

RESEARCH

Open Access



GRSR: a tool for deriving genome rearrangement scenarios from multiple unichromosomal genome sequences

Dan Wang¹ and Lusheng Wang^{1,2*}

From 13th International Symposium on Bioinformatics Research and Applications (ISBRA 2017)
Honolulu, Hawaii, USA. 30 May - 2 June 2017

Abstract

Background: Genome rearrangements describe changes in the genetic linkage relationship of large chromosomal regions, involving reversals, transpositions, block interchanges, deletions, insertions, fissions, fusions and translocations *etc.* Many algorithms for calculating rearrangement scenarios between two genomes have been proposed. Very often, the calculated rearrangement scenario is not unique for the same pair of permutations. Hence, how to decide which calculated rearrangement scenario is more biologically meaningful becomes an essential task. Up to now, several mechanisms for genome rearrangements have been studied. One important theory is that genome rearrangement may be mediated by repeats, especially for reversal events. Many reversal regions are found to be flanked by a pair of inverted repeats. As a result, whether there are repeats at the breakpoints of the calculated rearrangement events can shed a light on deciding whether the calculated rearrangement events is biologically meaningful. To our knowledge, there is no tool which can automatically identify rearrangement events and check whether there exist repeats at the breakpoints of each calculated rearrangement event.

Results: In this paper, we describe a new tool named GRSR which allows us to compare multiple unichromosomal genomes to identify “independent” (obvious) rearrangement events such as reversals, (inverted) block interchanges and (inverted) transpositions and automatically searches for repeats at the breakpoints of each rearrangement event. We apply our tool on the complete genomes of 28 *Mycobacterium tuberculosis* strains and 24 *Shewanella* strains respectively. In both *Mycobacterium tuberculosis* and *Shewanella* strains, our tool finds many reversal regions flanked by a pair of inverted repeats. In particular, the GRSR tool also finds an inverted transposition and an inverted block interchange in *Shewanella*, where the repeats at the ends of rearrangement regions remain unchanged after the rearrangement event. To our knowledge, this is the first time such a phenomenon for inverted transposition and inverted block interchange is reported in *Shewanella*.

Conclusions: From the calculated results, there are many examples supporting the theory that the existence of repeats at the breakpoints of a rearrangement event can make the sequences at the breakpoints remain unchanged before and after the rearrangement events, suggesting that the conservation of ends could possibly be a popular phenomenon in many types of genome rearrangement events.

Keywords: Comparative genomics, Genome rearrangement, Reversal, Transposition, Block interchange

*Correspondence: cswangl@cityu.edu.hk

¹Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., Hong Kong, People's Republic of China

²University of Hong Kong Shenzhen Research Institute, Shenzhen Hi-Tech Industrial Park, Nanshan District, Shenzhen, People's Republic of China



Background

Genome rearrangements involve gross changes of chromosomes and play important role in speciation. The problem of sorting signed genomic permutations arises in the study of species evolution via genome rearrangement. In this problem, chromosomes of interest are denoted by permutations of signed and ordered integers with each integer represents a conserved region (synteny block) in chromosomes. The order of the integers describes the order of synteny blocks on the chromosome and the sign (+ or -) of each integer indicates the transcriptional orientation. In the problem of sorting signed genomic permutations, given a pair of permutations and a set of rearrangement operations, we need to compute the minimum number of rearrangement operations in the set required to transform one permutation into another. The set of rearrangement operations can include one or more kinds of rearrangement operations, such as reversal, transposition, block interchange, fusion, fission, etc.

Many tools or algorithms have been proposed for sorting signed genomic permutations. In 1995, Hannenhalli and Pevzner provided a polynomial time algorithm for computing the minimum number of reversals, translocations, fissions, and fusions that would transform one multi-chromosomal genome to another [1]. Bafna and Pevzner proposed an algorithm for sorting permutations by transposition [2]. In 2002, Tesler improved the algorithm of Hannenhalli and Pevzner and proposed an efficient algorithm for multichromosomal genome rearrangements [3] and developed a tool named GRIMM to implement his algorithm [4]. SPRING is a tool for the analysis of genome rearrangement using reversals and block-interchanges [5].

In fact, given a pair of permutations, there are often more than one optimal rearrangement scenarios, especially when the rearrangement distance between this permutation pair is large. And sometimes, for the same pair of permutations, the computed rearrangement scenarios using different tools are not consistent. Hence, how to know whether the calculated scenarios are solid and biologically meaningful becomes an essential task.

Up to now, several mechanisms for genome rearrangements have been reported [6, 7]. Statistics analyzes showed that breakpoints are often associated with repetitive elements [8, 9]. There was evidence showing that a reversal can be mediated by a pair of inverted repeats (IRs) [10]. Rajaraman et al. suggested that rearrangements could be driven by the Insertion Sequences (ISs) and the positions of the reversal breakpoints in their study were also highly correlated with IS [11]. Darmon and Leach reviewed many examples of prokaryotic genomic rearrangements which were induced by natural transposable elements and pointed out that recombination between IRs can result in a reversal of the internal DNA sequence

[6]. The association between IR and genome rearrangement breakpoints was also reported in previous studies on mammals and drosophila genomes [12–14]. Hence, whether there exist repeats at the breakpoints of rearrangement events may give us a clue on whether the calculated rearrangement scenarios are biologically meaningful. To our knowledge, there is no tool which can automatically identify rearrangement events and check whether there exist repeats at the breakpoints of each calculated rearrangement event.

In this paper, we describe a new tool named GRSR for deriving genome rearrangement scenarios from multiple unichromosomal genome sequences and checking whether there are repeats at the breakpoints of each calculated rearrangement event. The input of the GRSR tool is a set of unichromosomal genome sequences and the output is pairwise rearrangement scenario which is a series of transpositions, block interchanges and reversals. Besides, for each calculated rearrangement event, GRSR checks whether there exist repeats which may mediate this rearrangement event. We applied the GRSR tool on complete genomes of 28 *Mycobacterium tuberculosis* strains and 24 *Shewanella* strains, respectively. We found many examples for supporting the theory that the existence of repeats at the breakpoints of the rearrangement can make the sequences at the breakpoints remain unchanged before and after the rearrangement events.

Methods

The GRSR tool can derive the pairwise rearrangement scenario and find repeats which may mediate a genome rearrangement event. The GRSR tool is comprised of four primary steps:

- 1 a multiple sequence alignment of the input genomes using Mugsy [15];
- 2 extraction of the coordinates of core blocks (sequences shared by all of the input genomes);
- 3 construction of synteny blocks and generating a signed permutation for each of the input genomes using GRIMM [4];
- 4 calculation of pairwise rearrangement scenarios using GRSR specific codes and GRIMM [4] and identification of repeats which may mediate rearrangement events using BLAST [16].

The input of the GRSR tool consists of multiple unichromosomal genomes (one file per genome). GRSR provides a Shell wrapper script for each of the four primary steps. Firstly, we use Mugsy to conduct a multiple sequence alignment of the input genomes and the alignment result is in an MAF file. Secondly, as transpositions, block interchanges and reversals happen on sequences which are shared by genomes, we extract the coordinates of

core blocks (shared by all of the input genomes) from the MAF file. Thirdly, we utilize the coordinates of core blocks to construct synteny blocks using GRIMM [4] and each input genome will be represented by a signed permutation describing the synteny block order on its chromosome. Lastly, we implement a novel method to compute the pairwise rearrangement scenario which is a series of rearrangement events involved in transforming one genome's permutation into another. The computed rearrangement scenarios will only include rearrangement events which happen on a single chromosome, such as transposition, block interchange and reversals. Once getting a rearrangement event, the GRSR tool will check whether there are repeats at the breakpoints of this event. The GRSR tool writes the rearrangement scenarios and whether there are repeats at the breakpoints of each rearrangement event into the *report.txt* file.

Multiple sequence alignment

We use Mugsy to conduct multiple sequence alignment of the input genomes. The alignment result is written to a file in the MAF format (See the “[Availability of data and materials](#)” section). In the MAF file, there are *core blocks* which include sequences shared by all of the input genomes and *accessory blocks* which include sequences shared by a subset of the input genomes.

Extraction of the coordinates of core blocks

As transposition, block interchange and reversals happen on sequences shared by genomes, we write a Shell script to extract the coordinates of core blocks on the input genomes. The coordinates of core blocks are written into a file named *core_coords.txt* (See the “[Availability of data and materials](#)” section) in the format as required by GRIMM.

Construction of synteny blocks and generating signed permutations

By utilizing the coordinates of core blocks, the GRIMM-SYNTENY tool inside the GRIMM package can construct synteny blocks and generate signed permutations for the input genomes. The coordinates of the synteny blocks on the input genomes are written into a file named *blocks.txt* and the signed permutations for the input genomes are saved in a file named *mgr_macro.txt* (See the “[Availability of data and materials](#)” section).

Calculation of pairwise rearrangement scenarios and finding repeats at the breakpoints of rearrangement events

For every pair of the input genomes, the GRSR tool will produce a rearrangement scenario. Hence, for n input

genomes, the GRSR will produce rearrangement scenarios for C_n^2 pairs of genomes.

Given a pair of signed permutations s and d , the GRSR tool calculate rearrangement scenario from s to d in three steps:

Step 1: merging blocks which are in the same order on s and d . For example, if

$$\begin{aligned} s &= 1, 2, 3, 4, 5, 6, 7, 8, 9 \\ d &= 1, 2, 3, 7, -6, -5, -4, 8, -9 \end{aligned} \quad (1)$$

after merging blocks,

$$\begin{aligned} s &= 1, 4, 7, 8, 9 \\ d &= 1, 7, -4, 8, -9 \end{aligned} \quad (2)$$

In the above example, as Block 1, 2 and 3 in Eq. 1 are in the same order on s and d , GRSR uses Block 1 in Eq. 2 to replace Block 1, 2 and 3 in Eq. 1. For the same reason, as Block 4, 5 and 6 in Eq. 1 are in the same order on s and d , the GRSR tool uses Block 4 in Eq. 2 to replace Block 4, 5 and 6 in Eq. 1 on s and use Block -4 in Eq. 2 to replace Block -6, -5 and -4 in Eq. 1 on d . We can find that merging blocks which are in the same order on s and d will not change the rearrangement scenarios from s to d .

Step 2: detecting independent transposition and block interchange. In a rearrangement scenario between two permutations, blocks which are involved in only one transposition or block interchange event are defined as blocks which are involved in an *independent* transposition or block interchange. After merging blocks in Step 1, the GRSR tool detects and removes blocks which are involved in *independent* transpositions or block interchanges before computing the reversal events using GRIMM.

Here we show all the cases of *independent* transpositions and block interchanges and represent *permutation s* as:

$$s = \dots, \pi_{i-1}, \pi_i, \pi_{i+1}, \dots, \pi_{j-1}, \pi_j, \pi_{j+1}, \dots \quad (3)$$

where π_i and π_j are arbitrary synteny blocks at respective position i and j on *permutation s* and $i \neq j$ and $1 \leq i, j \leq n$ (n is the total number of synteny blocks on *permutation s*). If π_i is π_1 , then π_{i-1} will be π_n . If π_i is π_n , then π_{i+1} will be π_1 .

We can say π_i is involved in an *independent* transposition from s to d once we find π_i on *permutation d* such that

$$\begin{aligned}
 d &= \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_j, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \pi_{i+1}, \dots, -\pi_j, \underline{\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_j, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, -\pi_{i-1}, \dots, -\pi_j, \underline{-\pi_i}, -\pi_{j-1}, \dots
 \end{aligned} \tag{4}$$

We can say π_i is involved in an *independent* inverted transposition from s to d once we find π_i on *permutation* d such that

$$\begin{aligned}
 d &= \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_j, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \pi_{i+1}, \dots, -\pi_j, \underline{-\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_j, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, -\pi_{i-1}, \dots, -\pi_j, \underline{\pi_i}, -\pi_{j-1}, \dots
 \end{aligned} \tag{5}$$

We can say π_i and π_j are involved in an *independent* block interchange from s to d once we find π_i and π_j on *permutation* d such that

$$\begin{aligned}
 d &= \dots, \pi_{i-1}, \underline{\pi_j}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \underline{-\pi_j}, \pi_{i+1}, \dots, -\pi_{j+1}, \underline{\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{\pi_j}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{-\pi_j}, -\pi_{i-1}, \dots, -\pi_{j+1}, \underline{-\pi_i}, -\pi_{j-1}, \dots
 \end{aligned} \tag{6}$$

We can say π_i and π_j are involved in an *independent* inverted block interchange from s to d once we find π_i and π_j on *permutation* d such that

$$\begin{aligned}
 d &= \dots, \pi_{i-1}, \underline{-\pi_j}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \underline{\pi_j}, \pi_{i+1}, \dots, -\pi_{j+1}, \underline{-\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{-\pi_j}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{\pi_j}, -\pi_{i-1}, \dots, -\pi_{j+1}, \underline{\pi_i}, -\pi_{j-1}, \dots
 \end{aligned} \tag{7}$$

We can say π_i and π_j are involved in an *independent* half inverted block interchange from s to d once we find π_i and

π_j on *permutation* d such that

$$\begin{aligned}
 d &= \dots, \pi_{i-1}, \underline{\pi_j}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \underline{-\pi_j}, \pi_{i+1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \underline{\pi_j}, \pi_{i+1}, \dots, -\pi_{j+1}, \underline{\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, \pi_{i-1}, \underline{-\pi_j}, \pi_{i+1}, \dots, -\pi_{j+1}, \underline{-\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{-\pi_j}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{-\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{\pi_j}, -\pi_{i-1}, \dots, \pi_{j-1}, \underline{\pi_i}, \pi_{j+1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{\pi_j}, -\pi_{i-1}, \dots, -\pi_{j+1}, \underline{-\pi_i}, -\pi_{j-1}, \dots \\
 \text{or} \\
 d &= \dots, -\pi_{i+1}, \underline{-\pi_j}, -\pi_{i-1}, \dots, -\pi_{j+1}, \underline{\pi_i}, -\pi_{j-1}, \dots
 \end{aligned} \tag{8}$$

Based on Eqs. 3, 4, 5, 6, 7 and 8, we developed a program in the GRSR tool to find and keep an record of independent transposition and block interchange events from s to d and search for repeats at the breakpoints of independent transposition and block interchange. The GRSR tool will then remove blocks involved in independent transposition and block interchange events in s and d before sorting by inversions using GRIMM. For example, in transforming the *permutation* s into *permutation* d in Eq. 2, Block 4 is found to be involved in an independent inverted transposition event using our GRSR tool. Since Block 4 substitutes Block 4, 5, and 6 in Eq. 1 in Step 1, the GRSR will eventually output Block 4, 5 and 6 are involved in a inverted transposition event. After writing this transposition event into the output Text file, GRSR will remove Block 4 in *permutation* s and d in Eq. 2, so s and d will become:

$$\begin{aligned}
 s &= 1, 7, 8, 9 \\
 d &= 1, 7, 8, -9
 \end{aligned} \tag{9}$$

For the purpose of using GRIMM in the next steps, GRSR make the permutation of s as 1, 2, 3, ..., n and change the permutation of d accordingly. Hence *permutation* s and d in Eq. 9 will become:

$$\begin{aligned}
 s &= 1, 2, 3, 4 \\
 d &= 1, 2, 3, -4
 \end{aligned} \tag{10}$$

In Eq. 10, we use Block 2, 3 and 4 to replace Block 7, 8 and 9 in Eq. 9, respectively. *Permutation* s and d like those in Eq. 10 will be the input of next step.

Step 3: sorting by reversals using GRIMM.

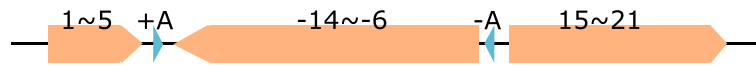


Fig. 1 Permutation for Strain 8, 17 and 20. Each orange block stands for one or several consecutive syntenic blocks. The integers above each orange block indicate the included syntenic blocks, for example, -14~-6 means the orange block includes the syntenic blocks from Block -14 to -6 on the permutation. Inverted repeat +A and -A is represented by triangles respectively. The arrow directions indicate the sign of each block

Kurono, KZN 4207, K, EAI5, H37Rv (NC_018143), H37Rv (NC_000962), Haarlem, KIT87190, CTRI-2, 7199-99, KZN 1435, 96075, 49-02, KZN 605, Erdman (ATCC35801), CCDC5180, CDC1551, EAI5/NITR206, H37Ra; ATCC 25177, HKBS1, ZMC13-264, Beijing/NITR203, respectively. There are a total of 122.9 Mbp for these 28 genomes.

The Mugsy tool completed the multiple sequence alignment of these 28 unichromosomal genomes in <2 days and the alignment result was saved in an MAF file. The coordinates of core blocks were then extracted and written into the *core_coords.txt* file.

By utilizing the coordinates of core blocks, GRIMM-Syntenic constructed 21 syntenic blocks and generated 28 permutations for each input genomes, which took less than 2 minutes. The 28 permutations are listed in Table 1. The coordinates of the syntenic blocks on each input genomes were saved in the *blocks.txt* file and the permutations were written in the *mgr_macro.txt* file.

Next, The genome rearrangement scenarios of 378 pairs of genomes were calculated in less than 10 minutes (there are 378 distinct pairs for 28 input genomes). And for each rearrangement event, the GRSR tool reported whether there were repeats flanking the rearranged regions. Here is a rearrangement scenario generated by the GRSR tool:

- Strain 1 to 8: total 1 rearrangement step(s):
 Step 1: Block 6-14, reversal. (Strain 1: Not found.
 Strain 8: Found, Length = 1355, Similarity = 100%)

According to the above results, we can see from Strain 1 to 8, there is a total of 1 rearrangement step which is a reversal of Block 6-14 and in Strain 8, the reversal region (Block 6-14) is flanked by a pair of inverted repeats with length = 1355 bp and similarity = 100%.

From the result generated by GRSR, we found that in Strain 8, 17, 20, the reversal region 6-14 are flanked by a pair of inverted repeats with length = 1355 bp and similarity = 100% as shown in Fig. 1. We can find that this pair of inverted repeats +A and -A can make sure the ends of the reversal region remain unchanged before and after the reversal event.

Shewanella

There are totally 24 strains with complete genomes for the *Shewanella* Genus in NCBI GenBank when we conducted this experiment. We downloaded all these 24 *Shewanella* strains and applied our GRSR tool on these 24 strains. There is a total 121 Mbp for these 24 input genomes.

The GRSR tool achieved 35 syntenic blocks for these 24 *Shewanella* genomes. The 24 genomes are divided into 21 groups according to their permutations. Genomes of the same permutation are in the same group. The permutations of the 21 groups (denoted as G1 to G21) are shown in Fig. 2. Group 6 consists of two *Shewanella baltica* strains OS155 and OS117. Group 15 includes three strains which are *Shewanella baltica* strains OS678, OS195 and OS185. Each of the remaining groups only

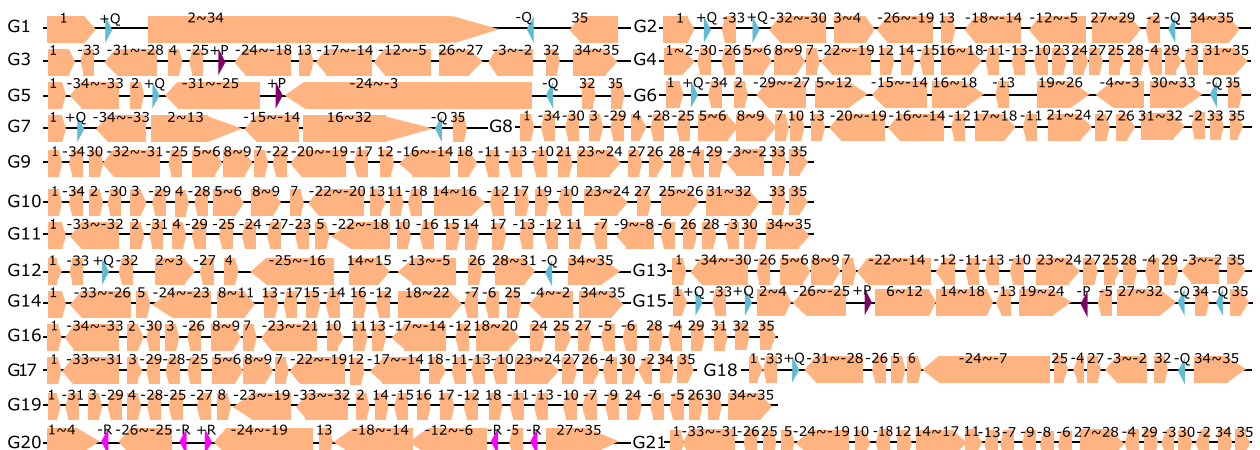


Fig. 2 Twenty one kinds of permutations for the 24 *Shewanella* strains. Repeats P, Q and R are represented by triangles respectively. The arrow directions indicate the sign of each block

Table 2 Summary of the characteristics of reversals mediated by IR with reversal distance = 1

sG	dG	rev _d	reversal	len (Mbp)	IR	r _d
1	2*	1	(34,1)	0.1530	Q(1)	3
1	5	1	(3,31)	3.4407	Q(2)	2
1	7*	1	(35,1)	0.0834	Q(2)	2
1	20	1	(6,24)	2.2358	R(1)	3
2	15*	1	(34,-33)	0.3553	Q(2)	2
3	15	1	(-24,-6)	2.6924	P(1)	3
5	15	1	(-24,-6)	2.4606	P(1)	4
6	15*	1	(35,1)	0.0564	Q(2)	3
12	18*	1	(34,-33)	0.2948	Q(2)	2
15	20	1	(6,24)	2.7479	P(0)	2
					R(1)	

Column sG and dG are the source and destination group. Rearrangement scenario is calculated from the permutation of the source group to the destination group. For dG with a asterisk, the reversal event is calculated from the permutation of sG to permutation of dG in the negative strand. For example, if the permutation of dG is 1, 2, 3, then the permutation of dG in the negative strand is -3, -2, -1. Column rev_d indicates the reversal distance between sG and cG after eliminating other independent rearrangement events. Column r_d indicates the distance of independent rearrangement events other than reversals. len is the length (in mbp) of reversal. Column IR lists which pair of inverted repeats (P, Q or R) flanks the reversal. The numeric code: 0 indicates the respective IR was found only in the source group, 1 indicates the IR was found only in the closest group, 2 indicates the IR was found in both groups

has one strain: Group 1 to Group 5 corresponds to *Shewanella putrefaciens* CN-32, *Shewanella baltica* BA175, *Shewanella* sp. ANA-3, *Shewanella halifaxensis* HAW-EB4, *Shewanella* sp. W3 18-1 respectively; Group 7 to Group 14 contains *Shewanella putrefaciens* 200, *Shewanella violacea* DSS12, *Shewanella woodyi* ATCC 51908, *Shewanella sediminis* HAW-EB3, *Shewanella denitrificans* OS217, *Shewanella* sp. MR-7, *Shewanella pealeana* ATCC 700345, *Shewanella oneidensis* MR-1 respectively; Group 16 to Group 20 contains *Shewanella amazonensis* SB2B, *Shewanella loihica* PV-4, *Shewanella* sp. MR-4, *Shewanella piezotolerans* WP3, and *Shewanella baltica* OS223 respectively.

Ten pairs of groups with reversal distance = 1 are found. As the chromosome of *Shewanella* is circular, the reversal region can contain the two ends of a permutation. For example, for reversal of Block 34~1 from Group 1 to 2, denoted as (34,1) in Table 2, the blocks involved in the

reversal events are Block 34, 35 and 1. All the 10 reversals (with lengths 56404 bp to 3440716 bp) are flanked by a pair of IRs (as listed in Table 2). In the 10 reversals, we found a total of three pairs of inverted repeats +P/-P, +Q/-Q and +R/-R. The locations of these three inverted repeats on the permutations are shown in Fig. 2. Six reversals are flanked by +Q/-Q, two reversals are mediated by +P/-P, one reversal is associated with +R/-R and the remaining one reversal between Group 15 to Group 20 is special because the inverted segment (6,24) is flanked by +P/-P in strains of Group 15 but by +R/-R in strains of Group 20. Sequence P is about 195 bp length, Q is about 5597 bp and R is about 1218 bp. Their lengths vary slightly in different strains.

We also find repeats at the breakpoints of inverted transpositions and inverted block interchanges in *Shewanella* strains. For example, from the permutation of G15 to that of G20, Block -33 is involved in an inverted transposition (See Fig. 3). The repeats denoted as +L/-L at the breakpoints of the transposition are also shown in Fig. 3. We can find that the three existence of repeat L can make sure the ends of the transposition remain unchanged before and after this inverted transposition.

Figure 4 shows repeats at the breakpoints of an inverted block interchange. From the permutation of G2 to G20, Block 3 to -25 (the gray region in Fig. 4) and Block -5 to 29 (the yellow region in Fig. 4) are involved in an inverted block interchange event. We can find that the existence of two pairs of inverted repeats +M/-M and +N/-N make the ends of the interchanged regions remain unchanged.

Discussion

Lots of algorithms have been proposed for computing rearrangement events with different kinds of rearrangement events such as reversals, transpositions, block interchanges, etc. However, it is hard to estimate in what sense the computed results reflect the true evolutionary history due to multiple solutions (especially when a large number of rearrangement events are involved) and different kinds of models for incorporating different kinds of rearrangement events. The GRIMM tool sorts permutations by reversal events only. As one transposition or block interchange event can be replaced by 3 reversal events, and one inverted transposition or inverted block interchange event can be replaced as 2 reversal events, the calculated reversal steps by GRIMM may not be the real reversal

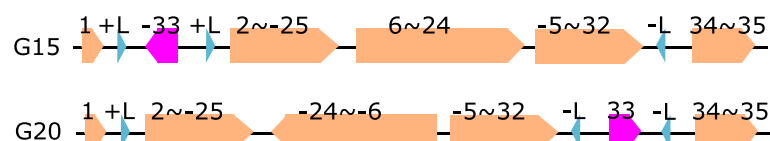


Fig. 3 Inverted transposition of Block -33 from G15 to G20. +L and -L are repeats at the breakpoints of this transposition

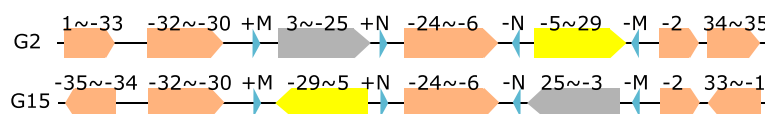


Fig. 4 Inverted block interchange between the gray and yellow region. +M/-M and +N/-N are repeats at the breakpoints of this block interchange

events. As a result, to better study the association between repeats and rearrangement events, our tool is designed to identify solid and obvious (independent) rearrangement events before searching for repeats at the breakpoints of rearrangement events

For current methods, computing the rearrangement distance between two chromosomes also suffers from multiple solutions. Incorporating the mechanism that rearrangement events are associated with repeats will reduce the chance of multiple solutions.

In this paper, we only consider the three operations reversals, block interchanges and transpositions where only one chromosome is involved. Further study will include rearrangement operations such as translocations, fissions and fusions where two chromosomes are involved.

Conclusions

From the results generated by the GRSR tool, we observed that many reversal events were flanked by a pair of inverted repeats so that the two ends of the reversal region remained unchanged before and after the reversal event. We also observed that in other rearrangement operations such as transposition and block interchange, there existed repeats (not necessarily inverted) at the breakpoints, where the ends remained unchanged before and after the rearrangement operations. This suggests that the conservation of ends could possibly be a popular phenomenon in many types of genome rearrangement events.

Abbreviations

IR: Inverted repeat; Mbp: Million base pair

Acknowledgements

We thank the reviewers for their helpful suggestions and comments. The abridged abstract of this work was previously published in the Proceedings of the 13th International Symposium on Bioinformatics Research and Applications (ISBRA 2017), Lecture Notes in Computer Science: Bioinformatics Research and Applications [17]

Funding

This work is supported by National Science Foundation of China (NSFC 61373048 to L.W. and D.W.) and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 11256116 to L.W. and D.W.). Publication of this article is also funded by these two grants (NSFC 61373048 and CityU 11256116).

Availability of data and materials

The datasets supporting the conclusions of this article are available in the GenBank of NCBI (<https://www.ncbi.nlm.nih.gov/genbank/>). The accession numbers of the 28 *Mycobacterium tuberculosis* genomes are NZ_CP009101, NZ_CP002883, NC_021251, NC_009565, NZ_CP002882, NC_017522, NZ_AP014573, NC_016768, NZ_CP007803, NC_021740, NC_018143, NC_000962, NC_022350, NZ_CP007809, NC_017524, NC_020089, NC_012943,

NZ_CP009426, NZ_HG813240, NC_018078, NC_020559, NZ_CP002885, NC_002755, NC_021194, NC_009525, NZ_CP002871, NZ_CP009100 and NC_021054. The accession numbers of the 24 *Shewanella* genomes are NC_009438, NC_017571, NC_008577, NC_010334, NC_008750, NC_009052, NC_017566, NC_014012, NC_010506, NC_009831, NC_007954, NC_008322, NC_009901, NC_004347, NC_016901, NC_017579, NC_008700, NC_009092, NC_008321, NC_009997, NC_011566, NC_009665, NC_011663 and NC_008345.

Source codes for the GRSR tool are available at <https://github.com/DanwangJessica/GRSR>. Through this link, you can find an example which includes input and output files of each of the four steps of GRSR.

About this supplement

This article has been published as part of *BMC Bioinformatics* Volume 19 Supplement 9, 2018: Selected articles from the 13th International Symposium on Bioinformatics Research and Applications (ISBRA 2017): bioinformatics. The full contents of the supplement are available online at <https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-19-supplement-9>.

Authors' contributions

DW collected the datasets, designed and developed the software, performed and analyzed the experiments, and wrote the paper. LW directed this research, designed the methods, analyzed the experimental results, and wrote the paper. Both authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published: 13 August 2018

References

- Hannenhalli S, Pevzner PA. Transforming men into mice (polynomial algorithm for genomic distance problem). In: Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium On. USA: IEEE; 1995. p. 581–92.
- Bafna V, Pevzner P. Sorting permutations by transpositions. In: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms. USA: Society for Industrial and Applied Mathematics; 1995. p. 614–23.
- Tesler G. Efficient algorithms for multichromosomal genome rearrangements. *J Comput Syst Sci.* 2002;65(3):587–609.
- Tesler G. Grimm: genome rearrangements web server. *Bioinformatics.* 2002;18(3):492–3.
- Lin YC, Lu CL, Liu Y-C, Tang CY. Spring: a tool for the analysis of genome rearrangement using reversals and block-interchanges. *Nucleic Acids Res.* 2006;34(suppl 2):696–9.
- Darmon E, Leach DR. Bacterial genome instability. *Microbiol Mol Biol Rev.* 2014;78(1):1–39.
- Gray YH. It takes two transposons to tango: transposable-element-mediated chromosomal rearrangements. *Trends Genet.* 2000;16(10):461–8.

8. Longo MS, Carone DM, Green ED, O'Neill MJ, O'Neill RJ, et al. Distinct retroelement classes define evolutionary breakpoints demarcating sites of evolutionary novelty. *BMC Genomics*. 2009;10(1):334.
9. Sankoff D. The where and wherefore of evolutionary breakpoints. *J Biol*. 2009;8(7):1.
10. Small K, Iber J, Warren ST. Emerin deletion reveals a common X-chromosome inversion mediated by inverted repeats. *Nat Genet*. 1997;16:96–9.
11. Rajaraman A, Tannier E, Chauve C. Fpsac: fast phylogenetic scaffolding of ancient contigs. *Bioinformatics*. 2013;29(23):2987–94.
12. Thomas A, Varré J-S, Ouangraoua A. Genome dedoubling by dcj and reversal. *BMC Bioinformatics*. 2011;12(9):1.
13. Bailey JA, Baertsch R, Kent WJ, Haussler D, Eichler EE. Hotspots of mammalian chromosomal evolution. *Genome Biol*. 2004;5(4):1.
14. Armengol L, Pujana MA, Cheung J, Scherer SW, Estivill X. Enrichment of segmental duplications in regions of breaks of synteny between the human and mouse genomes suggest their involvement in evolutionary rearrangements. *Hum Mol Genet*. 2003;12(17):2201–8.
15. Angiuoli SV, Salzberg SL. Mugsy: fast multiple alignment of closely related whole genomes. *Bioinformatics*. 2011;27(3):334–42.
16. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol*. 1990;215(3):403–10.
17. Cai Z, Daescu O, Li M. *Bioinformatics Research and Applications: 13th International Symposium, ISBRA 2017, Honolulu, HI, USA, May 29–June 2, 2017, Proceedings*. vol. 10330. Berlin, Heidelberg: Springer; 2017.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

