IMPROVING SENSITIVITY TO APPROXIMATE REPETITIVE DNA

by

Charlotte E. Schaeffer

A thesis proposal submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

MIAMI UNIVERSITY

Oxford, OH

2014

Abstract

Improving Sensitivity to Approximate Repetitive DNA

by

Charlotte E. Schaeffer, Master of Science

Miami University, 2014

Identifying repetitive sequences within a genome is one of the fundamental problems in bioinformatics. Repetitive DNA is a significant portion of most genomes, especially eukaryotic genomes [1]. This proposal seeks to improve an already existing repeat finding tool, known as Rapid Ab Initio Detection of Elementary Repeats (RAIDER) [2]. We hope to improve upon this tool's current ability to detect approximate repeats through the an in-depth study of spaced seeds, which are commonly used to detect approximate matches in strings. RAIDER currently only allows for a single spaced seed to be used in repeat detection, and this seed is chosen arbitarily. Therefore, the sensitivity of RAIDER to the detection of approximate repeats could be improved through the use of multiple seeds, more careful seed design and analysis, and making changes to the algorithm in order to make it more amenable to the use of spaced seeds.

(15 pages)

CONTENTS

Introduction

Repetitive DNA makes up a significant portion of most genomes, especially those of eukaryotic organisms. Repetitive DNA accounts for over one-third of the genetic material of higher organisms [3]. In fact, the Human Genome Project revealed that over one half of the DNA in the human genome is composed of these repetitive sequences [4].

Due to the inexact nature of repeats, their detection has become a significant problem in bioinformatics [5]. Over long periods of time, these repetitive sequences have accumulated mutations. Therefore, identification of repeats cannot be limited to the identification of identical subsequences in the genome. Rather, there needs to be a concept of sequence similarity, allowing us to measure the similarity of two sequences and determine whether or not they are a "match" (i.e. descend from the same initial repeat sequence).

This proposal seeks to improve an already existing repeat finding tool, known as Rapid Ab Initio Detection of Elementary repeats (RAIDER) [5]. Through an in-depth study of this tool and of approaches to finding approximate repeats, we hope to significantly improve upon RAIDER's current ability to detect approximate repeats.

We will go over the necessary background associated with the approximate repeat-finding problem. This will include a cursory discussion of the genome and DNA as strings, and then we will go in depth into concepts such as string matching, as well as the characterization of both exact (identical) repeats and approximate (similar) repeats.

Following a review of the common techniques employed in repeat-finding tools, we will focus on background and definitions associated with k-mer and spaced seed approaches, which are both used in RAIDER. Following this, we will discuss our proposed study of improving RAIDER's sensitivity to finding approximate repeats and give an expected timeline of our work.

Background

2.1 Biological Background

Every living organism inherits hereditary information from its parents that affect the organism's distinguishing traits [6]. This information is embedded inside an organism's genetic material, a molecule known as deoxyribonucleic acid (DNA). An organism's *genome* is the set of all DNA sequences associated with that organism.

Each sequence of DNA is composed of a chain of nucleotides. There are four nucleotides found in DNA: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) [7]. DNA can therefore be represented as a finite string $s = s_0 s_1 ... s_{n-1}$ over the alphabet $\Sigma = \{A, C, G, T\}$ of nucleotides.

Repetitive DNA is defined as a set of discrete DNA sequences in the same genome that are similar or identical to one another [8]. As previously mentioned, it makes up a significant portion of most genomes, especially in eukaryotic organisms. We will go on to define these repetitive sequences (repeats) more concretely.

2.2 String Matching

String matching is fundamental to the process of finding repetitive DNA, and therefore must be discussed prior to going into the more detailed definitions associated with repetitive data. String matchings can be categorized into one of two types: exact matchings and inexact matchings [9]. An *exact matching* is when two strings are the same size and are composed of the same sequence of characters. While this type of matching is of use to this discussion, the more relevant type of matching is known as *inexact* or *approximate matching*.

In the case of approximate matching, two strings s and t can differ to some degree and still be a "valid" match. There are a variety of ways in which two strings can differ [10]. Such differences include, but are not limited to, (1) *substitutions* (replacements) and (2) *indels* (also known as deletions and insertions).

The differences between strings are also used to characterize the operations that can be used to sequentially transform a source sequence, s, into a target sequence, t. These operations (substitution, insertion, and deletion) are known as the *elementary edit operations*. For example, consider strings s = BAT and t = BET. We could transform s into t *substituting* the character 'A' in the middle of string s with the character 'E'. Similarly, consider strings s =BAT and t =BATS. We could transform s into t by *inserting* the character 'S' at the end of s.

There are two widely used values we can calculate in order to quantify the overall distance or difference between two strings [10].

1. The *Hamming distance* between two strings is the minimum number of substitutions needed to transform the first string into the second.

2. The *edit distance* or *Levenshtein distance* [11] between two strings is the minimum number of elementary edit operations (substitutions, insertions, and deletions) needed in order to transform the first string into the second.

Consider the scenario depicted in Figure 2.1, where we are attempting to transform source string s = INDUSTRY into target string t = INTEREST. We can transform s into t using a minimum of 6 substitution operations, so the Hamming distance between s and t is 6 (left). It also takes a minimum of 6 elementary edit operations transform s into t, so the Levenshtein distance between s and t is 6 (right). In this case, the ability to use insertions and deletions did not reduce the distance between the strings.

Hamming Distance		Levenshtein Distance	
INDUSTRY → INTUSTRY	Substitute 'D' by 'T'	INDUSTRY → INDUSTR	Delete 'Y'
→ INTESTRY	Substitute 'U' by 'E'	→ INDUST	Delete 'R'
→ INTERTRY	Substitute 'S' by 'R'	→ INRUST	Substitute 'D' by 'R'
→ INTERERY	Substitute 'T' by 'E'	→ INREST	Substitute 'U' by 'E'
→ INTERESY	Substitute 'R' by 'S'	→ INTREST	Insert 'T'
→ INTEREST	Substitute 'Y' by 'T'	→ INTEREST	Insert 'E'

Figure 2.1: It takes a minimum of 6 substitution operations to transform s into t (left). In fact, it takes a minimum of 6 elementary edit operations (substitutions, deletions, and insertions) to transform s into t (right).

2.3 Defining Repeats

As previously mentioned, we can treat DNA sequences as strings over the alphabet $\Sigma = \{A, C, G, T\}$. For this reason, similar to string matchings, we can classify repeats as either exact or approximate. Zheng and Lonardi [12] proposed a bottom-up approach to defining repeats, through the definition of *elementary repeats*. Elementary repeats are sequences that (1) meet the prescribed minimum length and frequency thresholds to be considered a repeat, and (2) do not contain any subsequences that would also satisfy (1). The following are the precise definitions for both exact and approximate elementary repeats, as described by Zheng and Lonardi.

2.3.1 Exact Repeats

Definition 2.3.1 (Exact Elementary Repeat)**.** Let S be an input genomic sequence. Let l, f be some fixed thresholds for the minimum length and frequency, respectively. A subsequence A of S is an **exact elementary repeat** if:

1. $|A| \geq l$
2. $freq(A) \geq f$
3. $\forall\ i, j \in [0, |A| - 1] s.t. j - i \geq l,\ freq(A[i, j]) = freq(A)$
4. $\nexists\ A'$ s.t. A is a subsequence of A' and $freq(A') = freq(A)$

From Definition 2.3.1, we see that A must have sufficient length and frequency. Additionally, no subsequence of A that satisfies these length and frequency requirements can occur outside of A. Lastly, A must be maximal, meaning that A is not a subsequence of some other sequence A' with equal frequency in the genomic sequence.

2.3.2 Approximate Repeats

In order to go on to define an approximate elementary repeat, we must revisit the idea of inexact matching. We have two ways to quantify the distance or difference between two sequences [12, 13].

Notation 2.3.1. Suppose we have two sequences, A and A'.

- **Hamming Distance.** Let $R^* = \{r_1, \ldots, r_p\}$ be a minimum length sequence from the set of all possible replacement operation sequences that will transform A into A'. Then,

1. The Hamming distance between A and A' is denoted $d_H(A, A') = |R^*|$. We say that A forms a *k-mismatch* with A' if $d_H(A, A') \leq k$ for some constant k.
2. Suppose A is a subsequence of S and k is some fixed constant. Let $S_{H,k}(A)$ be the set of all non-overlapping subsequences of S that form a k-mismatch with A, and denote the size of this set as $f_{H,k}(A)$.
3. For any substring B of A, the string B' that results from performing the replacement operations from R^* on B is called the *H-image* of B induced by A.

- **Edit Distance.** Let $E^* = \{e_1, \ldots, e_q\}$ be a minimum length sequence from the set of all possible edit operation sequences that will transform A into A'. Then,

1. The edit distance between A and A' is denoted $d_E(A, A') = |E^*|$. We say that A forms a *k-difference* with A' if $d_E(A, A') \leq k$ for some constant k.
2. Suppose A is a subsequence of S and k is some fixed constant. Let $S_{E,k}(A)$ be the set of all non-overlapping subsequences of S that form a k-difference with A, and denote the size of this set as $f_{E,k}(A)$.
3. For any substring B of A, the string B' that results from performing the edit operations from E^* on B is the *E-image* of B induced by A.

Definition 2.3.2 (k-Mismatches Approximate Elementary Repeat)**.** Let S be an input genomic sequence. Let l, f, k be some fixed thresholds for the minimum length, frequency, and matching, respectively. A subsequence A of S is a **k-mismatches approximate elementary repeat** if:

1. $|A| \geq l$
2. $f_{H,k}(A) \geq f$
3. $\forall\, i, j \in [0, |A|-1]$ *s.t.* $j - i \geq l$, every H-image of $A[i, j]$ induced by A must form a k-mismatch with $A[i, j]$ and $f_{H,k}(A[i, j]) = f_{H,k}(A)$
4. $\nexists\, A'$ s.t. A is a subsequence of A', every H-image of A induced by A' forms a k-mismatch with A, and $f_{H,k}(A) = f_{H,k}(A')$

From Definition 2.3.2, we see that A must have sufficient length and that there must be a certain number of sequences that form a k-mismatch with A. Additionally, no subsequence of A that satisfies the length and k-mismatch frequency requirements can occur outside of A. Lastly, A must be maximal, meaning that A is not a subsequence of some other sequence A' with equal k-mismatch frequency in the genomic sequence.

Definition 2.3.3 (k-Differences Approximate Elementary Repeat)**.** Let S be an input genomic sequence. Let l, f, k be some fixed thresholds for the minimum length, frequency, and matching, respectively. A subsequence A of S is a **k-differences approximate elementary repeat** if:

1. $|A| \geq l$
2. $f_{E,k}(A) \geq f$
3. $\forall\, i, j \in [0, |A|-1]$ *s.t.* $j - i \geq l$, every E-image of $A[i, j]$ induced by A must form a k-difference with $A[i, j]$ and $f_{E,k}(A[i, j]) = f_{E,k}(A)$
4. $\nexists\ A'$ s.t. A is a subsequence of A', every E-image of A induced by A' forms a k-difference with A, and $f_{E,k}(A) = f_{E,k}(A')$

From Definition 2.3.3, we see that A must have sufficient length and that there must be a certain number of sequences that form a k-difference with A. Additionally, no subsequence of A that satisfies the length and k-difference frequency requirements can occur outside of A. Lastly, A must be maximal, meaning that A is not a subsequence of some other sequence A' with equal k-difference frequency in the genomic sequence.

2.4 Survey of Repeat Finding Tools

Over the years following the discovery of the abundance of repetitive DNA, a variety of repeat finding algorithmic approaches and tools have been developed [14]. Repeat finding tools can be broken down into two main categories: library-based and ab initio.

Library Based Tools. Library-based tools compare input genomic data to an existing library of repeat sequences in order to identify known repeats. RepeatMasker [15] is currently the most widely used library-based repeat finding tool [14]. It compares the consensus sequences from known repeat

families, stored in a database called RepBase, to search for new members of each family based on similarity.

Ab Initio Tools. Ab initio tools attempt to identify repeats without using any pre-existing knowledge of known repeat sequences or motifs. While both of these techniques are widely used, *ab initio* tools are becoming increasingly important due to the rise in number and diversity of sequences coming from genome sequencing projects.

The approaches that have been used so far in the ab initio identification of repeats and are of interest to this research can be grouped into two basic categories [14].

1. k-mer approaches find all exact substrings that have a frequency equal to or greater than some defined threshold. Examples of tools that are based on this approach include REPuter [13] and RepeatScout [16].

2. Spaced seed approaches are similar to k-mer approaches, but they use spaced seeds when matching two strings in order to allow some predefined amount of differences between the strings (including substitutions, insertions, and deletions). PatternHunter [17] was the first tool to use this approach. It was later followed by PatternHunter II [18], which allowed for the use of multiple spaced seeds.

2.5 Lmer (k-Mer) Approach

One approach to repeat-finding that is of particular interest is the k-mer approach, which will henceforth be referred to as the *Lmer* approach. An Lmer is simply a subsequence of the genomic sequence of length equal to L, some fixed constant. We will begin to describe this topic using some observations and definitions from Figueroa [5].

Lemma 2.5.1. *Every Lmer belongs to at most one elementary repeat family.*

Definition 2.5.1. Given strings x and y, define $\mathbf{x \circ y}$ as the string abc, where $x = ab$, $y = bc$, and b is the longest substring that is both a suffic of x and a prefix of y.

Definition 2.5.2 (Lmer Series)**.** An **Lmer series** is a sequence of Lmers $x_0, x_1, \ldots x_{k-1}$ in the query sequence such that the start coordinate of x_i is one greater than the start coordinate of x_{i-1}. A sequence F is **composed** from an Lmer series if $F = x_0 \circ x_1 \circ \cdots x_{k-1}$.

Definition 2.5.3 (Lmer Decomposition)**.** Given a subsequence F of the query sequence, the **Lmer decomposition** of F is the Lmer series $x_0, x_1, \ldots x_{k-1}$ such that F is composed of the series.

Using these definitions, we can go on to redefine elementary repeats using Lmers.

2.6 Spaced Seed Approach

Now we begin to consider a method that has been used in both sequence alignment and repeat-finding tools in order to improve the identification of approximate repeats. Spaced seeds are basically patterns describing what positions in two strings must match and what positions in two strings can be a match or a mismatch (don't care positions). It is fairly easy to see that spaced seeds can aide in the identification of k-mismatches approximate repeats, but not in k-difference approximate repeats because they do not provide a mechanism to care about which character is at position i in one of the strings without caring about which character is at position i in the other string. We begin with a more formal definition of a spaced seed.

Definition 2.6.1 (Spaced Seed)**.** A **spaced seed** is a string π over the alphabet $\Sigma = \{1, *\}$, where a position with value 1 is a match and a position with value * is a "wildcard position" that can be either a match or a mismatch [19].

A spaced seed π is inherently defined by an ordered list of matching positions $M_\pi = \{i_1 \ldots i_w\}$ [20]. The number of matching positions is the seed's *weight*, denoted w_π. The *length* or *span* of the seed is denoted $|\pi|$.

As previously mentioned, we can use spaced seeds when determining whether or not two sequences form a k-mismatch for some threshold k and therefore can be categorized as a match. We will go on to define a match between two lmers in terms of spaced seeds, and then use this to build the definition of a match between two longer sequences with lmer decompositions.

Definition 2.6.2. Let π be a spaced seed of length L with matching positions $M_\pi = \{i_1 \ldots i_w\}$. Let q and t be genomic sequences of length L. We say that t **matches** q with respect to π if $q_i = t_i \ \forall i \in M_\pi$. Further, if the previous condition is met and $w_\pi \geq L - k$ we can say that t **k-mismatches** q with respect to π.

Definition 2.6.3. Let π be a spaced seed of length L with matching positions $M_\pi = \{i_1 \ldots i_w\}$. Let Q and T be two genomic sequences of length n with Lmer decompositions $x_1, x_2, \ldots, x_k$ and

$y_1, y_2, \ldots, y_k$, respectively (where $k = n - L + 1$). We say that T **matches** Q with respect to π if $\forall\ i \in \{0, n)\ \exists j \in \{0, n - L)$ such that $j \leq i < j + L$ and x_j matches y_j with respect to π.

We say that two genomic sequences Q, T match one another if every position $i \in \{0, n)$ corresponds to Lmers $x_j \in Q$ and $y_j \in T$ spanning positions $\{j, j + L)$ that match one another with respect to some spaced seed π.

2.7 RAIDER

This proposal seeks to improve an already existing repeat finding tool, known as Rapid Ab Initio Detection of Elementary repeats (RAIDER) [5]. RAIDER is a linear time repeat-finding tool that employs the use of Lmers, hashing, and spaced seeds. The results it gives have been compared to those of RepeatScout, a widely used repeat finding tool, and from this comparison it seems that the tools display similar sensitivity and specificity when detecting repeats. However, RAIDER currently only allows for a single spaced seed to be used in repeat detection, and this seed is chosen arbitarily [2]. The sensitivity of RAIDER to the detection of approximate repeats could be improved through a more in-depth study of spaced seeds and approximate repeats.

Proposed Research

As previously mentioned, we propose to do an in-depth study of a pre-existing tool, RAIDER [2], in order to both inspect and improve upon its current ability to detect approximate repeats. Many of the established and emerging repeat-finding tools seem to employ the use of k-mers, spaced seeds, or both in order to most effectively detect repeats [14]. Additionally, RAIDER is currently employing both of these approaches in its algorithmic approach. For these reasons, we will start this study by trying to use a fundamentally similar approach to RAIDER and investigating how to make RAIDER better use spaced seeds. Additionally, we will look into how to make RAIDER best use multiple spaced seeds, as is used in PatternHunter II [18].

We will also look into seed design and analysis, attempting to determine what kind of spaced seed is "best" for repeat-finding purposes. An optimized spaced seed design could allow for a few optimized spaced seeds to be as sensitive to repeat finding as many non-optimized spaced seeds.

This kind of information could allow for an improved detection of repeats without significantly affecting the space complexity of the RAIDER algorithm.

Along the way, it may become evident that the best way to improve RAIDER's detection of approximate repeats will be to pivot away from the use of spaced seeds in favor of some other approach. We remain open to this idea, especially since it seems that spaced seeds are limited in that they can only be used to determine k-mismatches approximate repeats, not k-differences approximate repeats.

Timeline

Task	Finish (End of listed month)
Research methods of spaced seed design and analysis in order to better understand how to approach optimizing spaced seeds for repeat-finding.	10/14
Determine how to best use spaced seeds in RAIDER and potentially make some changes to the algorithm to make it more amenable to spaced seed usage.	12/14
Work on optimizing spaced seeds for use in RAIDER.	01/14
Have one or several hypothesized optimal spaced seeds for RAIDER. Analyze the different results retrieved using these seeds.	02/14
Either continue trying to optimize individual spaced seeds or start working on ways to best use multiple spaced seeds.	04.5/14
Prepare results for final presentation.	05.5/14

Bibliography

[1] P. A. Pevzner, H. Tang, and G. Tesler, "De novo repeat classification and fragment assembly," *Genome research*, vol. 14, no. 9, pp. 1786–1796, 2004.

[2] N. Figueroa, X. Liu, J. Wang, and J. Karro, "Raider: Rapid ab initio detection of elementary repeats," in *Advances in Bioinformatics and Computational Biology*, pp. 170–180, Springer, 2013.

[3] R. Britten and D. Kohne, "Repeated sequences in dna," *Science*, vol. 161, pp. 529–540, August 1968.

[4] E. Lander, L. Linton, B. Birren, and et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 02 2001.

[5] N. Figueroa, "Raider: Rapid ab initio detection of elementary repeats," Master's thesis, Miami University, 2013.

[6] B. Lewin, J. Krebs, E. Goldstein, and S. Kilpatrick, *Lewin's Genes XI*, vol. 11. Jones & Bartlett Learning, 2014.

[7] M. Elloumi and A. Zomaya, *Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications*. Wiley Series in Bioinformatics, Wiley, 2011.

[8] T. J. Treangen and S. L. Salzberg, "Repetitive dna and next-generation sequencing: computational challenges and solutions," *Nature Reviews Genetics*, vol. 13, pp. 36–46, 01 2012.

[9] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[10] J. Kruskal, "An overview of sequence comparison: Time warps, string edits, and macromolecules," *SIAM Review*, vol. 25, no. 2, pp. 201–237, 1983.

[11] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet physics doklady*, vol. 10, p. 707, 1966.

[12] J. Zheng and S. Lonardi, "Discovery of repetitive patterns in dna with accurate boundaries," in *Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on*, pp. 105–112, IEEE, 2005.

[13] S. Kurtz, J. V. Choudhuri, E. Ohlebusch, C. Schleiermacher, J. Stoye, and R. Giegerich, "Reputer: the manifold applications of repeat analysis on a genomic scale," *Nucleic acids research*, vol. 29, no. 22, pp. 4633–4642, 2001.

[14] S. Saha, S. Bridges, Z. V. Magbanua, and D. G. Peterson, "Computational approaches and tools used in identification of dispersed repetitive dna sequences," *Tropical Plant Biology*, vol. 1, no. 1, pp. 85–96, 2008.

[15] A. F. Smit and P. Green, "Repeatmasker," *Published on the web at http://www. repeatmasker. org*, 1996.

[16] A. L. Price, N. C. Jones, and P. A. Pevzner, "De novo identification of repeat families in large genomes," *Bioinformatics*, vol. 21, no. suppl 1, pp. i351–i358, 2005.

[17] B. Ma, J. Tromp, and M. Li, "Patternhunter: faster and more sensitive homology search," *Bioinformatics*, vol. 18, no. 3, pp. 440–445, 2002.

[18] M. Li, B. Ma, D. Kisman, and J. Tromp, "Patternhunter ii: Highly sensitive and fast homology search," *GENOME INFORMATICS SERIES*, pp. 164–175, 2003.

[19] K. Chao and L. Zhang, *Sequence Comparison: Theory and Methods*. Computational Biology, Springer, 2008.

[20] J. Buhler, U. Keich, and Y. Sun, "Designing seeds for similarity search in genomic dna," *J. Comput. Syst. Sci.*, vol. 70, pp. 342–363, May 2005.