These tandem repeat regions were chosen for analysis due to their long pattern and their large copy number. Second, early analysis revealed shadows associated with each flocculation gene located 10 to 14 kbp away. This spawned an extension for identifying non-contiguous regions of similarity. Third, the complete chromosome does not represent an isolated tandem repeat region but rather represents a viable biological entity, a chromosome. Fourth, the complete chromosome enables testing of the algorithm for long sequences.

GenBank Locus: SMOSATC. This sequence represents a well-conserved VLTR region having a pattern composed of a variable copy, **CAGT**-pattern, coupled with a 26 bp non-SSR region. The SSRs have two to four copies that contain errors such that only some SSRs are identified. Thus, SSRs are "visible" in only some of the copies. The challenge is being able to characterize the region as a VLTR without requiring that each copy contain an occurrence of an SSR while ensuring that enough SSRs occur to justify the VLTR characterization.

This region could be identified using a simple pattern structure by inserting gaps in several locations to compensate for the variable copy number but this would not properly characterize the region. Thus, other algorithms will locate the region but not properly characterize the region.

GenBank Locus: U00144. This sequence a SSR cluster having **AT**, **GT**, **ACAG**, **AGGG** and **CCGGG** patterns. It was chosen for identification of several SSRs having the same pattern length.

Table 3.3.C: Purpose for inclusion of individual sequences in the GenBank collection. The basic description of each sequence in Table 3.3.A is given together with an explanation of the purpose for including the sequence in the collection.

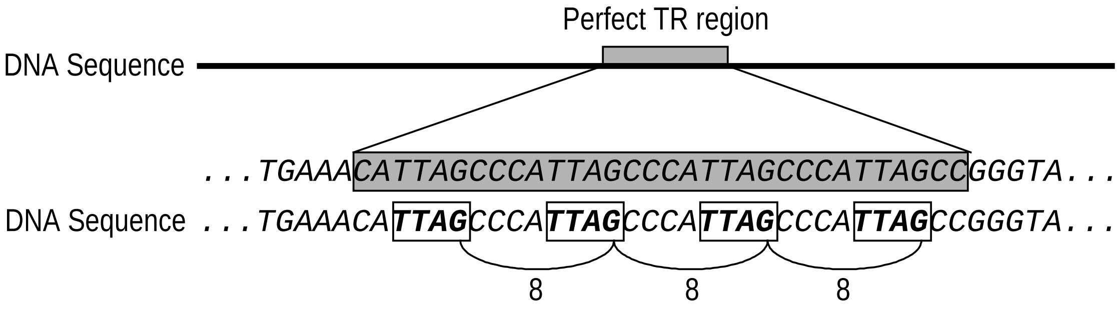
4. Algorithms and data structures

Tandem repeat identification involves locating regions containing tandem repetition and determining the recurring pattern that characterizes the tandem repetition. In practice, two complications hinder identification: imperfect pattern conservation and complex pattern structures. The broad set of issues that result from these complications cannot be addressed by a single technique. Thus, this algorithm is a collection of techniques, each designed to address fundamental biological issues observed in tandem repeat regions in DNA sequences.

This section describes the algorithm at four levels. Section 4.1 is an overview that presents a high level, informal description of the behavior of the algorithm using a series of examples. Section 4.2 describes the fundamental biological issues associated with tandem repeat identification. Section 4.3 explains the computational design decisions necessary to bring the fundamental biological complications to fruition. Finally, Section 4.4 gives a formal description of the algorithms and data structures that identify tandem repeat regions.

4.1. Overview

Suppose a DNA sequence contains a perfect tandem repeat region composed of several, consecutive copies of the same subsequence. Here is a region composed of four, consecutive

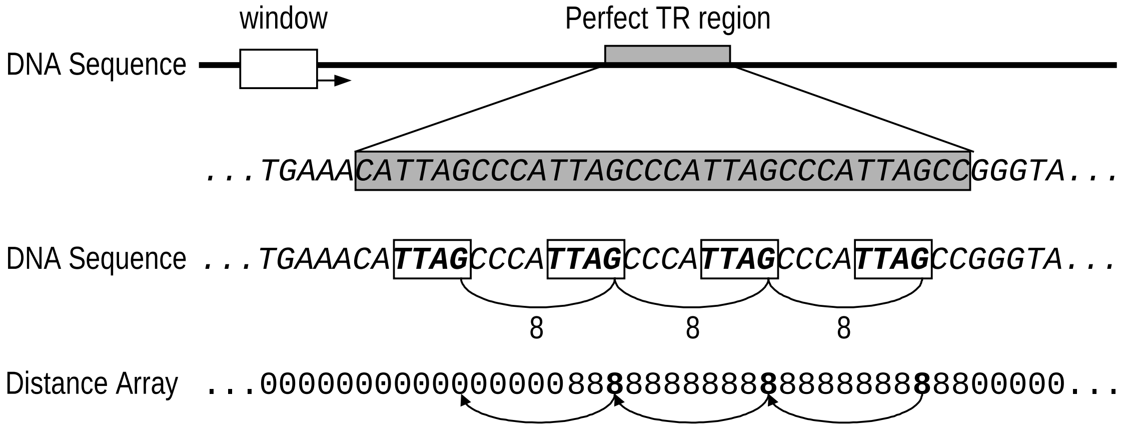

Perfect TR region
DNA Sequence
. . . TGAAACATTAGCCCATTAGCCCATTAGCCCATTAGCCGGGTA . . .
DNA Sequence . . . TGAAACATTAGCCCATTAGCCCATTAGCCCATTAGCCGGGTA . . .
8
8
8

occurrences of **CATTAGCC**. Suppose one looks in this region at all overlapping subsequences composed of $\boldsymbol{k}$ consecutive nucleotides, termed a ***k*-word** or simply **word**. Here, a series of overlapping words recurs every 8 positions. For words with $\boldsymbol{k} = 4$, the words are **CATT**, **ATTA**, **TTAG**, **TAGC**, **AGCC**, **GCCC**, **CCCA** and **CCAT**. Thus, identical words occur once every 8 positions, as shown in the example for the word **TTAG**. This recurrence of the same word every 8 positions reflects the pattern length or ***period*** associated with the subsequence **CATTAGCC**. Furthermore, the length of the word, $\boldsymbol{k}$, is independent of the length of the pattern, i.e. the period, thereby enabling identification of tandem repeat regions for any pattern length using any $\boldsymbol{k}$-length word. In practice, short words (e.g. $2 \le \boldsymbol{k} \le 5$) enable a greater sensitivity to imperfect pattern conservation and long words (e.g. $6 \le \boldsymbol{k} \ge 20$) enable a greater ability to span large periodicities. The word size tradeoff is discussed in Section 5.2.5.3.

The tandem repeat identification algorithm involves three primary steps. First, identify a potential tandem repeat region by finding regions of the DNA sequence that exhibit periodicity signatures characteristic of tandem repeats (see Section 4.3.2 and Section 4.4.1). Second, construct a pattern corresponding to the periodicity signature of the region (see Section 4.3.4 to Section 4.3.7 and Section 4.4.4 to Section 4.4.7). Third, characterize the region by aligning a pattern to the region (see Section 4.3.8 and Section 4.4.8).

Imperfect pattern conservation and complex pattern structures complicate the analysis of tandem repeat regions. Strong, well-conserved regions having simple, fixed length patterns generate strong periodicity signatures. Weak, poorly conserved regions and those having complex, variable length patterns generate weaker periodicity signatures. This algorithm performs an extensive period analysis to amplify weak periodicity signatures by sifting through noisy periodicity data and boosting the periodicity signature to sufficient levels for identification.

The algorithm performs an analysis of a DNA sequence by sliding a ***k***-length window across the sequence. During a pause at each position along the sequence, the ***k***-word in the window is processed. Once that word is known, the distance to the previous occurrence of the same word is calculated. This distance is recorded in a distance array at the same index as the current DNA sequence position. This creates a linear representation of the distances between identical words.



Here, a window of length $\boldsymbol{k} = 4$ slides across a perfect tandem repeat region in order to determine the words composing the region. In this example, the same words recur every 8 positions as depicted by distances between **TTAG** occurrences. The distance array records the distance from the word in the window to the rightmost previous occurrence of the same word.

The signature of a tandem repeat region in the distance array is two-fold. First, the distance between occurrences of a word reflects the periodicity of the tandem repeat region.

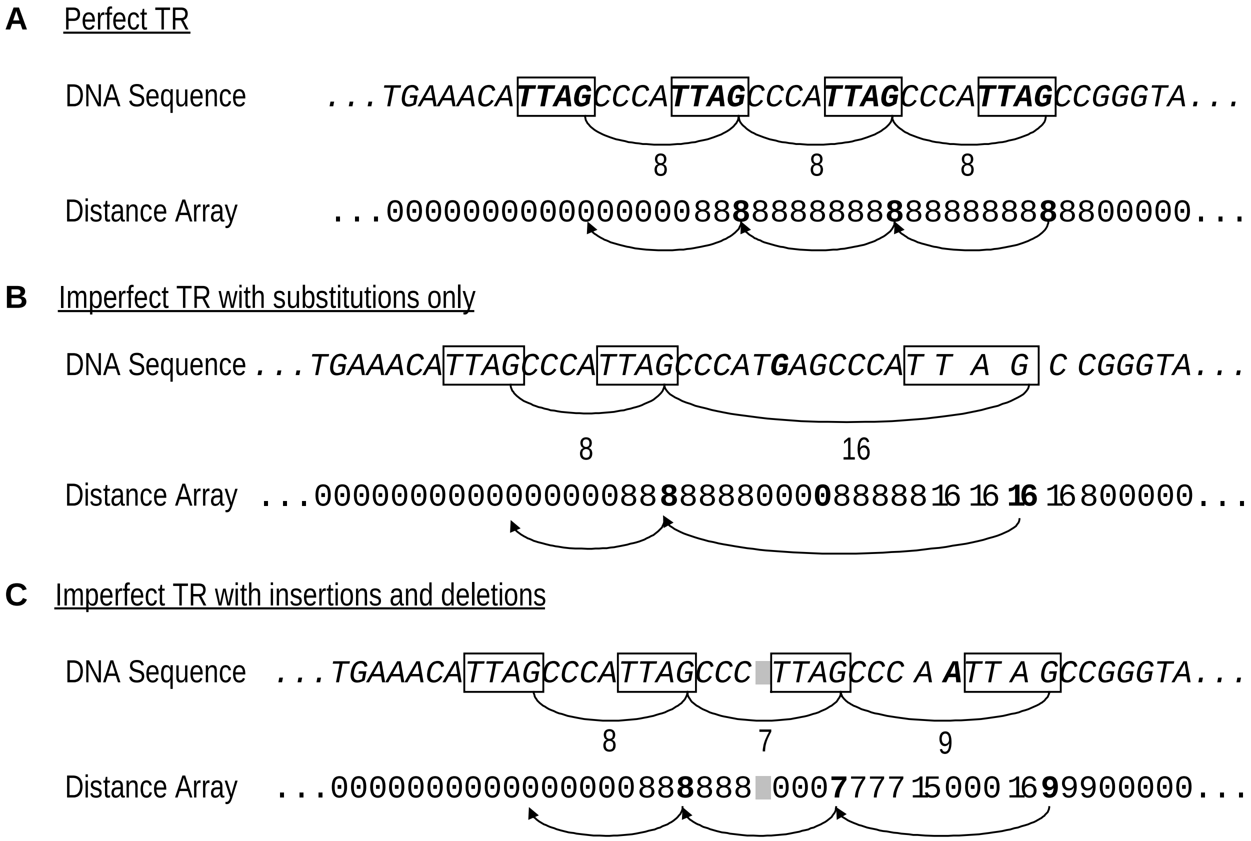
Here, the periodicity of 8 reflects the pattern of length 8. Second, tandem repeats generate ***runs*** of consecutive, identical distances in the distance array. Here, a run of consecutive 8s in the distance array, i.e. an 8-run, corresponds to the tandem repeat region.

Phase I: Use distance runs to locate potential tandem repeats. The algorithm uses analysis of the distance array as the primary method for locating potential tandem repeat regions. First, runs of identical distances act as a filter to remove noise associated with non-repetitive sequences in the overall sequence and with inexact conservation within a tandem repeat region. The positions associated with identical distance runs are placed in a histogram according to distance. A distance (or period) independent, threshold based approach selects the strongest periodicities. Once a periodicity is selected, the positions associated with the periodicity locate the tandem repeat region on the DNA sequence. In all, this process enables selection of fairly weak tandem repeat regions.

Second, all distances in the distance array proximal to the positions associated with the selected periodicity are analyzed extensively. This process groups related periodicities together forming a periodicity signature for the region that provides a more complete description of the region’s pattern structure and conservation level (see examples on next page). Additional analysis of the periodicity signature determines the complexity of the pattern structure such as whether the region’s signature corresponds to a variable length tandem repeat (VLTR) region or a multi-periodic tandem repeat (MPTR) region (see examples on next page).

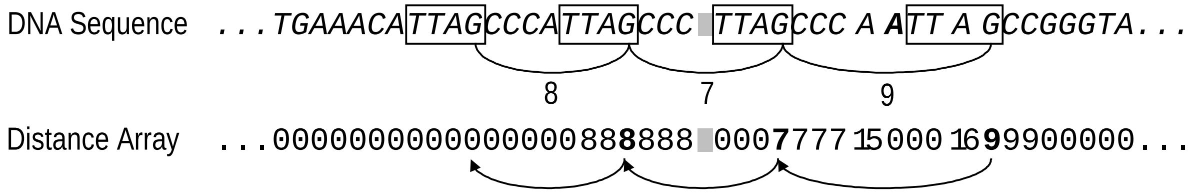
Finally, the analysis selects either a single periodicity or a set of periodicities to describe the basic periodicity of the region. The ***base period(s)*** capture the expected length of a single occurrence of the pattern. For well conserved regions, the base period often mimics a region’s periodicity signature. As shown in the examples that follow, imperfect pattern conservation and complex pattern structures create a periodicity signature that includes the basic periodicities and various additional periodicities.

Phase I examples: The impact of imperfect and complex patterns on the distance array. The impact of pattern imperfections is shown on the following page by a set of three regions. Here, a perfect tandem repeat region (see Panel **A**) is modified to introduce pattern imperfections containing substitutions (see Panel **B**) or containing insertions and deletions (see Panel **C**). The corresponding impact on the distance arrays is shown. Descriptions for panels **B** and **C** follow.


A Perfect TR
DNA Sequence ...TGAAACATTAGCCCATTAGCCCATTAGCCCATTAGCCGGGTA...
8 8 8
Distance Array ...00000000000000008888888888888888888800000...
B Imperfect TR with substitutions only
DNA Sequence ...TGAAACATTAGCCCATTAGCCCATGAGCCCAT T A G C CGGGTA...
8 16
Distance Array ...0000000000000000888888000888816 16 16 16800000...
C Imperfect TR with insertions and deletions
DNA Sequence ...TGAAACATTAGCCCATTAGCCC TTAGCCC A ATT A GCCGGGTA...
8 7 9
Distance Array ...0000000000000000888888 000777715000169900000...

The region in Panel **B** contains a single substitution of a **T** to a **G** (boldfaced above) causing the **TTAG** word to become **TGAG**. The substitution creates two changes in the distance array. First, the words that overlap the substitution no longer match words in the region. Second, distances calculated by referring to words that overlap the substitution are modified to reflect a distance across two copies, or 16 positions. Thus, two 8-runs and one 16-run occur in the distance array.

Imperfect pattern conservation caused solely by substitutions generates a cluster of proximal runs having the same distance or a multiple of the same distance, as well as, individual positions having the same or multiple of the distance. Multiples of a distance represent two occurrences of a pattern and are discarded leaving a single distance to represent the region's base period.



This region (same as Panel **C** on the previous page) contains an insertion of an **A** nucleotide (boldfaced above) and a deletion of an **A** nucleotide (denoted above by a gray box). This generates three changes that impact the distances between occurrences. First, the words that overlap the insertion or deletion no longer match words in the region. Second, the copy length changes; insertions increase the copy length as shown by a distance of 9 between the third and fourth **TTAG** occurrences and deletions decrease the copy length as shown by a distance of 7 between the second and third **TTAG** occurrences. Third, distances calculated by referring to words which overlap the insertion or deletion are modified to reflect a distance across two copies.

Imperfect pattern conservation due to nucleotide insertion or deletion generates a cluster of nearby runs which may or may not have the same distance but which are related since the runs result from distances between the *same* words. In particular, the distances differ by the length of the insertion or deletion or the sum of the length of consecutive copies. Here, the 7-run, the 8-run and the 9-run represent the lengths of a single pattern and are generated by distances between the same words. The 15 and 16 periodicities represent two consecutive copies but in this example, their quantity is insignificant. Thus, the region base periodicities are 7, 8 and 9.

The distance array reflects complex pattern structures that occur in tandem repeat regions. Three examples describe the impact of multi-periodic tandem repeat (MPTR), nested tandem repeats and variable length tandem repeat (VLTR) regions on the distance array. Analysis of the regions uses a window of length $\boldsymbol{k} = 4$. A description for each example follows.

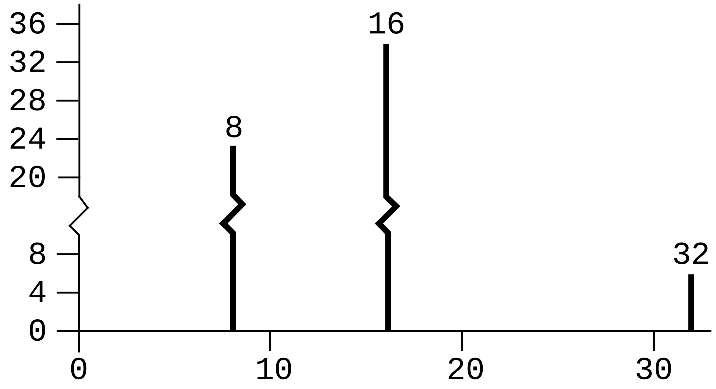
A DNA Sequence: Sequential Copies in Region

T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*A*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*A*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*

B Distance Array: Sequential Copies in Region

0	0	0	0	0	0	0	0
0	0	0	0	0	8	8	8
8	16	16	16	16	8	8	8
8	16	16	16	0	0	0	0
8	16	16	16	16	16	16	16
8	16	16	16	32	8	8	8
8	16	16	16	16	8	8	8
8	16	16	16	32	32	32	32
8	16	16	16	16	16	16	16
8	16	16	16	32	8	8	8

C Distance Histogram: All Distances in Region



This region is a multi-periodic tandem repeat (MPTR) composed of three distinct, yet related copies: **TTAGCCCA**, **TGAGCCCA** and **TGAGACCA**. Differences between the copies occur in the second and fifth positions. Words in the region that do not overlap the second and fifth positions reflect the length of a single copy, a periodicity of 8. Words that overlap the second position but not the fifth position reflect a recurring pattern across two copies, a periodicity of 16. Words overlapping the fifth position reflect patterns across one, two and four copies, periodicities of 8, 16 and 32. In all, the region generates many runs in the distance array with the 16-runs dominating.

This region is located initially via the strong periodicity of 16. Subsequent analysis of the periodicities reveals multiple, related periodicities at 8, 16 and 32. The analysis selects the periodicity of 8 as the region's base periodicity. Chapter 5 presents analysis of several sequence containing MPTR regions, e.g. GenBank locus MM102B5 in Section 5.1.11 and GenBank locus HSVDJSAT in Section 5.1.10.

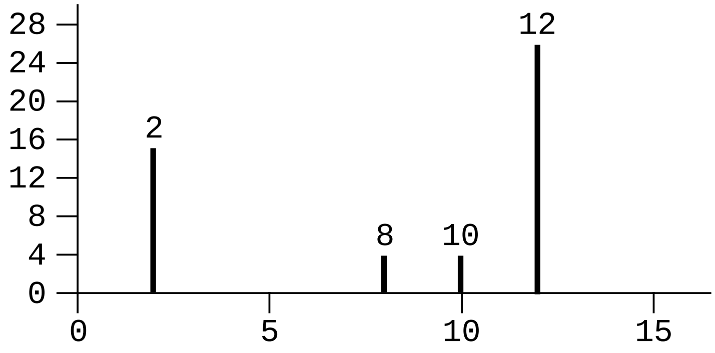
A DNA Sequence: Sequential Copies in Region

T	*T*	*A*	*G*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*
T	*T*	*A*	*G*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*
T	*T*	*A*	*G*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*
T	*T*	*A*	*G*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*
T	*T*	*A*	*G*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*

B Distance Array: Sequential Copies in Region

0	0	0	0	0	0	0	0	0	2	2	2
0	0	0	12	12	12	12	8	10	2	2	2
12	12	12	12	12	12	12	8	10	2	2	2
12	12	12	12	12	12	12	8	10	2	2	2
12	12	12	12	12	12	12	8	10	2	2	2

C Distance Histogram: All Distances in Region



This region is a nested tandem repeat region composed of a **TTAG** and a simple sequence repeat (SSR) having a **CA** pattern. The 2-runs indicate the presence of an SSR with a pattern length of 2. The 12-runs indicate the length of the overall pattern, **TTAGCACACACA** or

TTAG(CA)$_4$, including the nested SSR. In all, the region generates four 12-runs and five 2-runs in the distance array. Both the 2 and 12 periodicities are significant as they capture different pattern recurrences.

This region is located initially via the periodicity of 12. A base period of 12 yields a proper alignment of the region but does not capture the presence of the nested SSR. Thus, a base period of 4 represents the sequence not associated with the nested SSR. An independent SSR analysis selects a **CA** pattern to characterize the SSRs.

The next example shows a challenge associated with identifying nested tandem repeats.

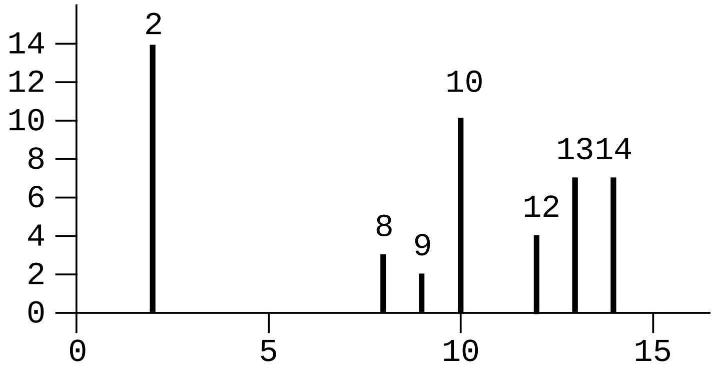
A DNA Sequence: Sequential Copies in Region

T T A G C A C A C A C A
T T A G C A C A C A
T T A G C A C A C A C A C
T T A G C A C A C A C A C A
T T A G C A C A C A

B Distance Array: Sequential Copies in Region

0	0	0	0	0	0	0	0	0	2	2	2
0	0	0	12	12	12	12	8	10	2		
10	10	10	10	10	10	10	8	10	2	2	2
13	13	13	13	13	13	13	9	9	2	2	2
14	14	14	14	14	14	14	8	10	2		

C Distance Histogram: Distance Array in Panel B



This region is a variable length tandem repeat (VLTR) region similar to Panel B except the number of copies in the SSR varies, **TTAG(CA)**$_{3-5}$ or **TTAG(CA)**$_{*}$. As before, the 2-run reflects the presence of the SSR having the **CA** pattern. The 10-run, 12-run, 13-run and 14-run indicate the length of individual copies in the region. Since the number of copies in the nested SSR varies, the overall periodicity of the VLTR region varies.

This region is initially located via the periodicity of 10. Subsequent periodicity analysis reveals a similar set of words common to the periodicities at 10, 12, 13 and 14. In particular, the words **TTAG**, **TAGC**, **AGCA** and **GCAC** occur in each of the periods and the words **ACAT**, **CATT** and **ATTA** occur in all but one of the periods. The analysis creates a collection of periodicities for periods produced by a similar set of words. In addition to the collection, this region contains a periodicity of 2 indicating the occurrence of several SSRs.

The collection of related periodicities together with multiple occurrences of a SSR having the same pattern triggers the VLTR analysis. This analysis strips the SSR off of each copy in order to determine the base periodicity of the sequence not associated with the SSR. Here, the base period is 4 for the sequence not associated with the SSRs. An independent SSR analysis selects a **CA** pattern to characterize the SSRs. Chapter 5 presents analysis of several sequences containing VLTR regions, e.g. GenBank locus BOVTGN in Section 5.1.4.

In tandem repeat regions with complex pattern structures, the periodicity signature of the tandem repeat region is diluted and spread across several periodicities. This weakens the signature and makes identification more difficult. This algorithm groups periodicities together based on a set of similar words associated with each periodicity. This boosts the strength of the signature and enables identification of the entire region.

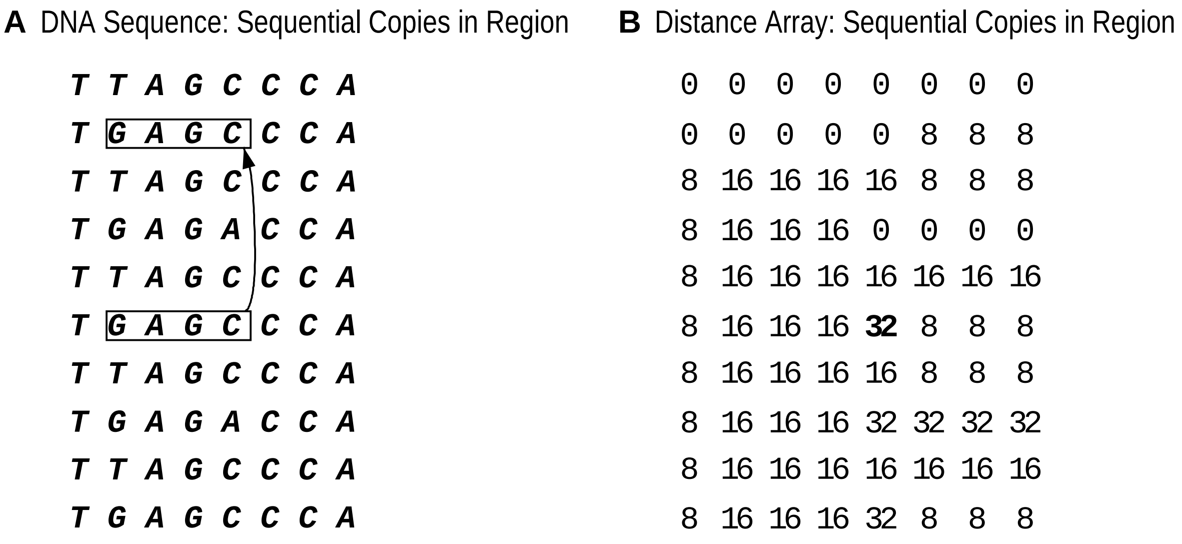
Phase II: Formation of a region pattern. The base periodicity determined in Phase I represents the expected length of a single occurrence of the region's pattern. This provides the basis for selecting or constructing a region pattern. For MPTR regions or regions having a simple pattern structure, an initial pattern is selected from the region using the base periodicity to select a "best" copy. For VLTR regions, an initial pattern is constructed of the pattern associated with the nested SSR appended to a "best" copy selected using the base periodicity. Here, "best" is a region copy or partial copy which best matches the base periodicity and contains the most distances in the distance array equal to the base periodicity.

For all regions, the initial pattern is aligned to the region. A consensus pattern is formed from the alignment. The consensus pattern provides the pattern for characterizing the tandem repeat region.

Phase III: Region characterization by alignment of a pattern to the tandem repeat region. A region is characterized by aligning a pattern to the approximate position of the tandem repeat region on the DNA sequence using a modified wraparound dynamic programming approach (see Section 2.2.4.3, Section 4.3.8 and Section 4.4.8). The alignment sequentially pairs a position in the DNA sequence with a position in the pattern. When necessary, a position in either the pattern or DNA sequence is skipped, i.e. a gap is paired with the position. A scoring function assigns a value to each pair, a positive score for pairs that match and negative scores for pairs that mismatch or are paired with a gap. The alignment chosen to characterize the region maximizes the score producing a "best" alignment of the pattern to the DNA sequence.

The wraparound portion of the dynamic programming algorithm allows a single pattern to be aligned repeatedly to the DNA sequence and is ideal for identifying tandem repeats. The algorithm in use is an extension to standard wraparound dynamic programming algorithms in that the pattern is a complex expression, not a simple sequence. Patterns are denoted as regular expressions though the definitions differ slightly to accommodate biological meanings, e.g. a partial match is allowed for the sequence within a closure element.

Phase IV: Locate cross-copy patterns associated with multiple, related periods. The periodicity signature for a multi-periodic tandem repeat (MPTR) region contains two types of periodicities: those associated with the base periodicity and those associated with integer multiples of the base periodicity. Here is a MPTR region having a base periodicity of 8 with cross copy patterns across two and four copies thereby creating a periodicity signature containing periodicities of 8, 16 and 32.



A DNA Sequence: Sequential Copies in Region

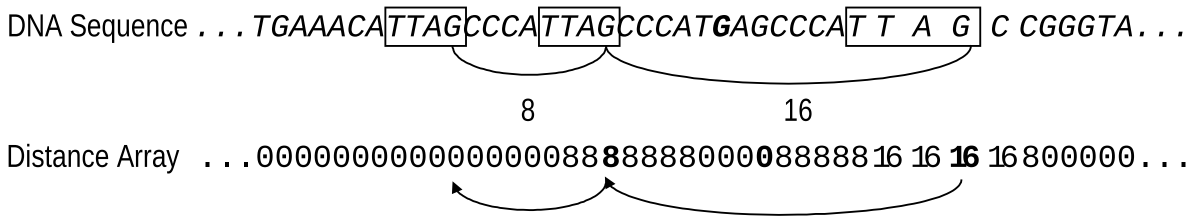
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*A*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*A*	*C*	*C*	*A*
T	*T*	*A*	*G*	*C*	*C*	*C*	*A*
T	*G*	*A*	*G*	*C*	*C*	*C*	*A*

B Distance Array: Sequential Copies in Region

0	0	0	0	0	0	0	0
0	0	0	0	0	8	8	8
8	16	16	16	16	8	8	8
8	16	16	16	0	0	0	0
8	16	16	16	16	16	16	16
8	16	16	16	**32**	8	8	8
8	16	16	16	16	8	8	8
8	16	16	16	32	32	32	32
8	16	16	16	16	16	16	16
8	16	16	16	32	8	8	8

In this example, the second column alternates between a **T** and **G** nucleotide creating a pattern across two copies. The distance array echoes this pattern with distances of 16. Also, the fifth column depicts a pattern across four copies generating a distance of 32 in the distance array.

Substitutions cause multi-copy periodicities to occur. For MPTR regions, these substitutions recur consistently creating a cross-copy pattern but random substitutions arise as well creating multi-copy periodicities but not cross-copy patterns. For example, this region is composed of several occurrences of an 8 bp pattern.


DNA Sequence . . . TGAAACATTAGCCCATTAGCCCATGAGCCCAT T A G C CGGGTA. . .
8
16
Distance Array . . .0000000000000000888888800000888881616 16 16800000. . .

One copy of the pattern contains a single substitution from a **T** to a **G**. As a result, several words differ in the copy causing distances of 16 to occur in the distance array instead of the distances of 8 present elsewhere in the region. Since this is the only substitution in the region, it is clear that a cross-copy pattern is not present.

This algorithm analyzes non-conserved columns in an alignment to identify patterns across copies. A single column is treated as a column sequence and analyzed to create a distance array. As before, the distances reflect recurring words in the column sequence such that the recurring distance of 2 indicates a pattern across two copies. The following example illustrates this analysis.

A DNA Sequence: Sequential Copies in Region

T	T	A	G	C	C	C	A
T	G	A	G	C	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	A	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	C	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	A	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	C	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	A	C	C	A
T	T	A	G	C	C	C	A
T	G	A	G	C	C	C	A

B Distance Arrays: One for Each Column

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	2	1	1	0	1	1	1
1	2	1	1	0	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1
1	2	1	1	4	1	1	1

This region is a MPTR region (same region as previous MPTR example). Each column in Panel **A** is treated as a sequence. A window of length $k = 3$ slides across the column sequence and creates a distance array. Each column in Panel **B** is the distance array for the corresponding column sequence in Panel **A**.

Columns 1, 3, 4, 6, 7 and 8 are conserved as reflected by the homogeneous nucleotide content in Panel **A** and the homogeneous distances of 1 in Panel **B**. Column 2 alternates between **T** and **G**. This alteration corresponds to distances of 2 and indicates a pattern across two copies. Similarly, column 5 contains an **A** recurring every four nucleotides. This corresponds to distances of 4 and a pattern across four copies.

Each pattern across copies represents a level of recurrence in the region. For a pattern across two copies, a pattern is constructed of two consensus base patterns that are modified to reflect the cross-copy pattern.

In this example, two patterns are constructed. The consensus pattern for the base period is **T(T/G)AG(C/A)CCA** or **TKAGMCCA** using ambiguity characters. A pattern reflecting a pattern across two copies (**2X**) combines two consensus patterns and modifies the second column to reflect the alternation between **T** and **G**: **TTAGMCCA TGAGMCCA**. A pattern reflecting a pattern across four copies (**4X**) combines four consensus patterns and modifies them in two ways. First, modify the pattern to reflect cross-copy patterns for factors of 4, e.g. the **T/G** alteration. Second, modify the pattern to reflect the pattern across four copies: **TTAGCCCA TGAGACCA TTAGCCCA TGAGCCCA**. Each constructed pattern is considered a multiple of the base period. Each is aligned to the region to create its own region characterization.

Summary. The identification algorithm works by finding runs of identical distances in the distance array. These distances correspond to the periodicity of the region and locate potential tandem repeats. Each potential tandem repeat is analyzed to determine a base periodicity. The base periodicity allows selection or construction of a pattern. The region is characterized by aligning the pattern to the region on the DNA sequence. Analysis of the alignment determines whether cross-copy patterns occur. For each significant cross-copy pattern, a pattern is constructed of multiple consensus patterns and modified to reflect the cross-copy patterns. Each pattern is aligned to the region to create its own region characterization.

4.2. Fundamental biological issues

It is known that DNA exhibits high levels of sequence duplication, often in tandem repetition. The duplication is retained in the sequence and allowed to gradually mutate. In some instances, the types of mutation are constrained, e.g. basic protein structures are constrained by the triplet encoding associated with codons. In other cases, complete genes are duplicated in tandem and then mutate to have slightly different function. In others, mutation proceeds unencumbered. Thus, an understanding of the duplication and mutational events in DNA sequences is critical to the development of algorithms for identifying tandem repeat regions in real DNA sequences.

In this section, three fundamental biological issues are discussed: region periodicity (Section 4.2.1), pattern conservation (Section 4.2.2) and the role of similarities and differences in sequence data (Section 4.2.3). All three are likely associated with sequence duplication and mutation events. Understanding each issue is critical to developing a solid algorithm for identifying tandem repeat regions.

4.2.1. Region periodicity

The single most crucial decision associated with tandem repeat analysis is the selection of a periodicity that characterizes a tandem repeat region. The periodicity reflects the duplication history that created the region. The current working theory is that a single subsequence randomly duplicates to create two tandem occurrences of the same subsequence. Over the course of time, additional duplication events increase the number of copies in a tandem repeat region. One mechanism is slippage of one strand relative to the other strand. A mispairing of copies in the region creates unpaired bulges in both strands. DNA repair mechanisms either excise the unpaired bulge or break the shorter strand and insert additional nucleotides to pair with the bulge. This mechanism accounts for rapid expansion or contraction of a region, tends to add or remove complete copies to a region and maintains a consistent tandem repetition throughout a region; a feature exhibited by a region’s periodicity.

Mutation causes imperfections to occur in the region. Single nucleotide substitutions are most common. Insertions, deletions and partial duplications of a copy are less prevalent. On the other hand, substitutions do not change the length of a copy and thus do not affect a region's periodicity. Insertions and deletions while less frequent cause individual copies to increase or decrease in length, thereby affecting a region's periodicity. Furthermore, large deletions and insertions do occur and dramatically affect perceived local periodicities. Thus, a good analysis of a region's periodicity must take into account length variations.

Duplication and mutation events interlace to create complex pattern structures. For instance, suppose a subsequence is tandemly duplicated, modified by a single nucleotide mutation in one of the copies and followed by another duplication event occurs that duplicates the entire region. This creates a nucleotide alteration in one position of the pattern. Similar events with insertions and deletions create complex pattern structures involving copies of different lengths. Complex interlacing of copies with varying lengths creates very complex pattern structures with unusual periodicities. Many other combinations of duplication and mutation events occur and create simple or complex pattern structures.

This algorithm uses an approach that begins with a single region periodicity. This periodicity reflects the dominant distance between identical words. The same words are present in other periodicities reflecting mutations within the region. Thus, the single region periodicity is expanded to a collection of periodicities involving a similar set of identical words. This collection of related periodicities associated with a region is called the region ***periodicity signature*** (see Section 4.3.5 and Section 4.4.5).

4.2.2. Pattern conservation

The conservation model in use by an algorithm dictates how a base pattern can vary within a region. Models typically assume either no pattern changes, allow mismatches, i.e. substitutions, or allow differences, i.e. substitutions, deletions and insertions. In addition, a conservation model determines the severity of pattern changes on a local and global scale.

A common approach, the ***k***-mismatch approach, allows copies to vary by ***k*** substitutions. Similarly, the ***k***-difference approach allows ***k*** substitutions, insertions and deletions. Both approaches limit the pattern variation on a local scale, i.e. at most ***k*** positions can differ in a local copy. It also limits the amount of pattern variation on a global scale since all copies must have at most ***k*** positional changes (see Section 2.2.6 and Section 2.3.1).

During analysis of real sequence data, a challenging tandem repeat class was revealed in which a poorly conserved copy is surrounded by moderate to well conserved copies. The ***k***-mismatch and ***k***-difference approaches have difficulty spanning across a poorly conserved copy. In the best scenario for these approaches, a region is divided into two separate regions having essentially identical patterns, located on either side of the poorly conserved copy.

Some regions with complex pattern structures have a similar conservation issue. For example, a region has a structure where every four copies a large subsequence insertion occurs. Each copy where the large insertion occurs generates a high degree of local variation yet, for a larger periodicity, i.e. the periodicity across four copies, the degree of conservation is high.

This algorithm uses a conservation model that allows significant local pattern variation while capping global pattern variation. The algorithm locates matches between copies in a region (see Section 4.4.1). The degree of local pattern variation is irrelevant as long as a sufficient number of good matches occur to locate and properly classify the region. Global pattern variation is capped during the alignment phase when a scoring function maximizes positive scores for matches against negative scores for mismatches and gaps.

4.2.3. Role of similarities and differences in sequence data

Sequence similarities reveal the global structure of a region while sequence differences reveal the local nuances that characterize a region. For example, visual scans of sequence data locates tandem repeat regions by finding sequence similarities. Often, the sequence similarities define the basic repeating pattern in the region. With closer inspection, recurring differences in the sequence identify higher order patterns in the region. As a result, an increase in the size of the pattern may occur.

Sequence similarities locate tandem repeat regions having either simple or complex pattern structures. Sequence differences in simple pattern structure regions are simply random noise. Sequence differences in complex pattern structures are either random noise, or an intricate part of the complex structure. Often, a single position in a pattern varies but the variation recurs repeatedly and rhythmically creating a regular variation across pattern copies. Thus, the design of identification algorithms must assign the role of sequence similarities and differences to the proper identification task.

This algorithm uses both sequence similarities and differences to identify tandem repeat regions. Similarities locate a region (see Section 4.4.4). Periodicity differences together with sequence similarities isolate a periodicity signature and select a base periodicity for a region. Sequence differences isolate periodicities associated with cross-copy patterns in multi-periodic tandem repeat (MPTR) regions (see Section 4.4.9).

4.3. Computational design decisions

The primary design of the algorithm is based upon the fundamental biological issues described in Section 4.2. In this section, the goal is to explain the reasoning behind the design of the algorithm specifically, to describe the biological reasoning for a specific computational design decision.

4.3.1. Identical words

The algorithm uses the distance between identical words to locate a tandem repeat region. There is a give and take that occurs between identifying identical words and similar words. *Identifying* similar words computationally takes considerable effort. *Analyzing* the results of identifying the similar words takes considerable effort. Identifying identical words requires the least amount of identification and the least analysis. The price for doing less work is a loss of information. So, the question becomes; is the loss of information worth the computational cost? Are there other ways of acquiring the information?

For DNA sequences, the mutation and duplication rates are such that regions rarely degrade so far that there are no identical words remaining. Often by the time the regions have degraded this far, the region can no longer be considered a tandem repeat region. Thus, this algorithm assumes that at least a few identical words remain in the region and uses a few occurrences to locate the region. Of course, the more identical words, the easier the region is to find and the more confident one is of identifying it and properly classifying it.

4.3.2. Locating a tandem repeat region: The distance filter and periodicity significance

Distances between identical words are placed in a distance array. For tandem repeats, two observations are made regarding the distance array. First, the distances in the array reflect the region's periodicity. Second, tandem repeats generate runs of consecutive, identical distances in the distance array. A position is a member of a run if the distance in the current position is identical to the distance in the position to the left or to the right. This algorithm uses a period-independent filter to select positions contained in consecutive, identical distance runs and to place the positions in a histogram such that positions associated with the same distance are in the same periodicity group.

This filter removes noise generated by the presence of non-repetitive sequence in the overall sequence and inexact conservation within the region. The role of the filter is to concentrate the positions corresponding to tandem repeat regions. It is not critical to include all positions in a tandem repeat region nor is it crucial to exclude all positions not associated with tandem repeat regions. Rather, enough positions need to be included in order to *locate* a tandem repeat region.

This algorithm requires five positions associated with runs of the same distance in order to locate a tandem repeat region. Intuitively, this threshold, $T_1 = 5$, corresponds to (1) a well-conserved region composed of a single, long run of at least five identical distances, (2) a moderately conserved region composed of at least a weak, run of two positions and a stronger run of three positions or (3) a weak region composed of at least three runs of two positions. This threshold does not allow a single run of a few consecutive, identical distances, as this tends to correspond to a single recurrence of an isolated substring. An exception is made for short distances where it is known that a least two full exact copies occur, i.e. $2\boldsymbol{d} \leq \boldsymbol{d} + \boldsymbol{k} + |\boldsymbol{d\ run}|$ (see Section 4.3.3 and Section 4.4.2). Likewise, two runs of two positions each is not allowed as this often corresponds to two independent substrings a little longer than the word size. The threshold allows all other combinations of runs of identical distances.

A search through the histogram reveals periodicity groups of identical distances that exceed the threshold. An analysis splits the group into proximal positions on the DNA sequence. A split occurs in a group if (1) a gap of at least twice the distance associated with the group, i.e. a distance of at least two copies of the current periodicity and (2) a gap of at least 100 positions occurs, i.e. a constraint that allows groups having small periodicities to remain as a single group. A group of proximal positions is considered a putative tandem repeat region and during analysis represents the core group of positions associated with a region. In addition, the group defines the *initial position* on the DNA sequence of the region, the *initial periodicity* of the region and the *initial set of identical words* associated with a region. Section 4.4.1 describes additional details for this algorithm.

4.3.3. Identifying simple sequence repeats (SSRs)

Simple sequence repeats (SSRs) are tandem repeats having a short pattern at most six nucleotides long. SSR identification occurs during the construction of the distance array where every distance, $\boldsymbol{d},\ \boldsymbol{d} \leq \boldsymbol{k}$, indicates a SSR since the region length $\geq (\boldsymbol{k} + \boldsymbol{d}) \geq (2 \times \boldsymbol{d})$, i.e. at least two copies of the pattern. This approach is reasonable since the distance array construction algorithm uses a default window size of $\boldsymbol{k} = 6$. For example, a distance of 1 indicates that a string of seven identical nucleotides occurs while a distance of 6 indicates that two tandem occurrences of the same six-nucleotide subsequence occur. Thus, this algorithm labels all single distance runs up to and including the current window size as a perfect SSR.

All perfect SSRs are analyzed to determine whether expansion of the region identifies a larger imperfect SSR region. The pattern associated with the perfect SSR is aligned to the region surrounding the perfect SSR using RegWDP (see Section 4.4.8.1). Section 4.4.2 describes the specifics of the SSR identification algorithm.

4.3.4. Defining an initial region

The goal of this analysis is to roughly define the edges of a region on the DNA sequence. This requires expanding the group of proximal positions to include all or most copies in the region without overextending beyond the bounds of the actual tandem repeat region. The approach expands the initial set of positions, the initial set of identical words and the original periodicity associated with the group of proximal positions (see Section 4.3.2) to sets of positions, identical words and periodicities that define an initial region. This process compensates for copy-length variations, random substitutions and cross-copy patterns associated with complex pattern structures.

The analysis starts with the initial set of identical words and searches for periodicities having a similar set of identical words. These periodicities together with the original periodicity form a set of related periodicities. An analysis of this set determines whether identical words common within the set of periodicities is absent from the set of identical words and if necessary adds the words to the set. This analysis repeats the cycle identifying additional related periodicities and common identical words. A final step expands the set of positions by locating all proximal positions of words in the set of identical words. This analysis generates a set of related periodicities, a set of identical words and a set of positions.

The techniques expand the initial sets associated with a group of proximal positions to sets representing the entire region. The analysis assumes that a similar set of words compose most or all of the copies in a region. Furthermore, it assumes that proximal positions of identical words, proximal positions in the same periodicity and different periodicities composed of a similar set of identical words all correspond to the same region. The final approach is the result of extending and expanding many ideas by empirical analysis of sequence data.

Initial analysis considered two techniques. First, use the set of identical words to identify related periodicities in the histogram, i.e. those periodicities containing enough positions associated with identical words. Second, use the set of identical words to extend the set of positions along the sequence to all proximal occurrences of the identical words. Both techniques performed well on a subset of regions and failed miserably at identifying other regions. The regions identified for each differed. Thus, a new approach using both techniques performed significantly better than either technique independently.

Unfortunately, several key problems remained. First, some regions had a set of positions that extended beyond the range of the actual tandem repeat region. Often in these cases, “islands” of identical words associated with the region occurred at the edge of a region. Occasionally, two tandem repeat regions were jammed together in one region. Second, some regions defined by two copies of a long pattern were not located. In these regions, it is obvious that a word occurs only twice, once in each copy of the pattern. Unfortunately, the similarity thresholds required more occurrences of identical words in order for the word to be added to the set of identical words. Third, some regions with highly variable copy lengths were either partially located or not located at all. This was particularly true of variable length tandem repeat regions where each copy length differs. Here, an identical word occurs multiple times in a region but for a single periodicity, only two occurrences of an identical word occur. Other problems remained as well.

The final approach is as follows. The information associated with the group of proximal positions initializes the set of related periodicities, the set of identical words and the set of positions and are referred to as the primary (1°) data. New periodicities (2° periodicities) are added to the set of periodicities using the criteria that at least T_1 (i.e. 5) positions in the periodicity must be associated with a word in the set of identical words. All words associated with all periodicities in the set of periodicities are added to the set of identical words and the set is pruned to remove weak identical words (details below). This process occurs again adding new periodicities and identical words (3° data) to the corresponding sets. In addition, all proximal positions of words in the set of identical words are added to the set of positions. The final sets of positions and identical words are pruned to remove “islands” of identical words unrelated to the rest of the region and to remove positions which conflict with previous, identified regions.

Pruning of the set of identical words eliminates words that do not provide a significant contribution to the region. Regions vary considerably such that some regions contain many copies and others contain only two copies. A variable threshold approach repeatedly trims the set of identical words and stops the pruning process before too many words are removed. In particular, the process performs a tentative pruning and measures the impact of the procedure before completing the pruning process. Tentative pruning starts with pruning single occurrences of an identical word and proceeds to at most four occurrences of an identical word such that all identical words with T_1 (i.e. five) occurrences remain in the set of identical words. The measure for whether to complete a pruning step compares the number of new words added to the set of identical words just prior to the pruning procedure and ensures that at least 30% of the new words remain in the set of identical words. This threshold is purposely set as a low hurdle and works well in practice.

Final region pruning removes positions from the set of positions and words from the set of identical words. First, the pruning procedure removes "islands" of positions composed of identical words not found elsewhere in the region. Often, these correspond to random occurrences of identical words separated by a periodicity identical to a periodicity in the region. Also, cases involve two adjacent, unrelated, tandem repeat regions with the same periodicity. Second, the pruning procedure removes positions associated with tandem repeat regions previously identified. This eliminates obvious redundant identification of the same region while allowing continued analysis for a similar, proximal region.

After the expansion and pruning stages, three sets remain: the set of positions, the set of related periodicities and the set of identical words. The set of related periodicities is denoted as the region's periodicity signature. The set of positions defines the initial edges of the region. The set of identical words are used to define putative copies within the region. Section 4.4.4 describes additional details for this algorithm.

4.3.5. Determining a region's base periodicity

The initial periodicity associated with a group of proximal positions represents the dominant periodicity within the region but may not represent the base periodicity of the region. Since the selection of a base periodicity is crucial to proper characterization of the region, this analysis accounts for simple and complex pattern structures as well as sequence differences associated with insertions, deletions and substitutions. Analysis targets the periodicity signature of the region, the presence of SSRs in the region and all the periodicities present in the region. Analysis is limited to the initial edges defined for the region.

The approach involves three separate procedures. The first step checks whether the region is a VLTR region. Section 4.3.6 describes this procedure. The second step corrects the periodicity when the dominant periodicity represents two or more copies of the region. This is the result of a substantial substitution rate within the region or a recurring cross-copy pattern associated with a multi-periodic tandem repeat. The same correction procedure applies to both cases. The third step expands the base periodicity to proximal periodicities to compensate for slight copy length variations. These steps are described in reverse order. The base periodicity associated with the group of proximal positions initializes the process.

The flexibility and constraints within this analysis are the result of empirical testing.

Step 3: Adjusting for copy length variation. The selection procedure expands the base periodicity to proximal periodicities to compensate for copy length variation by creating a halo region around the base periodicity and independently testing each periodicity within the halo region exceeds a threshold of three occurrences. The halo region includes all periodicities within three of the initial base periodicity. The analysis places all successful periodicities including the initial base periodicity in a set of base periodicities. The base periodicity in the set with the most occurrences is designated as the primary base periodicity. Section 4.4.5 describes additional details for this algorithm.

Step 2: Correcting for a periodicity across multiple copies. The correction procedure tests all periodicities in the initial region smaller than the initial base periodicity to determine whether an integer multiple of one of these periodicities is equal to the dominant periodicity. A little flexibility is given and a few constraints are applied. The constraint is that the integer multiple must be at most four, i.e. the periodicity must be at least one-fourth the size of the dominant periodicity. The flexibility involves a halo region of three around the initial base periodicity such that an integer multiple does not need to precisely match the periodicity. For example, an initial base periodicity of 19 is satisfied by a periodicity of 8 through 11, i.e. a halo exists around 19 from 16 through 22 such that two copies of length 8 through 11 fit within the halo region. Section 4.4.5 describes additional details for this algorithm.

4.3.6. Testing for a VLTR region

The algorithm tests each potential tandem repeat region to determine whether it might be a variable length tandem repeat (VLTR) region. A VLTR region contains a nested SSR region in each copy of the VLTR region such that the number of copies associated with each SSR region varies. A VLTR region's periodicity signature reflects periodicities for both the nested SSR regions and the variable copy length of the VLTR region.

This procedure performs two tests to determine whether a region is a potential VLTR region. First, a procedure checks the SSRs in the region to determine whether multiple SSRs have the same pattern. Second, a procedure removes all occurrences of this SSR in the VLTR region and tests the periodicity signature to determine whether the periodicity variability decreases. If sufficient SSRs with the same pattern occur and their removal decreases the variable periodicity then the region is classified as a potential tandem repeat region. An analysis of the non-SSR sequence determines a set of base periodicities for the non-SSR portion of the VLTR region. The pattern for the SSR represents the SSR portion. Section 4.4.6 describes additional details for this algorithm.

Select qualifying SSRs. The analysis checks all SSR regions in the proximity of the potential VLTR region, selects the SSR regions having most occurrences of the same pattern and checks whether the number of SSR regions is sufficient to qualify. The ideal scenario is that either each copy has a distinct, strong SSR or that no SSRs occur within the region. Instead, it is rare for each copy of a VLTR region to contain a strong, distinct SSR. At the same time, it is not uncommon for a single SSR region to occur in a tandem repeat region. Thus, a balance needs to be struck between having sufficient SSR regions to merit further analysis and having flexibility to allow only some of the copies to have SSRs. Through empirical testing, the analysis requires the number of SSR regions having the same pattern to meet or exceed two SSR regions and the quantity equivalent to $\boldsymbol{V} = 40\%$ of the VLTR region copies.

Remove qualified SSRs and test periodicity signature for variability decrease. Removal of perfect SSR regions is straightforward but, removal of short, shadowy SSR regions and imperfect copies of a SSR region pose difficulties. This procedure compensates for these difficulties by removing all obvious SSR regions and some of the shadowy SSR regions and testing whether this removal *decreases* the size of the dominant periodicity and whether the number of positions in the smaller periodicity is sufficiently large. It does not require that all SSR regions be removed.

Parallel to the DNA sequence and the distance array, construct a new distance array where the distances across perfect SSR regions and SSR shadows are reduced by the number of positions in each SSR region. Use this new distance array to construct a new periodicity signature using distances associated with the region's set of positions. Choose the periodicity in the new signature associated with the most positions. Ensure that the new dominant periodicity is *smaller* than the original periodicity and that the number of positions in the new periodicity is at least ***V***(40%) of the original periodicity. Qualifying regions are reclassified as potential VLTR regions.

Select VLTR region's base periodicity. A VLTR pattern contains an SSR portion and a non-SSR portion. During this portion of the analysis, the pattern of the SSR is known while the non-SSR sequence is unknown. This analysis selects a single base periodicity for the non-SSR sequence. The base periodicity is the new dominant periodicity chosen above, i.e. from the periodicity signature associated with the distance array that reflects SSR removal.

4.3.7. Selecting a region pattern

Construction of a quality base pattern is quite easy for some regions and very difficult for other regions. This approach chooses the "best" copy in a region as an initial base pattern, aligns the initial base pattern to the region and forms a consensus pattern from the alignment that serves as the region's base pattern. The best copy is usually quite good but, when a region has significant mutation levels, it does not suffice for characterizing a region. Thus, forming a consensus from all copies in the initial region enables a better region characterization.

Analysis of the sets of identical words and positions associated with the initial region enables the initial region to be split into copies. The identical word that most often recurs in the region determines the break point between copies. This procedure assumes that the most frequent word occurs at most once in each copy.

Select a "best copy" pattern. Two criteria decide which copy is a region's "best" copy: copy length relative to the base periodicity and the number of distances equal to the base periodicity associated with positions in the copy including distances linking to smaller positions (positions to the left) and distances linked from larger positions (positions from the right). Selecting a copy equal to the base periodicity is of supreme importance. The criteria for maximizing the number of distances equal to the base periodicity reflects the conservation level of the copy and allows a copy to be selected that best captures the base periodicity. If multiple copies tie, choose any of the equivalent copies as an initial base pattern. This is the process for selecting a base pattern for regions with simple pattern structures as well as multi-periodic tandem repeat region. Section 4.4.7 describes additional details for this algorithm.

Construct a VLTR pattern. A VLTR pattern has two components, the SSR portion and the non-SSR portion. The pattern for the SSR portion is known while the non-SSR portion is unknown. Selection of a non-SSR pattern uses the best copy selection procedure coupled with the non-SSR "base periodicity" to chose an initial pattern for the non-SSR portion. Construction of a VLTR pattern prepends the non-SSR pattern to a SSR, i.e. a closure element containing a SSR pattern, $\{\text{SSR pattern}\}_*$. Section 4.4.7 describes additional details for this algorithm.

Construct MPTR patterns for integer multiples of the base pattern. Analysis of the base pattern region characterization determines whether cross-copy patterns occur that are integer multiples of the base pattern. If a column in the base pattern alignment alternates between two nucleotides, then a pattern across two copies occurs. Likewise, if a column contains a particular nucleotide every four copies and all other copies contain another nucleotide, then a pattern across four copies occurs.

This procedure constructs patterns for cross-copy patterns. For a pattern across two copies, two base patterns are concatenated. If the nucleotide alteration occurs in the fifth column then the fifth position in each base pattern is modified to reflect the alteration. This cross-copy pattern is denoted the 2X pattern. For a pattern across four copies, four base patterns are concatenated. Here, two types of modifications occur. First, cross-copy patterns that are factors of four modify positions in the base pattern. Second, the column generating the pattern across four copies modifies the pattern. This approach works for well-conserved, cross-copy pattern structures but does not work for pattern structures where occasional copies are absent or extra copies are present.

The problem with regions where occasional copies are absent or extra copies are present is that region copies align to the wrong modified base pattern. Since the modified base patterns often have only small changes, skipping an entire modified base pattern is too costly during the alignment phase. So, it is important to use a procedure that applies little or no penalty for aligning to any modified base pattern. Thus, a modified construction procedure defines a pattern where each copy of a region has an equal ability to align to any of the modified base patterns.

The modified procedure links the modified base patterns using a union element which allows alternative patterns and is commonly known as the “OR” element. Under this schema, a region copy aligns to any modified base pattern and aligns the next region copy to any of the other modified base patterns. As a practical addition, this approach uses a slight incentive for aligning the next copy to the next modified base pattern but otherwise applies no penalty for “skipping” modified base patterns. Section 4.4.9 describes additional details for this algorithm.

4.3.8. Characterizing a region using a pattern

Region characterization involves alignment of a pattern to a tandem repeat region on the DNA sequence using a regular expression based, wraparound dynamic programming algorithm (RegWDP). Section 4.3.8.1 discusses the RegWDP algorithm. Section 4.3.8.2 discusses the process of determining the edges of the alignment. Section 4.3.8.3 discusses the process of forming a consensus pattern. Section 4.4.8 describes additional details for this algorithm.

4.3.8.1. Forming an initial alignment: Wraparound dynamic programming using regular expressions (RegWDP)

The RegWDP algorithm represents a significant computational extension to existing wraparound dynamic programming (WDP) algorithms. First, typical WDP algorithms use a two-pass algorithm but this algorithm introduces a one-pass algorithm. Second, RegWDP allows complex pattern descriptions using regular expressions (see Section 2.2.3). The standard regular expression representation varies slightly with this implementation in order to adhere to the biological requirements of tandem repeat regions (see Section 3.2).

The goal of this section is to describe pattern expressions, alignment calculations and how this approach handles various dynamic programming issues. First, this section describes how to construct the scoring matrix using a one-pass algorithm. Second, the basic scoring function associated with matches, mismatches and gaps is given. Next, a description of three pattern expressions and their calculation transitions are given: the pattern expression for a region with a simple pattern structure, the pattern expression for a nested tandem repeat region and the pattern expression for cross-copy patterns associated with multi-periodic tandem repeat regions. Finally, backtracking through the scoring matrix to produce an initial alignment is described.

The scoring matrix: A one-pass algorithm. Traditional WDP algorithms employ a two-pass approach to construct the scoring matrix $\boldsymbol{T}$ for the alignment of a sequence of length $\boldsymbol{n}$ to a pattern of length $\boldsymbol{m}$. The first pass for aligning a single sequence position to each position in the pattern determines the scores for row $\boldsymbol{i}$ in matrix $\boldsymbol{T}$ and uses the traditional dynamic programming (DP) approach involving three neighboring pre-calculated cells: $\boldsymbol{T}(\boldsymbol{i}-1,\boldsymbol{j}-1)$, $\boldsymbol{T}(\boldsymbol{i}-1,\boldsymbol{j})$ and $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j}-1)$. The second pass determines the wraparound calculation by copying the score in the last column, $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{m})$, to a shadow column to the left of the first pattern column, $\boldsymbol{T}(\boldsymbol{i},0)$ and updating the scores in cells across the row, i.e. updating a cell's dependency on $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j}-1)$. Section 2.2.4 describes the standard DP and WDP algorithms.

The one-pass algorithm delays the second pass wraparound calculation associated with row $\boldsymbol{i}-1$ until the calculation for row $\boldsymbol{i}$. Two different implementations are possible for the one-pass algorithm. The first implementation places a shadow column to the left of the first pattern position. When calculating the scores for row $\boldsymbol{i}$, the first action is to copy cell

$T(i-1,m)$ to $T(i-1,0)$. Then, the three neighboring cell algorithm proceeds as normal. The second choice is no shadow column. Instead, each cell is row i has a new dependency on $T(i-1,m)$. This implementation uses the second choice since the addition of regular expressions also requires the addition of cell dependencies.

The one-pass approach has two concessions. First, a slight, temporary inconsistency occurs in the scoring matrix. This is not a concern since the path through the inconsistent cells is never traversed. Second, the DP calculation typically accesses only the three neighboring cells. Incorporation of the wraparound calculation from the previous position, i.e. the previous row, stretches the algorithm to access non-proximal cells. This is not a concern for two reasons. One, the non-proximal cell is easy to locate. Two, the algorithm needs to be stretched to non-proximal cells in order to accommodate the regular expression calculation. Section 4.4.8.1 describes the specifics of the algorithm.

Basic scoring: Matches, mismatches and gaps. A "character" scoring matrix provides the score for aligning each possible character in a sequence against each possible character in a pattern, including the nucleotides {**A**, **C**, **G**, and **T**}, the ambiguity characters {**B**, **D**, **H**, **K**, **M**, **N**, **R**, **S**, **V**, **W**, **X** and **Y**}, and a gap for a single position. A *match* occurs when a nucleotide in the sequence is aligned to the same nucleotide in the pattern, e.g. aligning an **A** to an **A**. A *mismatch* occurs when different nucleotides are aligned, e.g. aligning an **A** to a **C**. A single *gap* in either the sequence or the pattern is a gap. Furthermore, ambiguity characters achieve partial matches, e.g. a **R** represents an **A** or a **G** and aligning a **R** to an **A** is $0.5\times$match, aligning a **R** to a **R** is a full match and aligning a **R** to a **C** is a full mismatch. Finally, the specific scores for matches, mismatches and gaps change within the algorithm but the ratio for match to mismatch to gap does not change. The default ratio is `1:−1:−3` for match to mismatch to gap.

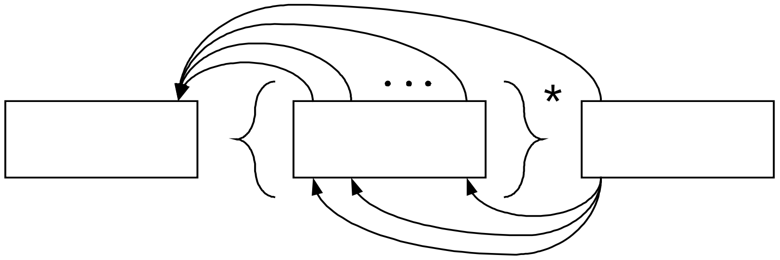
The basic pattern expression: regions with a simple pattern structure. Tandem repeat regions with a simple pattern structure have a pattern that is a sequence of nucleotides (see Section 3.2). A closure element denotes the consecutive recurrence of a string or, in this case a pattern, and is classic notation for regular expressions (see Section 2.2.3). For regular expressions, the closure element denotes that the enclosed string occurs zero or more times. Here, the algorithm uses a modified closure definition that reflects the biology of tandem repeat regions. Specifically, the modification allows a tandem repeat region *to begin at any position* in the pattern, to cycle through the positions in the pattern any number of times and *to end at any position* in the pattern. This differs from the standard closure definition that begins alignment at the first position in the pattern and ends alignment at the last position in the pattern.

For this implementation, it is the closure element that triggers the wraparound condition associated with the WDP algorithm. So for a region with a simple pattern structure, a pattern within a closure element is aligned using a one-pass WDP algorithm. Section 4.4.8.1 describes the specifics of the algorithm.

Pattern expressions using closure components: nested tandem repeat regions. A nested tandem repeat region is a tandem repeat nested inside the pattern of another tandem repeat (see Section 3.2). The complex expressions for these regions involve a second closure component nested inside the primary closure component. For standard regular expression closure definitions, an implementation of nested closure components is straightforward. For the modified closure definition just described, an implementation is complicated by the addition of dependencies associated with the many start and stop positions in the internal closure component.

This section describes the dependencies associated with nested closure components and the current implementation for realizing the dependencies. The approach makes no assumptions about the number of closure components associated with a pattern, nor the depth to which closure components are nested. Rather, both the quantity and nesting depth are unlimited.

The transitional dependencies for a nested closure component involve each position in the closure component. Below is a schematic of a closure component within a pattern.

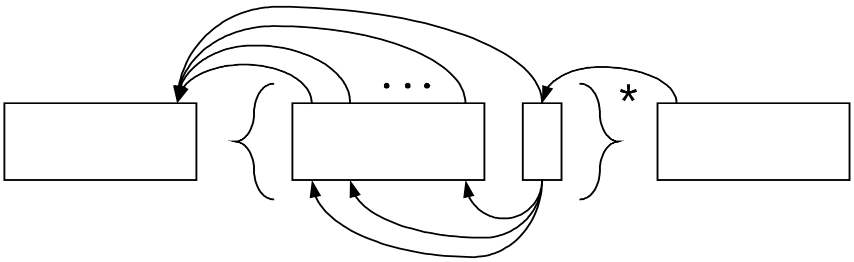


The arrows indicate dependencies of positions within the closure component upon previous positions in the pattern, dependencies of subsequent positions in the pattern upon positions in the closure component and the dependency of the position subsequent to the closure component to the position previous to the closure component. Not shown are the dependencies upon multiple positions in the sequence, i.e. the tip of each arrow represents a dependency upon cells in both the current and previous sequence positions.

The above schematic is deceivingly simple in that it represents the simplest nested closure component scenario. A more difficult scenario is an internal closure component located at the rightmost edge of the primary closure component. The wraparound dependency for the primary closure component traditionally relies on the last position in the pattern. Here, the potential last position in the primary closure component is every position in the internal closure component and the position just prior to the internal closure component. Thus, the number of wraparound dependencies explodes especially when coupled with the one-pass algorithm used in this implementation.

Another scenario that increases the number of dependencies is multiple closure components nested one inside the next. Here, a position within a closure component nested several levels deep has dependencies associated with each closure component in which it is nested. Tracking, updating and applying these dependencies requires additional procedures.

The introduction of a shadow column reduces the number of cell dependencies into a closure component from positions outside a closure component and simplifies the dependency that jumps the closure component. The shadow column occurs at the end of a closure component, carries the dependencies leading back into each position in the component and to the position just prior to the closure component and thereby consolidates all external dependencies coming into the component in one location. A shadow column is not necessary in front of the closure component since all dependencies are consolidated into one position, i.e. the position just prior to the closure component. This implementation simplifies the dependencies into, out of and across a closure component and makes the component act like a single position in the pattern.



The implementation minimizes dependencies for many calculations. It simplifies calculations for patterns having an internal closure component by reducing the dependencies into the internal closure component from positions in the rest of the pattern to a single cell dependency. Also, the wraparound dependency into a closure component returns to a single dependency.

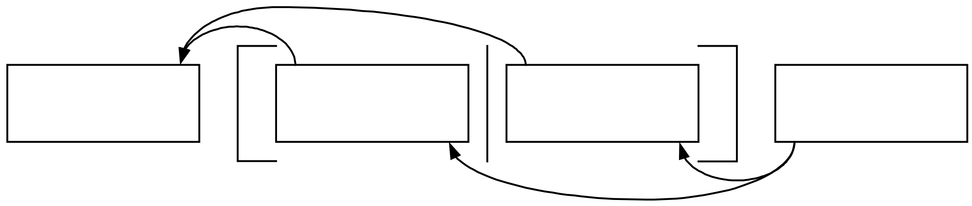
The introduction of a stack data structure keeps track of dependencies associated with multiple closure components nested one inside the next. Under this scenario, a position nested several levels down inside of closure components is dependent upon the wraparound criteria for each closure level. The stack tracks the dependencies for each closure component. Upon entering a closure component, an item is added to the top of the stack. When leaving the closure component, the item is removed from the top of the stack. In this manner, all the relevant dependencies are easily accessible. Section 4.4.8.1 describes the details for this algorithm.

The traditional method for calculating a score for an individual cell is to poll each and every dependency at the point when the score is calculated. Another method is to pre-process groups of cell dependencies that are repeatedly polled for a single cell calculation and reduce the dependencies for the entire group to a single "best" dependency. This pre-processing method works well in two situations associated with the closure component. First, the shadow column at the end of the closure component pre-processes the positions in the closure component and the position just prior to the closure component. In particular, the shadow column is a pointer to the best cell dependency. Second, upon entering a closure component, the stack of closure components adds a new item to the top. Analysis of the stack pre-processes the dependencies and selects the best dependencies. Upon leaving the closure component, the stack reverts to a previous best dependency. Using this method, pre-processing reduces the overall number of dependencies calculated during the course of constructing a scoring matrix.

Pattern expressions using union components: multi-periodic tandem repeat regions. A multi-periodic tandem repeat (MPTR) region has multiple, related patterns that characterize the region (see Section 3.2). The identification algorithm for MPTR regions constructs patterns using multiple, modified base patterns that are placed in a union component that, in turn, is placed directly inside a closure component. The linear order of modified base patterns is the expected cross-copy pattern according to the analysis procedure. The union component enables a region copy to choose which modified base pattern best aligns with the copy. The closure component allows multiple copies of a region to align to the pattern.

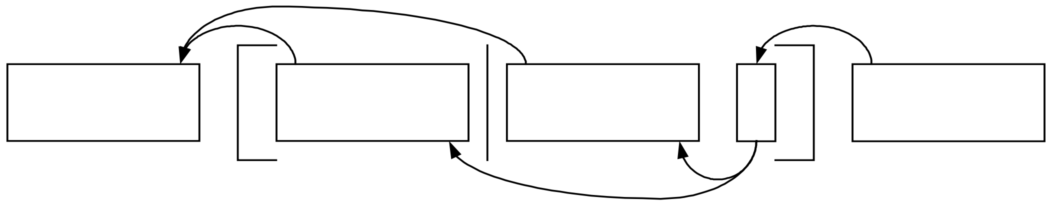
This section describes the dependencies associated with union components and the current implementation for realizing the dependencies. This approach makes no assumptions about the number of alternatives in the component associated with a pattern, nor the types of components nested within the union component. Rather, both the quantity and depth of nested components are unlimited. The identification algorithm that constructs MPTR patterns does assume that a union component is placed directly inside a closure component but the alignment procedure has no such requirement.

The transitional dependencies for a union component involve the first and last position of each alternative in the component. Below is a schematic of a union component



within a pattern. The arrows indicate dependencies of positions with the union component upon previous positions in the pattern and dependencies of subsequent positions in the pattern upon positions in the union component. Not shown are the dependencies upon multiple positions in the sequence, i.e. the tip of each arrow represents a dependency upon cells in both the current and previous sequence positions.

As with the closure component, the introduction of a shadow column reduces the number of cell dependencies into a union component from positions outside a union component. The shadow column occurs at the end of a union component, carries the dependencies leading back into the last position of each alternative and thereby consolidates all external dependencies coming into the component in one location. A shadow column is not necessary in front of the union component since all dependencies are consolidated into one position, the position just prior to the union component. This implementation simplifies the dependencies into and out of a



union component and makes the component act like a single position in the pattern. Section 4.4.8.1 describes the details for this algorithm.

The union component implementation is broad but MPTR patterns are the only usage in the current identification algorithm, namely a union component nested inside a closure component. This double component structure allows the first position of any alternative to follow the last position of any other alternative without suffering a gap penalty. The union component provides the means for alternative copy patterns and the closure component provides for aligning the subsequent region copy. In this scheme, the first position of each alternative relies on the wraparound dependency associated with the closure component. The closure component wraparound dependency relies on the shadow column of the union component. The shadow column of the union component relies on the last position of each alternative.

All alternatives in the union component are not equivalent. The implementation does include a small incentive for aligning the next copy with the next alternative but the incentive is only one-tenth of a match and is easily outbid by an alternative with a better alignment. This small incentive allows the region copies to align to the alternatives in the expected fashion while allowing alignments with better alternatives to occur unencumbered.

Backtracking through the scoring matrix. After computing the scoring matrix, the algorithm backtracks through the scoring matrix to determine an initial alignment of the pattern to the sequence. The extra shadow columns and closure stack are not involved in this process. In fact, the current implementation does not include either the shadow columns or the closure stack in the scoring matrix. Thus, the traditional backtracking scheme determines an initial alignment defined by a series of matches, mismatches and gaps. This implementation splits the alignment into a series of alignments, one for each copy of the pattern in the region. Section 4.4.8.1 describes the details for the RegWDP alignment algorithm.

4.3.8.2. Defining the final alignment: finding the region edges

An alignment represents consecutive occurrences of a pattern that align to a tandem repeat region on a DNA sequence. This procedure analyzes an initial alignment and refines the edges of the alignment so as to define the edge positions of the tandem repeat region on the sequence. The alignment analysis knows *a priori* the pattern used to generate the initial alignment, the edges of the original "best" copy used to create the pattern and the edges of the initial alignment.

The first step looks at alignment scores for each component in the pattern across all copies in the initial alignment, smoothes the scores to spread the discrepancies across multiple scores and determines the core edges using non-smoothed scores and the borderline edges using the smoothed scores. The second step selects borderline and core edges for the region by choosing the outward-most left and right edges for the individual component borderline edges and the inward-most left and right edges for the individual component core edges. All positions outward of the borderline edges are removed and a consensus pattern is formed (see Section 4.3.8.3). The final step verifies the sequence between the borderline edge and the core edge by calculating the alignment score relative to the new consensus pattern of sequence positions starting at the outmost edge and working inward. Poor sequence conservation causes the score to drop below a threshold and the outmost sequence segment to be removed from the region.

The initial procedure for defining a final alignment was simply to use the initial alignment. Various flaws in these alignments indicated a need for additional analysis. The alignment split into region copies minimizes the impact of a single poorly conserved copy.

Similarly, analyzing the individual components minimizes the impact of a poor component within a copy on the entire copy. Analysis using the list of scores occasionally chose an edge that chopped off several region copies. Analysis using the list of smoothed scores occasionally chose an edge that added several extra copies. Thus, a new procedure uses the list of scores to create a core edge and the list of smoothed scores to create a borderline edge knowing that the true edge lies somewhere in between these two edges. So, to find the final edges, a measure of sequence quality was necessary. A consensus pattern of the region itself seems to work well. Scores are re-calculated for the sequence positions between a borderline and core edge using the new consensus pattern and a threshold function based on the number of sequence positions in the sequence segment determines whether a segment is removed. Section 4.4.8.2 describes the details for this algorithm.

4.3.8.3. Forming a consensus pattern from an alignment

An alignment represents consecutive occurrences of a pattern that align to a tandem repeat region on a DNA sequence. A tally procedure counts the number of nucleotide occurrences in the alignment that align with each position in the alignment's pattern. A matrix records the tally such that each position in the pattern individually records the occurrence of gaps and the nucleotides {**A**, **C**, **G** and **T**}. Ambiguous nucleotide characters, {**B**, **D**, **H**, **K**, **M**, **N**, **R**, **S**, **V**, **W**, **X** and **Y**}, that occur in the sequence are evenly split between the appropriate nucleotides. The internal representation for a consensus template is the matrix. For display purposes, the consensus pattern is shown as a linear sequence containing ambiguity characters. Section 4.4.8.2 describes the details of the algorithm.

4.3.9. Identifying regions of similarity (ROS)

Regions of similarity are pairs of non-adjacent regions with strong similarity. Identification occurs by examining groups of proximal positions that fail to be defined as a tandem repeat region. Early analysis found ROSs over 10,000 nucleotides apart. Unfortunately, it was difficult to find the gems in the noisy data. Thus, the following criteria remove the noise associated with small, short or weak regions and retain the interesting regions.

Analysis applies the criteria to the failing groups of proximal positions (see Section 4.3.2 and Section 4.4.1). First, the group must have a periodicity of $\boldsymbol{d} \geq 66$. Second, the group of proximal positions must span 20 nucleotides. Finally, the region contains either 25 positions associated with the region's periodicity or at least 3 ***d*** *runs*. These criteria do not capture the intricacies that exist in sequence data. Rather, it represents a first filtering pass that reveals intriguing and useful results. Section 4.4.3 describes the details for this algorithm.

4.4. A programmer's guide to algorithms and data structures

This section focuses on the algorithms, data structures and other elements necessary for implementing the identification algorithm. This algorithm requires only a DNA sequence, ***S***, in order to identify tandem repeat regions having simple and complex pattern structures. In addition, a series of constants, thresholds and ratios are utilized within the algorithm and can easily be modified via command line arguments. Of these, the most interesting constant to modify is the window size, ***k***. The values for these constants and ratios are given in the appropriate sections. Furthermore, Section 4.3 explains the computational design decisions for the computational details described here and Section 4.2 explains the biological requirements behind this identification approach.

This section is organized in the following manner. Section 4.4.1 describes how to process the DNA sequence and locate potential tandem repeat regions. Section 4.4.2 locates simple sequence repeats (SSRs). Section 4.4.3 locates regions of similarity (ROS). Section 4.4.4 discusses the process for defining an initial region. Section 4.4.5 determines a region's base periodicity. Section 4.4.6 tests whether the region is a variable length tandem repeat (VLTR) region. Section 4.4.7 selects or constructs a region pattern. Section 4.4.8 characterizes a region using the pattern. Section 4.4.9 analyzes the alignment for multi-periodic tandem repeat (MPTR) patterns. Section 4.4.10 describes the algorithm for processing sequences having more than one tandem repeat region. Section 4.4.11 presents algorithms for processing long sequences. Section 4.4.12 describes the data files generated for the user interface by the identification algorithm. Finally, Section 4.4.12 describes the web-based user interface.

4.4.1. Process the DNA sequence and locate potential tandem repeat regions

Problem: Given the DNA sequence ***S*** of length ***n***, over the alphabet Σ and the window size, ***k***, construct parallel to ***S***, the word array, ***W***, and the distance array, ***D***, such that both arrays begin at position ***k*** and end at position ***n***, thereby creating arrays of length $\boldsymbol{n}-\boldsymbol{k}+1$. Also, construct two histograms, ***H*** and ***F*** where ***H*** contains all ***d*** in ***D*** and ***F*** contains all ***d runs*** in ***D*** such that positions of identical distances, ***d***, are placed in the same bin or histogram peak.

Definition: A ***distance run*** or ***d run*** is at least two consecutive positions in ***D*** containing identical distances, ***d***.

Solution: In addition to ***W*** and ***D***, construct a word occurrence array, ***O***, which records the position of the most recent occurrence of a word. ***O*** is an array from 0 to $|\Sigma|^{k}-1$ representing all possible $|\Sigma|^{k}$ words with each element initialized to 0.

Place the ***k***-length window over positions $S_1..S_k$. The position, ***i***, of the word, ***w***, in the window is based on the rightmost index of ***S*** in the window. Convert, w_i, into an integer, $I, 0..|\Sigma|^{k}-1$, and record it in the ***i***th position of ***W***. Then, lookup ***I*** in ***O***, O_I. If O_I=0, then $d_i = 0$ and ***i*** is the leftmost position of ***w*** in ***S***. If $O_I > 0$ then calculate $d_i = i - O_I$. Record d_i in the ***i***th position in the distance array, ***D***. Update O_I to be ***I***. Next, if $d_i > k$, then add ***i*** to the d_i bin in the histogram, ***H***. If $d_i > k$ and d_i starts or extends a *d run*, then add it to the histogram, ***F***. Verify that d_{i-1} of the *d run* is in ***F***. If $d \leq k$, then process d_i as a simple sequence repeat (see Section 4.4.2 for the procedure). Finally, slide the window to the right one position and repeat the procedure until all words in ***S*** are processed.

The default window size is $k = 6$. The default sequence alphabet, Σ**,** is both the nucleotides**,** {**A**, **C**, **G** and **T**}, and the ambiguous nucleotide counterparts, {**B**, **D**, **H**, **K**, **M**, **N**, **R**, **S**, **V**, **W**, **X** and **Y**}. All words containing ambiguous characters are ignored in this initial phase of the algorithm but utilized in later phases of the identification algorithm (see Section 4.4.8.1).

Problem: Given the histogram ***F***, select proximal groups, ***G***, associated with the same peak, ***p***, in ***F*** such that each group, ***g***, in ***G*** is composed of at least five positions. Also, denote the edges, $\boldsymbol{g_e}$, as the leftmost and rightmost position in ***g***.

Solution: For each peak, ***p***, in ***F*** having at least $T_1 = 5$ positions, $|\, \boldsymbol{p} \,| \geq \boldsymbol{T_1}$, place the positions into groups, ***G***, based upon a maximum gap between positions of the minimum of 100 and $2 \times \boldsymbol{p}$. For each group, ***g***, having leftmost and rightmost position edges, $\boldsymbol{g_e}$, verify that the span of ***g*** is at least $|\boldsymbol{g_e}| \geq 0.5 \times \boldsymbol{p}$ or $|\boldsymbol{g_e}| \geq \boldsymbol{p} - 5$. Remove each ***g*** where the number of positions, $|\, \boldsymbol{g} \,| < \boldsymbol{T_1}$. Process each ***g***, $|\, \boldsymbol{g} \,| \geq \boldsymbol{T_1}$ independently (for explanation of thresholds, see Section 4.3.2).

4.4.2. Locate simple sequence repeats (SSRs)

Problem: Given a distance array, ***D***, identify perfect SSR regions, ***R***, and construct a list of SSR regions, $\boldsymbol{R^{\circ}_{SSR}}$.

Solution: SSRs are typically defined as tandem repeats with a pattern of at most six nucleotides. The distance array locates identical *k*-length words and determines the distance between consecutive occurrences of a word. SSR regions correspond to distances, $\boldsymbol{d} \leq 6$. The default value of ***k*** is ***6***. Thus, each $\boldsymbol{d} \leq \boldsymbol{k}$ indicates a SSR since the region length $\geq (\boldsymbol{k} + \boldsymbol{d}) \geq (\boldsymbol{d} + \boldsymbol{d}) \geq 2\boldsymbol{d}$, i.e. at least two copies of a pattern. This algorithm identifies all single instances of $\boldsymbol{d} \leq \boldsymbol{k}$ and identical distance runs of $\boldsymbol{d} \leq \boldsymbol{k}$ as perfect SSRs and records in $\boldsymbol{R^{\circ}_{SSR}}$, the pattern, $\boldsymbol{P_{SSR}}$, the location, ***E***, and the distance, ***d***, of each region.

Problem: Given a perfect SSR region, $\boldsymbol{R}$, with edges, $\boldsymbol{E}$, on the sequence having a pattern, $\boldsymbol{P}$, identify imperfect SSRs that expand $\boldsymbol{R}$ to the region, $\boldsymbol{R'}$, denoted by the edges $\boldsymbol{E'}$ on the sequence. $\boldsymbol{E'}$ is larger than and contains $\boldsymbol{E}$.

Solution: Align the pattern, $\boldsymbol{P}$, to the sequence using the RegWDP algorithm (see Section 4.4.8.1) starting at a position at least 50 and no more than $|\boldsymbol{E}|$positions to the left of $\boldsymbol{E}$. Determine the region, $\boldsymbol{R'}$, and the edges, $\boldsymbol{E'}$, using the standard procedure for determining the final alignment of a pattern to a region. If $\boldsymbol{R'}$ is larger than $\boldsymbol{R}$ then define an imperfect SSR, $\boldsymbol{R'}$.

Problem: Given an imperfect SSR region, $\boldsymbol{R'}$, with edges, $\boldsymbol{E'}$, on the sequence having a pattern, $\boldsymbol{P}$, construct a list of imperfect SSR regions, $\boldsymbol{R^{\circ}_{NSSR}}$, that records the pattern, $\boldsymbol{P}$, the location, $\boldsymbol{E'}$, and the distance, $\boldsymbol{d}$, for each region.

Solution: For each region, $\boldsymbol{R'}$, add an item to $\boldsymbol{R^{\circ}_{NSSR}}$ containing the pattern, $\boldsymbol{P_{SSR}}$, the location, $\boldsymbol{E'}$, and the distance, $\boldsymbol{d}$, of each region.

4.4.3. Locate regions of similarity (ROS)

Problem: Given groups, $\boldsymbol{g}$ in $\boldsymbol{G}$, that fail to become tandem repeats due to less than two copies of a pattern, the associated periodicity, $\boldsymbol{p}$, for each $\boldsymbol{g}$, the associated group edges, $\boldsymbol{g_e}$, and a distance array, $\boldsymbol{D}$, identify regions of similarity, $\boldsymbol{R_{ROS}}$, and place them in a list, $\boldsymbol{R^{\circ}_{ROS}}$.

Solution: Regions of similarity are defined as pairs of non-adjacent regions with strong similarity. Examine groups of proximal positions, $\boldsymbol{g}$ in $\boldsymbol{G}$, that fail to become a tandem repeat region due to less than two copies of a pattern according to the following criteria. First, the periodicity, $\boldsymbol{p}$, associated with $\boldsymbol{g}$, must satisfy, $\boldsymbol{p} \geq 66$. Second, the span of $\boldsymbol{g}$ on $\boldsymbol{S}$ must satisfy, $|\boldsymbol{g_e}| \geq 20$. Finally, at least 3 identical *d runs* in $\boldsymbol{D}$ or at least 25 occurrences in $\boldsymbol{D}$ where $\boldsymbol{d} = \boldsymbol{p}$ within the region $\boldsymbol{g_e}$ on $\boldsymbol{D}$. For each group, $\boldsymbol{g}$, that satisfies the criteria, place an item, $\boldsymbol{R_{ROS}}$, in the list, $\boldsymbol{\mathring{R}_{ROS}}$ such that $\boldsymbol{R_{ROS}}$ contains the periodicity, $\boldsymbol{p}$, and the positions in $\boldsymbol{g}$.

4.4.4. Define an initial region

Problem: Given the word array, $\boldsymbol{W}$, the distance array, $\boldsymbol{D}$, the histogram, $\boldsymbol{H}$, and a selected peak, $\boldsymbol{p}$, construct a list of significant words, $\boldsymbol{S_w}$, and significant periodicities, $\boldsymbol{S_p}$.

Solution: The selected peak, $\boldsymbol{p}$, is placed in $\boldsymbol{S_p}$ and referred to as the **1° peak**. The words in $\boldsymbol{W}$ associated with positions in the 1° peak are placed in the list of words, $\boldsymbol{C_w}$, and $\boldsymbol{C_w}$ is pruned (details described below). A new set of related peaks (**2° peaks**) is selected and added to $\boldsymbol{S_p}$ (details described below) and associated words are placed in $\boldsymbol{C_w}$ and then pruned. This process is repeated with extension to **3° peaks** and associated words.

Pruning of $\boldsymbol{C_w}$ eliminates words that do not provide a significant contribution to the region. Since a region may contain a few or many copies of a pattern, the measure of significance varies from region to region. Thus, this algorithm varies the cut-off threshold from one to four occurrences for elimination of words from $\boldsymbol{C_w}$, i.e. occurrences of at least $\boldsymbol{T_1} = 5$ are not pruned. Begin by removing words having one occurrence and successively increase the cut-off up to $\boldsymbol{T_1} - 1 = 4$ occurrences. Stop before 70% of the words added as a result of the new peaks are eliminated. Furthermore, each word having at least $T_1 = 5$ occurrences remains in C_w. For an explanation of the thresholds, see Section 4.3.4.

New peaks are selected composed of positions associated with words that intersect C_w. A new peak has at least $T_1 = 5$ positions associated with words that intersect C_w.

To ensure all significant words are included, all words and positions associated with 1°, 2° and 3° peaks are added to S_w and pruned according to the C_w pruning rules described. In most instances, S_w and C_w are identical.

Definition: The left and right position edges, S_e, are the leftmost and rightmost positions in S_w.

Problem: Given the list of significant periodicities, S_p, the list of significant words, S_w and the edges associated with S_w, S_e, construct R_p; the positions of S_w in $\boldsymbol{R}$. Note: R_p must represent the core of $\boldsymbol{R}$ but, does not need to span all positions in $\boldsymbol{R}$. For optimization, ensure that most positions either to the left or right are included.

Solution: This process is composed of an expansion phase and a pruning phase. To begin the expansion phase, add to R_p all positions within S_e which correspond to a word in S_w. Extend R_p to the left of S_e by following the distance in $\boldsymbol{D}$ associated with the leftmost position of each $\boldsymbol{w}$ in S_w such that the maximum distance from the current edge is the maximum of (1) 5, (2) "largest peak in $\boldsymbol{S_p}$" and (3) 3×"strongest peak in $\boldsymbol{S_p}$," i.e. the peak having the most number of positions, |peak|. $\boldsymbol{R_p}$ is extended to the right by scanning each position beyond the current edge until the maximum distance is reached but, to optimize the algorithm, the current implementation does not perform this extension.

The pruning phase has two parts. In the first part, the edges are tested for islands. An island is a collection of one or more words such that all positions associated with the collection do not overlap words and positions associated with the rest of the region. Words and positions associated with islands are pruned from $\boldsymbol{R_p}$ and $\boldsymbol{S_w}$.

The second part of the pruning phase eliminates positions that conflict with previous, identified regions. This process may split $\boldsymbol{R_p}$ into multiple regions. Each region is processed as an independent region.

Definition: The region edges, $\boldsymbol{R_e}$, are the leftmost and rightmost positions in $\boldsymbol{R_p}$.

Definition: The region periodicity signature is the list of significant periodicities, $\boldsymbol{S_p}$.

4.4.5. Determine a region's base periodicity

Problem: Given $\boldsymbol{D}$ and $\boldsymbol{g_e}$, construct a distance histogram, $\boldsymbol{H_g}$, for all $\boldsymbol{d_i}$ in $\boldsymbol{D}$ for $\boldsymbol{i}$ in $\boldsymbol{g_e}$.

Solution: For each $\boldsymbol{d_i}$ in $\boldsymbol{D}$ starting at the left edge of $\boldsymbol{g_e}$ and ending at the right edge of $\boldsymbol{g_e}$, add $\boldsymbol{i}$ to the $\boldsymbol{d_i}$ bin in $\boldsymbol{H_g}$.

Problem: Given $\boldsymbol{D}$ and $\boldsymbol{R_e}$, construct a distance histogram, $\boldsymbol{H_e}$, for all $\boldsymbol{d_i}$ in $\boldsymbol{D}$ for $\boldsymbol{i}$ in $\boldsymbol{R_e}$.

Solution: For each $\boldsymbol{d_i}$ in ***D*** starting at the left edge of $\boldsymbol{R_e}$ and ending at the right edge of $\boldsymbol{R_e}$, add ***i*** to the $\boldsymbol{d_i}$ bin in $\boldsymbol{H_e}$.

Problem: Given ***D*** and $\boldsymbol{R_p}$, construct a distance histogram, $\boldsymbol{H_p}$, for all $\boldsymbol{d_i}$ in ***D*** for ***i*** in $\boldsymbol{R_p}$.

Solution: For each ***i*** in $\boldsymbol{R_p}$ and referencing $\boldsymbol{d_i}$ in ***D***, add ***i*** to the $\boldsymbol{d_i}$ bin in $\boldsymbol{H_p}$.

Problem (select initial base periodicity): Given the selected periodicity, ***p***, and the distance histogram, $\boldsymbol{H_e}$, determine whether the selected periodicity, ***p***, or a smaller periodicity, $\boldsymbol{p'} = \boldsymbol{m} \times \boldsymbol{p}$ where ***m*** is an integer, is the best initial base periodicity, $\boldsymbol{\bar{b}}$. In particular, look for a peak, $\boldsymbol{p'}$, in $\boldsymbol{H_e}$, $\boldsymbol{p'} < \boldsymbol{p}$, such that $|\boldsymbol{p'}| \geq (0.10 \times |\boldsymbol{p}|)$ and $(\boldsymbol{p} - \boldsymbol{h}) \leq (\boldsymbol{m} \times \boldsymbol{p'}) \leq (\boldsymbol{p} + \boldsymbol{h})$, for an integer ***m***, $2 \leq \boldsymbol{m} \leq 4$, and a halo of $\boldsymbol{h} = 3$.

Solution: Begin with the assumption that the periodicity, ***p***, represents the best periodicity, $\boldsymbol{\bar{b}} = \boldsymbol{p}$. Correct for periodicity across multiple copies by checking the peaks, $\boldsymbol{p'}$, in $\boldsymbol{H_e}$ starting with the smallest peak that satisfies $\boldsymbol{p'} < \boldsymbol{\bar{b}}$, and testing whether $\boldsymbol{p'}$ satisfies the criteria $|\boldsymbol{p'}| \geq (0.10 \times |\boldsymbol{\bar{b}}|)$ and $(\boldsymbol{\bar{b}} - \boldsymbol{h}) \leq (\boldsymbol{m} \times \boldsymbol{p'}) \leq (\boldsymbol{\bar{b}} + \boldsymbol{h})$ for an integer ***m***, $2 \leq \boldsymbol{m} \leq 4$ and a halo of $\boldsymbol{h} = 3$. If $\boldsymbol{p'}$ is found that satisfies the criteria then set $\boldsymbol{\bar{b}} = \boldsymbol{p'}$. Otherwise continue testing successively largest peaks, $\boldsymbol{p'}$, in $\boldsymbol{H_e}$, $\boldsymbol{p'} < \boldsymbol{\bar{b}}$. For an explanation of the thresholds, see Section 4.3.5.

Problem (adjust for copy length variation): Given the initial base periodicity, $\boldsymbol{\bar{b}}$, and the distance histogram, $\boldsymbol{H_p}$, construct a set of periodicities proximal to $\boldsymbol{\bar{b}}$, $\boldsymbol{R_b}$, for the region, $\boldsymbol{R}$, and designate the periodicity having the most occurrences in $\boldsymbol{H_p}$ as the base periodicity, ***b***.

Solution: For each peak, $\bar{\boldsymbol{p}}$, in $\boldsymbol{H_p}$, $(\bar{\boldsymbol{b}} - \boldsymbol{h}) \leq \bar{\boldsymbol{p}} \leq (\bar{\boldsymbol{b}} + \boldsymbol{h})$ and $\bar{\boldsymbol{p}} \geq \boldsymbol{T}_0$, add $\bar{\boldsymbol{p}}$ to $\boldsymbol{R_b}$. Designate the strongest peak in $\boldsymbol{R_b}$ as the base periodicity, ***b***. The default constants are $\boldsymbol{h} = 3$ and $\boldsymbol{T}_0 = 3$.

4.4.6. Test for a variable length tandem repeat (VLTR) region

Definition: $\boldsymbol{R'_e} = (\boldsymbol{R_e} - |\boldsymbol{R_e}| \times 0.20)..(\boldsymbol{R_e} + |\boldsymbol{R_e}|\ x 0.20)$.

Problem: Given the region edges, $\boldsymbol{R_e}$ and $\boldsymbol{R'_e}$, the region's collection of base periodicities, $\boldsymbol{R_b}$, and the region base periodicity, ***b***, look for SSRs having identical patterns, $\boldsymbol{R_{SSR}}$, across at least ***V*** of the region copies.

Solution: Calculate the approximate number of region copies, $\boldsymbol{c_a} = \frac{\boldsymbol{R_e}}{\boldsymbol{b}}$. Locate all SSRs within or overlapping $\boldsymbol{R'_e}$. For all SSRs having an identical pattern, $\boldsymbol{R_{SSR}}$, count the number of regions, ***r***, and verify whether $\boldsymbol{r} \geq \boldsymbol{V} \times \boldsymbol{c_a}$. The default value for ***V*** is 40%.

Problem: Given the region edges, $\boldsymbol{R'_e}$, the SSR regions, $\boldsymbol{R_{SSR}}$, within $\boldsymbol{R'_e}$ and the distance array, ***D***, construct a distance array, $\boldsymbol{D_r}$, which reflects removal of $\boldsymbol{R_{SSR}}$ from ***R***.

Solution: Construct a cancellation array, $\boldsymbol{D_c}$, for positions in $\boldsymbol{R'_e}$. Begin by initializing a factor, $\boldsymbol{f} = 0$. Then, process each position, ***i***, in $\boldsymbol{R'_e}$ from left to right, as follows. At each position, (1) if the position, ***i***, is within an $\boldsymbol{R_{SSR}}$, then decrease ***f*** by one, i.e.

$f = f - 1$, and (2) set the ith position of D_c to the current f. Next, construct another array, D_r, which reflects the removal of each R_{SSR} from D, $D_r = D + D_c$ for each position i in R'_e.

Problem: Given the distance array, D_r, and the region positions, R_p, construct a distance histogram, H_r, for all d_i in D_r for i in R_p.

Solution: For each i in R_p and referencing d_i in D_r, add i to the d_i bin in H_r.

Problem: Given the selected periodicity, p, and the distance histograms, H_p and H_r, let b_{VLTR} be the strongest peak in H_r, i.e. the number of positions in b_{VLTR}, $|b_{VLTR}|$, is greater than any other peak. Determine whether R is a potential VLTR by (1) testing whether $b_{VLTR} < p$ and (2) comparing $|b_{VLTR}|$ to the number of positions of p in H_p, $|p|$, such that $|b_{VLTR}| \geq V \times p$ thereby indicating that V of the copies have decreased in length.

Solution: Test whether $b_{VLTR} < p$ and $|b_{VLTR}| \geq V \times p$. The default value for V is 40%.

Definition: The base period, b_{VLTR}, represents the best length or strongest period of the VLTR region absent the R_{SSR}.

4.4.7. Select or construct a region pattern

Problem: Given the word array, W, and the region positions, R_p, construct a list of breakpoints, R_{BP}, which breaks the region into copies and a list of copy lengths, L.

Solution: Find the word, ***w***, in ***W*** at positions in $\boldsymbol{R_p}$ having the most occurrences. Record in $\boldsymbol{R_{BP}}$ in increasing order, the position of each occurrence of ***w*** in $\boldsymbol{R_p}$. Construct a list of copy lengths, $\boldsymbol{L}$, such that $\boldsymbol{L_i} = \boldsymbol{R_{BP\,i+1}} - \boldsymbol{R_{BP_i}} + 1$ for all consecutive pairs of positions, ***i*** and $\boldsymbol{i}+1$, in $\boldsymbol{R_{BP}}$.

Problem: Given the distance array, ***D***, the region base periodicity, ***b***, and the region breakpoints, $\boldsymbol{R_{BP}}$, construct a list, $\boldsymbol{O_c}$, of the number of occurrences of $\boldsymbol{d} = \boldsymbol{b}$ in ***D*** either at positions in a copy or at positions referring to a copy.

Solution: Construct the occurrence count list, $\boldsymbol{O_c}$, such that

$$\boldsymbol{O_{c_i}} = \sum_{j=\boldsymbol{R_{BP_i}}+1}^{\boldsymbol{R_{BP_{i+1}}}} \boldsymbol{f}\,(\boldsymbol{d_j},\boldsymbol{b}) \text{ where } \boldsymbol{f}\,(\boldsymbol{d_j},\boldsymbol{b}) = \left\{\begin{matrix} 1 \text{ for } (\boldsymbol{d_j} = \boldsymbol{b}) \\ 0 \text{ for } (\boldsymbol{d_j} \neq \boldsymbol{b}) \end{matrix}\right\} \text{ for all } \boldsymbol{i} = 1..(|\boldsymbol{R_{BP}}|-1).$$

4.4.7.1. Regions with a simple pattern structure and the base pattern for MPTR regions

Problem (select best copy): Given the base periodicity, ***b***, the region breakpoints, $\boldsymbol{R_{BP}}$, the list of copy lengths, $\boldsymbol{L}$, and the occurrence list, $\boldsymbol{O_c}$, select a best copy, $\boldsymbol{c_b}$, having edges, $\boldsymbol{c_e}$.

Solution: Use two criteria to select a best copy, $\boldsymbol{c_b}$. First, choose the copy in which the copy length is closest to ***b***, i.e. ***l*** in ***L*** such that $\boldsymbol{abs}(\boldsymbol{l} - \boldsymbol{b})$ is minimized. When more than one copy has a best copy length, choose the copy that maximizes the occurrence of $\boldsymbol{d} = \boldsymbol{b}$ in ***D*** in a copy. If a tie remains, choose any of the remaining copies.

The breakpoints, $\boldsymbol{R_{BP}}$, used to determine the copy lengths, $\boldsymbol{L}$, assume that the word, ***w***, which occurs most often in a region is the best word to split the region into copies. Since ***w*** may occur more than once in a copy, the copy lengths, ***l***, in ***L*** may not represent a complete copy. To partially compensate, combine consecutive "copies" in ***L*** and in $\boldsymbol{O_c}$ and test to determine whether the above two criteria are better satisfied with pairs of "copies." So, form $\boldsymbol{L'}$ such that $\boldsymbol{L'_i} = \boldsymbol{L_i} + \boldsymbol{L_{i+1}}$ and $\boldsymbol{O'_c}$ such that $\boldsymbol{O'_{c_i}} = \boldsymbol{O_{c_i}} + \boldsymbol{O_{c_{i+1}}}$ and perform the above using $\boldsymbol{L'}$ and $\boldsymbol{O'_c}$. Choose the best copy using both the single copy and the paired copy procedure. Back calculate to determine the breakpoints in $\boldsymbol{R_{BP}}$ used, $\boldsymbol{R_{BP_L}}$ and $\boldsymbol{R_{BP_R}}$ and define $\boldsymbol{c_e}$ as $\boldsymbol{R_{BP_L}} + 1$ and $\boldsymbol{R_{BP_R}}$.

4.4.7.2. Regions with a VLTR pattern structure

Problem: Given the DNA sequence, ***S***, distance array, ***D***, the edges of the group of proximal positions, $\boldsymbol{g_e}$, the base periodicity of the non-SSR portion, $\boldsymbol{b_{VLTR}}$, and the locations, ***L***, of $\boldsymbol{R_{SSR}}$, occurrences, select a best copy, $\boldsymbol{c_b}$, having edges $\boldsymbol{c_e}$ which abuts but does not overlap an occurrence of $\boldsymbol{R_{SSR}}$.

Solution: Best copies are selected from sequence located between locations ***l*** in ***L***, i.e. positions $\boldsymbol{l_{i_R}}..\boldsymbol{l_{i+1_L}}$ such that the copy length is $\boldsymbol{c_l} = \boldsymbol{l_{i+1_L}} - \boldsymbol{l_{i_R}} + 1$ for $\boldsymbol{i} = 1$ to $|\boldsymbol{L}| - 1$. I use three criteria to select a best copy, $\boldsymbol{c_b}$. First, choose the copy in which $\boldsymbol{c_l}$ is closest to $\boldsymbol{b_{VLTR}}$, i.e. minimize $\text{abs}(\boldsymbol{c_l} - \boldsymbol{b_{VLTR}})$. Second, when more than one copy has a best copy length, choose the copy that occurs within $\boldsymbol{g_e}$. Third, when more than one copy remains, choose the copy that maximizes the occurrence of any individual ***d*** in the copy. If a tie remains, choose any of the remaining copies. So, $\boldsymbol{c_e}$ represents the selected $\boldsymbol{c_l}$ and $\boldsymbol{c_b}$ is constructed of positions $\boldsymbol{c_e}$ on $\boldsymbol{S}$.

Problem: Given the best copy, $\boldsymbol{c_b}$, and the pattern of the SSR, $\boldsymbol{p_{SSR}}$, construct a pattern, $\boldsymbol{p_{VLTR}}$ for the VLTR region.

Solution: A VLTR pattern is a complex pattern structure composed of a nested TR having the pattern $\boldsymbol{p_{SSR}}$. Place $\boldsymbol{p_{SSR}}$ is a closure, represented as $\{\boldsymbol{p_{SSR}}\}_*$. The pattern, $\boldsymbol{p_{VLTR}}$, is a concatenation of $\boldsymbol{c_b}$ and $\{\boldsymbol{p_{SSR}}\}_*$ and since we are looking for tandem occurrences of $\boldsymbol{p_{VLTR}}$, everything is placed in a closure. Thus, $\boldsymbol{p_{VLTR}} = \boldsymbol{c_b}\{\boldsymbol{p_{SSR}}\}_*$ and the VLTR region is determined using $\{\boldsymbol{c_b}\{\boldsymbol{p_{SSR}}\}_*\}_*$.

Problem: Given a pattern, ***P***, construct a list of pattern component sizes, $\boldsymbol{P_S}$, containing the length of each component in ***P***.

Solution: Iterate through all components, $\boldsymbol{c_p}$, in ***P***. Add in order to $\boldsymbol{P_S}$ the component sizes, $|\boldsymbol{c_p}|$, such that the first item in the list corresponds to the first component in ***P***.

4.4.8. Characterize a region using the pattern

4.4.8.1. Form an initial alignment using wraparound dynamic programming algorithm with regular expressions (RegWDP)

4.4.8.1.1. Basic scoring

Problem: Given two alphabets, $\Sigma_{\boldsymbol{P}}$ for ***P*** and $\Sigma_{\boldsymbol{S}}$ for ***S***, a match score, $\boldsymbol{s_m}$, a mismatch score, $\boldsymbol{s_{mm}}$, and a gap score, $\boldsymbol{s_g}$, construct a two-dimensional matrix, $\boldsymbol{M_s}$, to represent the score for aligning a character in ***S*** to a character in ***P***.

Solution: Construct a two-dimensional matrix, $\boldsymbol{M_s}$, such that one dimension represents the possible characters in ***P***, $\{\Sigma_{\boldsymbol{P}}, \boldsymbol{gap}\}$ and the other dimension represents the possible characters in ***S***, $\{\Sigma_{\boldsymbol{S}}, \boldsymbol{gap}\}$. The element, $\boldsymbol{M_s}(\boldsymbol{i}, \boldsymbol{j})$, represents the score for aligning a character ***i*** in $\{\Sigma_{\boldsymbol{S}}, \boldsymbol{gap}\}$ with the character ***j*** in $\{\Sigma_{\boldsymbol{P}}, \boldsymbol{gap}\}$ as follows:

$$\boldsymbol{M_s}(\boldsymbol{i},\boldsymbol{j}) = \left\{\begin{array}{ll} (\boldsymbol{i} = \boldsymbol{gap}, \boldsymbol{j} = \boldsymbol{gap}) & 0 \\ (\boldsymbol{i} = \boldsymbol{gap}, \boldsymbol{j} \text{ in } \Sigma_{\boldsymbol{P}}) & \boldsymbol{s_g} \\ (\boldsymbol{i} \text{ in } \Sigma_{\boldsymbol{S}}, \boldsymbol{j} = \boldsymbol{gap}) & \boldsymbol{s_g} \\ (\boldsymbol{i} = \boldsymbol{j}), \boldsymbol{i} \text{ in } \Sigma_{\boldsymbol{S}}, \boldsymbol{j} \text{ in } \Sigma_{\boldsymbol{P}} & \boldsymbol{s_m} \\ (\boldsymbol{i} \neq \boldsymbol{j}), \boldsymbol{i} \text{ in } \Sigma_{\boldsymbol{S}}, \boldsymbol{j} \text{ in } \Sigma_{\boldsymbol{P}} & \boldsymbol{s_{mm}} \end{array}\right\}$$

The elements, $\boldsymbol{M_s}(\boldsymbol{i}, \boldsymbol{j})$, for aligning ambiguous nucleotides, {**B**, **D**, **H**, **K**, **M**, **N**, **R**, **S**, **V**, **W**, **X** and **Y**}, to one another and to the nucleotides **A**, **C**, **G** and **T** are modified to reflect the percentage of nucleotide commonality between nucleotides. For instance, **R** represents **A** or **G**. Aligning **R** to **A** achieves a score of $0.5 \times \boldsymbol{s_m}$. Aligning **R** to itself achieves a perfect match score of s_m. Aligning **R** to **C** achieves a mismatch score of s_{mm}.

4.4.8.1.2. The scoring matrix

Definition: The ***initial sequence position*** in an alignment, S_i, is $R_{e_L} - \max(50, 5 \times b)$ where ***b*** is the region base periodicity.

Definition: The ***best score*** in an alignment is associated with a cell denoted ***B*** and a row denoted S_B.

Definition: The ***pattern length*** is denoted $|P|$.

Definition: The ***final sequence position*** in an alignment, S_f, is $\max(R_{e_R}, S_B + r)$ where R_{e_R} is the right edge of the region edges, S_B is the best score in an alignment and $r = 20 \leq |P| \leq 100$.

Problem (generic pattern with regular expressions): Given a pattern, ***P***, of length ***m*** denoted using regular expressions and a region, ***R***, of length ***n*** denoted on a sequence, ***S***, by the edges R_e, construct a scoring matrix, $T(n, m)$, such that $T(i, j)$ represents the edit distance between the positions on the two strings ***S***, $S_i..i$, and ***P***, $1..j$ where $S_i \leq i \leq S_f$ and $1 \leq j \leq m$. Each cell in the matrix, $T(i, j)$, contains four items: a *score* indicating the cumulative alignment score, a *position indicator* indicating the dependency rule used to generate the score, a *column* which together with the position indicates the previous cell in the alignment and an *optional backtrack label* indicating cells previously visited during the backtracking phase.

Solution: Compute the ***i***th row of ***T*** for $\boldsymbol{j} = 1..\boldsymbol{m}$ using a one-pass wraparound dynamic programming algorithm modified to align a regular-expression based pattern to a sequence. Regular expressions describe patterns using concatenation, closure and union. Traditional dynamic programming (DP) algorithms handle concatenation of characters in the pattern and sequence. Wraparound dynamic programming (WDP) algorithms enable the pattern to be enclosed by a single closure component. This algorithm allows unlimited nesting of closure and union components using a closure definition modified to allow partial matches of the enclosed sequence. In addition, the identification algorithm generates MPTR patterns that use a union component that is always enclosed by a closure component, the only usage of a union component.

This algorithm calculates cell scores using one or more sets of cell dependencies. All cells require the standard neighbor cell dependencies.

$$\left.\begin{array}{l} \boldsymbol{T}(\boldsymbol{i}-1, \boldsymbol{j}-1) + \boldsymbol{M_s}(\boldsymbol{i}, \boldsymbol{j}) \\ \boldsymbol{T}(\boldsymbol{i}-1, \boldsymbol{j}) + \boldsymbol{s_g} \\ \boldsymbol{T}(\boldsymbol{i}, \boldsymbol{j}-1) + \boldsymbol{s_g} \end{array}\right\} \begin{array}{l} \text{Three neighbor dependencies} \\ \text{(concatenation rules)} \end{array}$$

The use of closure and union components in the pattern requires expansion of the three neighbor cell dependencies to include dependencies on the previous component, $\boldsymbol{p}$,

$$\left.\begin{array}{l} \boldsymbol{T}(\boldsymbol{i}-1, \boldsymbol{p_R}) + \boldsymbol{M_s}(\boldsymbol{i}, \boldsymbol{j}) \\[1em] \boldsymbol{T}(\boldsymbol{i}, \boldsymbol{p_R}) + \boldsymbol{s_g} \end{array}\right\} \begin{array}{l} \text{Previous component neighbor dependencies} \\[1em] \text{(extended concatenation rules)} \end{array}$$

where $\boldsymbol{p_R}$ is the last position of the previous component and $\boldsymbol{p_R} < \boldsymbol{j}$ such that the $\boldsymbol{p_R}$ dependencies are equivalent to the $\boldsymbol{j}-1$ neighbor cell dependencies.

A union component contains several alternative patterns. The score of each alternative is independent of the score of the other alternatives. Thus, the first position of each alternative is dependent on the last position of the component prior to the union component and the last position of each alternative represents the last position of the union component. This implementation uses a shadow position to represent the last position of the union component. After scoring all the alternatives, the shadow position points to the best score present in the last position of an alternative.

A closure component allows the sequence to align to the pattern in the component, zero or more times. Each position in the component is subject to the three neighbor dependencies. Also, the first position in the component is subject to the previous component neighbor dependencies. Furthermore, all positions within a closure component are subject to wraparound dependencies.

This algorithm generalizes the wraparound portion of the WDP algorithm to allow unlimited nesting of closure components. For each closure component, $\boldsymbol{c}$, in which a cell is nested, a cell is dependent on the last position of the component, $\boldsymbol{c_R}$, of the previous row with a gap penalty based on distance from the component's first position, $\boldsymbol{c_L}$, to the current position, $\boldsymbol{i}$.

$$\boldsymbol{T}(\boldsymbol{i}-1,\boldsymbol{c_R})+(\boldsymbol{j}-\boldsymbol{c_L})\times \boldsymbol{s_g}+\boldsymbol{M_s}(\boldsymbol{i},\boldsymbol{j}) \quad \Big\} \quad \text{Wraparound dependencies (closure rules)}$$

This implementation uses a stack data structure to realize unlimited nesting of components. A stack is necessary for wraparound dependencies and for previous component dependencies when more than one nesting depth is in use. For simplicity, the stacks record all nesting depths for wraparound dependencies and previous component dependencies.

The wraparound dependency stack works in the following manner. Before scoring the first position in a closure component, a new item is added to the top of the wrap stack. After scoring the last position in the closure component, the item is removed from the top of the stack. The item contains the position and score of the wraparound dependency, the current number of gaps incurred and optimization variables. The number of gaps is a counter that is initialized to zero and incremented after scoring each position.

The previous component stack works in a similar though slightly different manner. Before scoring a row, an item is added to the stack indicating no previous component scores. When a closure or union component occurs in the pattern, a new item is added to the top of the stack initialized with the information from the item directly below it. After scoring the component, the last position of the component updates the item directly below it and the item is removed from the top of the stack. The item placed on the stack contains the score and position for the last position in the previous component for both the current row and the previous row. Since union components have multiple alternatives, each capable of representing the best last position, the score and position is stored independently for the current row and for the previous row. In addition, optimization variables are stored in the stack.

Definition: A ***tandem repeat pattern*** uses the standard regular expression notation of concatenation, closure and union but uses a meaning slightly different than the standard regular expression meaning. Closure components in tandem repeat patterns allow partial pattern matches upon entering and exiting the component with a match beginning and ending at any position in the component. Union components represent alternative copy patterns associated with multi-periodic tandem repeats and are always enclosed within a closure component. The combined union/closure component forces each copy pattern to align as a unit to the sequence yet allows each sequence copy to align to the best copy pattern. In addition, the order of alternatives in a union component is the expected order of copies in the sequence. Thus, several identical alternatives are allowed in a union component.

Problem (tandem repeat patterns with regular expressions): Given a tandem repeat pattern, $\boldsymbol{P}$, of length $\boldsymbol{m}$ and a region, $\boldsymbol{R}$, of length $\boldsymbol{n}$ denoted on a sequence, $\boldsymbol{S}$, by the edges $\boldsymbol{R_e}$, construct a scoring matrix, $\boldsymbol{T(n,m)}$, such that $\boldsymbol{T(i,j)}$ represents the edit distance between the positions on the two strings $\boldsymbol{S}$, $\boldsymbol{S_i..i}$, and $\boldsymbol{P}$, $1..\boldsymbol{j}$ where $\boldsymbol{S_i} \le \boldsymbol{i} \le \boldsymbol{S_f}$ and $1 \le \boldsymbol{j} \le \boldsymbol{m}$. Enable partial pattern matches between closure components in the pattern and the sequence. Also, enable preferential alignment to the next copy pattern when identical alternatives exist for the best copy pattern.

Solution: Apply the solution for *generic pattern with regular expressions*. In addition, enable partial pattern matches in closure components and preferential alignment to the next copy pattern when identical alternatives exist.

A partial pattern match allows alignment to begin and end at any position in the closure component without incurring a gap penalty. This implementation accomplishes partial matching at the beginning of a closure component by having each position in the closure component depend upon the last position of the previous component. Likewise, each position in the closure component and the last position of the previous component can represent the last position of the closure component. This implementation uses a shadow position to represent the last position of the closure component. After scoring all positions in the component, the shadow position points to the best score present in that row of the component.

A second portion of the partial pattern match procedure is the introduction of a use fee for aligning to a closure component. This discourages short partial matches with poor alignment. This implementation charges $s_g + (0.5 \times s_m)$ upon entering a closure component. Intuitively, a match within a closure is before than a gap outside a closure, but not by much. Skipping across the closure component incurs no charge.

Preferential alignment to the next copy pattern within a union component enables choosing the next copy pattern over other identical alternatives, all of which represent the best copy pattern. This implementation uses a small next copy pattern incentive of $0.10 \times s_m$ or 10% of a match score. This means that the score must improve by a match score every `10` positions in a region. A new dependency adds this incentive to the last position of the previous alternative as a transition into another alternative.

4.4.8.1.3. Backtracking through the alignment

Problem (backtracking through an alignment): Given a scoring matrix, $\boldsymbol{T}$, construct an initial alignment, $\hat{\boldsymbol{A}}$, between a pattern, $\boldsymbol{P}$, and some or all of a sequence, $\boldsymbol{S}$. The alignment is an ordered list based on the linear order of $\boldsymbol{S}$ aligned to $\boldsymbol{P}$. It consists of a list of items, each containing three pieces of information: the *score* for the item, *position-pairs* for the left and right edge each defined using pattern and sequence positions and a *label* designating the item as a match, a mismatch, a gap in the pattern or a gap in the sequence.

Solution: Construct an initial alignment, $\hat{\boldsymbol{A}}$, by finding a path through the scoring matrix, $\boldsymbol{T}$, from right to left through positions in the sequence, $\boldsymbol{S}$, by following the dependencies invoked during construction of $\boldsymbol{T}$. The backtracking procedure begins at the cell having the best score, $\boldsymbol{B}$, follows the path backwards through $\boldsymbol{T}$ to create an initial alignment, $\hat{\boldsymbol{A}}$ and stops when it reaches a cell containing a cumulative score of zero.

An initial alignment is valid if the path overlaps sequence positions in the initial region as designated by $\boldsymbol{R_e}$. If an initial alignment is invalid, then decrement the row containing $\boldsymbol{B}$, $\boldsymbol{S_B}$, select the best score in the row and designate it as $\boldsymbol{B}$. Re-apply the procedure for backtracking through $\boldsymbol{T}$. This process repeats until a valid alignment is identified or no additional alignments exist.

During the backtracking procedure, the cells in the path are labeled with the current path indicator. When a labeled cell is reached, the procedure stops since the path has already been traversed. A special item is placed on the end of the path indicating the position-pair and the label of the intersecting path.

4.4.8.2. Define the edges of the final alignment and construct a consensus pattern

Problem (split the alignment into copies): Given an initial alignment, $\hat{\boldsymbol{A}}$, between a pattern, ***P***, and a sequence, ***S***, split $\hat{\boldsymbol{A}}$ into copies such that each component of ***P*** is represented at most once. Place each copy in the list of copies, $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$, using the same data structure as $\hat{\boldsymbol{A}}$.

Solution: Process the list, $\hat{\boldsymbol{A}}$, in a linear manner beginning at the smallest sequence position. Construct a new list to represent a copy, $\boldsymbol{c}$, of ***P*** in ***S***. Begin by processing the first item in $\hat{\boldsymbol{A}}$ and proceed to process each subsequent item in the list. If an item spans multiple copies of ***P*** in ***S***, create multiple items, one for each copy of ***P*** the item overlaps. Replace the single item in $\hat{\boldsymbol{A}}$ with the new items. If the first pattern position of an item is larger than the last pattern position in $\boldsymbol{c}$, then add the item to $\boldsymbol{c}$. If the first pattern position of an item is smaller than the last pattern position in $\boldsymbol{c}$, add $\boldsymbol{c}$ to the end of the $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$ list and start a new copy, $\boldsymbol{c}$. Add the item to $\boldsymbol{c}$. Repeat until all items in $\hat{\boldsymbol{A}}$ are processed.

Definition: The best copy, $\boldsymbol{c_b}$, used to construct the pattern, ***P***, has edges, $\boldsymbol{c_e}$.

Definition: The initial alignment, $\hat{\boldsymbol{A}}$, has edges, $\hat{\boldsymbol{A}}_{\boldsymbol{e}}$.

Problem (define borderline and core edges): Given an initial alignment, $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$, between a pattern, ***P***, and some or all of a sequence, ***S***, the edges, $\boldsymbol{c}_{\boldsymbol{e}}$, used to construct the pattern, ***P***, and the edges, $\hat{\boldsymbol{A}}_{\boldsymbol{e}}$, denoting the edges of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$, determine two sets of initial region edges, the borderline edges, $\boldsymbol{B}_{\boldsymbol{e}}$, and the core edges, $\boldsymbol{C}_{\boldsymbol{e}}$. The four edge sets are related such that $\boldsymbol{c}_{\boldsymbol{e}}$ are the inmost edges; the core edges, $\boldsymbol{C}_{\boldsymbol{e}}$, contain the sequence region denoted by $\boldsymbol{c}_{\boldsymbol{e}}$; the borderline edges, $\boldsymbol{B}_{\boldsymbol{e}}$, contain the sequence region denoted by $\boldsymbol{C}_{\boldsymbol{e}}$; and the alignment edges, $\hat{\boldsymbol{A}}_{\boldsymbol{e}}$, contain the sequence region denoted by $\boldsymbol{B}_{\boldsymbol{e}}$.

Solution: The procedure is a two-step process. First, the analysis looks at alignment scores in $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$ associated with each component within the pattern ***P*** across all copies of ***P*** in $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$ and determines the borderline and core edges for each component. Second, the analysis combines the data from the individual components to select a best set of borderline and core edges.

For each component, perform five steps to select borderline and core edges for the component. First, for each copy in $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$, collect the total score per copy for the component. Place these scores in an ordered list, $\boldsymbol{c}_{\boldsymbol{s}}$, based on copy order. Second, smooth the scores by applying a smoothing function to create a list of smoothed scores, $\boldsymbol{c}'_{\boldsymbol{s}}$, containing one score corresponding to each score in $\boldsymbol{c}_{\boldsymbol{s}}$. The smoothing function evens out the sequence conservation level of a single copy across several copies by summing the scores across three to five scores and dividing by the number of scores in the sum, i.e. the function is

$$\boldsymbol{c}'_{\boldsymbol{s}_{\boldsymbol{i}}} = \frac{\sum_{\boldsymbol{j}=\boldsymbol{i}-2}^{\boldsymbol{i}+2} \boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{j}}}}{\#\,\text{of } \boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{j}}} \text{ in sum}} \text{ for all } 1 \le \boldsymbol{j} \le \left(\#\,\text{in } \boldsymbol{c}_{\boldsymbol{s}}\right).$$

Third, determine the largest score, $\boldsymbol{s}$, in $\boldsymbol{c}'_{\boldsymbol{s}}$. If the largest score occurs only once, then assign the second largest score to $\boldsymbol{s}$. Calculate a threshold of half the height between a base score of zero and $\boldsymbol{s}$, $\boldsymbol{T}_2 = \boldsymbol{s}/2$. This is a common smoothing approach that seems to work here. Finally, define the borderline and core edges for the component. Begin with the copy in $\boldsymbol{c}_{\boldsymbol{s}}$ associated with the edges $\boldsymbol{c}_{\boldsymbol{e}}$. Expand outward in $\boldsymbol{c}_{\boldsymbol{s}}$ to the left and to the right until $\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{i}}} < \boldsymbol{T}_2$. Define the core edges, $\boldsymbol{C}_{\boldsymbol{e}}$, for the component as the outmost edges of the previous copy, i.e. the last score which satisfies $\boldsymbol{c}_{\boldsymbol{s}_{\boldsymbol{i}}} \ge \boldsymbol{T}_2$. Switch from the $\boldsymbol{c}_{\boldsymbol{s}}$ list to the $\boldsymbol{c}'_{\boldsymbol{s}}$ list and continue expanding outward until $\boldsymbol{c}'_{\boldsymbol{s}_{\boldsymbol{i}}} < \boldsymbol{T}_2$. Define the borderline edges, $\boldsymbol{B}_{\boldsymbol{e}}$, for the component as the outmost edges of the previous copy, i.e. the last score which satisfies $\boldsymbol{c}'_{\boldsymbol{s}_{\boldsymbol{i}}} \ge \boldsymbol{T}_2$.

Once the borderline edges, $\boldsymbol{B}_{\boldsymbol{e}}$, and the core edges, $\boldsymbol{C}_{\boldsymbol{e}}$, are selected for each component, select the borderline and core edges for the entire region. The region's borderline edges are the outward most edges associated with the components. The region's core edges are the inward most edges associated with the components.

Problem (columns in the alignment): Given an alignment, $\hat{\boldsymbol{A}}$ or $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$, construct a list of column lists, $\hat{\boldsymbol{A}}_{\boldsymbol{C}}$, that records the contents of a column in the alignment. Specifically, the primary list indicates the pattern position and the internal list is an ordered list of sequence positions; the sequence positions aligned to a specific position in the pattern. In addition, record gaps in the sequence as "`-1`" in the column that is skipped.

Solution: Process each item in the alignment independently. If the item is a gap in the pattern then the item is ignored. If the item is a gap in the sequence then for each pattern position in the item, add a "`-1`" to the end of the appropriate list to indicate a gap. If the item is a match or mismatch then for each sequence position in the item, place the sequence position at the end of the corresponding pattern position list.

Problem (trim an alignment): Given an alignment, ***A***, and a set of edges, ***E***, remove sequence positions in ***A*** not contained within ***E***.

Solution: Linearly process the alignment, ***A***. If the match, mismatch or gap item is completely contained within ***E*** then keep the item in ***A***. If the item partially overlaps the positions in ***E*** then split the item into two items such that one item contains the alignment overlapping the positions in ***E*** and the other item contains all other positions. If the item does not overlap the sequence positions contained within ***E*** then remove the item from ***A***.

Problem (trim a list of pattern position columns): Given a list of pattern position columns, $\hat{\boldsymbol{A}}_{\boldsymbol{C}}$, and a set of edges, ***E***, remove sequence positions in $\hat{\boldsymbol{A}}_{\boldsymbol{C}}$ not contained within ***E***.

Solution: Process each pattern position column in $\hat{A}_C$. Remove sequence positions not contained in $\boldsymbol{E}$ by linearly processing the column list. If a "`-1`" indicating a gap occurs between two sequence positions not in $\boldsymbol{E}$, then remove it. Otherwise, retain it.

Problem (consensus pattern formation): Given an alignment, $\hat{A}_C$, between a pattern, $\boldsymbol{P}$, and a sequence, $\boldsymbol{S}$, construct a consensus pattern using a two-dimensional matrix, $\hat{P}_C$, that records the nucleotide occurrence for each pattern position. Specifically, the matrix records the number of occurrences of each gap and nucleotide, {**A**, **C**, **G** and **T**}, in $\boldsymbol{S}$ that align to a specific position in $\boldsymbol{P}$ according to the $\hat{A}_C$.

Solution: The alignment, $\hat{A}_C$, is a list of pattern position columns aligning sequence positions to a particular pattern position. For each column in $\hat{A}_C$, update the column in $\hat{P}_C$ corresponding to the same pattern position. Process each sequence position in the column by accessing the nucleotide at the position in $\boldsymbol{S}$ and incrementing the appropriate nucleotide in the column of $\hat{P}_C$. Process each "`-1`" by incrementing the gap in the column of $\hat{P}_C$.

Problem (final edge definition): Given the borderline edges, B_e, the core edges, C_e, an initial alignment, $\hat{A}_R$, between a pattern, $\boldsymbol{P}$, and a sequence, $\boldsymbol{S}$, trimmed to contain only sequence positions in B_e, and a consensus pattern, $\hat{P}_C$, based on the trimmed $\hat{A}_R$, determine the final edges, $\boldsymbol{A_e}$ for the alignment and trim $\hat{\boldsymbol{A}}_{\boldsymbol{R}}$ to form the final alignment, $\boldsymbol{A_R}$.

Solution: The basic approach is to begin at the borderline edges and proceed inward to the core edges. The outward most edge adjusts inward as sequence segments on the edges prove to be too weak to retain within the region. To determine the strength of a sequence segment, the consensus pattern, $\hat{\boldsymbol{P}}_{\boldsymbol{C}}$, rescores each sequence position based on the sequence position's alignment to the pattern, $\boldsymbol{P}$. The process stops when the core edges are reached and the edges that remain represent the final alignment, $\boldsymbol{A_R}$, and the final region edges.

Begin at an outmost borderline edge, e.g. the left edge of $\boldsymbol{B_e}$, with an initial score of zero and no sequence positions in the segment. Access the next item in $\boldsymbol{A_R}$. If it is a gap in the pattern, add it to the segment. If it is a gap in the sequence, a match or a mismatch, then rescore each sequence position in the item based on the pattern position, add each sequence position individually to the segment and determine whether the total score of the segment drops below a $\boldsymbol{T}_3$ where $\boldsymbol{T}_3 = (0.2 \times \boldsymbol{s_m}) \times$"# of sequence positions in the segment." If a segment drops below $\boldsymbol{T}_3$ then remove the sequence positions in the segment from the alignment and reinitialize the segment and its total score. Continue until all sequence positions between $\boldsymbol{B_e}$ and $\boldsymbol{C_e}$ are processed. The alignment that remains is the final alignment, $\boldsymbol{A_R}$ having the edges, $\boldsymbol{A_e}$.

Definition: The list of pattern position columns, $\hat{\boldsymbol{A}}_{\boldsymbol{C}}$, trimmed to remove sequence positions not contained in the edges, $\boldsymbol{A}_{\boldsymbol{e}}$, is denoted $\boldsymbol{A}_{\boldsymbol{C}}$. The algorithm is given earlier in this section (see "*trim a list of pattern position columns*").

Definition: The consensus pattern, $\hat{\boldsymbol{P}}_{\boldsymbol{C}}$, is updated to reflect the region edges $\boldsymbol{A}_{\boldsymbol{e}}$ and the list of pattern position columns, $\boldsymbol{A}_{\boldsymbol{C}}$, and is denoted $\boldsymbol{P}_{\boldsymbol{C}}$.

4.4.8.3. Adjust and characterize the final alignment

Problem (merge copies to compact alignment): Given an alignment, $\boldsymbol{A}_{\boldsymbol{R}}$, between a pattern, ***P***, and a sequence, ***S***, merge copies in $\boldsymbol{A}_{\boldsymbol{R}}$ that corresponds to the same occurrence of the pattern ***P***.

Solution: Merge copies in $\boldsymbol{A}_{\boldsymbol{R}}$ associated with nested closure components and union/closure component combinations that represent a single occurrence within the pattern, ***P***. Merge two consecutive copies, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, in the $\boldsymbol{A}_{\boldsymbol{R}}$ list where $\boldsymbol{c}_1$ occurs in the sequence before $\boldsymbol{c}_2$, if one of the following two criteria applies. First, if the largest sequence position in $\boldsymbol{c}_1$ is smaller than the smallest sequence position in $\boldsymbol{c}_2$, then merge $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ into a single copy. Second, if $\boldsymbol{c}_1$ overlaps $\boldsymbol{c}_2$ only in a nested closure component then, merge $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ into a single copy. This second criteria requires using the pattern structure associated with ***P*** to determine whether the largest sequence position in $\boldsymbol{c}_1$ ends in a nested closure component and whether the smallest sequence position in $\boldsymbol{c}_2$ corresponds to a position in the same nested closure component.

Problem (color conserved columns in an alignment): Given the list of pattern position columns, $\boldsymbol{A_C}$, and the consensus pattern, $\boldsymbol{P_C}$, both which reflects the region edges $\boldsymbol{A_e}$, determine which pattern positions are conserved and indicate this in $\boldsymbol{P_C}$ and construct an array, $\boldsymbol{S_C}$, parallel to $\boldsymbol{S}$ that colors the conserved sequence positions.

Solution: A column associated with a pattern position is conserved if one of two criteria is satisfied. The first criteria requires a single nucleotide to exhibit conservation of at least $T_4 = 80\%$ of the items in the column. The second criteria requires $\boldsymbol{T_5} = 5$ consecutive nucleotides in the column to be identical.

Analysis of $\boldsymbol{P_C}$ determines whether a nucleotide is sufficiently conserved in the column to meet or exceed $\boldsymbol{T_4}$. For each position in $\boldsymbol{P_C}$, determine the ratio of occurrence for each nucleotide and gap using

$$\text{ratio of } \boldsymbol{j} = \frac{|\boldsymbol{i}|}{\sum\limits_{\boldsymbol{i}=\{\boldsymbol{A},\boldsymbol{C},\boldsymbol{G},\boldsymbol{T},\boldsymbol{gap}\}} |\boldsymbol{i}|} \text{ for all } \boldsymbol{j} = \{\boldsymbol{A},\boldsymbol{C},\boldsymbol{G},\boldsymbol{T},\boldsymbol{gap}\}.$$

For each nucleotide ratio, when its percentage meets of exceeds $\boldsymbol{T_4}$, declare the pattern position to be conserved and declare the nucleotide with the best nucleotide ratio as the conserved nucleotide.

Analysis of $\boldsymbol{A_C}$ determines whether sufficient consecutive nucleotides exist in a given column corresponding to a pattern position. For each column in $\boldsymbol{A_C}$, sequentially process the sequence positions. For each position, access the corresponding nucleotide in $\boldsymbol{S}$ and determine whether it matches the previous nucleotide. Keep an indicator defining the nucleotide of the most recent sequence positions and a counter indicating the number of consecutive occurrences of the nucleotide. When the counter meets or exceeds $\boldsymbol{T}_5$, declare the pattern position to be conserved and declare the nucleotide with the best nucleotide ratio as the conserved nucleotide.

For each conserved pattern position, traverse the corresponding column in $\boldsymbol{A_C}$. For each sequence position in the column, access the corresponding nucleotide in ***S***. If the nucleotide corresponds to the conserved nucleotide then color the sequence position in $\boldsymbol{S_C}$.

4.4.9. Analyze the alignment for MPTR regions

Definition: The largest period in $\boldsymbol{R_b}$ is $\max(\boldsymbol{R_b})$.

Definition: The smallest period in $\boldsymbol{R_b}$ is $\min(\boldsymbol{R_b})$.

Problem: Given the distance array, ***D***, and the region edges, $\boldsymbol{R_e}$, construct a distance histogram, $\boldsymbol{H_e}$, for all $\boldsymbol{d_i}$ in ***D*** for ***i*** in $\boldsymbol{R_e}$.

Solution: For each $\boldsymbol{d_i}$ in ***D*** starting at the left edge of $\boldsymbol{R_e}$ and ending at the right edge of $\boldsymbol{R_e}$, add ***i*** to the $\boldsymbol{d_i}$ bin in $\boldsymbol{H_e}$.

Problem: Given the distance histogram, $\boldsymbol{H_e}$, the region base periodicities, $\boldsymbol{R_b}$, and the largest and smallest region base periodicity, $\min(\boldsymbol{R_b})$ and $\max(\boldsymbol{R_b})$, construct a list, $\boldsymbol{M_R}$, containing periodicities in $\boldsymbol{H_e}$ having at least $\boldsymbol{T}_1 = 5$ occurrences which lie within $\boldsymbol{h} = 3$ of being an integer multiple of $\boldsymbol{R_b}$.

Solution: For each peak, $\boldsymbol{p}$, in $\boldsymbol{H_e}$ where $|\boldsymbol{p}| \geq \boldsymbol{T}_1$, determine the largest integer, $\boldsymbol{m}$, such that $\boldsymbol{m} \times \min(\boldsymbol{R_b}) \leq \boldsymbol{p}$. Use $\boldsymbol{m}$ to calculate $\boldsymbol{p}' = \boldsymbol{m} \times \max(\boldsymbol{R_b})$. If $\boldsymbol{p}' \geq \boldsymbol{p}$ then add $\boldsymbol{m}$ to $\boldsymbol{M_p}$. Decrement $\boldsymbol{m}$, $\boldsymbol{m} = \boldsymbol{m} - 1$ and re-test whether $\boldsymbol{p}' \geq \boldsymbol{p}$. If so, then add $\boldsymbol{m}$ to $\boldsymbol{M_p}$. Repeat until $\boldsymbol{p}' < \boldsymbol{p}$. Add the pair $(\boldsymbol{p}, \boldsymbol{M_p})$, ordered by $\boldsymbol{p}$, to $\boldsymbol{M_R}$.

Problem: Given the sequence, ***S***, the pattern, ***P*** and the list of pattern position columns, $\boldsymbol{A_C}$, construct a list of sequences, $\boldsymbol{S_C}$, one for each column in $\boldsymbol{A_C}$.

Solution: Each pattern position column in $\boldsymbol{A_C}$ corresponds to the sequence positions in order that align to the corresponding pattern position. The column contains sequence positions indicating positions in the sequence, ***S*** and "-1" indicating a gap in the sequence at this pattern position. So, for each pattern position column in $\boldsymbol{A_C}$, construct a sequence, ***C***, that preserves the order of the pattern position column. For each sequence position in the column, access the sequence, ***S***, to determine the corresponding nucleotide. For each "-1", insert a gap in the sequence, ***C***. Construct ***C*** for each pattern position column in $\boldsymbol{A_C}$.

Problem: Given the list of sequences, $\boldsymbol{S_C}$, for each column in $\boldsymbol{A_C}$, construct a series of distance arrays, ***C***, such that for each sequence, ***j***, in $\boldsymbol{S_C}$ and each window size $\boldsymbol{k} = 3,4,5,6$, there is a separate distance array, $\boldsymbol{C_{j,k}}$.

Solution: For each sequence, ***j***, in $\boldsymbol{S_C}$, $\boldsymbol{S_{C_j}}$, construct four distance arrays, one for each $\boldsymbol{k} = 3,4,5,6$, $\boldsymbol{C_{j,k}}$, using the distance array construction procedure (see Section 4.4.1).

Problem: Given the distance array, $\boldsymbol{C_{j,k}}$, construct a distance histogram, $\boldsymbol{H_{j,k}}$, for each column, ***j***, and each window size, ***k*** as given by $\boldsymbol{C_{j,k}}$.

Solution: For each column, ***j***, each window size, ***k***, and each position ***i*** in $\boldsymbol{C_{j,k}}$ containing $\boldsymbol{d_i}$, add 1 to the $\boldsymbol{d_i}$ bin in $\boldsymbol{H_{j,k}}$.

Problem: Given the distance histogram, $\boldsymbol{H_{j,k}}$, determine significant multiples, $\boldsymbol{S_M}$ for each fixed ***k***, $\boldsymbol{S_{M_k}}$.

Solution: Construct a new histogram, $\boldsymbol{H_k}$, by combining histograms for all ***j*** associated with a fixed ***k***, $\boldsymbol{H_{j,k}}$. For each peak in $\boldsymbol{H_k}$, ***p***, $|\boldsymbol{p}| \geq 3$, add ***p*** to a list of significant multiples, $\boldsymbol{S_{M_k}}$.

Problem: Given the significant multiples, $\boldsymbol{S_M}$, and the distance histogram, $\boldsymbol{H_{j,k}}$, construct an array of lists, $\boldsymbol{M_C}$. The array represents each multiple, ***m***, in $\boldsymbol{S_M}$ from a multiple of one up to the largest ***m***, in $\boldsymbol{S_M}$. Each list in the array represents each column, ***j***, which strongly exhibits a periodicity of ***m*** regardless of the window size, ***k***, used to analyze the column.

Solution: For a fixed ***k*** and each ***m***, in $\boldsymbol{S_M}$, find the largest number of occurrences, $|\boldsymbol{p_m^*}|$, associated with a column, i.e. iterate thru $\boldsymbol{H_{j,k}}$ to find the column ***j*** which maximizes $\boldsymbol{p} = \boldsymbol{m}$ in $\boldsymbol{H_{j,k}}$. Next, calculate a threshold, $\boldsymbol{h_m} = \max(3, 0.5 \times |\boldsymbol{p_m^*}|)$. Now, construct an array of lists, $\boldsymbol{M_{C_k}}$, for a fixed ***k*** such that if $|\boldsymbol{p}| \geq \boldsymbol{h_m}$ for $\boldsymbol{p} = \boldsymbol{m}$ then add the column $\boldsymbol{j}$ to the list at index $\boldsymbol{m}$ in $\boldsymbol{M_{C_k}}$. Construct $\boldsymbol{M_{C_k}}$ for each $\boldsymbol{k}$ in $\boldsymbol{H_{j,k}}$. Construct an array of lists, $\boldsymbol{M_C}$, by combining lists at the same index, $\boldsymbol{m}$, from each $\boldsymbol{M_{C_k}}$ such that a column $\boldsymbol{j}$ occurs at most once in each list in $\boldsymbol{M_C}$.

Problem: Given the significant peaks in $\boldsymbol{H_e}$ which are multiples of the region base periodicities, $\boldsymbol{R_b}$, the list of periodicity multiples, $\boldsymbol{M_R}$, and the significant columns which correspond to multi-copy periodicity, $\boldsymbol{M_C}$, construct a list of multiples, $\boldsymbol{M}$, composed of a $(\boldsymbol{m}, \boldsymbol{p})$ pair where $\boldsymbol{m}$ represents the multiple and $\boldsymbol{p}$ represents the periodicity.

Solution: For each index, $\boldsymbol{m}$, in $\boldsymbol{M_C}$ having at least one associated column, test whether a peak, $\boldsymbol{p}$, in $\boldsymbol{M_R}$ contains $\boldsymbol{m}$. If $\boldsymbol{m}$ exists in any $\boldsymbol{p}$ in $\boldsymbol{M_R}$, then place $\boldsymbol{m}$ in a list of integers, $\boldsymbol{M'}$. For each peak, $\boldsymbol{p}$, in $\boldsymbol{M_R}$, determine which multiple $\boldsymbol{m}$ in $\boldsymbol{M_{R_p}}$ is present in $\boldsymbol{M'}$ and contains the most occurrences, $\boldsymbol{m}^*$, for this $\boldsymbol{p}$. Update $(\boldsymbol{m}, \boldsymbol{p})$ in $\boldsymbol{M}$ to ensure that the $\boldsymbol{p}$ having the largest $\boldsymbol{m}^*$ is present.

4.4.9.4. Construct MPTR patterns

Problem: Given the alignment, $\boldsymbol{A}$, the distance array, $\boldsymbol{D}$, and the multiples, $\boldsymbol{M}$, for each $(\boldsymbol{m}, \boldsymbol{p})$ pair in $\boldsymbol{M}$, select the best $\boldsymbol{m}$ consecutive copies, $\boldsymbol{c_m}$, in $\boldsymbol{A}$ based on distances in $\boldsymbol{D}$ having the most occurrences of $\boldsymbol{p}$.

Solution: For each $(\boldsymbol{m}, \boldsymbol{p})$ pair in $\boldsymbol{M}$, select the best $\boldsymbol{m}$ consecutive copies in $\boldsymbol{A}$ using three criteria. First, choose the $\boldsymbol{m}$ consecutive copies in $\boldsymbol{A}$ in which the sequence length, $\boldsymbol{l}$, across the copies is closest to p, i.e. $\mathrm{abs}(-p)$. Second, when more than one set of copies has the best l, choose the set that maximizes the occurrence of $d = p$ in D for those sequence positions in the set. Third, when more than one set remains, choose the set that maximizes the number of conserved sequence positions associated with conserved columns in A. If a tie remains, choose any of the remaining sets to designate as c_m for this m.

Problem: Given the sequence, S, the base pattern, P, the alignment of P to S, A, the multiples, M, and the sets of best copies for multiples in M, c_m, construct a pattern, P_m, for each multiple, m, in M.

Solution: For each (m, p) pair in M, construct a union, P_m, composed of m copies of P such that each P is a separate alternative, e.g. for $m = 2$, construct $[P \mid P]$. Using columns in the mth index of M_C, M_{C_m}, use each column j in M_{C_m} to modify each P in P_m such that the jth column of each copy in c_m replaces the jth column of the corresponding copy in P_m. This replace operation allows gaps from c_m to replace nucleotides from P.

Problem: Given the sequence, S, the pattern P_m and the region edges, A_e, associated with the alignment, A, characterize the region by forming an alignment, A_m, between P_m and S proximal to the region denoted by the edges A_e.

Solution: Use the standard alignment procedure (see Section 4.4.8) to align P_m to S.

4.4.10. Sequences having more than one tandem repeat region

Problem: For each tandem repeat region identified using the above analysis, store the information for each region; the region edges, $\boldsymbol{R_e}$, the region base periodicities, $\boldsymbol{R_b}$, the region periodicity signature, $\boldsymbol{S_P}$, the region multiples and associated periodicities, $\boldsymbol{M}$, and for each region characterization the list of pattern component sizes, $\boldsymbol{P_S}$. Sequentially assign each region a region number, $\boldsymbol{N_R}$. Use individual lists to store information such that the position in the list corresponds to the region number. The individual lists store the location on the sequence, $\boldsymbol{R_e^\circ}$, the base periods, $\boldsymbol{R_b^\circ}$, the periodicity signatures, $\boldsymbol{S_P^\circ}$, the multiples and associated periods, $\boldsymbol{M}^\circ$, and the pattern component sizes, $\boldsymbol{P_S^\circ}$.

Solution: Initialize $\boldsymbol{N_R} = 0$. For each tandem repeat region identified, increment $\boldsymbol{N_R}$ by 1, $\boldsymbol{N_R} = \boldsymbol{N_R} + 1$. Add R_e to the end of the list $\boldsymbol{R_e^\circ}$. Add $\boldsymbol{R_b}$ to the end of the list $\boldsymbol{R_b^\circ}$. Add $\boldsymbol{S_P}$ to the end of $\boldsymbol{S_P^\circ}$. Add $\boldsymbol{M}$ to the end of the list $\boldsymbol{M}^\circ$. Add $\boldsymbol{P_S}$ to the end of $\boldsymbol{P_S^\circ}$.

Problem: Given the region edges list, $\boldsymbol{R_e^\circ}$, and a potential tandem repeat region defined by either the group of proximal positions, $\boldsymbol{g}$ in $\boldsymbol{G}$ bounded by the edges $\boldsymbol{g_e}$, or the more refined region denoted by positions, $\boldsymbol{R_p}$ and bounded by the edges $\boldsymbol{R_e}$, ensure that the region does not overlap previously identified regions bounded by the edges in $\boldsymbol{R_e^\circ}$. Specifically, remove positions from the potential region that overlap previously identified regions by adjusting the proximal positions, $\boldsymbol{g}$ and $\boldsymbol{g_e}$, or the more refined region, $\boldsymbol{R_p}$ and $\boldsymbol{R_e}$.

Solution: For each potential tandem repeat region, restrict initial region analysis to positions not present in identified tandem repeat regions. Perform the analysis twice, during formation of a group of proximal positions, $\boldsymbol{g}$ in $\boldsymbol{G}$, and once $\boldsymbol{g}$ is expanded to a more refined region defined by the positions $\boldsymbol{R_p}$ and bounded by $\boldsymbol{R_e}$. Remove positions in $\boldsymbol{g}$ or $\boldsymbol{R_p}$ that overlap identified tandem repeat regions bounded by $\boldsymbol{R_e^i}$ in $\boldsymbol{R_e^\circ}$.

Begin search at end of $\boldsymbol{R_e^\circ}$ and proceed forward through the list until the region edges are to the left of the current region. For each tandem repeat region bounded by the edges, $\boldsymbol{R_e^i}$ in $\boldsymbol{R_e^\circ}$, test whether the identified region overlaps the current potential region. If $\boldsymbol{g_{e_L}} < \boldsymbol{R^i_{e_L}} \le \boldsymbol{g_{e_R}}$ then remove the positions $\boldsymbol{g_{e_L}} .. \boldsymbol{R^i_{e_L}}$ from $\boldsymbol{g_e}$. If $\boldsymbol{g_{e_L}} \le \boldsymbol{R^i_{e_R}} < \boldsymbol{g_{e_R}}$, then remove positions $\boldsymbol{R^i_{e_R}} .. \boldsymbol{g_{e_R}}$ from $\boldsymbol{g_e}$. If $\boldsymbol{R^i_{e_L}} \le \boldsymbol{g_{e_L}} < \boldsymbol{g_{e_R}} \le \boldsymbol{R^i_{e_R}}$, then remove all positions from $\boldsymbol{g_e}$. The same overlap conditions apply for the larger region bounded by $\boldsymbol{R_e}$, i.e. substitute $\boldsymbol{R_e}$ for $\boldsymbol{g_e}$.

Problem: Given the region edges, $\boldsymbol{R_e}$, the region periodicity signature, $\boldsymbol{S_p}$, the region base periodicities, $\boldsymbol{R_b}$, the region multiples and associated periodicities, $\boldsymbol{M}$, and the distance histograms, $\boldsymbol{H}$ and $\boldsymbol{F}$, remove positions from the distance histograms that represent significant periodicities associated with this region.

Solution: Restrict removal of positions to the region bounded by $\boldsymbol{R_e}$. For each periodicity in the periodicity signature, $\boldsymbol{S_p}$, remove the positions in ***H*** and ***F*** bounded by $\boldsymbol{R_e}$. For each base periodicity in $\boldsymbol{R_b}$, remove the positions in ***H*** and ***F*** bounded by $\boldsymbol{R_e}$. For each periodicity in ***M***, remove the positions in ***H*** and ***F*** bounded by $\boldsymbol{R_e}$.

4.4.11. Long sequences

Problem: Given a data file containing the sequence, ***S***, a block size, ***B***, a block overlap size, $\boldsymbol{B_O}$, and a block halo number, $\boldsymbol{B_N}$, construct a list of sequence blocks, $\boldsymbol{S_B}$, and process the sequence as a series of blocks. Initial analysis occurs within a primary block that is flanked by up to $\boldsymbol{B_N}$ secondary blocks to either side of the primary block. Identify initial tandem repeat regions within the primary block. Identify the final region within the primary and secondary blocks. ***B***, $\boldsymbol{B_O}$ and $\boldsymbol{B_N}$ are user-defined variables.

Solution: Process the sequence, ***S***, and store it as sequence blocks in $\boldsymbol{S_B}$. Intersperse construction of $\boldsymbol{S_B}$ with analysis of $\boldsymbol{S_B}$ and identification of tandem repeat regions and regions of similarity. Previous sections describe this analysis and region identification. Here, the construction of $\boldsymbol{S_B}$ is described.

The $\boldsymbol{S_B}$ data structure is a list of at most $(2\times\boldsymbol{B_N})+1$ blocks, each of length ***B***, representing a primary block flanked on each side by $\boldsymbol{B_N}$ secondary blocks. Initialize $\boldsymbol{S_B}$ with $\boldsymbol{B_N}+1$ blocks. Each item in the list is an array demarked with an initial and final sequence position. Adjacent blocks contain a short duplicated sequence representing a block overlap of $\boldsymbol{B_O}$. The block beginning at position one represents the first primary block. There are $\boldsymbol{B_N}$ secondary blocks to the right. There are no secondary blocks to the left, since no sequence exists to the left.

Analysis of the primary sequence block creates the distance array, $\boldsymbol{D}$, the word array, $\boldsymbol{W}$, and the distance histograms, $\boldsymbol{H}$ and $\boldsymbol{F}$ (see Section 4.4.1). The distance array relies on the word occurrence array, $\boldsymbol{O}$, that does carry a “memory” across from one primary sequence block to another (see Section 4.4.1). Potential tandem repeat regions and initial region definitions utilize information from the primary sequence block (see Section 4.4.4 through Section 4.4.7). Region characterization uses both the primary and the secondary blocks (see Section 4.4.8) and the MPTR analysis of the region alignment uses sequence that may be present in primary and secondary blocks (see Section 4.4.9).

Once all regions associated with the primary block are analyzed, “increment” the sequence block system by shifting the primary block forward to the next block, and if necessary, removing the $(\boldsymbol{B_N} + 1)$ th block to the left of the primary block and reading a new block from the sequence data file. Continue processing the sequence blocks until the primary block containing the last sequence position is processed.

The current implementation uses the following default values: $\boldsymbol{B} = 500$ nucleotides, $\boldsymbol{B_O} = 50$ nucleotides and $\boldsymbol{B_N} = 20$.

4.4.12. Data file output from identification algorithm to user interface

4.4.12.1. Tandem repeat regions (all regions)

Problem (The region tracking file): Given the list of region edges, $\boldsymbol{R}_{\boldsymbol{e}}^{\circ}$, the list of pattern component sizes, $\boldsymbol{P}_{\boldsymbol{S}}^{\circ}$, the list of region periodicity signatures, $\boldsymbol{S}_{\boldsymbol{P}}^{\circ}$, and multiples and associated periods, $\boldsymbol{M}^{\circ}$, output a data file for each sequence analyzed containing details of all non-SSR tandem repeat regions identified in the region. Information in this file must be sufficient to track all other data files generated for individual regions and region characterizations.

Solution: The regions output file (*.regions) is the primary file for tracking all major regions identified in a sequence. The creation of the file occurs after all regions have been identified. The file does not specify SSR regions but does record all other identified, tandem repeat regions. In particular, the first line of the file specifies the order to process the regions and the remaining lines represent the non-SSR regions.

The identification algorithm assigns a unique numerical identifier, $\boldsymbol{N}_{\boldsymbol{R}}$, to each region (see Section 4.4.10). Specifically, the identifier for the first region is 1, the second region, 2, and so forth. The order of identification dictates the numeral assigned to each region and may or may not correspond to the actual order of occurrence in the sequence. Thus, the first line of the file re-orders the regions according to the order of occurrence in the sequence without reassigning new numeric identifiers to the regions themselves. The remaining lines list the regions in the order of occurrence in the sequence. The files that accompany the region are available via the numerals listed in the first line.

Region Tracking File Format (*.regions)

ListOfRegions (1 line): reorders region numeric identifiers according to sequence occurrence

Regions (series of paired lines): ordered according to sequence occurrence

ListOfRegions

<# list length> <ordered list of regions>

Regions

VLTRRegion OR
MPTRRegion OR
Region

VLTRRegion

"V" 1<tab>RegionRange
<# of components><tab> <list of component sizes>

MPTRRegion

"M" <# of periods><tab>RegionRange
<# of periods><tab><list of periods>
<# of periods><tab><list of multiples>

Region

"X" "1"<tab>RegionRange
1 <period>

RegionRange

<# begin position> <# end position>

Region Tracking File: Abbreviated Example

```
7    1 2 3 4 5 6 7
X 1  219  277
1 11
V 1  309  703
2 24 2
M 4  4203 6647
4 1 2 4 5
4 135 270 540 675
X 1  11876     11935
1 27
X 1  12258     12327
1 21
X 1  12453     12839
1 48
X 1  12865     12904
1 9
…
```

Problem (Tandem repeat region table file): Given the list of perfect SSR regions, $\boldsymbol{R}^{\circ}_{\boldsymbol{SSR}}$, the list of imperfect SSR regions, $\boldsymbol{R}^{\circ}_{\boldsymbol{NSSR}}$, the list of region edges, $\boldsymbol{R}^{\circ}_{\boldsymbol{e}}$, the list of pattern component sizes, $\boldsymbol{P}^{\circ}_{\boldsymbol{S}}$, the list of region periodicity signatures, $\boldsymbol{S}^{\circ}_{\boldsymbol{P}}$, and the multiples and associated periods, $\boldsymbol{M}^{\circ}$. For each sequence analyzed, output a data file containing details of all tandem repeat regions identified in the region. Include all perfect and imperfect SSR regions, VLTR regions, MPTR regions and simple pattern structure regions. This file format is similar to the regions tracking file format except it contains perfect and imperfect SSRs and allows containment of an SSR within other SSR.

Solution: The tandem repeat regions table file (*.tr.table) contains all identified regions including SSR regions. The file format is similar to the region tracking file format (see previous problem and solution in this section) in that the first line specifies the order to process the regions and the remaining lines represent the regions. This file contains several additional region types; “S” for perfect SSR regions, “N” for imperfect SSR regions and “C” for region containment of perfect SSR regions inside imperfect SSR regions. The “C” designation enables individual regions to be clustered into one group for display and other purposes. Each cluster is designated by only one region in the first line of the file.

The identification algorithm assigns a unique numerical identifier to each non-SSR region but does not assign numerals to the perfect and imperfect SSR regions. Thus, a “0” in the list on the first line of the file indicates an SSR region. The format of an SSR region is given after the example.

Tandem Repeat Region Table File Format

ListOfRegions (1 line): reorders region numeric identifiers by sequence occurrence.

Regions (series of single & paired lines): ordered by sequence occurrence

ListOfRegions

<# list length> <ordered list of regions>

Regions

MPTRRegion OR

VLTRRegion OR

PerfectSSRRegion OR

ImperfectSSRRegion OR

Region OR

ContinueRegion

ContinueRegion

"C"

RegionRange

<# begin position> <# end position>

MPTRRegion

"M" <# of periods><tab>RegionRange

<# of periods><tab><list of periods>

<# of periods><tab><list of multiples>

VLTRRegion

"V" 1<tab>RegionRange

<# of components><tab> <list of component sizes>

PerfectSSRRegion

"S" 1<tab>RegionRange

"1" <# pattern length><tab><sequence>

ImperfectSSRRegion

"N" 1<tab>RegionRange

"1" <# pattern length><tab><sequence>

Region

"X" "1"<tab>RegionRange

1 <period>

Tandem Repeat Region Table File: Abbreviated Example

```
6 0 0 0 0 0 1
S 1  491  508
1 2 gt
S 1  537  556
1 2 gt
N 1  583  610
1 2 gt
C
S 1  585  608
1 2 gt
S 1  635  641
1 1 g S 1 641  690
1 2 gt
V 1  309  703
2 24 2
```

4.4.12.2. Tandem repeat regions (individual regions)

Problem: Given the sequence, $\boldsymbol{S}$, the colored conservation array, $\boldsymbol{S_C}$, that is parallel to the $\boldsymbol{S}$, the alignment, $\boldsymbol{A_C}$, that characterizes the region, $\boldsymbol{R}$, with the consensus pattern, $\boldsymbol{P_C}$, the region edges, $\boldsymbol{A_e}$, and the region periodicity signature, $\boldsymbol{S_P}$, create a data file (*.image) for the region characterization.

Solution: The image file (*.image) is a single region characterization "image" between a pattern, $\boldsymbol{P_C}$, and a region, $\boldsymbol{R}$, on a sequence, $\boldsymbol{S}$, and depicts the alignment, $\boldsymbol{A_C}$, of $\boldsymbol{P}$ to $\boldsymbol{R}$. $\boldsymbol{S}$ provides the nucleotides. $\boldsymbol{S_C}$ provides the nucleotide "color" indicating pattern position conservation. $\boldsymbol{P_C}$ provides the template. $\boldsymbol{A_C}$ provides the spacing information that aligns the nucleotides in $\boldsymbol{S}$ to the pattern positions in $\boldsymbol{P_C}$. $\boldsymbol{A_e}$ provides the positions bounding the region. In addition, the file contains sequence both before and after the region represents proximal sequence data and the region periodicity signature, $\boldsymbol{S_P}$.

An integer "color" indicates conservation. All region characterizations for a single region are assigned a unique color. VLTR regions and regions with a simple pattern structure have one characterization per region. MPTR regions have more than one characterization per region. Since an MPTR pattern reflects patterns that are factors of the current multiple, the conservation coloring reflects the conservation associated with the same factors (see Section 4.4.8.3). If a sequence nucleotide is conserved in more than one pattern, the sequence nucleotide is colored according to the pattern associated with the smallest factor. In order to properly display the image, all image files for factors of the current multiple are necessary. These are available via the region tracking file.

Image File Format

PeriodicitySignature (1 line)
SequenceBreakpoints (1 line)
AlignmentWidth (1 line)
ImageRange (1 line)
RegionRange (1 line)
Template (series of lines)
Factors (1 line)
Sequence (series of lines)

PeriodicitySignature

<# list length> <list of integer periodicities>

SequenceBreakpoints

“0”

AlignmentWidth

<# maximum width for alignment>

ImageRange

<# begin position> <# end position>

RegionRange

<# begin position> <# end position>

Template

TemplateBegin
TemplateSequence
TemplateEnd

Factors

“F” <tab> <list of integer factors>

Sequence

SequenceOnly OR
SequenceColored OR
SequenceSpace OR
SequenceLineBreak

Image File: Abbreviated Example

```
7    9   10   19   20
37   38   48
0
14
1115 1694
1129 1539
T    begin
T    {
4    ctgdga    6
gwk  3
4    g    1
T    }*
T    end
F    1
ggcacaagaatgtg 14
*
.    1
4    ctggga    6
gag  3
4    g    1
*
.    1
4    ctggga    6
.    1
tt   2
4    g    1
*
```

4.4.12.3. Regions of similarity (entire sequence)

Problem (Regions of similarity table file format): Given the list of regions of similarity, $\mathring{\boldsymbol{R}}_{\boldsymbol{ROS}}$, create a data file (*.sim.table) detailing each ROS.

Solution: For each region, $\boldsymbol{R}_{\boldsymbol{ROS}}$, in the list, $\mathring{\boldsymbol{R}}_{\boldsymbol{ROS}}$, place an entry in the regions of similarity table file (*.sim.table). The file format specifies sets of triplet lines separated by a blank line. The first line of a triplet specifies the distance between the pair of regions and corresponds to the periodicity, ***p***, in $\boldsymbol{R}_{\boldsymbol{ROS}}$. A numerical indicator follows the distance specifying an associated distance range: a "1" indicates a range below 1,000, a "2" indicates a range between 1,000 and 10,000 and a "3" is a range above 10,000. This is a visual aid for quickly determining the magnitude of the range. The second and third lines are each a list of positions in the distance array. The second line corresponds to the leftmost region in the pair. The third line corresponds to the rightmost region in the pair and the positions stored in $\boldsymbol{R}_{\boldsymbol{ROS}}$.

Regions of Similarity Table File Format

Regions (series of triplet lines) separated by blank lines

Regions

DistanceLine
RegionLine

RegionLine

BlankLine

DistanceLine

<distance><tab>DistanceRange

RegionLine

<list of sequence positions>

DistanceRange

If <distance> less than 1000 then “1”

If <distance> 1,000..10,000 then “2”

If <distance> greater than 10,000 then “3”

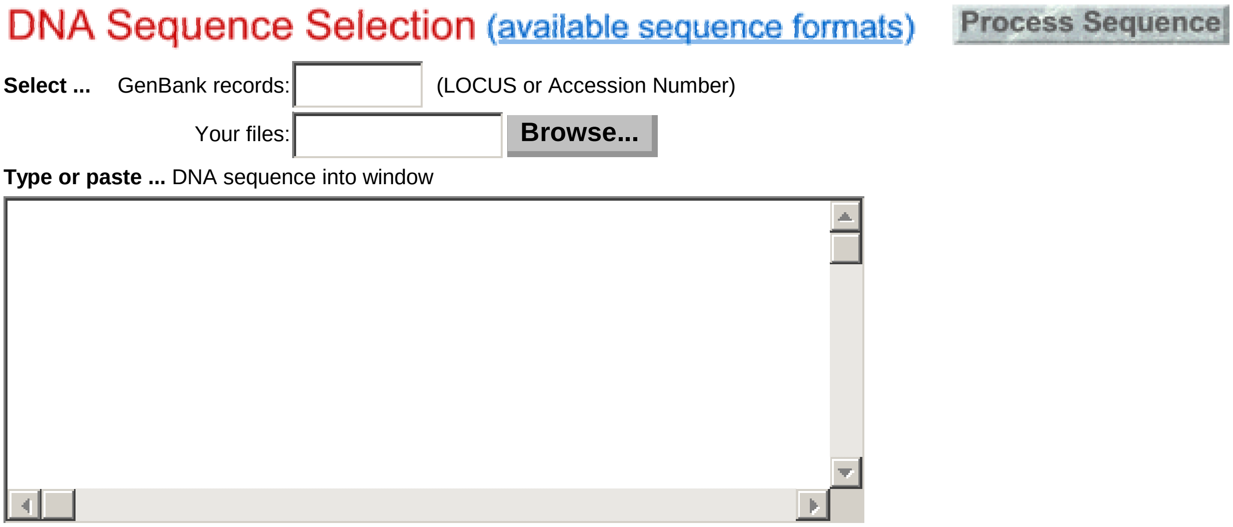
Regions of Similarity Table File: Abbreviated Example

```
153      1
11997    12030 12049 12050 12051 12061 12062 12083 12084 12103 12104
12150    12183 12202 12203 12204 12214 12215 12236 12237 12256 12257

13808    3
13544    13548 13583 13662 13663 13664 13710 13741
27352    27356 27391 27470 27471 27472 27518 27549

12278    12301 12302 12369 12408 12433 12444 12445 12447 12448 12467
   12470
24731    24754 24755 24822 24861 24886 24897 24898 24900 24901 24920
   24923

1707     2
33455    33469 33531 33532 33558 33596 33597 33617 33618 33664
35162    35176 35238 35239 35265 35303 35304 35324 35325 35371
```

4.4.12.4. Visual synopsis (entire sequence)

The identification algorithm generates many files for creating a visual synopsis of the identified regions. This information is a redundant representation of the data files previously described. The formats of these files will not be described.

4.4.13. User interface: a webpage


DNA Sequence Selection (available sequence formats)
Process Sequence
Select ... GenBank records:
(LOCUS or Accession Number)
Your files:
Browse...
Type or paste ... DNA sequence into window

4.4.13.1. Request for sequence data

Problem: Acquire sequence data for analysis.

Solution: Users input sequence data by indicating a locus or accession number from the GenBank DNA sequence database, by indicating a personal file or providing raw text. Several sequence data formats are available including well-known sequence formats, GenBank, EMBL and FASTA formats, and two simple formats, the simple comment and sequence only formats. The sequence only format contains unformatted sequence data or sequence data where the white space (spaces and tabs), carriage returns and numerals are present but ignored. The simple comment format allows comment lines at the top of the file denoted by a semicolon in the first position of the line. The remaining lines follow the sequence only format.

4.4.13.2. Process sequence data using identification algorithm

Problem: Process sequence data and identify tandem repeat regions and regions of similarity.

Solution: The HTML webpage submits requests to a Perl script running on a CGI server that retrieves and stores the sequence data in a data file and issues a command line execution of the primary identification algorithm. The identification algorithm identifies tandem repeat regions and regions of similarity and stores the results in data files. Once the identification procedure is complete, the script processes the result data files and displays the results as a HTML webpage.

The user's sequence data file is stored according to the date and time. All results data files are named by appending a file ending to the sequence data file name.

4.4.13.3. Display results of processing the sequence data

Problem: Display results of identification algorithm.

Solution: A Perl script processes the identification results after applying the identification algorithm to a sequence data file. A HTML webpage displays the results of analyzing the sequence including a visual synopsis of the tandem repeat regions and the regions of similarity, the complete sequence data with tandem repeat regions highlighted, alignments of individual tandem repeat regions with periodicity larger than the window size of $\boldsymbol{k} = 6$ and a tandem repeat region table containing region edges, pattern structure type and pattern information. Chapter 5 describes the format of the displays.

(This page left intentionally blank)

5. Analysis of sequence data

This section presents analysis of GenBank sequences using the identification algorithm given in Chapter 4 and discusses general analysis of regions in DNA sequences having a specific pattern structure. The algorithm analyzes a DNA sequence to identify tandem repeat regions and regions of similarity. The primary thrust is tandem repeat region identification which locates the position of the region in the sequence and determines the pattern structure and conservation of the region in order to characterize the region. Secondary analysis identifies non-contiguous recurrences such as interspersed repeats, distant, related genes and clusters of related genes. The identification algorithm uses a collection of techniques tailored to address issues and challenges for specific types of patterns and pattern structures. Section 5.1 describes analysis of individual sequences. Section 5.2 discusses analysis of pattern classes including algorithmic extensions to identify non-contiguous regions of similarity.

5.1. GenBank sequence collection: Analysis of individual sequences

This section describes the analysis of sequences in the collection of GenBank sequences (Section 3.3). Each subsection discusses the analysis for a single sequence. In particular, the discussion describes the prior knowledge acquired from the sequence annotation and the biological literature, the structure known from the visual analysis of the sequence, additional knowledge discovered during development of the identification algorithm and the capability of the identification algorithm to locate and characterize tandem repeat regions and to locate regions of similarity.

Sequence displays depict the success of the identification algorithm to identify tandem repeats. One displays the entire GenBank sequence and locates regions by coloring the positions indicating tandem repeat regions. The other shows only the positions in and near a single tandem repeat region, characterizes the region by aligning the region to a pattern and colors conserved positions in the tandem repeat region.

The display of the entire GenBank sequence locates **perfect SSRs**, **imperfect SSRs** and, **identified, non-SSR tandem repeat regions** using colors to indicate the sequence positions contained in a region. A sequence position that belongs to multiple regions of different types assigns the highest priority color to the position. Perfect SSRs have the highest coloring priority followed by imperfect SSRs, then non-SSR tandem repeat regions and finally sequence positions not present in a tandem repeat region. The sequence is broken into lines containing 50 nucleotides and placed consecutively one above the next. Each row is flanked by sequence positions corresponding to the first and last position in the row. Column position markers above and below the sequence indicate the horizontal position every 10 positions across a row. An example is given below.

```
            10        20        30        40        50
             |         |         |         |         |
  1 gaggaggatggaacactgggggagccgatacccaggacagggcagtcct  50
 51 ggaggcaaccgttatccacctcagggagggggtggctggggtcagcccca 100
101 tggaggtggctggggccagcctcatggaggtggctggggccaacctcatg 150
151 gaggtggctggggtcagccccatggtggtggctggggacagccacatggt 200
201 ggtggctggggacagccacatggtggtggaggctggggtcaaggtgtacc 250
251 c                                                  251
             |         |         |         |         |
```

The display for an individual non-SSR region characterization depicts both the alignment of a region to a pattern and the conservation of the region. Each region has a base pattern characterization and MPTR regions have additional region characterizations. This display includes the sequence and structure of the pattern, the alignment of sequence positions in the region to a pattern as a series of copies and the conservation of pattern positions across copies in the region.

First, the display depicts the pattern used to generate the alignment on a line above the aligned sequence copies. For the base pattern, the line begins with "**>1X**" and integer multiple patterns associated with MPTR regions substitute the integer multiple for the **1**, e.g. an integer multiple of 2 uses "**>2X**." For each pattern line, pattern position conservation colors over the **default coloring** such that each integer multiple has a distinct coloring. For integer multiples, several pattern lines may appear, one for each factor of the current integer multiple with its own region characterization.

Second, the display for each region characterization aligns the sequence to the pattern to create rows such that each row represents a copy of the pattern. These rows appear directly below the pattern line(s). Each row is flanked by sequence positions corresponding to the first and last position in the row. Black spaces representing gaps enable sequence positions to align to the appropriate pattern position. Vertical columns through the copies coincide with the pattern position to which all positions in the column align.

Third, the display depicts conservation in the region by indicating high conservation in individual columns through sequence copies that are associated with a particular pattern position. If the conservation criteria are met, then all occurrences of a particular nucleotide in the column are colored according to the assigned color for a pattern (for coloring rules, see Section 4.4.8.3). For base pattern region characterizations, only the base pattern colors the sequence. For integer multiple region characterizations, several patterns color the sequence, one for each factor of the current integer multiple with its own region characterization. Here, the color of the pattern with the smallest factor has the highest coloring priority.

Finally, many of the region characterizations are too wide to fit across the page. Thus, the alignment is split into segments at most 50 columns wide and the segments are displayed sequentially. The webpage depictions are substantially better since the segmentation does not occur. An example is given below.

```
              10        20        30
               |         |         |
 1X: {c tgdgagwkg}*
>2X:{[ctgdgagwgg|c tgdgagwag]}*
1195        gagg              1198
1199  ctgggagagg c tggga ttg  1217
1218  ctgggagagg c tgggag ag  1236
1237  ctgacagagg c tggga ttg  1255
1256  ctgggaaagg c tgggag ag  1274
1275  ctgggagagg cctgggag ag  1294
1295  ctgggagagg c tgtga ttg  1313
1314  ctgggagagg              1323
1324  ctgggagagg c tgggag ag  1342
1343  ctgggagagg c tgaga ttg  1361
1362  ctggga agg c tgggag ag  1379
1380  ctgggagagg c tgggag ag  1398
1399  ctgggagagg c tgtga ttg  1417
1418  ctgggagagg              1427
               |         |         |
```

5.1.1. GenBank locus: AF071126

Species: *Zea mays* (corn).

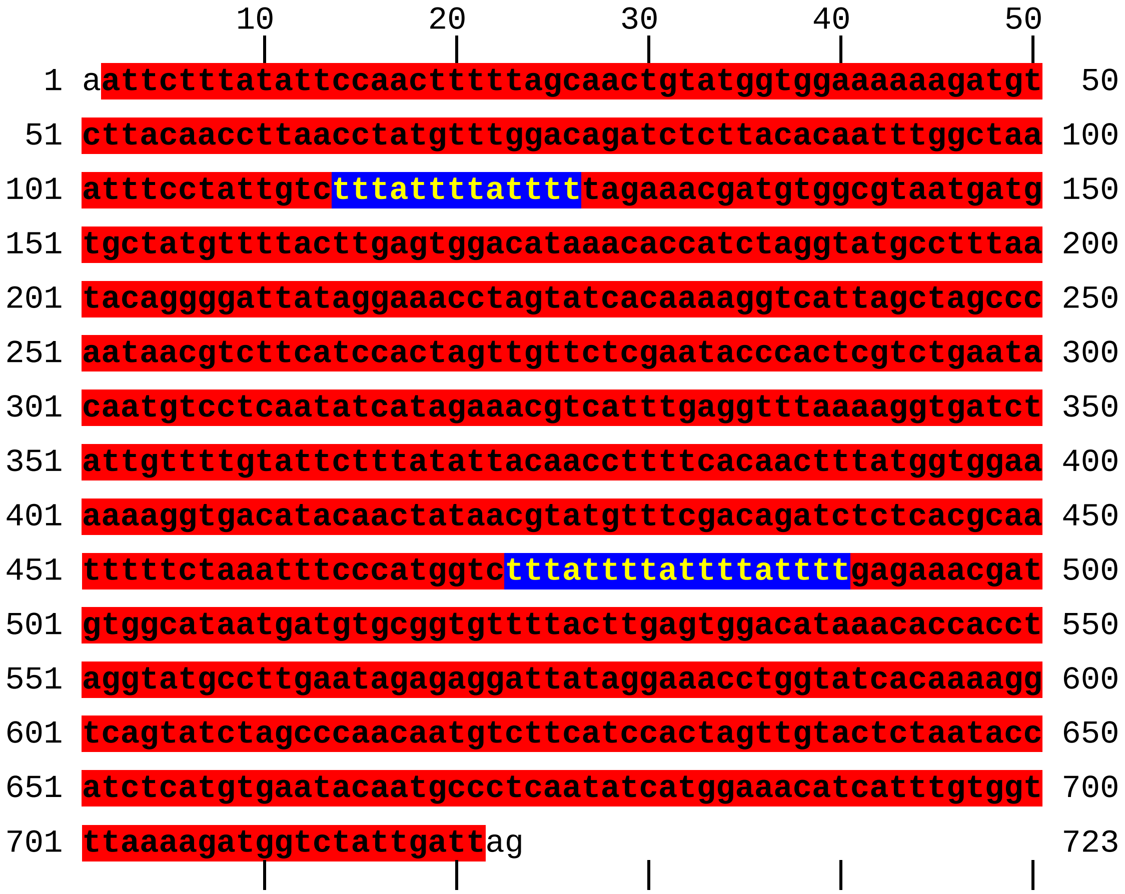
Sequence annotation. This sequence contains a dimer of two TR-1 elements each about 360 base pairs (bp) long. The dimer represents a small fragment of a larger 12 kbp region corresponding to a knob DNA segment (Ananiev *et al*. 1998). Here, the dimer represents a tandem repeat region having a long pattern and a copy number of two.

Visual analysis. The region contains two SSR regions having an **ATTTT** pattern that were discovered during development of the identification algorithm. These SSR regions occur within the dimer and have different copy numbers, thereby categorizing the region as a variable length tandem repeat (VLTR).

Algorithmic performance. The identification algorithm locates two nested SSR regions and the VLTR region. Figure 5.1.1.A displays the location of the SSR and VLTR regions within the sequence. The SSR regions have the same **ATTTT** pattern but different copy numbers. Figure 5.1.1.B displays the region characterization for the VLTR region.

This sequence was surprisingly difficult to analyze due to the unexpected presence of an internal tandem repeat with variable copy number. Early versions of the VLTR region analysis did not handle regions having a copy number of two. Thus, this sequence served to refine the VLTR identification process for regions with few copies.

This sequence served to improve the algorithmic performance for VLTR region analysis due to various difficulties encountered. First, regions having two copies generate one distance for each identical word, i.e. there are only two occurrences for each identical word. Second, VLTR regions have copy length variation that generates distances corresponding to each copy length, i.e. the region has more than one significant distance. In this region, the distances are 354, 359 and 364. The distances of 359 and 364 correspond to the two copy lengths. The distance of 354 corresponds to the distance between the end of one location of the internal tandem repeat and the beginning of another location of the same internal tandem repeat. Third, the design of the analysis enables identification of regions having slightly less than two copies, a



```
                10        20        30        40        50
                 |         |         |         |         |
  1 aattctttatattccaacttttagcaactgtatggtggaaaaaagatgt  50
 51 cttacaaccttaacctatgtttggacagatctcttacacaatttggctaa 100
101 atttcctattgtctttattttatttttagaaacgatgtggcgtaatgatg 150
151 tgctatgttttacttgagtggacataaacaccatctaggtatgcctttaa 200
201 tacaggggattataggaaacctagtatcacaaaaggtcattagctagccc 250
251 aataacgtcttcatccactagttgttctcgaatacccactcgtctgaata 300
301 caatgtcctcaatatcatagaaacgtcatttgaggtttaaaaggtgatct 350
351 attgttttgtattctttatattacaaccttttcacaactttatggtggaa 400
401 aaaaggtgacatacaactataacgtatgtttcgacagatctctcacgcaa 450
451 tttttctaaatttcccatggtctttattttattttattttgagaaacgat 500
501 gtggcataatgatgtgcggtgttttacttgagtggacataaacaccacct 550
551 aggtatgccttgaatagagaggattataggaaacctggtatcacaaaagg 600
601 tcagtatctagcccaacaatgtcttcatccactagttgtactctaatacc 650
651 atctcatgtgaatacaatgccctcaatatcatggaaacatcatttgtggt 700
701 ttaaaagatggtctattgattag                            723
                 |         |         |         |         |
```

Figure 5.1.1.A: Entire sequence for GenBank locus AF071126. The yellow on blue positions indicate a SSR. The black on red positions indicate a non-SSR tandem repeat region.

condition that identifies a single duplication of a long sequence section. In this region, positions associated with a single distance covers a little over one and a half copies, thereby not satisfying this criteria. Thus, a single distance cannot locate the region of a two copy VLTR region and the related distance analysis that utilizes identical words to identify related distances cannot expand the region to make it large enough to perform additional region analysis.

```
   1 a   1

             10        20        30        40        50
              |         |         |         |         |
>1X:{kagaaacgatgtggcrtaatgatgtgckrtgttttacttgagtggacata
   2                                                      2
 127 tagaaacgatgtggcgtaatgatgtgctatgttttacttgagtggacata 176
 491 gagaaacgatgtggcataatgatgtgcggtgttttacttgagtggacata 540
              |         |         |         |         |
             60        70        80        90       100
              |         |         |         |         |
>1X: aacaccatctaggtatgcctttaatacaggggattataggaaacctagta
   2                                                      2
 177 aacaccatctaggtatgcctttaatacaggggattataggaaacctagta 226
 541 aacaccacctaggtatgccttgaatagagaggattataggaaacctggta 590
              |         |         |         |         |
            110       120       130       140       150
              |         |         |         |         |
>1X: tcacaaaaggtcattagctagcccaataacgtcttcatccactagttgtt
   2                                                      2
 227 tcacaaaaggtcattagctagcccaataacgtcttcatccactagttgtt 276
 591 tcacaaaaggtcagtatctagcccaacaatgtcttcatccactagttgta 640
              |         |         |         |         |
            160       170       180       190       200
              |         |         |         |         |
>1X: ctcgaatacccactcgtctgaatacaatgtcctcaatatcatagaaacgt
   2                                                      2
 277 ctcgaatacccactcgtctgaatacaatgtcctcaatatcatagaaacgt 326
 641 ctctaataccatctcatgtgaatacaatgccctcaatatcatggaaacat 690
              |         |         |         |         |
```

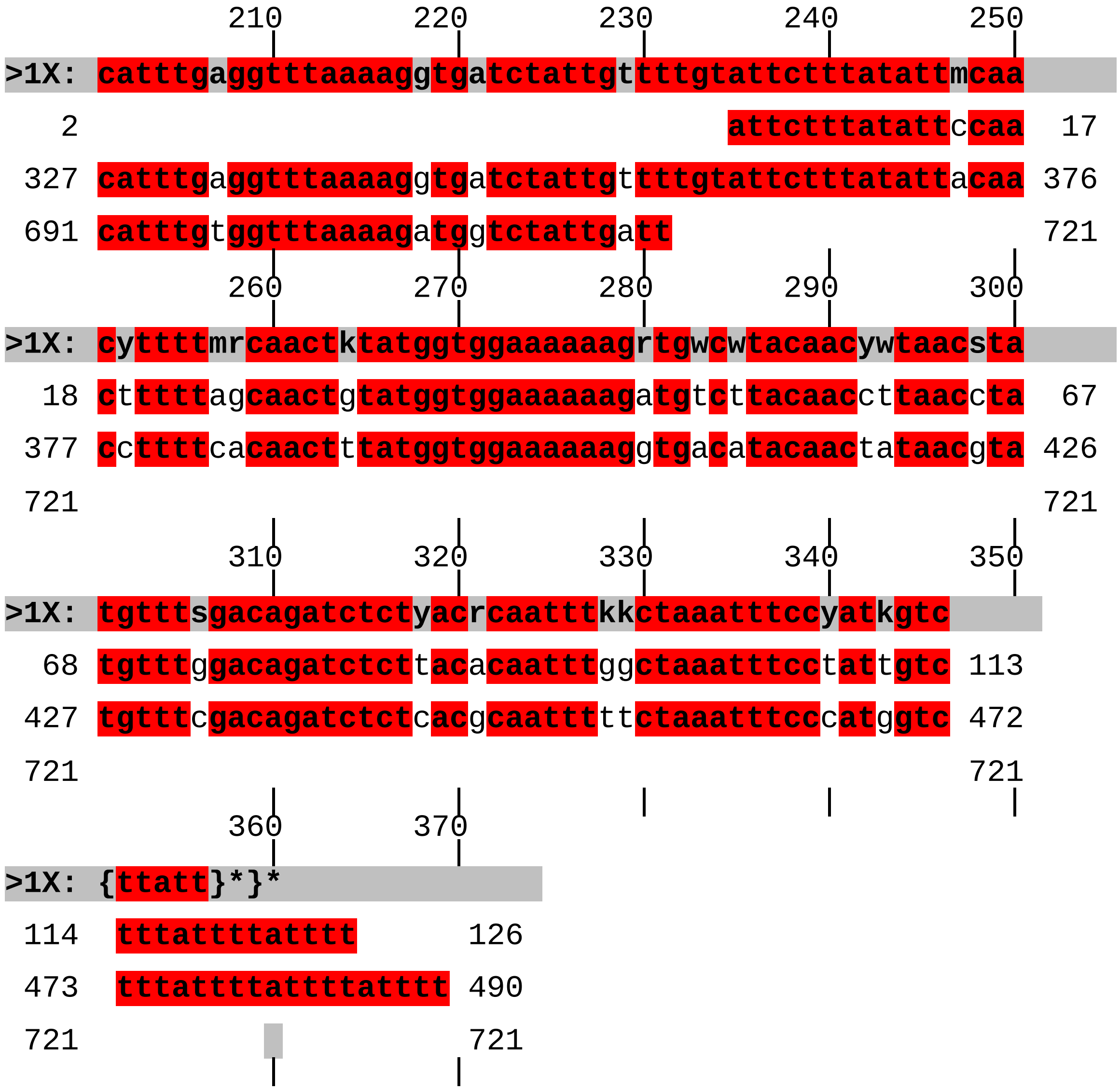

210 220 230 240 250
>1X: catttgaggtttaaaaggtgatctattgttttgtattctttatattmcaa
2 attctttatattccaa 17
327 catttgaggtttaaaaggtgatctattgttttgtattctttatattacaa 376
691 catttgtggtttaaaagatggtctattgatt 721
260 270 280 290 300
>1X: cyttttmrcaactktatggtggaaaaaagrtgwcwtacaacywtaacsta
18 cttttagcaactgtatggtggaaaaaagatgtcttacaacctaacccta 67
377 cctttcacaactttatggtggaaaaaaggtgacatacaactataacgta 426
721 721
310 320 330 340 350
>1X: tgtttsgacagatctctyacrcaatttkkctaaatttccyatkgtc
68 tgtttggacagatctcttacacaatttggctaaatttcctattgtc 113
427 tgtttcgacagatctctcacgcaatttttctaaatttcccatggtc 472
721 721
360 370
>1X: {ttatt}*}*
114 tttattttatttt 126
473 tttattttattttatttt 490
721 721

Figure 5.1.1.B: Region characterization for the VLTR region in the AF071126 sequence. This VLTR region contains two internal SSR regions having the pattern **ATTTT**, each having a different copy number. The non-SSR region is 346 bp long.

5.1.2. GenBank locus: AF091342

Species: *Bos taurus* (cow).

Sequence annotation. This sequence is a portion of the neuronal intermediate filament gene (NF-M). The gene contains a recurring KSP amino acid motif (lysine-serine-proline) known as a KSP repeat. In this species, the KSP repeat is KSPXX and is reflected in the DNA sequence as a tandem repeat region having a 15 nucleotide pattern.

Visual analysis. Initial analysis reveals two features. First, the sequence contains many SSRs most of which are tri-nucleotide repeats having the patterns **AAG** or **AGG** and all but two of which are comprised of the nucleotides **A** and **G**. Second, the sequence contains a tandem repeat region composed primarily of five, related, 15-nucleotide patterns that form a marginal multi-periodic tandem region (MPTR). The five patterns are **AAGTCCCCAACAGCA**, **AAGTCCCCAGAGGCA**, **AAGTCCCCAACGGCC**, **AAGTCCCCAGTGGCC** and **AAGTCCCCGACGGCC**.

Algorithmic performance. The algorithm locates both the SSR regions and the MPTR region. The SSRs have periodicities of 1, 3, 4, 5 and 6 with most being tri-nucleotide patterns containing the nucleotides A and G. Table 5.1.2.A details the specifics for each SSR region. The MPTR region has a moderately conserved, base pattern of 15 nucleotides as shown in the region characterization displayed in Figure 5.1.2.A. Integer multiples of 4 and 6 attempt to capture the marginal cross-copy patterns but it is clear from the region characterizations shown in Figure 5.1.2.B and Figure 5.1.2.C that the entire region does not fully support either periodicity.

Both the SSR regions and the MPTR region have an identification challenge. First, the SSR regions have distinct positions but many have the same periodicity signature and closely related patterns. This requires the algorithm to locate individual regions based not only on periodicity and pattern but also on position. Second, the MPTR region has a base periodicity of 15 but the region's pattern is variable with about five, 15 nucleotide sequences recurring within the region. The challenge is to determine whether the cross copy pattern that occurs in a portion of the region is sufficiently significant to define a cross-copy periodicity. This region offers the opportunity to hone identification parameters for marginal MPTR regions.

Sequence Location	Region Pattern	Sequence Location	Region Pattern
220 ..229	AGG	1886..1896	AAG
367 ..379	AGG	1931..1939	AGG
367..376	AGG	1984..1992	AGG
1151 ..1171	AAG	1999..2024	AAGGAG
1151..1159	AAG	2008..2024	AAGGAG
1162..1171	AAG	2045..2057	AGG
1222 ..1279	AAGAGG	2045..2053	AGG
1238..1247	AGG	2070..2080	AAGG
1256..1277	AAGAGG	2071..2080	AAGG
1529 ..1537	AAC	2103..2115	AAGGG
1781 ..1809	AAGGAG	2103..2113	AAGGG
1790..1801	AAGGAG	2925..2944	A
1808 ..1818	AAAAG	2925..2943	A
1812 ..1821	AAAG		

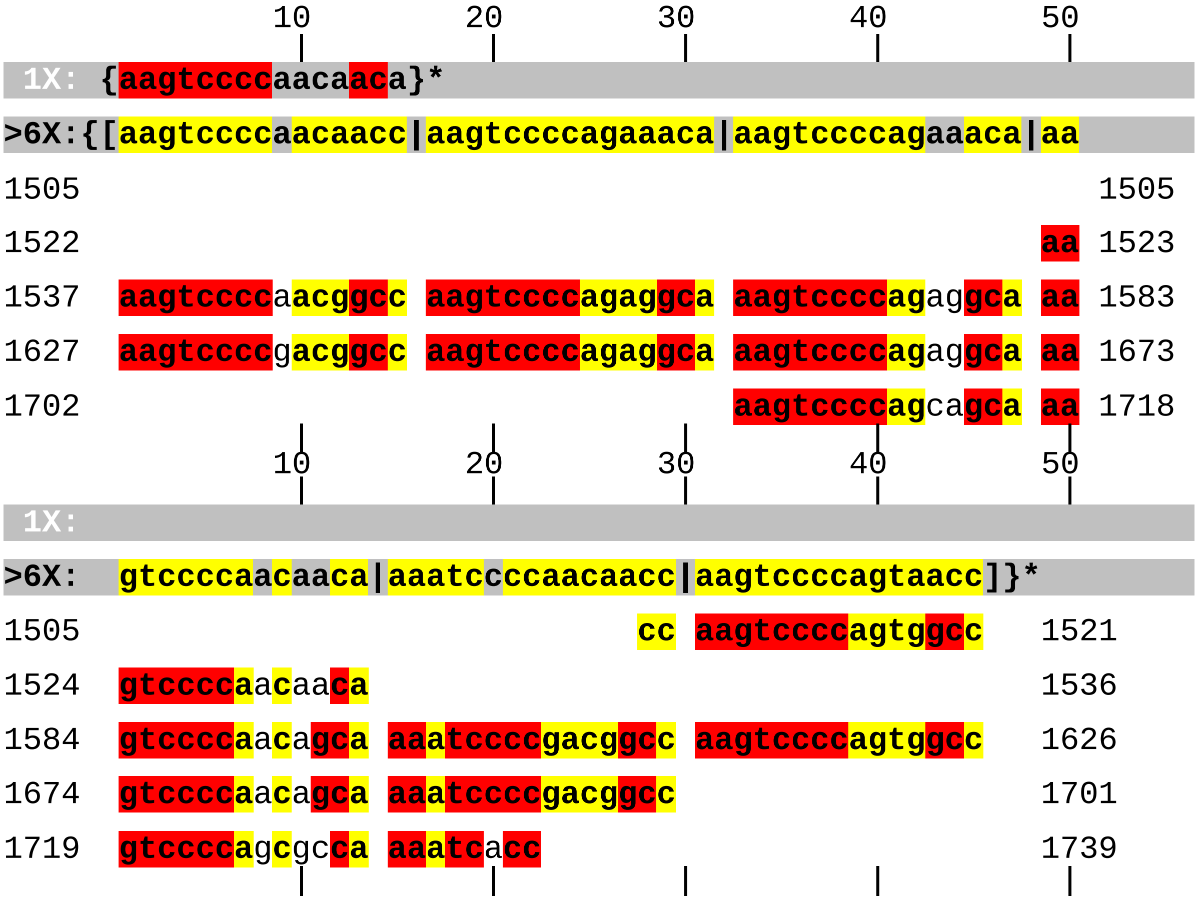
Table 5.1.2.A: SSRs present in the AF091342 sequence as determined by the identification algorithm.

```
             10        20        30        40        50
              |         |         |         |         |
1407 cgaagctaaggaggaaaagaaaatggaggaaaaggctgaagaagtggctc 1456
1457 caaaggaggagctggcggcagaagccaaggtggagaagccagagaaagcc 1506
              |         |         |         |         |
             10        20
              |         |
>1X:{aagtccccaacaaca}*
1507 aagtccccagtggcc   1521
1522 aagtccccaacaaca   1536
1537 aagtccccaacggcc   1551
1552 aagtccccagaggca   1566
1567 aagtccccagaggca   1581
1582 aagtccccaacagca   1596
1597 aaatccccgacggcc   1611
1612 aagtccccagtggcc   1626
1627 aagtccccgacggcc   1641
1642 aagtccccagaggca   1656
1657 aagtccccagaggca   1671
1672 aagtccccaacagca   1686
1687 aaatccccgacggcc   1701
1702 aagtccccagcagca   1716
1717 aagtccccagcgcca   1731
1732 aaatcacc          1739
              |         |
             10        20        30        40        50
              |         |         |         |         |
1740 tgtggaggaagtgaaacccaaagcagaagctggagctgagaaaggagaac 1789
1790 agaaggagaaggtggaggaagaaaagaaagaagcaaaggaatctcccaag 1839
              |         |         |         |         |
```

Figure 5.1.2.A: Region characterization for the base pattern of the MPTR region in the AF091342 sequence.

```
              10        20        30        40        50
               |         |         |         |         |
 1X: {aagtccccaacaaca}*
>4X:{[aagtccccaacaaca|aagtccccaacgaca|aagtccccagcgaca|aa
1507                                  aagtccccagtggcc    1521
1522  aagtccccaacaaca aagtccccaacggcc aagtccccagaggca    1566
1567                                  aagtccccagaggca    1581
1582  aagtccccaacagca aaatccccgacggcc aagtccccagtggcc    1626
1627                  aagtccccgacggcc aagtccccagaggca    1656
1657                                  aagtccccagaggca    1671
1672  aagtccccaacagca                                 aa 1688
1702  aagtccccagcagca                 aagtccccagcgcca    1731
1732                                  aaatcacc           1739
               |         |         |         |         |
              10        20
               |         |
 1X:
>4X:  gtccccaacgaca]]}*
1521                   1521
1566                   1566
1581                   1581
1626                   1626
1656                   1656
1671                   1671
1689  atccccgacggcc    1701
1731                   1731
1739                   1739
               |         |
```

Figure 5.1.2.B: Region characterization of the 4X pattern for the MPTR region in the AF091342 sequence.



```
                   10        20        30        40        50
                    |         |         |         |         |
 1X: {aagtccccaacaaca}*
>6X:{[aagtccccaacaacc|aagtccccagaaaca|aagtccccagaaaca|aa
1505                                                       1505
1522                                                    aa 1523
1537  aagtccccaacggcc aagtccccagaggca aagtccccagaggca aa 1583
1627  aagtccccgacggcc aagtccccagaggca aagtccccagaggca aa 1673
1702                                  aagtccccagcagca aa 1718
                    |         |         |         |         |
                   10        20        30        40        50
                    |         |         |         |         |
 1X:
>6X:  gtccccaacaaca|aaatccccaacaacc|aagtccccagtaacc]}*
1505                            cc aagtccccagtggcc   1521
1524  gtccccaacaaca                                  1536
1584  gtccccaacagca aaatccccgacggcc aagtccccagtggcc   1626
1674  gtccccaacagca aaatccccgacggcc                  1701
1719  gtccccagcgcca aaatcacc                         1739
                    |         |         |         |         |
```

Figure 5.1.2.C: Region characterization of the 6X pattern of the MPTR region in the AF091342 sequence.

5.1.3. GenBank locus: AMU73928

Species: *Apis mellifera* (honeybee)

Sequence annotation. The sequence annotation demarks a minisatellite region from position 76 to position 209 but does not specify a pattern for the tandem repeat region. The research is unpublished and additional effort to ascertain the pattern was fruitless.

Visual analysis. The sequence contains a well-conserved, variable length tandem repeat (VLTR) region having a 17 bp non-SSR region and a nested SSR with a **T** pattern.

Algorithmic performance. The algorithm successfully identifies the nested SSR regions and the VLTR region. The region edges noted in the annotation represent four copies of the pattern structure and the algorithm expands both edges to include partial copies before and after the annotated region. Figure 5.1.3.A displays the entire sequence and shows the specifics for the SSR regions. Figure 5.1.3.B displays the region characterization for the VLTR region in which only two sequence differences occur in over four copies of the pattern.

```
         10        20        30        40        50
          |         |         |         |         |
  1 cccatgtcccagcggcgtattgctttgcatcgcgaacgcactttcaatgt  50
 51 cccagcggcgtattgcttctattttataagtaccagctaaattttttttt 100
101 tttttttataagtaccagctaaaattttttttttttttttttataagtac 150
151 cagctaaaattttttttttttttttttataagtaccagctaaaatttttt 200
201 tttttttttataagttccagcggcgtattgctttctgaaatttaaaaaaa 250
251 aaaaaaaattttttttttaataatatattatata                 283
          |         |         |         |         |
```

Figure 5.1.3.A: Entire sequence for GenBank locus AMU73928. The sequence contains five SSR regions having a **T** pattern and one having an **A** pattern. Four of the **T** pattern SSR regions occur within a VLTR region.

```
         10        20        30        40        50
          |         |         |         |         |
  1 cccatgtcccagcggcgtattgctttgcatcgcgaacgcactttcaatgt  50
 51 cccagcggcgtattgcttcta                               71
          |         |         |         |         |
         10        20        30        40
          |         |         |         |
>1X:{ataagtaccagctaaaa{t}*}*
 72                  tttt            75
 76 ataagtaccagctaaat ttttttttttttttt     107
108 ataagtaccagctaaaa tttttttttttttttttt  142
143 ataagtaccagctaaaa tttttttttttttttttt  177
178 ataagtaccagctaaaa ttttttttttttttt     209
210 ataagttccagc                          221
          |         |         |         |
         10        20        30        40        50
          |         |         |         |         |
222 ggcgtattgctttctgaaatttaaaaaaaaaaaaaaatttttttttaata 271
272 atatattatata                                       283
          |         |         |         |         |
```

Figure 5.1.3.B: Region characterization for the VLTR region in the AMU73928 sequence.

5.1.4. GenBank locus: BOVTGN

Species: *Bos taurus* (cow).

Sequence annotation. This sequence contains a minisatellite region (denoted BTGL1) having variable copy number, nested SSR regions. The fragment containing BTGL1 is known to exhibit length variation and has been utilized for DNA fingerprinting (Nave *et al*. 1997). This sequence represents one of at least 13 alleles for BTGL1 (Kashi *et al*. 1994) and is composed of seven copies between 46 and 82 bp in length such that each copy contains a 29 bp sequence coupled with an SSR having a **GT** pattern (Kashi *et al*. 1990).

Visual analysis. Visual analysis confirms the region's features and the complex pattern structure that classifies the region as a variable length tandem repeat (VLTR). In addition, the sequence at the transition between the non-SSR sequence and the nested SSR is similar to the SSR pattern yet is imperfect. In the sequence following the perfect SSR, a **TG** occurs creating two consecutive **T** nucleotides. In the sequence preceding the perfect SSR, one copy contains a run of seven **G** nucleotides with three **G** nucleotide shadows in four of the other copies. The original researchers place all of these inconsistencies in the non-SSR portion of the pattern.

Algorithmic performance. The identification algorithm locates the SSR regions and the VLTR region within the sequence. Figure 5.1.4.A displays the entire sequence and locates the perfect SSR regions including seven **GT**-pattern SSR regions, one **G**-pattern SSR region and one **A**-pattern SSR region. The **GT**-pattern and **G**-pattern SSRs occur within the VLTR region. A **TGG** sequence precedes many of the SSR regions and a **TG** sequence secedes all of the SSR regions. The high cost of a gap close to the edge of the SSR region precludes inclusion of these in the

SSR. In this particular region, inclusion of these sequences in the non-SSR portion reinforces the recurring gap present throughout the region.

The VLTR region characterization in Figure 5.1.4.B uses a pattern with a 24 bp non-SSR sequence and a nested **GT**-pattern SSR to locate the VLTR region. This analysis includes the seceding **TG** in the non-SSR portion and the preceding **TGG** in the SSR portion. The pattern was chosen from an initial region that ends at position 550 due to the partitioning scheme described in Section 4.4.11. Figure 5.1.4.C shows the region characterization for the analysis without this arbitrary anomaly that uses a pattern with a 28 bp non-SSR sequence and a nested **GT**-pattern SSR. Here, the preceding **TGG** and the seceding **TG** sequences are in the non-SSR portion of the pattern. Both characterizations are imprecise but capture the major structure of the region.

```
              10        20        30        40        50
               |         |         |         |         |
  1 aagcttcacatcccgagaattccctcccagcgctcgtggtcccacagagg  50
 51 gctctgctggacctgcctcgggtcacatggcaggtctggggaggacacac 100
101 ctctccccggcagagaaatggccagaagccaggtctgctccacacgtgcc 150
151 ttctcccaatactctctaacttttaaaaaaactgccaaagaaaaagcggta 200
201 cgtaataacaagcgcacagatacgtaatttataatggctgacacggttgg 250
251 cagggaaatgtgttacgcaggaattatgtttttatttatgtgtgtcctgt 300
301 tttggagacagcataagtaatcatgggtgtgtgtgtgtgtgtgtgtgtgt 350
351 gttgcctgtctccagcgtaagtaatcatgtgtgtgtgtgtgtgtgtgtgt 400
401 tgcctgtctccagcgtaagtaatcgtgtgtgtgtgtgtgtgtgtgtgtgt 450
451 gtgtgtgtgtgttgcctgtctccagagtaagtaatcatgggtgtgtgtgt 500
501 gtgtgtgttgcctgtctccagcataagtaatcatgggtgtgtgtgtgtgt 550
551 gtgtgttgcctgtctccagcataagtaatcatgggtgtgtgtgtgtgtgt 600
601 gtgtgtgttgcctgtctccagcataagtaatcatgggggggtgtgtgtgt 650
651 gtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgttgcctgtctc 700
701 cagggacttttgtacagagaagctt                          725
               |         |         |         |         |
```

Figure 5.1.4.A: The entire sequence for GenBank locus BOVTGN. The sequence contains seven SSR regions having a **GT** pattern, one having a **G** pattern and one having an **A** pattern. The **GT**-pattern and **G**-pattern SSR regions occur within the VLTR region.

```
             10        20        30        40        50
              |         |         |         |         |
 259 tgtgttacgcaggaattatgttttatttatgtgtgtcctgttttggaga 308
              |         |         |         |         |
             10        20        30        40        50
              |         |         |         |         |
>1X:{tgcctgtctccagcgtaagtaatc{gt}*}*
 309           cagcataagtaatc atgggtgtgtgtgtgtgtgtgtg 347
 353 tgcctgtctccagcgtaagtaatc atgtgtgtgtgtgtgtgtgtgtgt  400
 401 tgcctgtctccagcgtaagtaatc gtgtgtgtgtgtgtgtgtgtgtgtg 449
 463 tgcctgtctccagagtaagtaatc atgggtgtgtgtgtgtgtgt    508
 509 tgcctgtctccagcataagtaatc atgggtgtgtgtgtgtgtgtgt  556
 557 tgcctgtctccagcataagtaatc atgggtgtgtgtgtgtgtgtgtg 605
 609 tgcctgtctccagcataagtaatc atggggggtgtgtgtgtgtgtgtg 657
 691 tgcctgtctccag                                     703
              |         |         |         |         |
             10        20        30        40
              |         |         |         |
>1X:
 348 tgtgt                               352
 400                                     400
 449 tgtgtgtgtgtgt                       462
 508                                     508
 556                                     556
 606 tgt                                 608
 658 tgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgt   690
 691                                     703
              |         |         |         |
             10        20        30
              |         |         |
 704 ggacttttgtacagagaagctt  725
              |         |         |
```

Figure 5.1.4.B: VLTR region characterization using first four copies to choose pattern for a region in the BOVTGN sequence.

```
             10        20        30        40        50
              |         |         |         |         |
 259 tgtgttacgcaggaattatgttttatttatgtgtgtcctgttttggaga 308
              |         |         |         |         |
             10        20        30        40        50
              |         |         |         |         |
>1X:{tgcctgtctccagcgtaagtaatcatgg{gt}*}*
 309           cagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgtg 347
 353 tgcctgtctccagcgtaagtaatcatgt gtgtgtgtgtgtgtgtgtgt   400
 401 tgcctgtctccagcgtaagtaatcgtgt gtgtgtgtgtgtgtgtgtgtg 449
 463 tgcctgtctccagagtaagtaatcatgg gtgtgtgtgtgtgtgtgt     508
 509 tgcctgtctccagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgt   556
 557 tgcctgtctccagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgtg 605
 609 tgcctgtctccagcataagtaatcatgg gggggtgtgtgtgtgtgtgtg 657
 691 tgcctgtctccag                                      703
              |         |         |         |         |
             10        20        30        40
              |         |         |         |
>1X:
 348 tgtgt                              352
 400                                    400
 449 tgtgtgtgtgtgt                      462
 508                                    508
 556                                    556
 606 tgt                                608
 658 tgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgt  690
 691                                    703
              |         |         |         |
             10        20        30
              |         |         |
 704 ggacttttgtacagagaagctt  725
              |         |         |
```

Figure 5.1.4.C: VLTR region characterization using the entire region in the BOVTGN sequence to choose a pattern.

5.1.5. GenBank locus: BTA132392

Species: *Bos taurus* (cow)

Sequence annotation. This sequence represents a segment extracted from a prion protein (PrP) gene (PRNP). The gene contains a tandem repeat region with numerous documented alleles that primarily differ by the number of copies in the region although minor point mutations also are evident (Goldmann *et al*. 1991, Schlapfer *et al*. 1999). This sequence contains an allele having seven copies of a 24 to 27 bp pattern (Schlapfer *et al*. 1999). The length difference occurs in an imperfect SSR having a **GGT** pattern nested within the 24 or 27 nucleotide pattern. The difference translates in the PrP protein into the presence or absence of a Gly amino acid in a string of Gly amino acids.

Visual analysis. The sequence documented in the literature is confirmed visually. Analysis of the nested SSRs determines that only two copies have 27 nucleotides, the first and last copies in the region, and both are imperfect. Furthermore, the nested SSRs have well positioned substitutions that make them very difficult to identify. Thus, this region is a variable length tandem repeat (VLTR) in which the nested SSR has poor conservation and the non-SSR sequence is well conserved.

Algorithmic performance. The algorithm locates several SSR regions and a tandem repeat region having six to seven copies using a pattern with a simple pattern structure. Figure 5.1.5.A displays the identified SSR regions. Figure 5.1.5.B displays the region characterization for the larger tandem repeat region. The simple pattern structure in this characterization does not capture the presence of the nested SSRs and as a result not only is the region not fully

```
                10        20        30        40        50
                 |         |         |         |         |
    1 gaggaggatggaacactgggggagccgatacccaggacagggcagtcct  50
   51 ggaggcaaccgttatccacctcagggagggggtggctggggtcagcccca 100
  101 tggaggtggctggggccagcctcatggaggtggctggggccaacctcatg 150
  151 gaggtggctggggtcagccccatggtggtggctggggacagccacatggt 200
  201 ggtggctggggacagccacatggtggtggaggctggggtcaaggtgtacc 250
  251 c                                                  251
                 |         |         |         |         |
```

Figure 5.1.5.A: The entire sequence for GenBank locus BTA132392. The sequence contains three SSR regions having a **TGG** pattern. These SSRs occur within a larger tandem repeat region having a 24 or 27 bp pattern.

```
                10        20        30        40        50
                 |         |         |         |         |
    1 gaggaggatggaacactgggggagccgatacccaggacagggcagtcct 50
   51 ggaggcaaccgttatccacctcaggga                        77
                 |         |         |         |         |
                10        20        30
                 |         |         |
>1X:{ctgggghcarcchcatggdggtgg}*
   78                ggg ggtgg   85
   86 ctggggtcagccccatggaggtgg  109
  110 ctggggccagcctcatggaggtgg  133
  134 ctggggccaacctcatggaggtgg  157
  158 ctggggtcagccccatggtggtgg  181
  182 ctggggacagccacatggtggtgg  205
  206 ctggggacagccacatggtggtgg  229
                 |         |         |
                10        20        30
                 |         |         |
  230 aggctggggtcaaggtgtaccc  251
                 |         |         |
```

Figure 5.1.5.B: The region characterization for a tandem repeat region using a simple pattern structure that occurs in the BTA132392 sequence.

characterized but both the leading and trailing edges are incorrect such that half a copy is excluded from the region at each edge. Nevertheless, the characterization does locate the region and it is straightforward to view the characterization and ascertain the correct pattern structure. As such, the region is minimally characterized.

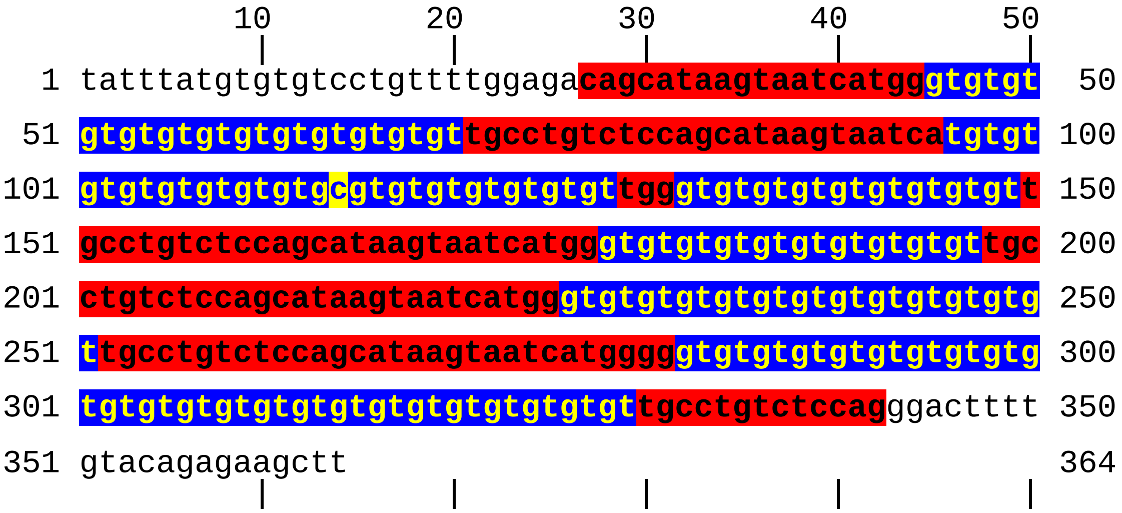
5.1.6. GenBank locus: BTU75906

Species: *Bos taurus* (cow)

Sequence annotation. This sequence contains another allele for the BTGL1 minisatellite region (additional details for the BTGL1 allele in the BOVTGN sequence are given in Section 5.1.4). This allele has essentially the same 29 bp non-SSR pattern and nested SSR as the BOVTGN allele although here the copies vary in length from 48 to 79 bp.

Visual analysis. A description of the VLTR region is confirmed visually. This second BTGL1 allele is a member of the sequence collection due to identification challenges associated with one of the nested SSR regions; the SSR region contains three sequence differences, two substitutions and one insertion or deletion. These imperfections in the middle of the SSR create three perfect SSR regions within one imperfect SSR region. Thus, this region serves as a test for VLTR analysis to ensure that analysis of imperfect SSR regions generates a good region characterization.

Algorithmic performance. The algorithm locates the SSR regions within the larger VLTR region. Figure 5.1.6.A displays the entire sequence and locates the five SSR regions having a **GT** pattern. Figure 5.1.6.B displays the region characterization for the VLTR region. The second SSR region is an imperfect SSR containing three perfect SSR regions. Initial analysis of this region tried to assign each perfect SSR to the role of the imperfect SSR thereby creating three "copies" for one actual copy. As such, this region serves to ensure that the analysis procedure is not too rigid when processing nested SSRs.



```
               10        20        30        40        50
                |         |         |         |         |
  1 tatttatgtgtgtcctgttttggagacagcataagtaatcatgggtgtgt  50
 51 gtgtgtgtgtgtgtgtgtgttgcctgtctccagcataagtaatcatgtgt 100
101 gtgtgtgtgtgtgcgtgtgtgtgtgtgttgggtgtgtgtgtgtgtgtgtt 150
151 gcctgtctccagcataagtaatcatgggtgtgtgtgtgtgtgtgtgttgc 200
201 ctgtctccagcataagtaatcatgggtgtgtgtgtgtgtgtgtgtgtgtg 250
251 ttgcctgtctccagcataagtaatcatgggggtgtgtgtgtgtgtgtgtg 300
301 tgtgtgtgtgtgtgtgtgtgtgtgtgtgttgcctgtctccagggactttt 350
351 gtacagagaagctt                                     364
                |         |         |         |         |
```

Figure 5.1.6.A: The entire sequence for GenBank locus BTU75906. The sequence contains five SSR regions having a **GT** pattern. All SSRs occur within the VLTR region.

```
              10        20        30
               |         |         |
   1 tatttatgtgtgtcctgttttggaga   26
               |         |         |
              10        20        30        40        50
               |         |         |         |         |
>1X:{tgcctgtctccagcataagtaatcatgg{gt}*}*
  27           cagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgtg  65
  71 tgcctgtctccagcataagtaatcatgt gtgtgtgtgtgtgtgcgtgtg 119
 150 tgcctgtctccagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgt  197
 198 tgcctgtctccagcataagtaatcatgg gtgtgtgtgtgtgtgtgtgtg 246
 252 tgcctgtctccagcataagtaatcatgg gggtgtgtgtgtgtgtgtgtg 300
 330 tgcctgtctccag                                      342
               |         |         |         |         |
              10        20        30
               |         |         |

  65 tgtgt                          70
 120 tgtgtgtgttgggtgtgtgtgtgtgtgtgt 149
 197                                197
 247 tgtgt                          251
 301 tgtgtgtg tgtgtgtgtgtgtgtgtgtgt 329
 342                                342
               |         |         |
              10        20        30
               |         |         |
 343 ggacttttgtacagagaagctt  364
               |         |         |
```

Figure 5.1.6.B: The region characterization for the VLTR region in the BTU75906 sequence.

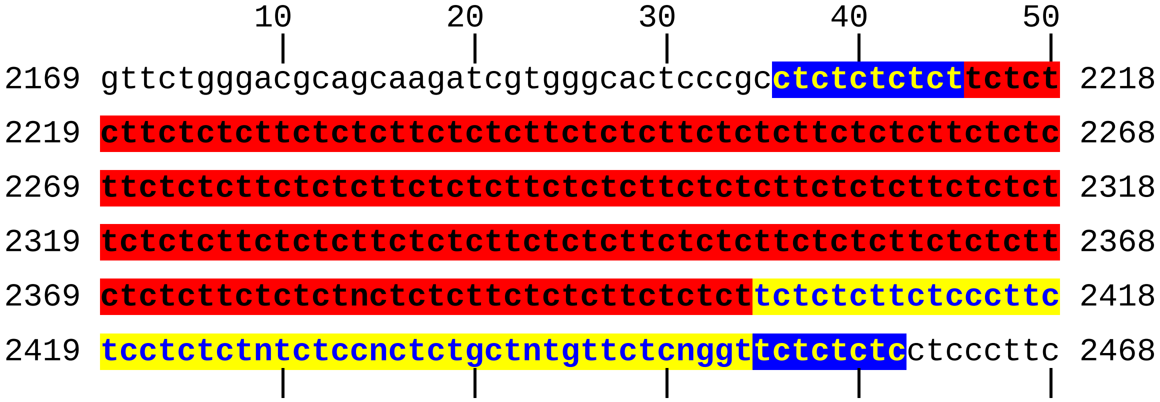
5.1.7. GenBank locus: DMPUGDMG1

Species: *Drosophila melanogaster* (fruitfly).

Sequence annotation. The fruitfly genome contains a gene encoding for the eye pigmentation enzyme, methylenetetrahydrofolate dehydrogenase-methenyl-tetrahydrofolate cyclohydroase-formyltetrahydrofolate synthetase (MTHFD). The allele, *pugilist^{Dominant}* (*pug^{D}*), exhibits a dominant mutant phenotype and consists of a one kbp piece of highly repetitive DNA fused to the N-terminus of the wildtype MTHFD (Rong & Golic 1998). Furthermore, the severity of the mutant phenotype depends on the length of the repetitive stretch requiring at least 500 bp to exhibit the dominant phenotype (Titen *et al*. 2002). This sequence contains a fragment of the allele including all of the wildtype sequence and a portion of the repeat. The repetitive sequence is a well-conserved tandem repeat region having a **TCTCTCT** pattern.

Visual analysis. The sequence contains a region with a high **C**/**T** alteration. Three distinct regions occur within the region. The leading edge has a short SSR region having a **CT** pattern. Next, a large region exists having 25 exact, consecutive copies of a **TCTCTCT** pattern that deteriorates into a non-structured nucleotide collection of **T**, **C** and the ambiguous **N** character representing any nucleotide. Finally, the trailing edge ends in a short SSR region having a **CT** pattern.

Algorithmic performance. The algorithm locates the two SSR regions having a **CT** pattern that flank the tandem repeat region having the simple pattern structure **TCTCTCT**. Figure 5.1.7.A displays a portion of the entire sequence that includes the two SSR regions having a **CT** pattern



```
             10        20        30        40        50
              |         |         |         |         |
2169 gttctgggacgcagcaagatcgtgggcactcccgcctctctctcttctct 2218
2219 cttctctcttctctcttctctcttctctcttctctcttctctcttctctc 2268
2269 ttctctcttctctcttctctcttctctcttctctcttctctcttctctct 2318
2319 tctctcttctctcttctctcttctctcttctctcttctctcttctctctt 2368
2369 ctctcttctctctnctctcttctctcttctctcttctctcttctcccttc 2418
2419 tcctctctntctccnctctgctntgttctcnggttctctctcctcccttc 2468
              |         |         |         |         |
```

Figure 5.1.7.A: A portion of the sequence display for GenBank locus DMPUGDMG1.

and the tandem repeat region that occurs between them. Figure 5.1.7.B displays the region characterization for the **TCTCTCT** pattern based tandem repeat region. The core of the region contains 25 consecutive copies that have perfect conservation. The sequence trailing the core degenerates. Six copies from the degenerate section are included in the region before the degeneracy becomes too severe.

Two challenges exist for characterizing this region; determining the pattern and determining the edges. Visual rendering deems ***T(CT)$_3$*** as the ideal pattern due to the 25 exact tandem copies. The **CT** pattern is also promising as the region is bounded by SSRs having this two-base pattern. Another pattern is a complex pattern structure that nests the **CT** pattern, ***T(CT)$_*$***.

Visual analysis determines clear edges for the section having perfect conservation of the **TCTCTCT** pattern. The leading edge of the region corresponds to the leading edge of the 25 perfect copies but it is uncertain where the trailing edge is due to a gradual deterioration of the pattern into a non-structured nucleotide collection mostly composed of **T**, **C** and the ambiguous **N**

```
             10        20        30        40        50
              |         |         |         |         |
2147 cttggggttgcctggagttaattcgccgttctggagtagagatcgccgga 2146
2197 gccagggctgtggttctgggacgcagcaagatcgtgggcactcccgcctc 2206
              |         |         |         |         |
             10                            10
              |                             |
>1X:{cttctcy}*                >1X:{cttctcy}*
2207   tctct   2211           2317 cttctct   2323
2212 cttctct   2218           2324 cttctct   2330
2219 cttctct   2225           2331 cttctct   2337
2226 cttctct   2232           2338 cttctct   2344
2233 cttctct   2239           2345 cttctct   2351
2240 cttctct   2246           2352 cttctct   2358
2247 cttctct   2253           2359 cttctct   2365
2254 cttctct   2260           2366 cttctct   2372
2261 cttctct   2267           2373 cttctct   2379
2268 cttctct   2274           2380 ctnctct   2386
2275 cttctct   2281           2387 cttctct   2393
2282 cttctct   2288           2394 cttctct   2400
2289 cttctct   2295           2401 cttctct   2407
2296 cttctct   2302           2408 cttctcc   2414
2303 cttctct   2309           2415 cttctcc   2421
                                            |
2310 cttctct   2316
              |
             10        20        30        40        50
              |         |         |         |         |
2422 tctctntctccnctctgctntgttctcnggttctctctcctcccttc 2468
              |         |         |         |         |
```

Figure 5.1.7.B: Region characterization for a tandem repeat region having a **TCTCTCT** pattern that occurs in the DMPUGDMG1 sequence.

character representing any nucleotide. So at one extreme, the trailing edge corresponds to the trailing edge of the 25 perfect copies and at the other extreme, the region includes the entire degenerate section. This analysis chooses an edge between the extremes such that several copies of the degenerate section occur in the final region characterization. Furthermore, the entire sequence display makes clear that the degeneracy is an imperfect match to the **CT**-pattern based SSR that secedes the degenerate region.

5.1.8. GenBank locus: DMU41809

Species: *Drosophila melanogaster* (fruitfly).

Sequence annotation. The sequence contains a portion of the cyclin J gene (CycJ) followed by two direct overlapping 27 bp copies (Finley & Brent 1994, Finley *et al*. 1996).

Visual analysis. Several short SSRs occur in the sequence as well as two tandem repeat regions each having approximately two copies.

Algorithmic performance. The identification algorithm locates four SSR regions and two tandem repeat regions. Figure 5.1.8.A displays the SSR regions identified in the sequence. The one has a **TCG** pattern with three perfect copies and one imperfect copy. Another has an **A** pattern. The two remaining SSRs have an **AT** pattern and occur within the 15 bp tandem repeat region. Figure 5.1.8.B displays a region characterization for a perfectly conserved tandem repeat region having slightly over two copies of a 22 bp pattern. Figure 5.1.8.C displays the region characterization of a tandem repeat region that has slightly less than two copies of a 15 bp pattern and that contains the two **AT** pattern based SSRs. Although the pattern exhibits perfect conservation, the region is considered to be a marginal tandem repeat region since two complete copies do not occur.

```
            10        20        30        40        50
             |         |         |         |         |
  31 gctgccgagcagaacatcttcgtcgtcgaccgcaaattgaagaagacttg   80
             |         |         |         |         |
1125 aaaccagcacccattgaaacacgccaaagtggagagtgtagccaaagatt 1174
1175 agtaaaagggaaaaaagaaactattaccatttggaggaaatccaaatagc 1224
1225 ttagaagtgaaatccaaatagcttagaagtgaaatgacatccataacgta 1274
1275 gatatggtcttgttcaattgagttgaacctcacccatataataatcgtga 1324
1325 tgttatgctttaatatatatgtatttaatatatatttgaaattatcttat 1374
1375 aaacttataaaatacaatatgtcaaaaactctaaaaaaaaaaaaaaaaaa 1424

             |         |         |         |         |
```

Figure 5.1.8.A: A portion of the sequence display for GenBank locus DMU41809.

```
            10        20        30        40        50
             |         |         |         |         |
1201 tgtagccaaagattagtaaaagggaaaaaagaaactattaccatttggag 1210
             |         |         |         |         |
            10        20        30
             |         |         |
>1X:{aaatagcttagaagtgaaatcc}*
1211                gaaatcc      1217
1218 aaatagcttagaagtgaaatcc      1239
1240 aaatagcttagaagtgaaat        1259
             |         |         |
            10        20        30        40        50
             |         |         |         |         |
1260 gacatccataacgtagatatggtcttgttcaattgagttgaacctcaccc 1309
             |         |         |         |         |
```

Figure 5.1.8.B: Region characterization for a tandem repeat region having a 22 bp pattern that occurs in the DMU41809 sequence.

```
             10        20        30        40        50
              |         |         |         |         |
1283 cttgttcaattgagttgaacctcacccatataataatcgtgatgttatgc 1332
              |         |         |         |         |
             10        20
              |         |
>1X:{tatgtatttaatata}*
1333       tttaatata   1341
1342 tatgtatttaatata   1356
1357 tat               1359
              |         |
             10        20        30        40        50
              |         |         |         |         |
1360 ttgaaattatcttataaacttataaaatacaatatgtcaaaaactctaaa 1409
              |         |         |         |         |
```

Figure 5.1.8.C: Region characterization for a tandem repeat region having a 15 bp pattern that occurs in the DMU41809 sequence.

5.1.9. GenBank locus: ECTRNYSU

Species: *Escherichia coli* (bacteria)

Sequence annotation. The sequence contains the *tyrT* operon. The primary product of the operon is the gene $\mathrm{tRNA}_1^{\mathrm{Tyr}}$. The sequence immediately following the tRNA gene is a well-conserved tandem repeat region composed of three copies of a 178 bp pattern (Egan & Landy 1978). Early work determined that this tRNA gene uses a rho-dependent termination process that is encoded within the tandem repeat region (Kupper *et al*. 1978). Each copy of the repeat encodes a termination site with the middle copy representing the primary termination site. Later work found a secondary RNA product (rtT RNA) within the transcript in the first copy of the tandem repeat region (Bösl & Kersten 1991). Furthermore, the rtT RNA contains a complete reading frame for a small basic peptide, Tpr, having unusual bacterial amino acid composition (Rossi *et al*. 1981). In addition, the $\mathrm{tRNA}_1^{\mathrm{Tyr}}$ gene in this sequence is a mutant, amber suppressor $\mathrm{tRNA}_1^{\mathrm{Tyr}}$ *su3*$^+$ allele. Although the biology is complex, the structure of the tandem repeat region is straightforward.

Visual analysis. Slightly over three copies of a 178 bp pattern are visually confirmed for this sequence. The nested SSR regions have an **ACC** pattern. Each SSR region has the same sequence; three perfect copies flanked on either side by an imperfect copy.

Algorithmic performance. The algorithm locates five SSR regions and the tandem repeat region having the 178 bp pattern. Figure 5.1.9.A is a sequence display that locates the SSR regions. One SSR region has an **A** pattern. The other four SSR regions occur within the larger tandem repeat region. They have an **ACC** pattern and each region has the same sequence, **CCC CCA CCA CCA TCA C**, which contains three perfect **ACC** copies with one imperfect copy on either side for a total of five copies. Figure 5.1.9.B displays the region characterization for the tandem repeat region using a pattern having a 162 bp non-SSR portion and a **ACC**-SSR pattern portion.

```
               10        20        30        40        50
                |         |         |         |         |
 501 gatatgatgcgccccggcttcccgataagggagcaggccagtaaaagcat  550
 551 taccccgtggtggggttcccgagcggccaaagggagcagactctaaatct  600
 601 gccgtcatcgacttcgaaggttcgaatccttccccccaccaccatcacttt  650
 651 caaaagtccctgaacttcccaacgaatccgcaattaaatattctgcccat  700
 701 gcggggaaggatgagaagcttcgaccaaggttcgactcgagcgccagcga  750
 751 gagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcagcgctga  800
 801 gtaatccttcccccaccaccatcactttcaaaagtccctgaactctcaag  850
 851 cgaatccgcaatcaaatattctgcccatgcggggaaggatgagaagcttc  900
 901 gactaaggttcggctcgagcgtcagcgagagagcgttgccgcaggcaacg  950
 951 acccgaagggcgaagcgcgcagcgctgagtaatccttcccccaccaccat 1000
1001 cactttcaaaagccccggaattctcaaacgaatccgcaatcaaatattct 1050
1051 gcccaagcggggaaggatgagaagcttcgaccaaggttcgacttgagcgc 1100
1101 cagcgagagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcag 1150
1151 cgctgagtaatccttcccccaccaccatcgcttccttaaataatcaaaac 1200
1201 ctgttactccgagctaattttgcccgatggcaccacgtttaccgggccta 1250
1251 caatatcctcccctgtaggtcggataagcgaagcgcatccgacaactcac 1300
                |         |         |         |         |
```

Figure 5.1.9.A: A portion of the sequence display for GenBank locus ECTRNYSU.

Figure 5.1.9.B (on next page): Region characterization for the nested tandem repeat region that occurs in the ECTRNYSU sequence.

```
             10        20        30        40        50
              |         |         |         |         |
 610 ggccaaagggagcagactctaaatctgccgtcatcgacttcgaaggttcg 624
              |         |         |         |         |
             10        20        30        40        50
              |         |         |         |         |
>1X:{tttcaaaagyccckgaaytyycmarcgaatccgcaatyaaatattctgcc

 625                                                    625
 648 tttcaaaagtccctgaacttcccaacgaatccgcaattaaatattctgcc  697
 826 tttcaaaagtccctgaactctcaagcgaatccgcaatcaaatattctgcc  875
1004 tttcaaaagccccggaattctcaaacgaatccgcaatcaaatattctgcc 1053
              |         |         |         |         |
             60        70        80        90       100
              |         |         |         |         |
>1X: catgcggggaaggatgagaagcttcgacyaaggttcgrctcgagcgycag

 625                                                    625
 698 catgcggggaaggatgagaagcttcgaccaaggttcgactcgagcgccag  747
 625 catgcggggaaggatgagaagcttcgactaaggttcggctcgagcgtcag  647
1054 caagcggggaaggatgagaagcttcgaccaaggttcgacttgagcgccag 1103
              |         |         |         |         |
            110       120       130       140       150
              |         |         |         |         |
>1X: cgagagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcagcgc

 625                                                    625
 748 cgagagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcagcgc  797
 926 cgagagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcagcgc  975
1104 cgagagagcgttgccgcaggcaacgacccgaagggcgaagcgcgcagcgc 1153
              |         |         |         |         |
            160       170       180
              |         |         |
>1X: tgagtaatcctt{ccc}*}*
 625      aatcctt cccccaccaccatcac  647
 798 tgagtaatcctt cccccaccaccatcac  825
 976 tgagtaatcctt cccccaccaccatcac 1003
1154 tgagtaatcctt cccccaccacc      1176
              |         |         |
             10        20        30        40        50
              |         |         |         |         |
1177 atcgcttccttaaataatcaaaacctgttactccgagctaattttgcccg 1226
              |         |         |         |         |
```

5.1.10. GenBank locus: HSVDJSAT

Species: *Homo sapiens* (human)

Sequence annotation. This sequence contains a minisatellite region composed of 36 closely related 9- and 10-mers. The three primary patterns are **CTGGGAGAGG**, **CTGGGAGAG** and **CTGGGATTG**, denoted **1**, **2** and **3** respectively (Boan *et al*. 1997). The patterns combine to form the 11 copy pattern **1 2 1 3 1 2 1 2 1 3 1**. *In vitro* intramolecular homologous recombination experiments suggest that this minisatellite could be a hotspot involved in meiotic homologous recombination (Boan *et al*. 1998). In addition, the region is flanked by two *Chi* motifs (**GCTGGTGG**) and has sequence identity to recombination signals in immunoglobulin VDJ joining and sequence similarity to immunoglobulin switch regions (Boan *et al*. 1997).

Visual analysis. The primary patterns and pattern structure of this region described above is confirmed visually.

Algorithmic performance. The algorithm locates numerous SSR regions (not detailed here) and a multi-periodic tandem repeat (MPTR) region with a 10 bp base pattern. Figure 5.1.10.A displays the region characterization for the base pattern. The alignment is moderate to well conserved with non-conserved columns indicating cross-copy patterns. Visual rendering of the region characterization observes an alternation between 9 bp and 10 bp copies. These alternating copies combine to form a 19 bp pattern. Figure 5.1.10.B displays the region characterization for a 2X pattern representing two base patterns. The 9/10 bp copy alternation is not sufficient to represent the pattern. Although the 10 bp pattern occurs after every 9 bp copy, the 9 bp patterns

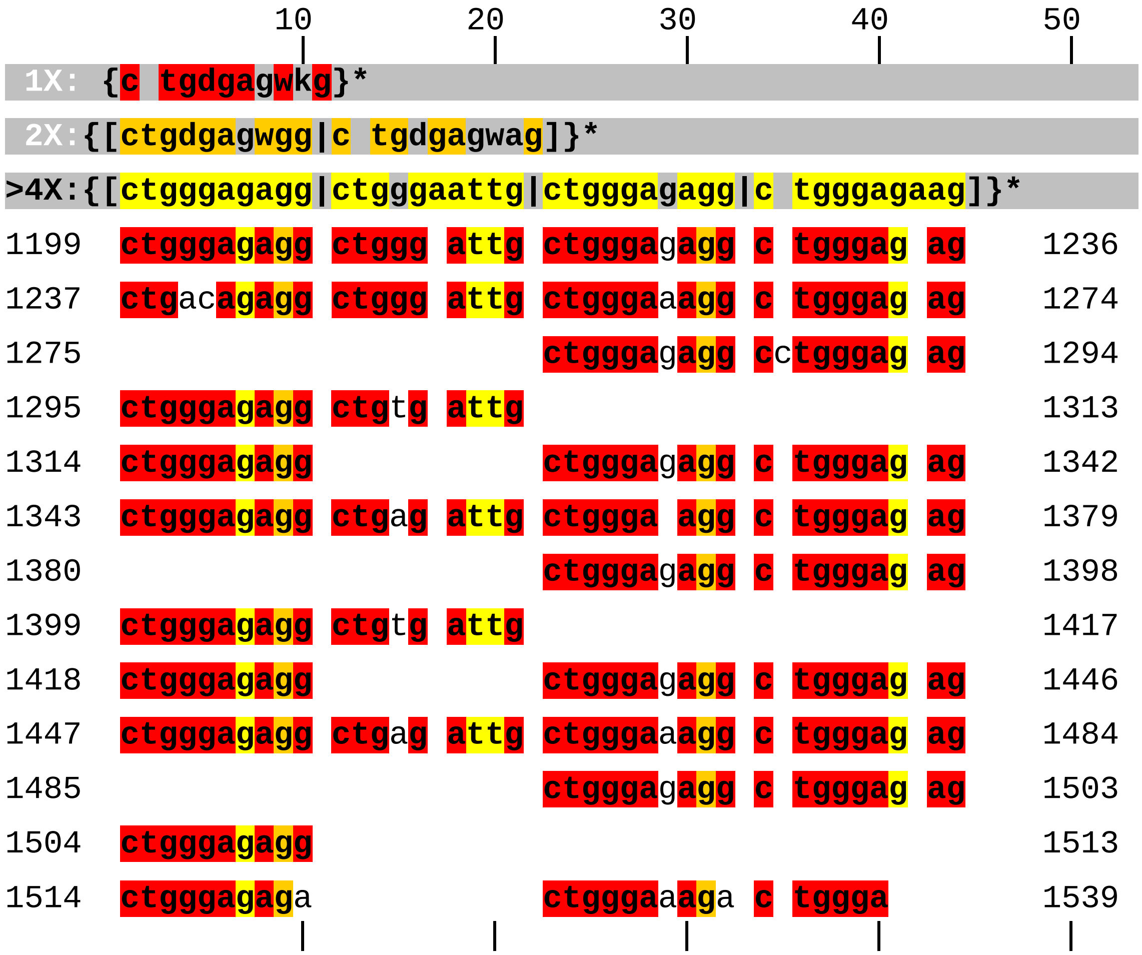
```
             10        20        30        40        50
              |         |         |         |         |
1185 ctaggacacccagcagggcagtgttgagagtgagcatcctggcagggctg 1194
              |         |         |         |         |
             10        20                  10        20
              |         |                   |         |
>1X:{c tgdgagwkg}*            >1X:{c tgdgagwkg}*
1195        gagg   1198       1371 c tgggaga g   1379
1199 c tgggagagg   1208       1380 c tgggagagg   1389
1209 c tggga ttg   1217       1390 c tgggaga g   1398
1218 c tgggagagg   1227       1399 c tgggagagg   1408
1228 c tgggaga g   1236       1409 c tgtga ttg   1417
1237 c tgacagagg   1246       1418 c tgggagagg   1427
1247 c tggga ttg   1255       1428 c tgggagagg   1437
1256 c tgggaaagg   1265       1438 c tgggaga g   1446
1266 c tgggaga g   1274       1447 c tgggagagg   1456
1275 c tgggagagg   1284       1457 c tgaga ttg   1465
1285 cctgggaga g   1294       1466 c tgggaaagg   1475
1295 c tgggagagg   1304       1476 c tgggaga g   1484
1305 c tgtga ttg   1313       1485 c tgggagagg   1494
1314 c tgggagagg   1323       1495 c tgggaga g   1503
1324 c tgggagagg   1333       1504 c tgggagagg   1513
1334 c tgggaga g   1342       1514 c tgggagaga   1523
1343 c tgggagagg   1352       1524 c tgggaaaga   1533
1353 c tgaga ttg   1361       1534 c tggga       1539
                                            |         |
1362 c tggga agg   1370
              |         |
             10        20        30        40        50
              |         |         |         |         |
1540 aagatggcataggccttgagccaggagtgtgagttcatgaagataggctg 1589
              |         |         |         |         |
```

Figure 5.1.10.A: Region characterization for the base pattern of the MPTR region that occurs in the HSVDJSAT sequence.

```
              10        20        30
               |         |         |
 1X: {c tgdgagwkg}*
>2X:{[ctgdgagwgg|c tgdgagwag]]}*
1195        gagg                  1198
1199  ctgggagagg c tggga ttg  1217
1218  ctgggagagg c tgggag ag  1236
1237  ctgacagagg c tggga ttg  1255
1256  ctgggaaagg c tgggag ag  1274
1275  ctgggagagg cctgggag ag  1294
1295  ctgggagagg c tgtga ttg  1313
1314  ctgggagagg              1323
1324  ctgggagagg c tgggag ag  1342
1343  ctgggagagg c tgaga ttg  1361
1362  ctggga agg c tgggag ag  1379
1380  ctgggagagg c tgggag ag  1398
1399  ctgggagagg c tgtga ttg  1417
1418  ctgggagagg              1427
1428  ctgggagagg c tgggag ag  1446
1447  ctgggagagg c tgaga ttg  1465
1466  ctgggaaagg c tgggag ag  1484
1485  ctgggagagg c tgggag ag  1503
1504  ctgggagagg              1513
1514  ctgggagaga              1523
1524  ctgggaaaga c tggga aag  1542
1543  atggcatagg              1552
1553  c                       1553
              |         |         |
```

Figure 5.1.10.B: Region characterization for the integer multiple of 2 having a 19 bp pattern associated with the MPTR region in the HSVDJSAT sequence.

alternate forming a four-copy oscillation of **CTGGGAGAGG CTGGGAGAG CTGGGAGAGG CTGGGATTG** to form a 38 bp pattern. Figure 5.1.10.C displays the region characterization for the 4X pattern. An occasional hiccup in the pattern structure inserts an extra **CTGGGAGAGG** in the copy oscillation creating a larger pattern structure every 11 copies. Figure 5.1.10.F displays the region characterization for the 11X pattern. This 11-copy oscillation forms a 104 bp pattern and corroborates with single position substitutions in another well-conserved column. Finally,



```
              10        20        30        40        50
               |         |         |         |         |
 1X: {c tgdgagwkg}*
 2X:{[ctgdgagwgg|c tgdgagwag]}*
>4X:{[ctgggagagg|ctgggaattg|ctgggagagg|c tgggagaag]}*
1199  ctgggagagg ctggg attg ctgggagagg c tgggag ag    1236
1237  ctgacagagg ctggg attg ctgggaaagg c tgggag ag    1274
1275                       ctgggagagg cctgggag ag    1294
1295  ctgggagagg ctgtg attg                          1313
1314  ctgggagagg            ctgggagagg c tgggag ag    1342
1343  ctgggagagg ctgag attg ctggga agg c tgggag ag    1379
1380                       ctgggagagg c tgggag ag    1398
1399  ctgggagagg ctgtg attg                          1417
1418  ctgggagagg            ctgggagagg c tgggag ag    1446
1447  ctgggagagg ctgag attg ctgggaaagg c tgggag ag    1484
1485                       ctgggagagg c tgggag ag    1503
1504  ctgggagagg                                     1513
1514  ctgggagaga            ctgggaaaga c tggga        1539
               |         |         |         |         |
```

Figure 5.1.10.C: Region characterization for an integer multiple of 4 having a 38 bp pattern associated with the MPTR region in the HSVDJSAT sequence.

analysis indicates significant cross-copy patterns every 5 and 6 copies. Figure 5.1.10.D displays the region characterization for the 5X pattern (48 bp) and Figure 5.1.10.E displays the region characterization for the 6X pattern (58 bp). Although there is a degree of recurrence at these multiples, it is clear that they reflect the 11-copy oscillation.

This region is difficult to analyze algorithmically. Identification of regions having this level of pattern structure complexity significantly stretched the development of the algorithm. Several issues contribute to the challenge including a periodicity stronger than the base periodicity, two base periodicities, three, closely related copy patterns, and occasional duplicate copies in the pattern structure that create a more complex structure. As a result, the region

```
              10        20        30        40        50
               |         |         |         |         |
 1X: {c tgdgagwkg}*
>5X:{[c tgggagaag|ctgggagagg|ctgtgaattg|ctgggagagg|ctggg
1190                                                     1190
1192  c t ggag  g ctgggagagg ctg ggattg            ctggg 1222
1228  c tgggag ag ctgacagagg ctg ggattg            ctggg 1260
1266                                  ctgggaga g ctggg 1279
1285  cctgggag ag ctgggagagg ctgtg attg ctgggagagg ctggg 1328
1334  c tgggag ag ctgggagagg ctgag attg            ctggg 1366
1371                                  ctgggaga g ctggg 1384
1390  c tgggag ag ctgggagagg ctgtg attg ctgggagagg ctggg 1432
1438  c tgggag ag ctgggagagg ctgag attg            ctggg 1470
1476                                  ctgggaga g ctggg 1489
1495                                  ctgggaga g ctggg 1508
1514                                  ctgggagaga ctggg 1528
1534  c tggga aag                                  atggc 1547
1553              c                                      1553
               |         |         |         |         |
              60
               |
 1X:
>5X:   agagg]}*
1190      gg   1191
1223   agagg   1227
1261   aaagg   1265
1280   agagg   1284
1329   agagg   1333
1367   a agg   1370
1385   agagg   1389
1433   agagg   1437
1471   aaagg   1475
1490   agagg   1494
1509   agagg   1513
1529   aaaga   1533
1548   atagg   1552
1553           1553
               |
```

Figure 5.1.10.D: Region characterization for an integer multiple of 5 having a 48 bp pattern associated with the MPTR region in the HSVDJSAT sequence.

characterization is not "clean." Rather, each level of recurrence captures a portion of the periodicity evident in the region and must handle significant deviations.

Figure 5.1.10.E (on next page): Region characterization for an integer multiple of 6 having a 58 bp pattern associated with the MPTR region in the HSVDJSAT sequence.

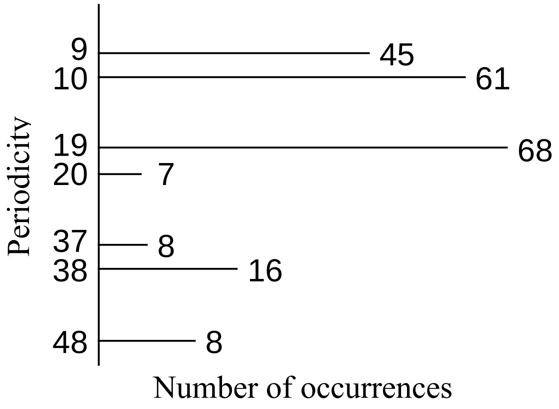
```
              10        20        30        40        50
               |         |         |         |         |
 1X: {c tgdgagwkg}*
 2X:{[ctgdgagwgg|c tgdgagwag]}*
>6X:{[c tgdgagaag|c  tgdgagagg|c tgdgaattg|ctg dgagagg|
1190                        gg                          1191
1192  c tg gag  g c  tgggagagg c tggg attg              1217
1218              c  tgggagagg                          1227
1228  c tgggag ag c  tgacagagg c tggg attg              1255
1256              c  tgggaaagg                          1265
1266  c tgggag ag c  tgggagagg                          1284
1285  cctgggag ag c  tgggagagg c tgtg attg              1313
1314              c  tgggagagg                          1323
1324              c  tgggagagg                          1333
1334  c tgggag ag c  tgggagagg c tgag attg              1361
1362              c  tggga agg                          1370
1371  c tgggag ag c  tgggagagg                          1389
1390  c tgggag ag c  tgggagagg c tgtg attg              1417
1418              c  tgggagagg                          1427
1428              c  tgggagagg                          1437
1438  c tgggag ag c  tgggagagg c tgag attg              1465
1466              c  tgggaaagg                          1475
1476  c tgggag ag c  tgggagagg                          1494
1495  c tgggag ag c  tgggagagg                          1513
1514              c  tgggagaga                          1523
1524              c  tgggaaaga                          1533
1534  c tggga aag a  tggcatagg                          1552
1553              ccttgagccagg agtgtgagttc atgaagatagg  1586
               |         |         |         |         |

ctgdgagagg|ctgdgagaag]}*
```

no sequence positions align to these alternatives

```
               10        20        30        40        50
                |         |         |         |         |
  1X: {c tgdgagwkg}*
>11X:{[ctgggagagg|ctgtgaattg|ctgggagagg|ctgggagagg|ctggga
 1199  ctgggagagg ctg ggattg            ctgggagagg ctggga 1233
 1295  ctgggagagg ctgtg attg ctgggagagg ctgggagagg ctggga 1339
 1399  ctgggagagg ctgtg attg ctgggagagg ctgggagagg ctggga 1443
 1504  ctgggagagg                                         1513
 1514  ctgggagaga                                         1523
                |         |         |         |         |
               60        70        80        90       100
                |         |         |         |         |
  1X:
>11X: gaag|ctgggagagg|ctgagaattg|ctgggaaagg|ctgggagaag|c
1234  g ag ctgacagagg ctg ggattg ctgggaaagg ctgggag ag c 1275
1340  g ag ctgggagagg ctgag attg ctggg aagg ctgggag ag c 1380
1444  g ag ctgggagagg ctgag attg ctgggaaagg ctgggag ag c 1485
1513                                                     1513
1524                             ctgggaaaga              1533
               |         |         |         |         |
             110       120       130
               |         |         |
  1X:
>11X: tgggagagg|c tgggagaag]}*
1276  tgggagagg cctgggag ag   1294
1381  tgggagagg c tgggag ag   1398
1486  tgggagagg c tgggag ag   1503
1513                          1513
1534            c tggga aag   1542
               |         |         |
```

Figure 5.1.10.F: Region characterization for an integer multiple of 11 having a 104 bp pattern associated with the MPTR region in the HSVDJSAT sequence.

Figure 5.1.10.G: Periodicity signature for the MPTR region in the HSVDJSAT sequence.



The region periodicity signature has many periodicities as shown in Figure 5.1.10.G. The strong periodicity at 19 is most significant as it corresponds to two copies, a 9 bp copy and a 10 bp copy. This scenario where he strongest periodicity is larger than the base periodicity is not unique but has been observed in several MPTR regions. The current analysis looks for significant periodicities in the periodicity signature that are factors of the strongest periodicity and chooses this as the base periodicity. In this manner, not only is the above scenario taken into account but also the strongest periodicity is not lost since it becomes an integer multiple of the base periodicity.

This region has multiple base periodicities, most notably, the 9 and 10 periodicities. Initial analysis chose a single base periodicity but, neither a periodicity of 9 nor a periodicity of 10 results in an integer multiple of 11 to represent the 104 bp pattern (12X @9 bp or 10X @10 bp). Only the combination of both the 9 and 10 bp periodicities selects the correct integer multiple (5 @ 9 bp pattern + 6 @ 10 bp pattern - 1 = 11X : 104 bp pattern). Thus, it became necessary to allow multiple base periodicities. This decision had a ripple affect throughout the analysis. Most notably, the procedure for pairing a periodicity to an integer multiple became a multi-step, complex procedure.

The three related though distinct copy patterns are fairly well conserved throughout the region. The patterns combine to form many complex structures, including by interleaving 9 and 10 bp patterns (2X), by alternating the 9 bp patterns to form a 9 10 9′ 10 pattern (4X) and by inserting an extra 10 bp pattern to form a 10 9′ 10 9 10 9′ 10 9′ 10 9 10 pattern (11X).

5.1.11. GenBank locus: MM102B5

Species: *Mus musculus* (mouse)

Sequence annotation. This sequence contains a portion of a gamma satellite region. The sequence results from an experiment designed to integrate MS32 tandem repeat regions. This sequence represents a portion of a transgenic lineage (102B) and represents the 5′ flanking region to the transgene comprised of gamma satellite DNA. The gamma satellite region is a tandem repeat region having a 234 bp pattern consisting of two subunits of length 116 and 118 that likely evolved from three 9 bp sequences, **GAAAAATGA**, **GAAAAAACT** and **GAAAAACGT** (Hastie 1989). More precisely, the gamma satellite 234 bp pattern is composed of eight subunits α_1 β_1 α_2 β_2 α_3 β_3 α_4 β_4 where the α subunit has a length of 28 bp and the β subunit has a length of 30 bp.

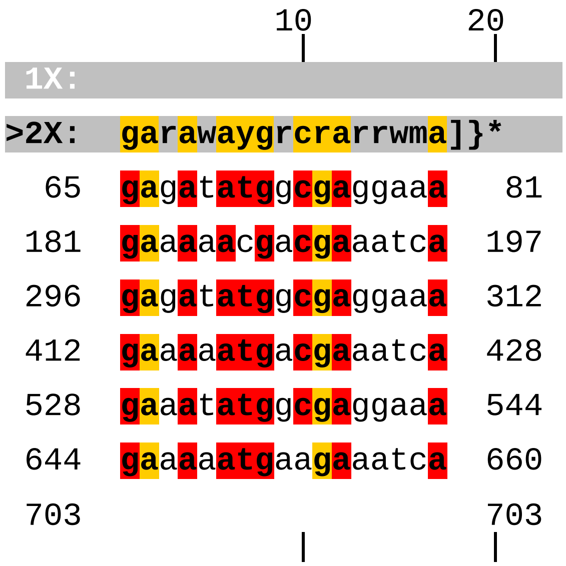
Visual analysis. A tandem repeat region having a 58 bp pattern recurs within the sequence and represents approximately one-fourth the size of the gamma satellite pattern. This corresponds to a combined α β subunit. Furthermore, cross-copy patterns are evident across every two copies and every four copies. Thus, this region is a multi-periodic tandem repeat region.

Algorithmic performance. The algorithm locates a multi-periodic tandem repeat (MPTR) region having a 58 bp base pattern with integer multiples of 2 and 4. Figure 5.1.11.A displays the region characterization for the base pattern. Figure 5.1.11.B displays the region characterization for the integer multiple of 2 having a 116 bp pattern. Figure 5.1.11.C displays the region characterization for the integer multiple of 4 having a 232 bp pattern. The conservation level of the base pattern region characterization is good. The region characterization for the integer multiple of 4 reveals a region having almost perfect conservation. Furthermore, the 232 bp pattern corresponds to the pattern for the gamma satellite region.

```
             10        20        30        40        50
              |         |         |         |         |
>1X:{ctgaaaawsrygdmaaatkaraaayrymcactddasgacbtgrrawaygr
   1                                    ggacctggaatatgg   15
  24 ctgaaaatcacggaaaatgagaaatacacactttaggacgtgagatatgg   73
  82 ctgaaaaaggtggaaaattagaaatgtccactgtaggacgtggaatatgg  131
 140 ctgaaaatcatggaaaatgaaaaacatccactagacgacttgaaaaacga  189
 198 ctgaaaaacgtgacaaatgagaaatgcacactgaaggacctg gatatgg  247
 255 ctgaaaatcacgtaaaatgagaaatacacactttaggacgtgagatatgg  304
 313 ctgaaaaaggtggaaaattagaaatgtggactgtaggacgtggaatatgg  362
 371 ctgaaaatcatggaaaatgagaaacatccatttgacgacttgaaaaatga  420
 429 ctgaaaaacgtgacaaatgagaaatgcacactgtaggaactggaatatgg  478
 487 ctgaaaatcacgtaaaatgagaaatacacactttaggacgtgaaatatgg  536
 545 ctgaaaaagttggaaaaattagaatgtccagtgtacgacgtggaatatgg  594
 603 ctgaaaatcatggaaaatgagaaacatccatttgacgacnagaaaaatga  652
 661 ctgaaaaacgtgaaaaatgagaaatgcacactgtagga              698
              |         |         |         |         |
               60
                |
>1X: crarrwma}*
  16 cgagaaaa   23
  74 cgaggaaa   81
 132 caagaaaa  139
 190 cgaaatca  197
 248 cgagaaaa  254
 305 cgaggaaa  312
 363 caagataa  370
 421 cgaaatca  428
 479 cgagaaaa  486
 537 cgaggaaa  544
 595 caagataa  602
 653 agaaatca  660
 698           698
               |
```

Figure 5.1.11.A: Region characterization for the 58 bp base pattern of a MPTR region that occurs in the MM102B5 sequence.

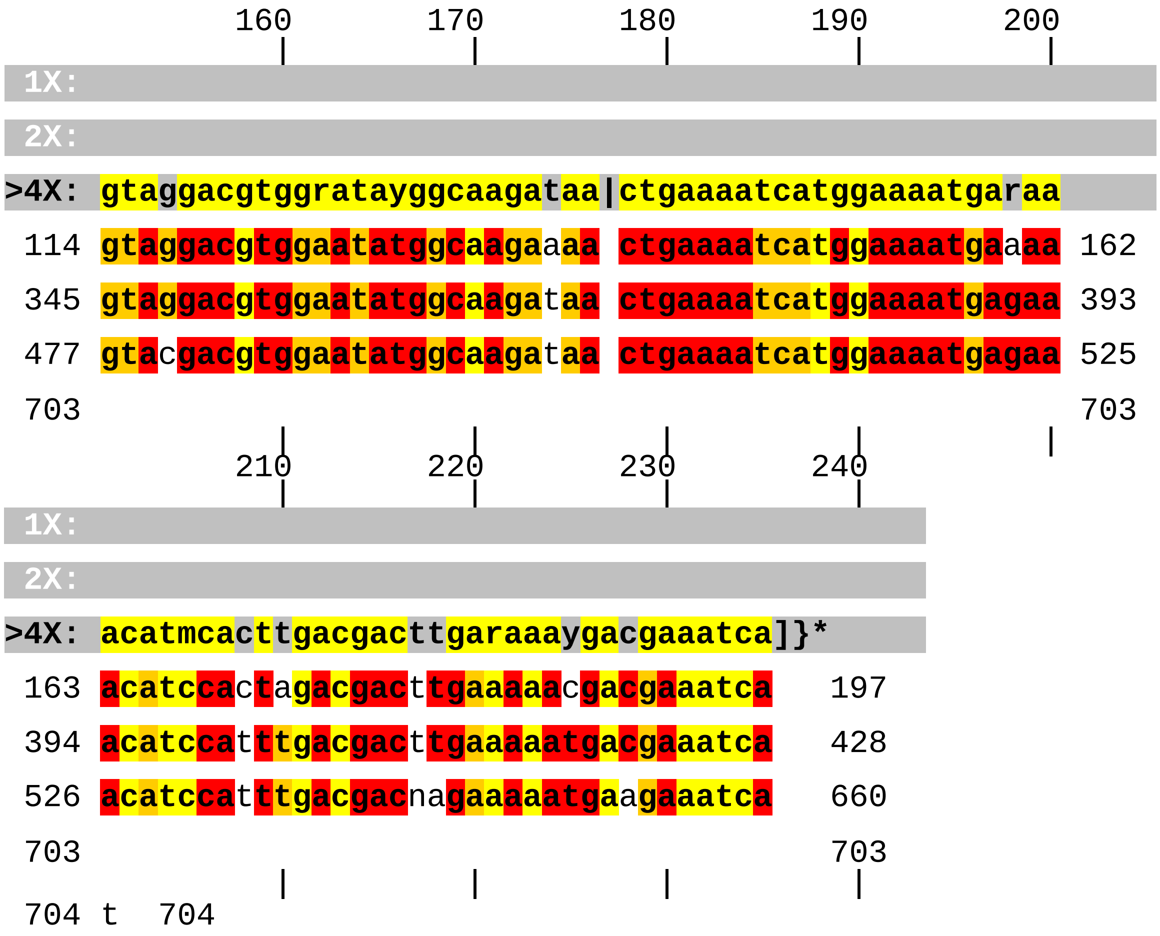
```
              10        20        30        40        50
               |         |         |         |         |
 1X: {ctgaaaawsrygdmaaatkaraaayrymcactddasgacbtgrrawaygr
>2X:{[ctgaaaaasgygdmaaatkaraaaygymcactgdasgacbtggrawaygr
   1                                     ggacctggaatatgg   15
  82  ctgaaaaaggtggaaaattagaaatgtccactgtaggacgtggaatatgg  131
 198  ctgaaaaacgtgacaaatgagaaatgcacactgaaggacctgg atatgg  246
 313  ctgaaaaaggtggaaaattagaaatgtggactgtaggacgtggaatatgg  362
 429  ctgaaaaacgtgacaaatgagaaatgcacactgtaggaactggaatatgg  478
 545  ctgaaaaagttggaaaaattagaatgtccagtgtacgacgtggaatatgg  594
 661  ctgaaaaacgtgaaaaatgagaaatgcacactgtaggagcagg         704
               |         |         |         |         |
              10        20        30        40        50
               |         |         |         |         |
 1X:  crarrwma}*
>2X:  crarrwma|ctgaaaatsaygdmaaatkaraaayaymcacttdasgacbt
  16  cgagaaaa ctgaaaatcacggaaaatgagaaatacacactttaggacgt   64
 132  caagaaaa ctgaaaatcatggaaaatgaaaaacatccactagacgactt  180
 247  cgagaaaa ctgaaaatcacgtaaaatgagaaatacacactttaggacgt  295
 363  caagataa ctgaaaatcatggaaaatgagaaacatccatttgacgactt  411
 479  cgagaaaa ctgaaaatcacgtaaaatgagaaatacacactttaggacgt  527
 595  caagataa ctgaaaatcatggaaaatgagaaacatccatttgacgacna  643
 704                                                      704
               |         |         |         |         |
```



```
                 10        20
                  |         |
 1X:
>2X:  garawaygrcrarrwma]}*
  65  gagatatggcgaggaaa    81
 181  gaaaaacgacgaaatca   197
 296  gagatatggcgaggaaa   312
 412  gaaaaatgacgaaatca   428
 528  gaaatatggcgaggaaa   544
 644  gaaaaatgaagaaatca   660
 703                      703
                  |         |
```

Figure 5.1.11.B: Region characterization for an integer multiple of 2 having a 116 bp pattern associated with the MPTR region in the MM102B5 sequence.

```
                10        20        30        40        50
                 |         |         |         |         |
 1X:  {ctgaaaawsrygdmaaatkaraaayrymcactddasgacbtgrrawaygr
 2X: {[ctgaaaaasgygdmaaatkaraaaygymcactgdasgacbtggrawaygr
>4X: {[ctgaaaaacgtgacaaatgaraaatgcmcactgaaggacctggrataygg
  1                                       ggacctggaatatgg  15
198    ctgaaaaacgtgacaaatgagaaatgcacactgaaggacctgg atatgg  246
429    ctgaaaaacgtgacaaatgagaaatgcacactgtaggaactggaatatgg  478
661    ctgaaaaacgtgaaaaatgagaaatgcacactgtaggagcagg         703
                 |         |         |         |         |
                60        70        80        90       100
                 |         |         |         |         |
 1X:   crarrwma}*
 2X:   crarrwma|ctgaaaatsaygdmaaatkaraaayaymcacttdasgacbt
>4X:   cgagaaaa|ctgaaaatcacgtaaaatgaraaatacmcactttaggacgt
 16    cgagaaaa ctgaaaatcacggaaaatgagaaatacacactttaggacgt  64
247    cgagaaaa ctgaaaatcacgtaaaatgagaaatacacactttaggacgt  295
479    cgagaaaa ctgaaaatcacgtaaaatgagaaatacacactttaggacgt  427
703                                                        703
                 |         |         |         |         |
               110       120       130       140       150
                 |         |         |         |         |
 1X:
 2X:   garawaygrcrarrwma]}*
>4X:   garatayggcgaggaaa|ctgaaaaaggtggaaaattaraaatgtmcact
 65    gagatatggcgaggaaa ctgaaaaaggtggaaaattagaaatgtccact  113
296    gagatatggcgaggaaa ctgaaaaaggtggaaaattagaaatgtggact  344
428    gaaatatggcgaggaaa ctgaaaaagttggaaaaattagaatgtccagt  476
703                                                        703
                 |         |         |         |         |
```



```
          160       170       180       190       200
           |         |         |         |         |
1X:
2X:
>4X: gtaggacgtggratayggcaagataa|ctgaaaatcatggaaaatgaraa
 114 gtaggacgtggaatatggcaagaaaa ctgaaaatcatggaaaatgaaaa 162
 345 gtaggacgtggaatatggcaagataa ctgaaaatcatggaaaatgagaa 393
 477 gtacgacgtggaatatggcaagataa ctgaaaatcatggaaaatgagaa 525
 703                                                    703
           |         |         |         |         |
          210       220       230       240
           |         |         |         |
1X:
2X:
>4X: acatmcacttgacgacttgaraaaygacgaaatca]}*
 163 acatccactagacgacttgaaaaacgacgaaatca   197
 394 acatccatttgacgacttgaaaaatgacgaaatca   428
 526 acatccatttgacgacnagaaaaatgaagaaatca   660
 703                                       703
           |         |         |         |
 704 t  704
```

Figure 5.1.11.C: Region characterization for an integer multiple of 4 having a 232 bp pattern associated with the MPTR region in the MM102B5 sequence.

5.1.12. GenBank locus: MMMSAT5

Species: *Mus musculus* (mouse).

Sequence annotation. This sequence contains a 270 bp tandem repeat region containing a mix of SSR regions having **AC**, **AT** and **GT** patterns.

Visual analysis. A 270 bp region is composed of numerous tandem repeat regions having di-nucleotide patterns. All but one position in the region corresponds to a set of six di-nucleotides. All but ten positions are members of two in the set of six di-nucleotides. The region contains large SSR regions composed of 21.5 copies as well as small regions composed of single di-nucleotide occurrences.

Algorithmic performance. The algorithm identifies a cluster of SSR regions. These are displayed in Figure 5.1.12.A and detailed in Table 5.1.12.A. In addition, a VLTR region is incorrectly identified due to an alteration between SSR regions having **AC** and **AT** patterns. The identification algorithm has not been modified to locate this type of mixed SSR region. Yet, identification of individual SSR regions allows visual identification of the multiple SSR region.

```
                 10        20        30        40        50
                  |         |         |         |         |
  1 ctccagcctacccacattcattacacacaccacacacatatgtgtgtgta  50
 51 ataatatatatatctatatacacacacacacatatatatatacacacaca 100
101 tatatatacacacacacatatatatacacacatatacaccatacacacac 150
151 acacatatatacatatatatatacatacacatatataccatatacacaca 200
201 tgtatatgtatatatatatatatatatatatatatatatatatatatata 250
251 tgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtgtatggaga 300
301 aagtgagtcatgagtaaaagtaatagcttatccagggccacttagctggt 350
351 taggaaaagagtaaatttgagttaacttagacctctgatggcggaggccg 400
401 tgcaagtgcttc                                       412
                  |         |         |         |         |
```

Figure 5.1.12.A: The entire sequence for GenBank locus MMMSAT5.

Sequence Location	Region Pattern	Sequence Location	Region Pattern
23 ..38	AC	100..108	AT
23..30	AC	108..118	AC
31..38	AC	118..126	AT
41 ..49	GT	143..155	AC
54 ..67	AT	155..173	AT
54..63	AT	163..173	AT
70 ..82	AC	198..263	ATATGT
82 ..92	AT	200..213	ATATGT
92 ..100	AC	209..251	AT
95..100	AC	251..293	GT

Table 5.1.12.A: SSRs present in MMMSAT5 as determined by the identification algorithm.

5.1.13. GenBank locus: NC_001133

Species: *Saccharomyces cerevisiae* (yeast)

Sequence annotation. The original sequence (GenBank accession number: L28920) contained a 5 kbp sequence representing a flocculation gene (FLO1) that contains several tandem repeat regions including a large tandem repeat region having a 135 bp pattern (Watari *et. al.* 1994). A larger 55 kbp sequence (GenBank locus: YSCCHR1RAA) contains this region and represents the right arm of chromosome I. The complete chromosome I sequence (GenBank locus: NC_001133) contains two flocculation genes (FLO1 and FLO9), one at each end of the chromosome, that each contain a tandem repeat region having a similar 135 bp pattern (Bussey *et al*. 1995). This analysis focuses on the FLO1 and FLO9 genes present on chromosome I.

The FLO1 gene covers a 4611 bp region and codes for a protein of 1537 amino acids. The gene contains numerous tandem repeat regions including a large tandem repeat region having a 135 bp pattern, a short tandem repeat region having a 153 bp pattern, one having a 27 bp pattern and several other proximal recurrences. The large tandem repeat region is the most striking region as it contains 18 tandem recurrences of the 135 bp pattern.

The FLO9 gene covers a 3969 bp region and codes for a protein of 1323 amino acids. The gene is smaller than the FLO1 gene such that the tandem repeat region having the 153 bp pattern is not present in the gene and the tandem repeat region having a 135 bp pattern has fewer copies. The identification algorithm identifies a multi-periodic tandem repeat region having slightly more than 13 copies of a 135 bp pattern.

Visual analysis. Visual analysis verified the presence of two tandem repeat regions, each having a periodicity of 135 bp but did not confirm the precise cross-copy pattern structure.

Algorithmic performance. The algorithm locates two tandem repeat regions having a 135 bp pattern corresponding to the expected FLO1 and FLO9 genes. Many other SSR and non-SSR tandem repeat regions occur in the sequence and are identified by the algorithm. Here, only the two MPTR regions are discussed.

Both identified regions have a 135 bp base pattern and integer multiples. Figure 5.1.13.A displays the base pattern region characterization for the FLO1 MPTR region. Analysis of this region characterization identifies valid multiples of **2**, **4** and **5**. Figure 5.1.13.B displays the **5X** region characterization. Figure 5.1.13.C displays the base pattern region characterization for the FLO9 MPTR region.

Both the FLO1 and FLO9 genes exhibit moderate conservation. Although the analysis determines that multiples exist, visual analysis of non-conserved columns in the base pattern region characterizations does not indicate cross-copy patterns. Rather, it suggests that a column sequence is likely to have a stretch of identical nucleotides followed by a stretch of another nucleotide and so forth, e.g. column 37, 55, 61, and 78.

Section 5.2.5 describes additional analysis for this chromosome involving non-contiguous regions of similarity that indicates that each of the flocculation genes, FLO1 and FLO9, has a gene shadow. Neither of the gene shadows contains the large tandem repeat region.

```
               10        20        30        40        50
                |         |         |         |         |
204354 tgtcagtaccactacaactacaacggaaccatggaccggtactttcactt 204303
204354 ctacatctactgaaatgaccaccgtcaccggtaccaacggcgttccaact 204303
                |         |         |         |         |
               10        20        30        40        50
                |         |         |         |         |
  >1X:{aactacaactgagccatggaacagcactttttacctctacwtctacygaaw
204304                                                    204304
204354 aactacaactgagccatggaacagcactttttacctctacttctaccgaat 204403
204489 aactacaactgagccatggaacagcactttttacctctacttctaccgaat 204538
204624 gactacaactcagccatggaacgacactttttacctctacttctaccgaat 204673
204759 gactacaactcagccatggaacgacactttttacctctacttctaccgaat 204808
204894 gactacaactcagccatggaacgacactttttacctctacatccactgaaa 204943
205029 gactacacctcagccatggaacgacactttttacctctacatccactgaaa 205078
205164 aactacaactgagccatggaacagcactttttacctctacatccactgaaa 205213
205299 aactacaactcagccatggaacgacactttttacctctacatccactgaaa 205348
205434 gactacaactcagccatggaacgacactttttacctctacatccactgaaa 205483
205569 gactacaactcagccatggaacgacactttttacctctacatccactgaaa 205618
205704 caccaccactgaaccatggactggtactttcacctctacatccactgaga 205753
205839 aaccaccactgaaccatggactggtactttttacttctacatctactgaaa 205888
205974 caccaccactgaaccatggactggtactttttacttctacatctactgaaa 206023
206109 cactacaacggaaccatggaccggtactttcacttctacatctactgaaa 206158
206244 caccaccactgaaccatggactggcactttcacttcgacttccactgagg 206293
206379 caccaccactgaaccatggactggtactttcacttctacatctactgaaa 206428
206514 aaccaccactgaaccatggactggtactttttacttcgacttccactgaaa 206563
                |         |         |         |         |
               60        70        80        90       100
                |         |         |         |         |
  >1X: tgaccacmgtcacyggyaccaayggygtwcsaactgacgaaaccrtcatt
204304                                    gacgaaaccgtcatt 204318
204404 tgaccacagtcactggcaccaatggtgtacgaactgacgaaaccatcatt 204453
204539 tgaccacagtcaccggtaccaatggtttgccaactgatgagaccatcatt 204588
204674 tgaccacagtcaccggtaccaatggtttgccaactgatgagaccatcatt 204723
204809 tgaccacagtcaccggtaccaatggtttgccaactgatgagaccatcatt 204858
204944 tcaccaccgtcaccggtaccaatggtttgccaactgatgagaccatcatt 204993
205079 tgaccaccgtcaccggtaccaacggtttgccaactgatgaaaccatcatt 205128
205214 tgaccaccgtcaccggtaccaacggtttgccaactgatgaaaccatcatt 205263
205349 tgaccaccgtcaccggtaccaacggtttgccaactgatgaaaccatcatt 205398
205484 tcaccaccgtcaccggtaccaccggtttgccaactgatgagaccatcatt 205533
205619 tgaccaccgtcaccggtaccaacggcgttccaactgacgaaaccgtcatt 205768
205754 tgaccaccgtcaccggtactaacggtcaaccaactgacgaaaccgtgatt 205803
205889 tgaccaccattactggaaccaacggcgttccaactgacgaaaccgtcatt 205938
206024 tgaccaccattactggaaccaatggtcaaccaactgacgaaaccgttatt 206073
206159 tgacgcacgtcaccggtaccaacggcgttccaactgacgaaaccgtcatt 206208
206294 ttaccaccatcactggaaccaacggtcaaccaactgacgaaactgtgatt 206343
206429 tgaccaccgtcaccggtactaacggtcaaccaactgacgaaaccgtgatt 206478
206564 tgtctactgtcactggaaccaatggcttgccaactgatgaaactgtcatt 206613
                |         |         |         |         |
              110       120       130       140
                |         |         |         |
  >1X: gtmatcagaacwccaacaacwgcyasyacyrycat}*
204319 gtcatcagaactccaacaactgctagcaccatcat  204353
204454 gtaatcagaacaccaacaacagccactactgccat  204488
204589 gtcatcagaacaccaacaacagccactactgccat  204623
204724 gtcatcagaacaccaacaacagccactactgccat  204758
204859 gtcatcagaacaccaacaacagccactactgccat  204893
204994 gtcatcagaacaccaacaacagccactactgccat  205028
205129 gtcatcagaacaccaacaacagccactactgccat  205163
205264 gtcatcagaacaccaacaacagccactactgccat  205298
205399 gtcatcagaacaccaacaacagccactactgccat  205433
205534 gtcatcagaacaccaacaacagccactactgccat  205568
205769 gtcatcagaactccaactagtgaaggtctaatcag  205703
205804 gttatcagaactccaaccagtgaaggttggttac  205838
205939 gtcatcagaactccaaccagtgaaggtctaatcag  205973
206074 gttatcagaactccaactagtgaaggtctaatcag  206108
206209 gtcatcagaactccaaccagtgaaggtctaatcag  206243
206344 gttatcagaactccaaccagtgaaggtctaatcag  206378
206479 gttatcagaactccaaccagtgaaggttggttac  206513
206614 gttgtcaaaactccaactactgccatctcatcca   206647
                |         |         |         |
               10        20        30        40        50
                |         |         |         |         |
206648 gtttgtcatcatcatcttcaggacaaatcaccagctctatcacgtcttcg 206727
206648 cgtccaattattaccccattctatcctagcaatggaacttctgtgatttc 206805
                |         |         |         |         |
```

Figure 5.1.13.A: Region characterization for the 135 bp base pattern associated with the MPTR region in the FLO1 gene that occurs in the NC_001133 sequence.

```
                   10        20        30        40        50
                    |         |         |         |         |
204043 gatggtaccactgtaagtgatgacttcgaagggtacgtctattcctttga 204122
204123 cgatgacctaagtcaatctaactgtactgtccctgacccttcaaattatg 204202
                    |         |         |         |         |
                   10        20        30        40        50
                    |         |         |         |         |
  1X: {aactacaactgagccatggaacagcactttttacctctacwtctacygaaw
>5X:{[aactacaactgagccatggaacagcactttttacctctacwtctacygaaw
204203                                                    204203
204489 aactacaactgagccatggaacagcactttttacctctacttctaccgaat 204538
205164 aactacaactgagccatggaacagcactttttacctctacatccactgaaa 205213
205839 aaccaccactgaaccatggactggtactttttacttctacatctactgaaa 205888
206514 aaccaccactgaaccatggactggtactttttacttcgacttccactgaaa 206563
                    |         |         |         |         |
                   60        70        80        90       100
                    |         |         |         |         |
  1X:  tgaccacmgtcacyggyaccaayggygtwcsaactgacgaaaccrtcatt
>5X:   tgaccacmgtcacyggyaccaayggygtwcsaactgacgaaaccrtcatt
204203                                                    204203
204539 tgaccacagtcaccggtaccaatggtttgccaactgatgagaccatcatt 204588
205214 tgaccaccgtcaccggtaccaacggtttgccaactgatgaaaccatcatt 205263
205889 tgaccaccattactggaaccaacggcgttccaactgacgaaaccgtcatt 205938
206564 tgtctactgtcactggaaccaatggcttgccaactgatgaaactgtcatt 206613
                    |         |         |         |         |
                  110       120       130       140       150
                    |         |         |         |         |
  1X:  gtmatcagaacwccaacaacwgcyasyacyrycat}*
>5X:   gtmatcagaacwccaacaacwgcyasyacyrycat|aactacaactgagc
204203                                                    204203
204589 gtcatcagaacaccaacaacagccactactgccat gactacaactcagc 204637
205264 gtcatcagaacaccaacaacagccactactgccat aactacaactcagc 205312
205939 gtcatcagaactccaaccagtgaaggtctaatcag caccaccactgaac 205987
206614 gttgtcaaaactccaactactgccatctcatcca                 206647
                    |         |         |         |         |
               160       170       180       190       200
                |         |         |         |         |
 1X:
>5X:   catggaacagcacttttacctctacwtctacygaawtgaccacmgtcacy
204203                                                    204203
204638 catggaacgacacttttacctctacttctaccgaattgaccacagtcacc 204687
205313 catggaacgacacttttacctctacatccactgaaatgaccaccgtcacc 205362
205988 catggactggtacttttacttctacatctactgaaatgaccaccattact 206037
206647                                                    206647
                |         |         |         |         |
               210       220       230       240       250
                |         |         |         |         |
 1X:
>5X:   ggyaccaayggygtwcsaactgacgaaaccrtcattgtmatcagaacwcc
204203                                                    204203
204688 ggtaccaatggtttgccaactgatgagaccatcattgtcatcagaacacc 204737
205363 ggtaccaacggtttgccaactgatgaaaccatcattgtcatcagaacacc 205412
206038 ggaaccaatggtcaaccaactgacgaaaccgttattgttatcagaactcc 206087
206647                                                    206647
                |         |         |         |         |
               260       270       280       290       300
                |         |         |         |         |
 1X:
>5X:   aacaacwgcyasyacyrycat|aactacaactgagccatggaacagcact
204203                                                    204203
204738 aacaacagccactactgccat gactacaactcagccatggaacgacact 204786
205413 aacaacagccactactgccat gactacaactcagccatggaacgacact 205461
206088 aactagtgaaggtctaatcag cactacaacggaaccatggaccggtact 206136
206647                                                    206647
                |         |         |         |         |
              310       320       330       340       350
                |         |         |         |         |
 1X:
>5X:   tttacctctacwtctacygaawtgaccacmgtcacyggyaccaayggygt
204203                                                    204203
204787 tttacctctacttctaccgaattgaccacagtcaccggtaccaatggttt 204836
205462 tttacctctacatccactgaaatcaccaccgtcaccggtaccaccggttt 205511
206137 ttcacttctacatctactgaaatgacgcacgtcaccggtaccaacggcgt 206186
206647                                                    206647
                |         |         |         |         |
              360       370       380       390       400
                |         |         |         |         |
 1X:
>5X:   wcsaactgacgaaaccrtcattgtmatcagaacwccaacaacwgcyasya
204203                                          ctgtcagta 204211
204837 gccaactgatgagaccatcattgtcatcagaacaccaacaacagccacta 204886
205512 gccaactgatgagaccatcattgtcatcagaacaccaacaacagccacta 205561
206187 tccaactgacgaaaccgtcattgtcatcagaactccaaccagtgaaggtc 206236
206647                                                    206647
                |         |         |         |         |
              410       420       430       440       450
                |         |         |         |         |
 1X:
>5X:   cyrycat|aactacaactgagccatggaacagcacttttacctctacwtc
204212 ccactac aactacaacggaaccatggaccggtactttcacttctacatc 204260
204887 ctgccat gactacaactcagccatggaacgacacttttacctctacatc 204935
205562 ctgccat gactacaactcagccatggaacgacacttttacctctacatc 205610
206237 taatcag caccaccactgaaccatggactggcactttcacttcgacttc 206285
206647                                                    206647
                |         |         |         |         |
              460       470       480       490       500
               |         |         |         |         |
 1X:
>5X:   tacygaawtgaccacmgtcacyggyaccaayggygtwcsaactgacgaaa
204261 tactgaaatgaccaccgtcaccggtaccaacggcgttccaactgacgaaa 204310
204936 cactgaaatcaccaccgtcaccggtaccaatggtttgccaactgatgaga 204985
205611 cactgaaatgaccaccgtcaccggtaccaacggcgttccaactgacgaaa 205660
206286 cactgaggttaccaccatcactggaaccaacggtcaaccaactgacgaaa 206335
206647                                                    206647
               |         |         |         |         |
              510       520       530       540       550
               |         |         |         |         |
 1X:
>5X:   ccrtcattgtmatcagaacwccaacaacwgcyasyacyrycat|aactac
204311 ccgtcattgtcatcagaactccaacaactgctagcaccatcat aactac 204359
204986 ccatcattgtcatcagaacaccaacaacagccactactgccat gactac 205034
205661 ccgtcattgtcatcagaactccaactagtgaaggtctaatcag caccac 205709
206336 ctgtgattgttatcagaactccaaccagtgaaggtctaatcag caccac 206384
206647                                                    206647
               |         |         |         |         |
              560       570       580       590       600
               |         |         |         |         |
 1X:
>5X:   aactgagccatggaacagcacttttacctctacwtctacygaawtgacca
204360 aactgagccatggaacagcacttttacctctacttctaccgaattgacca 204409
205035 acctcagccatggaacgacacttttacctctacatccactgaaatgacca 205084
205710 cactgaaccatggactggtactttcacctctacatccactgagatgacca 205759
206385 cactgaaccatggactggtactttcacttctacatctactgaaatgacca 206434
206647                                                    206647
               |         |         |         |         |
              610       620       630       640       650
                |         |         |         |         |
  1X:
 >5X:   cmgtcacyggyaccaayggygtwcsaactgacgaaaccrtcattgtmatc
204410 cagtcactggcaccaatggtgtacgaactgacgaaaccatcattgtaatc 204459
205085 ccgtcaccggtaccaacggtttgccaactgatgaaaccatcattgtcatc 205134
205760 ccgtcaccggtactaacggtcaaccaactgacgaaaccgtgattgttatc 205809
206435 ccgtcaccggtactaacggtcaaccaactgacgaaaccgtgattgttatc 206484
206647                                                    206647
                |         |         |         |         |
              660       670       680
                |         |         |
  1X:
 >5X:   agaacwccaacaacwgcyasyacyrycat]}*
204460 agaacaccaacaacagccactactgccat   204488
205135 agaacaccaacaacagccactactgccat   205163
205810 agaactccaaccagtgaaggtttggttac   205838
206485 agaactccaaccagtgaaggtttggttac   206513
206647                                 206647
                |         |         |
               10        20        30        40        50
                |         |         |         |         |
206648 gtttgtcatcatcatcttcaggacaaatcaccagctctatcacgtcttcg 206727
206648 cgtccaattattaccccattctatcctagcaatggaacttctgtgatttc 206727
                |         |         |         |         |
```

Figure 5.1.13.B: Region characterization for the integer multiple of 5 associated with the MPTR in the FLO1 gene that occurs in the NC_001133 sequence.

```
                 10        20        30        40        50
                  |         |         |         |         |
25295 acagaagttccattgctaggatagaatggggtaataattggacgcgcaga 25344
25345 cgtgataaagctggtgatttgtcctgaagaagatgacaaactggatgaga 25394
                  |         |         |         |         |
                 10        20        30        40        50
                  |         |         |         |         |
 >1X:{agaggtraaagtrycrktccatggytsagtkgtrgtsmtgrywrkasykk
25395                                                tgg  25397
25483 agaggtgaaagtaccggtccatggctcggttgtagttgtaaccaaacctt  25532
25618 agaggtgaaagtaccagtccatggttcagtggtggtgctgattagacctt  25667
25753 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  25802
25888 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  25937
26023 agaggtaaaagtgtcgttccatggctgagttgtagttatggcagtagtgg  26072
26158 agaggtaaaagtgctgttccatggctcagttgtagttatggcagtagtgg  26207
26293 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  26342
26428 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  26477
26563 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  26612
26698 agaggtaaaagtgtcgttccatggctgagttgtagtcatggcagtagtgg  26747
26833 agaggtaaaagtgctgttccatggctcagttgtagttatggcagtagtgg  26882
26968 agaggtaaaagtgctgttccatggctcagttgtagttatgatggtgctag  27017
27103 agaagtgaaagtaccggtccatggttccgttgtagttatggtagtactg   27151
                  |         |         |         |         |
              60        70        80        90       100
               |         |         |         |         |
>1X:  cwstwgttggwgttctgatgacaatgayggtytcrtcagttggmamrccg
25398 cagtagttggagttttgacaataatgacagtttcatcagttggttgaccg  25447
25533 cactggttggagttctgataacaatcacggtttcgtcagttggttgaccg  25582
25668 cactagttggagttctgatgacaatgacggtttcgtcagttggaacgccg  25717
25803 ctgttgttggtgttctgatgacaatgatggtctcatcagttggcaaaccg  25852
25938 ctgttgttggtgttctgatgacaatgatggtttcatcagttggcaaaccg  25987
26073 ctgttgttggtgttctgatgacaatgatggtttcatcagttggcaaaccg  26122
26208 ctgttgttggtgttctgatgacaatgatggtttcatcagttggcaaaccg  26257
26343 ctgttgttggtgttctgatgacaatgatggtctcatcagttggcaaacca  26392
26478 ctgttgttggtgttctgatgacaatgatggtctcatcagttggcaaacca  26527
26613 ctgttgttggtgttctgatgacaatgatggtctcatcagttggcaaacca  26662
26748 ctgttgttggtgttctgatgacaatgatggtctcatcagttggcaaacca  26797
26883 ctgttgttggtgttctgatttacaatgatggtttcgtcagttcgtacacca  26932
27018 cagttgttggagttctgatgacaatgacggtttcgtcagttggaacgccg  27067
27151                                                     27151
               |         |         |         |         |
             110       120       130       140
               |         |         |         |
>1X:  ttrgtaccggtgacggtggtsatytcagtggatgt}*
25448 ttggttccagtgatggtggtcatctcagtagatgt  25482
25583 ttagtaccggtgacggtggtcatctcagtggatgt  25617
25718 ttggtaccggtgacggtggtcatttcagtggatgt  25752
25853 ttggtaccggtgacggtggtgatttcagtggatgt  25887
25988 ttggtaccggtgacggtggtcatttcagtggatgt  26022
26123 ttggtaccggtgacggtggtcatttcagtggatgt  26157
26258 ttggtaccggtgacggtggtcatttcagtggatgt  26292
26393 ttggtaccggtgacggtggtgatttcagtggatgt  26427
26528 ttggtaccggtgactgtggtcaattcggtagaagt  26562
26663 ttggtaccggtgactgtggtcaattcggtagaagt  26697
26798 ttggtaccggtgactgtggtcaattcggtagaagt  26832
26933 ttggtgccagtgactgtggtcaattcggtagaagt  26967
27068 ttggtaccggtgacggtggtcatttcagtagatgt  27102
27151                                      27151
               |         |         |         |
              10        20        30        40        50
               |         |         |         |         |
27152 acagtataatttgaagggtctggaatggtacagtttggctggcttagatt 27201
27202 gttgtcaaaagtatatacgtacccttcaaagtcatcactaacggtagtgc 27251
               |         |         |         |         |
```

Figure 5.1.13.C: Region characterization for the 135 bp base pattern associated with the MPTR region in the FLO9 gene that occurs in the NC_001133 sequence.

5.1.14. GenBank locus: SMOSATC

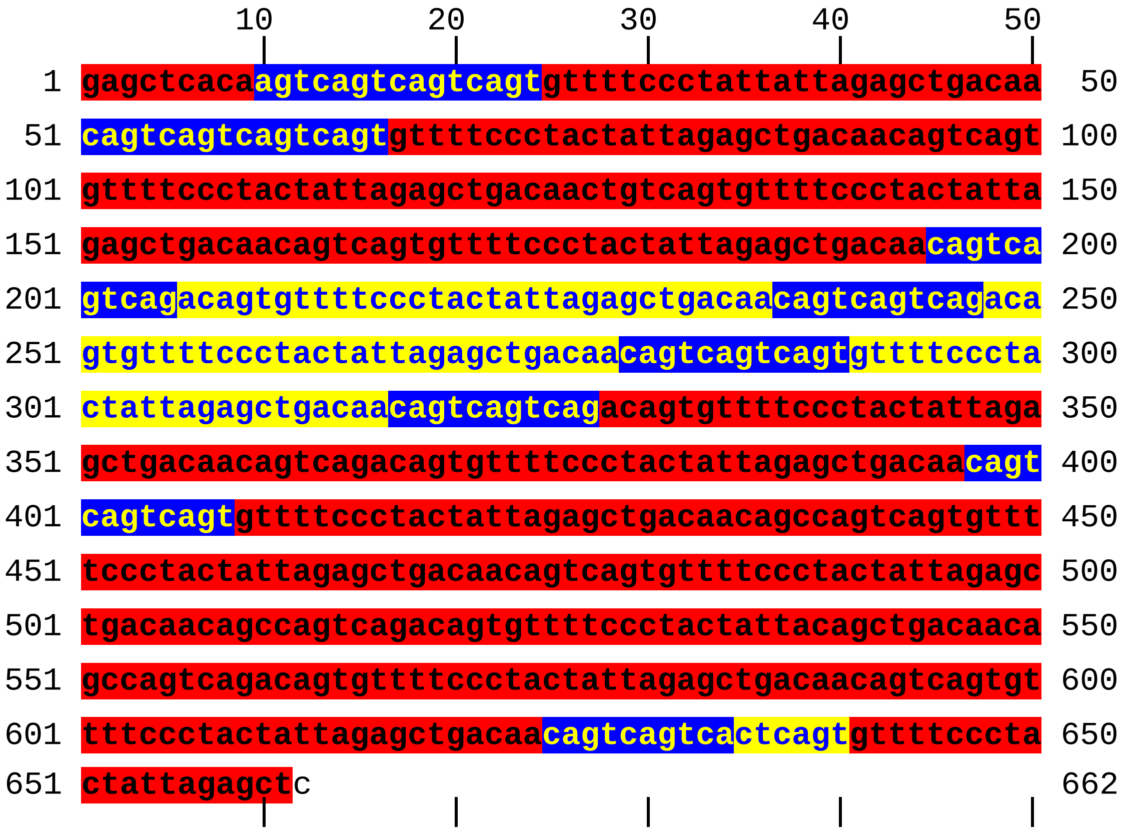
Species: *Salmo salar* (salmon)

Sequence annotation. This sequence represents 18 copies of a much larger minisatellite region. The region's pattern is approximately 40 bp and contains a variable copy number SSR having an **AGTC** pattern (Goodier & Davidson 1998).

Visual analysis. The region is a VLTR having a non-SSR portion with a 26 bp pattern and a SSR portion with an **AGTC** pattern. Some of the SSR regions are well conserved but others either have two copies, contain substitutions or both, thereby making many of the SSRs difficult to locate.

Algorithmic performance. The algorithm locates eight SSR regions within one VLTR region. Figure 5.1.14.A displays the SSR regions. All have the same pattern, **AGTC**. Figure 5.1.14.B displays the region characterization for the VLTR region having a 26 bp non-SSR portion and a SSR with an **AGTC** pattern. The SSRs in many copies of the VLTR region are not located due to the SSR having only two copies or the region having substitutions.

The occurrence of imperfect SSR regions that are identified means that VLTR analysis cannot assume that every copy or even most copies in the VLTR region have identifiable SSRs. Thus, the VLTR classification and analysis procedures are flexible, able to handle VLTR regions having a minimal number of identified SSR regions within the VLTR region. The analysis has at least three flexible procedures. First, a bare minimum of SSR regions triggers a full analysis of the region as a VLTR region. Second, VLTR analysis locates all SSRs having the identified pattern and locates all SSR shadows representing at least one copy of the pattern or any rotation of the pattern. Third, a region distance array containing all distances in the region is modified to test whether the removal of all SSRs and SSR shadows improves the periodicity distribution of the region. These procedures together with minimal thresholds enable flexible analysis of VLTR regions.



```
              10        20        30        40        50
               |         |         |         |         |
  1 gagctcacaagtcagtcagtcagtgttttccctattattagagctgacaa  50
 51 cagtcagtcagtcagtgttttccctactattagagctgacaacagtcagt 100
101 gttttccctactattagagctgacaactgtcagtgttttccctactatta 150
151 gagctgacaacagtcagtgttttccctactattagagctgacaacagtca 200
201 gtcagacagtgttttccctactattagagctgacaacagtcagtcagaca 250
251 gtgttttccctactattagagctgacaacagtcagtcagtgttttcccta 300
301 ctattagagctgacaacagtcagtcagacagtgttttccctactattaga 350
351 gctgacaacagtcagacagtgttttccctactattagagctgacaacagt 400
401 cagtcagtgttttccctactattagagctgacaacagccagtcagtgttt 450
451 tccctactattagagctgacaacagtcagtgttttccctactattagagc 500
501 tgacaacagccagtcagacagtgttttccctactattacagctgacaaca 550
551 gccagtcagacagtgttttccctactattagagctgacaacagtcagtgt 600
601 tttccctactattagagctgacaacagtcagtcactcagtgttttcccta 650
651 ctattagagctc                                       662
               |         |         |         |         |
```

Figure 5.1.14.A: The entire sequence for GenBank locus SMOSATC.

```
             10        20        30        40        50
              |         |         |         |         |
>1X:{gttttccctactattagagctgacaa{cagt}*}*
   1                 gagctcacaa   gtcagtcagtcagt   24
  25 gttttccctattattagagctgacaa cagtcagtcagtcagt   66
  67 gttttccctactattagagctgacaa cagtcagt          100
 101 gttttccctactattagagctgacaa ctgtcagt          134
 135 gttttccctactattagagctgacaa cagtcagt          168
 169 gttttccctactattagagctgacaa cagtcagtcagacagt  210
 211 gttttccctactattagagctgacaa cagtcagtcagacagt  252
 253 gttttccctactattagagctgacaa cagtcagtcagt      290
 291 gttttccctactattagagctgacaa cagtcagtcagacagt  332
 333 gttttccctactattagagctgacaa cagtcagacagt      370
 371 gttttccctactattagagctgacaa cagtcagtcagt      408
 409 gttttccctactattagagctgacaa cagccagtcagt      446
 447 gttttccctactattagagctgacaa cagtcagt          480
 481 gttttccctactattagagctgacaa cagccagtcagacagt  522
 523 gttttccctactattacagctgacaa cagccagtcagacagt  564
 565 gttttccctactattagagctgacaa cagtcagt          598
 599 gttttccctactattagagctgacaa cagtcagtcactcagt  640
 641 gttttccctactattagagct                        661
              |         |         |         |         |

 662 c  662
```

Figure 5.1.14.B: Region characterization for the VLTR region in the SMOSATC sequence.

5.1.15. GenBank locus: U00144

Species: *Bos taurus* (cow).

Sequence annotation. This sequence contains a cluster of SSR regions in an intron associated with a major histocompatibility complex gene (MHC-DRB). The SSR cluster is a hypervariable repeat region that is useful for typing alleles for this gene (Ammer *et al*. 1992). The cluster in this sequence has the structure $(\mathbf{GT})_2$ **GCTT** $(\mathbf{GT})_9$ **GAAA** $(\mathbf{GA})_{22}$ **GGGAGG** $(\mathbf{GA})_5$ **CA** $(\mathbf{GA})_2$ **AA** $(\mathbf{GA})_3$ $(\mathbf{CAGA})_3$.

Visual analysis. Visual analysis confirms the presence and structure of the SSR cluster.

Algorithmic performance. The algorithm locates all SSR regions in the cluster except the leading edge sequence $(\mathbf{GT})_2$ **GCTT**. Figure 5.1.15.A displays the SSR regions. Table 5.1.15.A details the specifics for each region. The first SSR region is not associated with the SSR cluster but is present in the sequence. Also, the SSR cluster has two perfect SSR regions having an **AG** pattern that surround a perfect SSR region having an **AGGG** pattern such that all three together form an imperfect SSR having an **AG** pattern. All other SSR regions require no explanation.

```
             10        20        30        40        50
              |         |         |         |         |
  1 agtataccaagaaagagtgtcatttcttcaacgggaccgagcgggtgcgg  50
 51 ttcctgaacagatacttccataatggagaagagttcgtgcgcttcgacag 100
101 cgactggggcgagtaccgggcggtgaccgagctagggcggccggacgccg 150
151 agtactggaacagccagaaggagatcctggagcggggccgggccgcggtg 200
201 gacacgtactgcagacacaactacggggtcggtgagagtttcactgtgca 250
251 gcggcgaggtgagggcgggggtggacggccagtgtggagcagtgtgcttg 300
301 tgtgtgtgtgtgtgtgtgaaagagagagagagagagagagagagagagag 350
351 agagagagagagagagggagggagagagagacagagaaagagagacagac 400
401 agacaga                                            407
              |         |         |         |         |
```

Figure 5.1.15.A: The entire sequence for GenBank locus U00144.

Table 5.1.15.A: SSRs present in U00144 using the identification algorithm.

Sequence Location	Region Pattern
182 ..197	CCGGG
185.. 195	CCGGG
299 ..318	GT
321 ..398	AG
321.. 366	AG
364.. 374	AGGG
372.. 381	AG
383 ..407	ACAG
393.. 406	ACAG

5.2. GenBank sequence collection: Analysis of pattern classes

Section 5.1 presents an analysis of tandem repeat regions present in the collection of GenBank DNA sequences (Section 3.3) using the identification algorithm (Chapter 4). This section discusses each pattern class identified by the algorithm; simple pattern structures (Section 5.2.1), SSRs (Section 5.2.2), VLTRs (Section 5.2.3) and MPTRs (Section 5.2.4). In addition, Section 5.2.5 discusses an extension to the identification algorithm that locates non-contiguous regions of similarity.

5.2.1. Simple pattern structures

Simple sequence patterns are the traditional representation of tandem repeat regions and correspond in this data model to the simple pattern structure, $\boldsymbol{T} = \boldsymbol{p}_{\boldsymbol{c}}$, where $\boldsymbol{p}$ is a pattern and $\boldsymbol{c}$ is the number of consecutive occurrences of the pattern in a region. Many algorithms exist for identifying tandem repeat regions having this structure including recent algorithms that identify a region without prior knowledge of either the pattern or the period of a region (Section 2.3.4). This includes an algorithm (Benson 1999) that resembles portions of this algorithm (Chapter 4).

The two algorithms have a similar overall ability to identify tandem repeat regions having simple pattern structures. There are minor differences in abilities. Benson’s algorithm performs slightly better at identifying poorly conserved regions having a 10 to 20 bp pattern due to use of smaller window sizes for analysis. This algorithm identifies long patterns that exceed Benson’s maximum periodicity, i.e. this algorithm has no upper limit on periodicity. Other slight differences occur as well based on the distribution of identical distances within a region.

5.2.2. Simple sequence repeats (SSRs)

The identification algorithm identifies simple sequence repeats (SSRs) as regions where the pattern length is at most the window size, $\boldsymbol{k}$ (see Section 4.4.2 for details). Any distance, $\boldsymbol{d} \le \boldsymbol{k}$, locates a perfect SSR. An imperfect SSR contains a perfect SSR and is found by performing a region characterization using the pattern of the perfect SSR.

The primary drawback of this approach is not identifying poorly conserved SSR regions. A region must have at least $\boldsymbol{k} + \boldsymbol{d}$ consecutive, conserved positions in order for this approach to identify a perfect SSR region. This is of particular concern when analyzing SSR regions nested within the pattern of a tandem repeat region having a complex pattern structure. Missing the SSR regions may cause improper characterization of the larger region.

5.2.3. Nested tandem repeats and variable length tandem repeats (VLTRs)

The identification algorithm identifies variable length tandem repeat (VLTR) regions as regions having a complex pattern structure such that the pattern has a variable copy number nested tandem repeat. The current implementation identifies VLTRs having nested SSRs. The algorithm analyzes each potential tandem repeat region to determine sufficient SSRs having the same pattern occur within the region to merit analysis for a nested SSR. The analysis looks for SSR regions having variable copy numbers occurring within and near the region by removing all identified SSR regions and SSR shadows and testing for an improvement of the region's periodicity signature. If the signature improves then the potential tandem repeat becomes a potential VLTR region (see Section 4.4.6 for details).

This procedure performs well for many VLTR regions. Regions having well-conserved nested SSRs and a non-SSR portion of fixed length have exceptional region characterizations.

Furthermore, the algorithm is able to characterize a region as a VLTR region when only some of a region's copies have identified SSR regions having the specified pattern. This flexibility allows for poor pattern conservation or short SSR regions.

The procedure does misclassify some regions. False VLTR designation usually involves the presence in the region of multiple SSR regions having different patterns and where at least one pattern corresponds to multiple SSR regions. Clusters of SSR regions composed of many SSR regions often are misclassified as VLTRs, e.g. a portion of the region in GenBank locus MMMSAT5 (analysis shown in Section 5.1.12). Failure to classify a region as a VLTR is usually due to not locating short, imperfectly conserved, nested SSR regions (see Section 5.2.2). In order to trigger a more intensive VLTR region analysis, the algorithm must locate a sufficient quantity of SSR regions having the same pattern in the region. Of the SSRs found, sufficient copy number variation must occur to merit VLTR designation. For example, the region in GenBank locus BTA132392 (analysis shown in Section 5.1.5) contains short SSR regions with imperfections such that half of the SSRs are not located. The half that is located contains only one SSR having a different copy number than the other SSR regions. Thus, this region is not designated as a VLTR even though visually it is observed to be a VLTR region.

The pattern construction procedure on occasion poorly characterizes a correctly designated VLTR region. The pattern for the nested SSR is usually correct but imperfections in the SSR region particularly near the SSR edges affect selection of the non-SSR portion of the pattern. The obvious choice is to assign the imperfections to the SSR portion. The other option is to add the imperfections to the non-SSR portion. The argument for the later is that if the sequence bordering the SSR resembles the SSR pattern but is conserved throughout all copies of the VLTR region then it should be explicitly separate from the SSR portion, e.g. the region in GenBank locus BOVTGN (analysis shown in Section 5.1.4). Also, sequence imperfections in the non-SSR portion near the nested SSR region may make the sequence appear more similar to the SSR pattern thereby suggesting inclusion with the SSR portion. Considerable effort was appropriated to select a good pattern and to develop a good alignment procedure for VLTR regions yet, it is clear that more effort is required to refine the procedure.

The current identification algorithm captures the basic concept associated with a VLTR region but it can be improved. Prime extensions involve the identification of regions having fixed copy number nested repeats, regions having non-SSR nested tandem repeats and regions having a complex pattern structure with more than one nested repeat. Below is an explanation of the current limitations for these extensions.

This algorithm characterizes regions with nested variable copy number SSRs (VLTRs) but does not explicitly characterize regions with fixed copy number nested SSRs. This is a compromise that enables identifying the elusive variable length copies associated with VLTR regions while simplifying the process of doing so. First, the process of removing SSRs and testing for an improvement in the periodicity signature is an important verification step for ensuring that the nested SSR pattern structure reflects the region's actual pattern structure. A fixed copy number SSR won't improve the region's periodicity signature. Second, since the current analysis allows only one SSR to be nested, it is more important to capture the SSRs having variable copy numbers than the fixed copy number SSRs. Fixed copy number SSRs introduce no periodicity signature complications and do not cause alignment difficulties. Variable copy number SSRs cause tandem repeat regions not to be identified or to be poorly characterized. Thus, the decision is to concentrate on the variable copy number SSRs. A verification scheme needs development in order to construct patterns for fixed copy number nested SSRs.

This algorithm allows nesting of SSRs but does not allow nesting of tandem repeat regions having larger patterns. Several factors contribute to this decision. First, SSR identification proceeds via a different procedure than regions having larger patterns. In fact, all SSR regions are known prior to identification of tandem repeats having the larger pattern. This simplifies the process of searching for nested SSRs since SSR identification is already complete. A more significant reason is that "related" SSR regions represent SSR regions having *identical patterns*. This is reasonable since the largest pattern is of length six. Regions having larger patterns require comparison of *similar patterns*. It is impractical to expect precisely the same pattern for larger patterns, e.g. one nucleotide difference between two 20 bp patterns should be classified as the "same" pattern. Furthermore, nesting of non-SSR tandem repeats requires additional algorithmic "passes" to identify each level of nesting. Thus, two procedures need development, one for pattern comparison and the other for additional nesting levels.

Finally, this algorithm allows a pattern structure to nest one repeat but does not allow nesting of multiple repeats. Some regions have pattern structures best captured by multiple nested SSRs, e.g. the region in GenBank locus AMU73928 clearly has an **A**-run and a **T**-run (analysis shown in Section 5.1.3). For other regions, it is not clear whether one or two nested repeats best captures the pattern structure, e.g. the region in GenBank locus BOVTGN clearly has a nested SSR having a **TG** pattern but, a **G**-run that occurs in multiple copies as a minor imperfection at the edge of the **TG**-based SSR becomes a full-fledged SSR in one copy (analysis shown in Section 5.1.4). Analysis procedures involve the SSR removal verification scheme and the RegWDP alignment. Multiple nested repeats significantly increases the complexity of both algorithms. Considerable effort was appropriated to develop a good alignment procedure for VLTR regions yet, it is clear that more effort is required to produce the desired alignment for structures having multiple nested repeats.

5.2.4. Multi-periodic tandem repeats (MPTRs)

The algorithm identifies multi-periodic tandem repeat (MPTR) regions as regions having a complex pattern structure such that integer multiples of a base pattern capture additional levels of recurrence within a region. The analysis locates the region and characterizes it with a base pattern using the procedures for regions having a simple pattern structure. Analysis of the base pattern region characterization looks at a column through the sequence copies that are aligned to a position in the pattern and determines whether cross-copy patterns occur. Cross-copy patterns that recur every third copy is considered a multiple of three. Selection of integer multiples of the base pattern requires both a periodicity in the region's periodicity signature and cross-copy patterns associated with the column analysis. New patterns are constructed for each integer multiple, ***m***, by combining ***m*** base patterns and then modifying the base patterns using cross-copy patterns associated with the column analysis (see Section 4.4.9 for details). Each new pattern creates a new region characterization.

This procedure is effective in finding well conserved and moderately conserved cross-copy patterns, e.g. the region in GenBank locus MM102B5 (analysis shown in Section 5.1.11). Some challenging regions are also well characterized. For example, the region in GenBank locus

HSVDJSAT (analysis shown in Section 5.1.10) contains three distinct patterns that interleave in a complicated pattern structure that creates cross-copy patterns every 2, 4, 5, 6 and 11 copies.

The procedure does not properly characterize all regions. The column analysis is quite flexible enabling just a hint of a cross-copy pattern to trigger a potential integer multiple. If enough random substitutions occur to create a periodicity associated with the multiple then a false multiple of the base pattern is created to characterize the region. This is less likely to occur for regions having few copies and more likely to occur for regions having many copies, e.g. the region having a 135 bp pattern in GenBank locus NC_001133 (analysis shown in Section 5.1.13). Thus, the analysis needs additional procedures to refine the multiple selection procedure.

The current identification algorithm captures the basic concept associated with a MPTR region but it can be improved. At present, the algorithm concentrates on identifying cross-copy patterns in fixed length copies. The prime extension involves the identification of regions having multiple, unrelated periodicities such as periodicities associated with variable length copies. As region pattern structures become more complex, the various types of regions in the data model converge. For example, the most credible scenario for a region having multiple unrelated periodicities in a nested repeat region having a variable copy number, non-SSR, nested tandem repeat. Thus, more complex MPTR regions are essentially more complex VLTR regions.

5.2.5. Regions of similarity (ROS)

During development of the identification algorithm, several alternative avenues for identifying recurrences became apparent. One avenue involves identifying similar genes, gene shadows, gene clusters as well as interspersed repeats by locating non-contiguous, potentially distant, regions of similarity. Section 4.4.3 presents a proof of concept algorithm for identifying regions of similarity. This preliminary algorithm is able to locate Alu sequences, long terminal repeats (LTRs), related tandem repeat regions, gene clusters and genes paired with open reading frames (ORFs) or gene shadows but does not locate all occurrences within a sequence. Analysis of two sequences shows the success of this approach.

5.2.5.1. GenBank locus: HS333B10

Species: *Homo sapiens* (human)

Sequence annotation. The sequence contains sequence homologies to 24 complete and 14 partial Alu elements using consensus sequence based analysis.

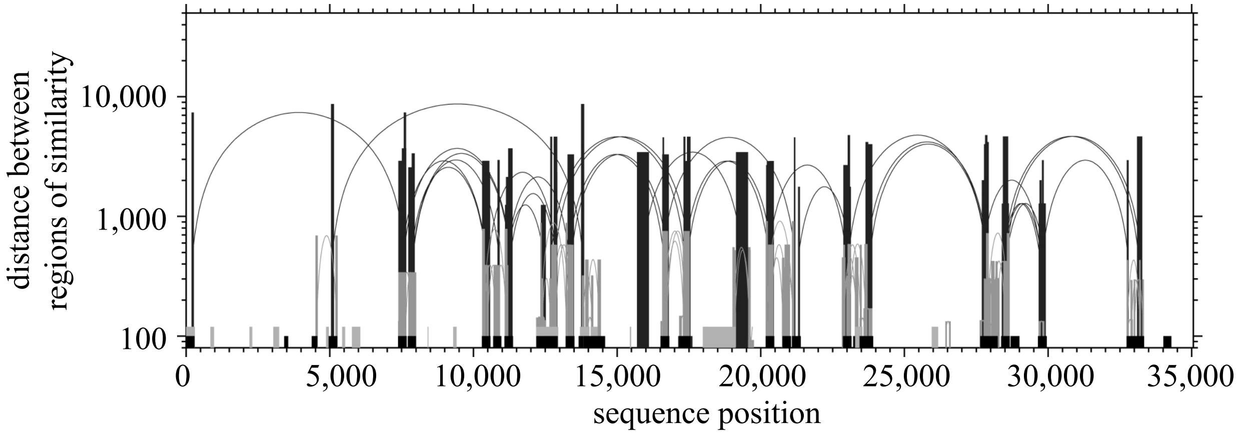


Figure 5.2.5.1.A: Analysis of GenBank locus HS333B10 using a window size of $\boldsymbol{k} = 6$. Each region of similarity is comprised of two vertical bars connected by an arc and is shown in dark gray and black. Light gray bars denote complete and partial Alu repeats present in the sequence annotation (vertically up to 100). Medium gray bars denote non-SSR tandem repeats identified by the algorithm (vertically up to 120).

Regions of similarity analysis. Figure 5.2.5.1.A displays links connecting all Alu regions except for four partial and one complete Alu region. An Alu repeat is a retroposon having a sequence structure that is a dimer. Some regions of similarity locate similarities within individual Alu elements. Other regions of similarity connect different Alu elements.

5.2.5.2. GenBank locus: NC_001133

Species: *Saccharomyces cerevisiae* (yeast)

Sequence annotation. Section 5.1.13 describes the flocculation genes, FLO1 and FLO9, present in yeast chromosome I. Both genes are known to have gene shadows located10 to 14 kbp distal to each gene. A Bdh1 gene for the enzyme (2R, 3R)-2, 3-butanediol dehydrogenase and a putative polyol dehydrogenase gene occur at positions 33 to 36 kbp. A transposon occurs at positions 160 to 166 kbp and includes flanking long terminal repeats (LTRs). A gene cluster occurs at positions183 and 190 kbp and links the Prm9 gene to five ORFs. Other genes and features are annotated but are not relevant for this discussion.

Regions of similarity analysis. Figure 5.2.5.2.A displays five regions of similarity clusters associated with this chromosome. The first two are the flocculation genes (FLO9 and FLO1) that occur one at each end of the chromosome (24 to 28 kbp and 203 to 208 kbp, respectively), with each having a linkage to a gene shadow (11 to 14 kbp and 202 to 222 kbp, respectively). The third is a gene cluster with numerous regions of similarities linking the Prm9 gene and five ORFs (183 to 190 kbp). Fourth, a Bdh1 gene for the enzyme (2R, 3R)-2, 3-butanediol dehydrogenase has a similarity in its first 200 bp with a putative polyol dehydrogenase (33 to 36 kbp). Finally, two LTRs, one at each edge of a transposon, link to one another (160 to 166 kbp).

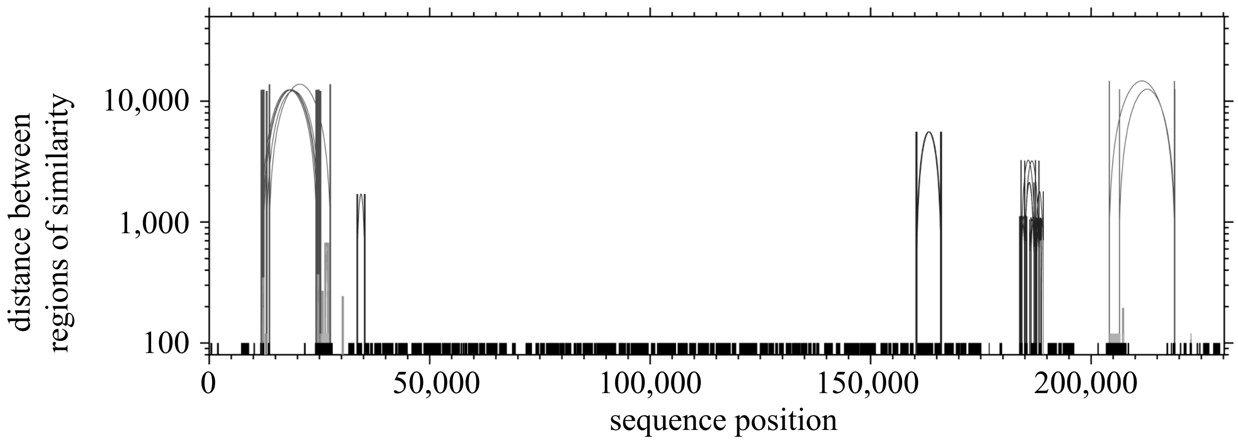


Figure 5.2.5.2.A: Analysis of yeast chromosome I using a window size of $\boldsymbol{k} = 6$. This display represents the entire chromosome and includes, paired regions of similarity, non-SSR tandem repeat regions and sequence annotations for genes and ORFs. This sequence contains FLO9 and FLO1 genes, each paired with a gene shadow more than 10 kbp away (10 to 30 kbp and 202 to 22 kbp), a gene cluster (183 to 190 kbp), two proximal genes (33 to 36 kbp) and two LTRs (160 to 166 kbp). Each region of similarity is comprised of two vertical bars connected by an arc and is shown in dark gray and black. The height of the vertical bar represents the distance between identical words in the regions. Light gray bars denote genes and ORFs present in the sequence annotation (vertically up to 100). Medium gray bars denote non-SSR tandem repeats identified by the algorithm (vertically up to 120). Light and medium gray bars are easier to distinguish in other displays.

The regions of similarity clusters associated with the flocculation genes and with the gene cluster merit additional discussion.

The flocculation genes and gene shadows represent structures that include short and long tandem repeat regions, and other internal recurrences as well as non-contiguous similarity between the gene and its shadow. Figure 5.2.5.2.B displays the five regions of similarity linking

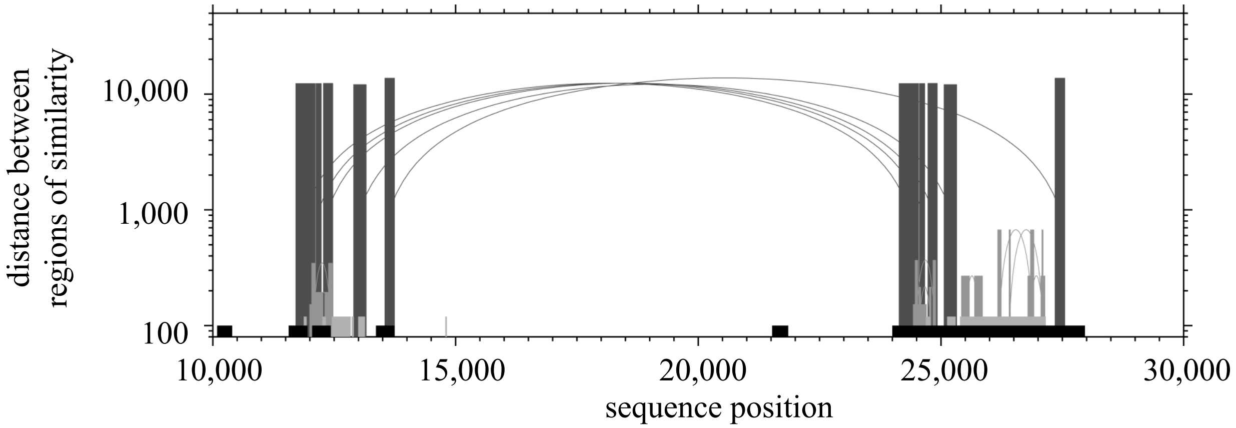


Figure 5.2.5.2.B: Analysis of the FLO9 gene (24 to 28 kbp) and its gene shadow (11.5 to 14 kbp) that occur in the NC_001133 sequence. The vertical bars connected by arcs indicate regions of similarity. Figure 5.2.5.2.A describes the sequence annotations and non-SSR tandem repeats shown in the display.

the FLO9 gene with its shadow. Four of the regions of similarity emanate from the left and one from the right of a large tandem repeat region having a 135 bp pattern (see Section 5.1.13 for analysis). The shadow contains three ORFs and several tandem repeat regions present in the gene but it does not contain the large tandem repeat having a135 bp pattern. The regions of similarity link tandem repeat regions in the shadow and the gene having the same pattern, ORFs in the shadow with portions of the FLO9 gene and unclassified portions of the shadow with portions of the FLO9 gene. Figure 5.2.5.2.C displays the regions of similarity linking the FLO1 gene with its shadow. Here, the linkage is substantially weaker. The regions of similarity in the gene begin either in or beside the large tandem repeat region having the 135 bp pattern and link to a single pseudogene in the shadow.

A gene cluster contains numerous regions of similarities that link the Prm9 gene to five ORFs. Figure 5.2.5.2.D displays these regions of similarities. Over half are pairs separated by

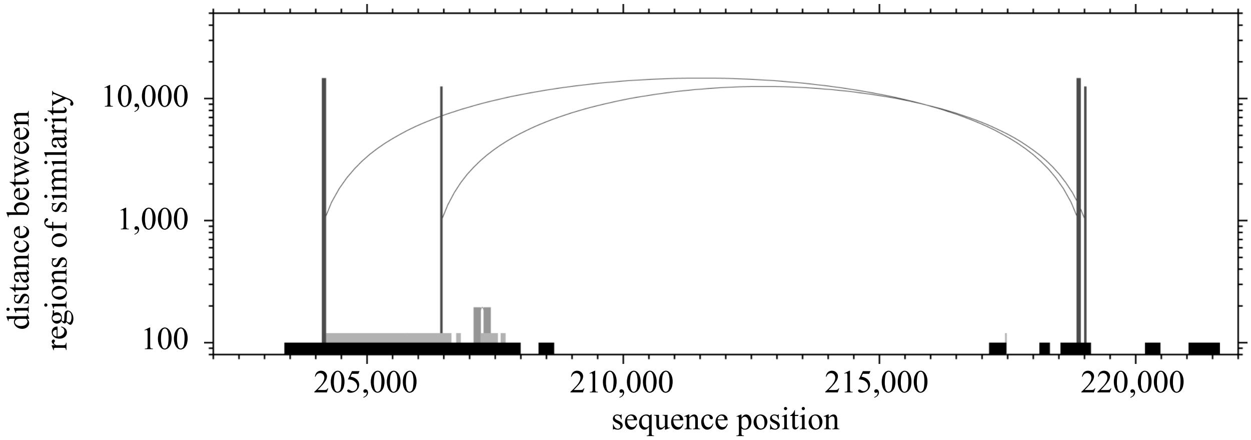


Figure 5.2.5.2.C: Analysis of the FLO1 gene (203.5 to 208 kbp) and its gene shadow (shown at 219 kbp) that occurs in the NC_001133 sequence. The vertical bars connected by arcs indicate regions of similarity. Figure 5.2.5.2.A describes the sequence annotations and non-SSR tandem repeats shown in the display.

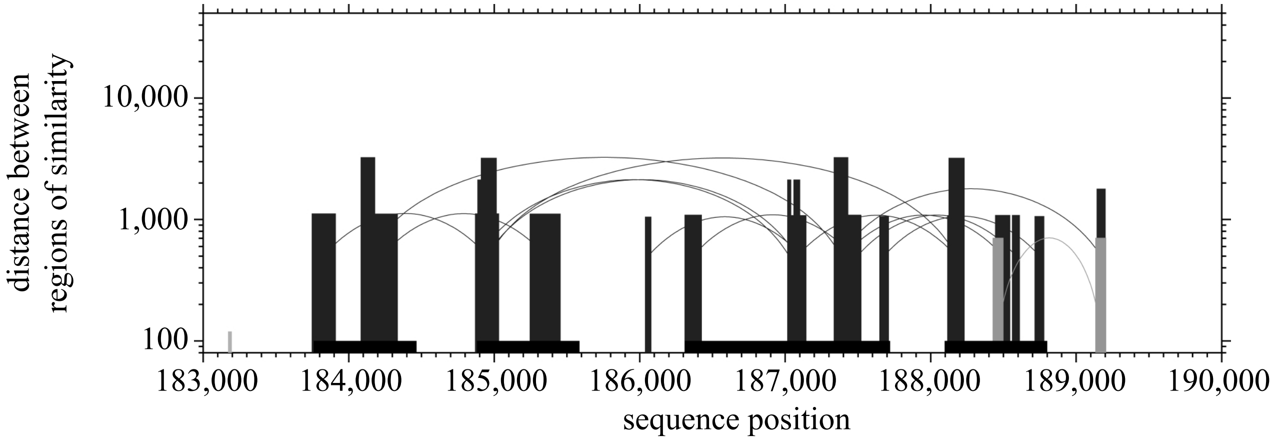


Figure 5.2.5.2.D: Analysis of a gene cluster linked by numerous regions of similarity that occurs in the NC_001133 sequence. The vertical bars connected by arcs indicate regions of similarity. Figure 5.2.5.2.A describes the sequence annotations and non-SSR tandem repeats shown in the display.

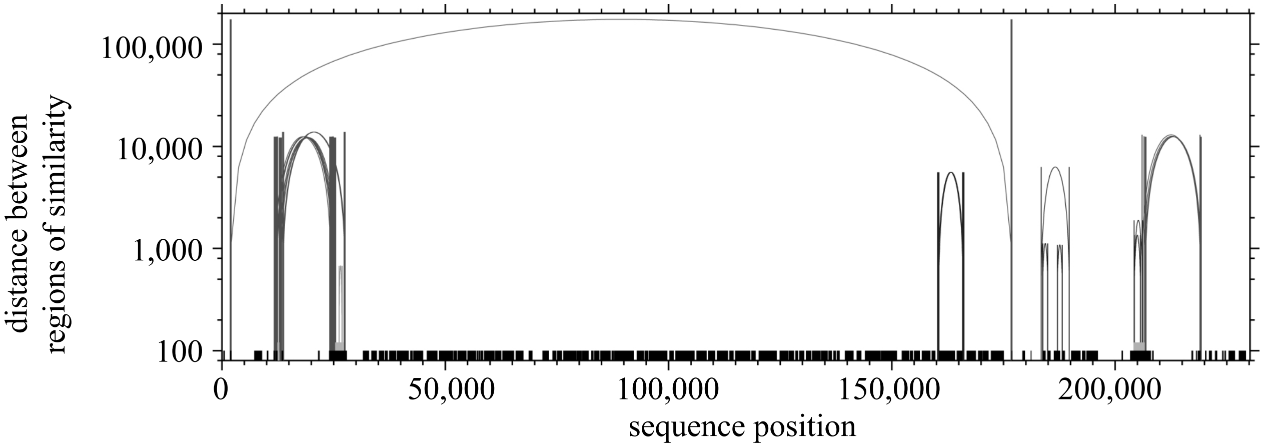


Figure 5.2.5.2.E: Analysis of yeast chromosome I using a window size of $k = 12$. The vertical bars connected by arcs indicate regions of similarity. Figure 5.2.5.2.A describes the sequence annotations and non-SSR tandem repeats shown in the display.

about 1,100 bp but some pairs are 3,250 bp apart. Some regions have similar regions at several distances, e.g. the region from sequence positions 18,4950 to 18,5000 is linked to regions 1,122bp, 2,130 bp and 3,215 bp away.

The analysis of yeast chromosome I provides a taste of how regions of similarity can identify non-contiguous features in DNA sequences. It is clear that varying the window size of the analysis significantly modifies the results of the analysis. Figure 5.2.5.2.E displays an analysis of the sequence using a window size of $k = 12$. Here, the gene cluster has significantly fewer regions of similarity. Interestingly, a region of similarity appears that spans 170 kbp and links a Pau7 gene (near 177 kbp) to an ORF (near 2 kbp).

5.2.5.3. Word size tradeoff

The current algorithm uses a single window size ($\boldsymbol{k} = 6$) to locate non-contiguous regions of similarity. The analysis at a window size of $\boldsymbol{k} = 12$ suggests that larger window sizes locate sequence similarity spanning larger distances, e.g. the Pau7 gene sequence similarity. Unfortunately, larger window sizes are less sensitive to sequence similarity, e.g. the gene cluster sequence similarity. This tradeoff between spanning distance and sensitivity to sequence similarity is directly tied to nucleotide ratios and word frequencies for the sequence.

Sequence similarity is by no means new. In fact, good algorithms already exist for identifying sequence similarity in complete genomes, e.g. Kurtz uses suffix tree-based algorithms to locate sequence similarity (see Section 2.3.1). His early algorithm uses a window size of $\boldsymbol{k} = 20$ (Kurtz 1999, Kurtz & Schleiermacher 1999) and recent algorithms use a window size of $10 \leq \boldsymbol{k} \leq 13$ (Kurtz *et al*. 2001). The problem is that these large window sizes rely on large stretches of sequence identity to locate sequence similarities and are not sensitive to sequence similarities based on short stretches of sequence identity. Furthermore, using a small window size with these algorithms would bump up against the statistical frequencies for word occurrences thereby generating a massive quantity of sequence similarities many of which would be spurious. Thus, these algorithms do not have the intelligence to utilize the small window size sensitivity and generate useful results.

5.3. Bibliography

Allen, M. J., Jeffreys, A. J., Surani, M. A., Barton, S., Norris, M. L. and Collick, A. (1994) Tandemly repeated transgenes of the human minisatellite MS32 (D1S8), with novel mouse gamma satellite integration. Nucleic Acids Research **22**(15):2976-2981.

Ammer, H., Schwaiger, F. W., Kammerbauer, C., Gomolka, M., Arriens, A., Lazary, S. and Epplen, J. T. (1992) Exonic polymorphism vs intronic simple repeat hypervariability in MHC-DRB genes. Immunogenetics **35**(5):332-340.

Ananiev, E. V., Phillips, R. L. and Rines, H. W. (1998) A knob-associated tandem repeat in maize capable of forming fold-back DNA segments: are chromosome knobs megatransposons? Proceedings of the National Academy of Science of the United States of America **95**(18):10785-10790.

Baron, B., Poirier, C., Simon-Chazottes, D., Barnier, C. and Guenet, J. L. (1992) A new strategy useful for rapid identification of microsatellites from DNA libraries with large size inserts. Nucleic Acids Research **20**(14):3665-3669.

Benson, G. (1999) Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Research **27**(2):573-580.

Boan, F., Gonzalez, A. I., Rodriguez, J. M. and Gomez-Marquez, J. (1997) Molecular characterization of a new human minisatellite that is able to form single-stranded loops in vitro and is recognized by nuclear proteins. FEBS Letters **418**(3):251-257.

Boan, F. Rodriguez, J. M. and Gomez-Marquez, J. (1998) A non-hypervariable human minisatellite strongly stimulates in vitro intramolecular homologous recombination. Journal of Molecular Biology **278**(3):499-505.

Bösl, M. and Kersten, H. (1991) A novel RNA product of the *tyrT* operon of *Escherichia coli*. Nucleic Acids Research 19(21):5863-5870.

Bussey, H., Kaback, D. B., Zhong, W., Vo, D. T., Clark, M. W., Fortin, N., Hall, J., Ouellette, B. F., Keng, T., Barton, A. B., Su, Y., Davies, C. J. and Storms, R. K. (1995) The nucleotide sequence of chromosome I from *Saccharomyces cerevisiae*. Proceedings of the National Academy of Science of the United States of America **92**(9):3809-3813.

Egan, J. and Landy, A. (1978) Structural analysis of the $\mathrm{tRNA}_1^{\mathrm{Tyr}}$ gene of *Escherichia coli*. A 178 base pair sequence that is repeated 3.14 times. The Journal of Biological Chemistry 253(10):3607-3622.

Finley, R. L. Jr. and Brent, R. (1994) Interaction mating reveals binary and ternary connections between Drosophila cell cycle regulators. Proceedings of the National Academy of Sciences of the United States of America **91**(26):12980-12984.

Finley, R. L. Jr., Thomas, B. J., Zipursky, S. L. and Brent, R. (1996) Isolation of Drosophila cyclin D, a protein expressed in the morphogenetic furrow before entry into S phase. Proceedings of the National Academy of Sciences of the United States of America **93**(7):3011-3015.

Goodier, J. L. and Davidson, W. S. (1998) Characterization of novel minisatellite repeat loci in Atlantic salmon (*Salmo salar*) and their phylogenetic distribution. Journal of Molecular Evolution **46**(2):245-255.

Goldmann, W., Hunter, N., Martion, T., Dawson, M. and Hope, J. (1991) Different forms of the bovine PrP gene have five or six copies of a short, G-C-rich element within the protein-coding exon. Journal of General Virology **72**:201-204.

Hastie, N.D. (1989) In Lyon, M. F. and Searle, A. G. (eds) Genetic variants and strains of the laboratory mouse. Oxford University Press. Pp 559-573.

Kashi, Y., Iraqi, F., Tikochinski, Y., Ruzinzki, B., Nave, A., Beckmann, J. S., Friedmann, A., Soller, M. and Gruenbaum, Y. (1990) (TG)n Uncovers a sex-specific hybridization pattern in cattle. Genomics **7**:31-36.

Küpper, H., Sekiya, T., Rosenberg, M., Egan, J. and Landy, A. (1978) A ρ-dependent termination site in the gene coding for tyrosine tRNA su_3 of *Escherichia coli*. Nature **272**(5652):423-428.

Kurtz, S. (1999) Reducing the space requirement of suffix trees. Software-Practice and Experience **29**(13):1149-1171.

Kurtz, S. and Schleiermacher, C. (1999) REPuter: fast computation of maximal repeats in complete genomes. Bioinformatics **15**(5):426-427.

Kurtz, S., Choudhuri, J.V., Ohlebusch, E., Schleiermacher, C., Stoye, J. and Giegerich, R. (2001) REPuter: the manifold applications of repeat analysis on a genomic scale. Nucleic Acids Research **29**(22):4633-4642.

McCorkle, G. M. and Altman, S. W. (1982) Large deletion mutants of *Escherichia coli* $\mathrm{tRNA}_1^{\mathrm{Tyr}}$. Journal of Molecular Biology **155**(2):83-103.

Nave, A., Kashi, Y. and Soller, M. (1997) Minisatellite and microsatellite length variation at a complex bovine VNTR locus. Animal Genetics **28**(1):52-54.

Rong, Y. S. and Golic, K. G. (1998) Dominant defects in Drosophila eye pigmentation resulting from a euchromatin-heterochromatin fusion gene. Genetics **150**(4):1551-1566.

Rossi, J., Egan, J., Hudason, L. and Landy, A. (1981) The *tyrT* locus: terminations and processing of a complex transcript.

Schlapfer, I., Saitbekova, N., Gaillard, C. and Dolf, G. (1999) A new allelic variant in the bovine prion protein gene (PRNP) coding region. Animal Genetics **30**(5):386-387.

Titen, S. W., Rong, Y. S. and Golic, K. G. (2002) A model system for expansion and contraction of repetitive DNA. 43rd Annual Drosophila Research Conference. **43**:275B.

Watari, J., Takata, Y., Ogawa, M., Sahara, H., Koshino, S., Onnela, M. L., Airaksinen, U., Jaatinen, R., Penttila, M. and Keranen, S. (1994) Molecular cloning and analysis of the yeast flocculation gene FLO1. Yeast **10**(2):211-225.

6. Beyond tandem repeats

This dissertation presents an algorithm for identifying tandem repeat regions in DNA sequences. The data model for tandem repeats represents a leap forward from regions defined by simple sequence patterns to regions defined by complex pattern structures. The analysis procedures realize this model by locating and characterizing tandem repeat regions having several complex pattern structures as well as the traditional simple sequence patterns. Furthermore, the identification algorithm compensates for common issues such as small copy numbers, poor conservation levels and factors contributing to noisy data.

Tandem repeats represent a repetitive feature that is straightforward to identify using computational algorithms that do not require prior sequence knowledge but do require prior knowledge of valid tandem repeat pattern structures. The identification techniques use a simple sequence similarity search to *locate* potential repeats and *validate* the repeat by applying knowledge of tandem repeat pattern structures. This approach of locating recurrences using sequence similarities and validating the recurrences using known structures should identify other repetitive DNA features. For instance,

- **Interspersed repeats.** Region of similarity (ROS) analysis locates Alu repeats (GenBank locus HS333B10) and a transposon (yeast chromosome I) by similarity between separate individual elements. Validation of potential repeats can use known structural knowledge for retroposons (Section 2.1.1.2.1) and proretroviral transposons (Section 2.1.1.2.2).
- **Palindromes, hairpins and stem loops.** Non-direct sequence similarities such as similarities between a sequence and its reverse complement can locate alternative

duplex structures. These structures require a high degree of sequence similarity sensitivity as stems can be rather short and imperfections can occur within the stems. The techniques in this dissertation likely can locate strong stem structures but not weak, short stems common to structures such as tRNAs.

- **Gene duplications and gene clusters.** ROS analysis of yeast chromosome I locates gene clusters and genes paired with gene shadows. An ORF based validation procedure can validate whether sequence similarities occur within genes.
- **Comparison of two sequences.** The techniques in this dissertation analyze a single sequence but the approach is amenable to comparison of multiple sequences. A natural extension is to compare all chromosomes in a genome to identify full genome recurrences. Other techniques exist to make this comparison but they utilize larger window sizes that are less sensitive to sequence similarities.

Other features are more difficult to identify using these techniques. Promoters and other signals typically have a 5 to 15 bp consensus sequence with numerous ambiguous nucleotides that are not amenable to similarity searches using identical words. Some similar protein structures are known to have very different amino acid sequences. Similar amino acid sequences can have a large degree of nucleotide sequence dissimilarity due to codon redundancy.

A powerful identification approach is one that both *locates* sequence similarities and *determines* the source of the similarity. The tandem repeat identification algorithm represents one such approach. Adapting this approach to other DNA features will assist genome projects identify common features present in DNA, identify sequence similar genes and generally provide another tool for more comprehensive sequence annotation.