CTRD: A Fast Applet for Computing Signed Translocation Distance between Genomes

WangSen Feng¹, Lusheng Wang² and Daming Zhu³

¹ Department of Computer Science, Peking University, Beijing, 100871, P. R. China
² Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
³ School of Computer Science and Technology, Shandong University, Jinan, Shandong, P.R. China

ABSTRACT
Summary: CTRD is a software for computing translocation distance between genomes. It takes two genomes as its input and tests whether one genome can be transformed into the other. If possible, it computes the translocation operation serial. We adopt the fastest known \(O(n^2 \log n)\) algorithm.

Our contributions include (1) give a necessary and sufficient condition to ensure that one genome can be transformed into the other for translocation operations, and (2) develop a software using the fastest known \(O(n^2 \log n)\) algorithm.

Contact: cswangl@cityu.edu.hk

1. INTRODUCTION
A computational approach to evolutionary studies based on rearrangements was pioneered by Sankoff et al. in 1990. A lot of work has been done in this field. For reversal distance, see [1],[2],[3],[6]. For transposition distance problem, see [4],[5]. Hannenhalli first studied the signed translocation distance in [7]. Zhu and Ma gave the fastest algorithm that runs in \(O(n^2 \log n)\) time in [8]. Given two signed genomes, it is not true that one genome can always be transformed into the other by using translocation operations. Previously, no one has seriously discussed the conditions that one signed genome can be transformed into the other for translocation operations. We give the necessary and sufficient conditions here and develop a software, CTRD, to compute the signed translocation distance between two genomes. We adopt the fastest known \(O(n^2 \log n)\) algorithm in [8]. CTRD takes two genomes (expressed as signed integer sequences) as input, and tests whether the source genome A can be transformed into the destination genome B using translocation operations. If yes, it computes the translocation distance between the two given genomes and outputs the translocation serial.

2. RELIMINARIES
A genome is a collection of chromosomes and a chromosome consists of a set of genes organized in a linear order. A gene is represented as a signed integer. A chromosome is a sequence of signed numbers. A genome is a set of sequences of signed numbers. For example, \(A=((1,2,3,4,5,6,7,-8,9,10,11,-12),(13,14,15,16,17))\) is a genome containing two chromosomes. The first chromosome has twelve genes: \(1,2, \ldots, 12\), and the second has five genes: \(13, 14, \ldots, 17\).

Translocation is one of the most common rearrangement events in mammalian evolution, it exchanges genetic material between different chromosomes. A translocation acting on chromosomes \(X=(X_1,X_2)\) and \(Y=(Y_1,Y_2)\) swaps the segments in the chromosomes and results two new chromosomes. Here we study the most common type of translocation, reciprocal translocation where each of the four segments, \(X_1, X_2, Y_1\) and \(Y_2\), is non-empty. A translocation is a prefix-prefix translocation if the prefix of one chromosome is swapped with the prefix of the other chromosome and a translocation is a prefix-suffix translocation, if the prefix of one chromosome is swapped with the suffix of the other chromosome. (See Figure 1.)

![Figure 1. Illustration of translocation operations.](image)

Prefix-prefix translocation

<table>
<thead>
<tr>
<th>X</th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y1</td>
<td>Y2</td>
</tr>
</tbody>
</table>

Prefix-suffix translocation

<table>
<thead>
<tr>
<th>X</th>
<th>X1</th>
<th>- Y1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y1</td>
<td>X2</td>
</tr>
</tbody>
</table>

3. TESTING CONDITIONS
Here we give the necessary and sufficient conditions that...
one genome can be transformed into the other.

**Theorem 1:** Considering the source genome $A$ and the destination genome $B$, $A$ can be transformed into $B$ by translocation if and only if:

1. The two genomes contain the same set of genes;
2. The two genomes contain the same number (must be at least 2) of chromosomes;
3. The two genomes have the same set of ending (either head or tail) genes;
4. For any gene $g$ that is an ending gene in $A$, (1) if $g$’s sign in $A$ is different from that in $B$, then $g$ must be a head in one genome and a tail in the other; (2) if $g$ has the same sign in both $A$ and $B$, then $g$ must be either a head in both genomes or a tail in both genomes.

4. IMPLEMENTATION

CTRD is a Java Applet developed with JDK 1.4.2_02 and its GUI is created with JFC/Swing. When you visit the web page, it is downloaded onto your machine and executed in your browser. We adopt the technique of Java Applet because it has two advantages (1) compatible with multiple platforms (CTRD can run on most operating systems and web browsers, including Unix and Windows, Netscape and IE), and (2) easily interact with users (GUI created with Swing is beautiful and user-friendly, all interactions are on client local computer without the need to communicate with the server).

A graphical user interface is provided. CTRD also provides on-line helps to users. The graphical user interface is shown in Figure 2.

![Figure 2](image)

Figure 2: The graphical user interface

CTRD accepts two genomes which are represented by integer sequences as its input. There are two input methods, you can either directly input source genome $A$ and destination genome $B$ in the “Input Area” by hand, (If the data is in a file, you can copy it to the “Input Area” as follows: (1) highlight the data and press “&C” and (2) move the focus to the input area and press “&V”:.) or click the “Examples” button to get a dialog and choose one item in the list, and click "OK" button to load the example.

After the two genomes are input, click the “Start” button to begin the computation. Then CTRD checks the possibility on whether the source genome $A$ can be transformed into the destination genome $B$ by translocation operations basing on Theorem 1. If yes, it will give the user the translocation distance between the two genomes and output the translocation operations. Users can copy the output result from the “Output Area” to his or her own text editor. The intermediate steps of genome transformation can be show in a separated window.

REFERENCES