Approximate Repetitive DNA Identification

Charlotte E. Schaeffer

Abstract

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].

CONTENTS

I. 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].

II. Background

A. Biological Background

Every living organism inherits hereditary information from its parents that affect the organism's distinguishing traits [5]. 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) [6]. DNA can therefore be represented as a finite string $s = s_0s_1...s_{n-1}$ 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 [7]. 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.

B. 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 [8]. 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 [9]. 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 [9].

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* [10] 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 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

INDUSTRY	→ INTUSTRY	Substitute ’D’ by ’T’
	→ INTESTRY	Substitute ’U’ by ’E’
	→ INTERTRY	Substitute ’S’ by ’R’
	→ INTERERY	Substitute ’T’ by ’E’
	→ INTERESY	Substitute ’R’ by ’S’
	→ INTEREST	Substitute ’Y’ by ’T’

LEVENSHTEIN DISTANCE

INDUSTRY	→ INDUSTR	Delete ’Y’
	→ INDUST	Delete ’R’
	→ INRUST	Substitute ’D’ by ’R’
	→ INREST	Substitute ’U’ by ’E’
	→ INTREST	Insert ’T’
	→ INTEREST	Insert ’E’

Fig. 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).

C. *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 [11] 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.

Definition II.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. $A \subset A'$ and $freq(A') = freq(A)$

From Definition II.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.

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], [11].

Notation II.1. For two sequences A and A',

1) Let $R = \{r_1, \ldots, r_d\}$ be any set of replacement operations that will transform A into A'. Then,
   a) The *Hamming distance* between A and A' is denoted $d_H(A, A') = \min |R|$.
   b) We say that A *k-mismatches* A' if $d_H(A, A') \leq k$ for some constant k.
   c) If A is a subsequence of S, let $S_H(k, A)$ be the set of all non-overlapping subsequences that form a k-mismatch with A i.e. A k-mismatches $B\ \forall B \in S_H(k, A)$.

   d) For any substring C in A, the string B' that results from the transformation operations in $R*$ (the set of replacement operations of minimum length) as the *H-image* of B induced by A and is denoted as $I_H(A, A', C)$.

2) Let $E = \{e_1, \ldots, e_d\}$ be any set of elementary edit operations operations that will transform A into A'. Then,

   a) The *edit distance* between A and A' is denoted $d_E(A, A') = \min |E|$.
   b) We say that A *k-differences* A' if $d_E(A, A') \leq k$ for some constant k.
   c) If A is a subsequence of S, let $S_E(k, A)$ be the set of all non-overlapping subsequences that form a k-difference with A i.e. A k-differences B $\forall B \in S_E(k, A)$.
   d) For any substring C in A, the string B' that results from the transformation operations in $E*$ (the set of replacement operations of minimum length) is the *E-image* of B induced by A.

Definition II.2 (k-Mismatches Approximate Elementary Repeat)**.** Let S be an input genomic sequence. Let l, f be some fixed thresholds for the minimum length and frequency, respectively. Additionally, let k be a fixed constant for matching. A subsequence A of S is a **k-mismatches approximate elementary repeat** if:

1) $|A| \geq l$
2) $|S_H(k, A)| \geq f$
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 $|S_H(k, A[i, j])| = |S_H(k, A)|$

Definition II.3 (k-Differences Approximate Elementary Repeat)**.** Let S be an input genomic sequence. Let l, f be some fixed thresholds for the minimum length and frequency, respectively. Additionally, let k be a fixed constant for matching. A subsequence A of S is a **k-differences approximate elementary repeat** if:

1) $|A| \geq l$
2) $|S_E(k, A)| \geq f$
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 $|S_E(k, A[i, j])| = |S_E(k, A)|$

D. 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 [13]. Repeat finding tools can be broken down into two main categories: library-based and ab initio.

Library-based tools compare input genomic data to an existing library of repeat sequences in order to identify known repeats. RepeatMasker [14] is currently the most widely used library-based repeat finding tool [13]. 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 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.

E. Rapid Ab Initio Detection of Elementary Repeats

This proposal seeks to improve an already existing repeat finding tool, known as Rapid Ab Initio Detection of Elementary repeats (RAIDER).

F. Spaced Seeds

Definition II.4. 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 [15].

A spaced seed π is defined by an ordered list of matching positions $M_\pi = \{i_1 \dots i_w\}$ [16]. The number of matching positions is the seed’s *weight*, denoted w_π. The *length* or *span* of the seed is denoted $|\pi|$.

Definition II.5. Let π be a spaced seed of length L with matching positions $M_\pi = \{i_1 \dots i_w\}$. Let q and t be genomic sequences of length L. We say that t **matches** q with respect to π if $q_i = t_i \forall i \in M_\pi$.

Definition II.6. Let π be a spaced seed of length L with matching positions $M_\pi = \{i_1 \dots i_w\}$. Let Q and T be genomic sequences of length n with Lmer decompositions $x_1, x_2, \dots, x_k$ 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.

III. Proposed Research

IV. Timeline

References

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

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

[7] 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.

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

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

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

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.