To whom it may concern,

In the following pages you can find the drafts for two of our papers:

1- Qu-SV: Quick Detection of Structural Variants in Haploid, Diploid, and Tumor Genomes.

2- UGALA: Ultrafast Genome Alignment based on Limited Anchoring.

I have to mention that some parts are missing and will be available on the official preprint. However, I am available for a complete presentation of the project. These drafts are not fully completed and I will update them gradually.

Amirhossein.

a.afshinfard@gmail.com

Qu-SV: Quick Detection of Structural Variants in Haploid, Diploid and Tumor Genomes

Amirhossein Afshinfard†, Damoun Nashta-ali†, Seyed Abolfazl Motahari

February 28, 2018

Abstract

Structural Variants are among the main contributors to different types of genomic disorders. Because of the many challenges in detection (such as repeat regions, sequencing biases and errors, heterozygous variations, etc.), proposing novel and versatile methods is still a topic of concern. In this study, we propose an ultrafast novel approach to detect these changes from sequencing reads, regardless of that the genome is a haploid, diploid or polyploid. In this approach, non-informative reads are being identified and eliminated very quickly and efficiently. Simultaneously, anchoring variation regions, informative read segments build up clusters of events along the genome with some events being connected because of sharing some identical reads. Parsing the resulted sparse graph of events, the algorithm discovers the exact location, type, and sequence of each alteration. Evaluations of this method were successful in both speed and accuracy and additionally showed its ability to detect complex variants.

[updating - please contact a.afshinfard@gmail.com for slides and online presentation - the content of this paper is not reviewed and corrected for academic writing and grammatical mistakes yet - we are updating the results and figures]

1 Introduction

Structural Variants (SVs) are generally defined as genetic variations in DNA sequence other than mutations which involve one or a few base pairs. SVs, despite single nucleotide polymorphisms (SNPs) or single nucleotide variants (SNVs), requires the disruption of the sugar-phosphate backbone of DNA and thus involve more base pairs [1]. In other words, Structural Variation (SV) refers to a variation that changes the structure of the genome. This variation may change the length of the chromosome (Unbalanced SV such as novel insertion, deletion, copy-number variation CNV) or only change the structure without affecting its length and content (balanced SV, including inversion and reciprocal translocation). Figure 1 shows three types of these variations.

Recent research in the last decade has revealed that SVs are much more frequent than previously thought [2, 3] and they build up a large fraction of the human genome variation, even more than SNPs [4, 5]. This opens a new field in genetic and genomic studies and motivates lots of research on methods for detection of structural variants as a primary step for further investigations. The next desired steps are revealing phenotypic impacts of SVs (disorder and non-disorder impacts) as well as studying their population genetics [6], their formation mechanisms, and genome evolution.

SVs, observed both as germ-line and as somatic variation, contribute to genomic disorders ranging from neurodevelopmental disorders (including schizophrenia [7] and autism [8, 9]) to childrens developmental disorders [10], blood diseases [11], diabetes

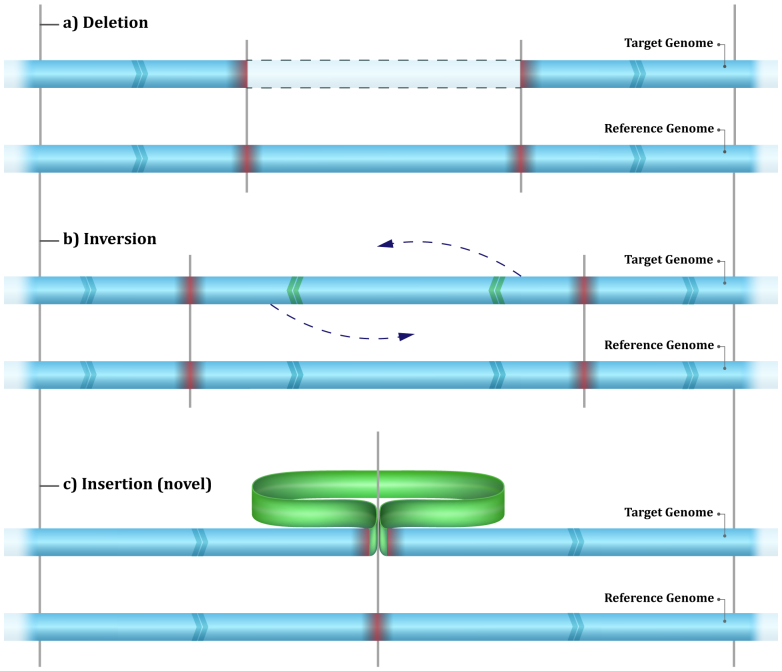

Figure 1: Structural Variation. a) a region on the target genome is deleted, b) an inverted region on the target genome (a balanced SV), and c) a novel sequence inserted.

[12], a wide range of cancers [13, 14, 15], etc. By influencing gene expression directly or indirectly [16], they revealed to involve not only genomic disorders but also other complex traits and phenotypes at various levels [17].

Structural variants are caused by different mutational mechanisms including DNA recombination, replication, and repair-associated processes [1]. Investigating breakpoints of structural variants at base pair resolution is crucial for understanding their formation mechanisms [18]. As SVs are likely responsible for gene and genome evolution [19], the discovery of new structural variants will help to study formation mechanisms and their population genetics to yield a better understanding of genome evolution.

2 Challenges

Detection of SVs is still a topic of concern, provided that there is an annual increase in the number of discovered alterations [20, 21] as there are biases in each method for detection of different types and various lengths. Many methods are limited to detection of longer SVs and some of them are able to search for specific types (e.g. many misses balanced SVs which are of great importance [22]) due to their nature. SV detection methods are encountered with many challenges such as repeat region, read mappability, and many fundametal biases in sequencing: length of reads, error rate, depth of sequencing [23], reduced read mappability or poor Genome Mappability Score (GMS) [24], especially in repeat-rich regions and thus reduced signal-to-noise ratios. In Addition to these difficulties, it is more challenging should we consider complex SVs, diploid genomes, and sequencing reads from tumor samples containing somatic mutations. [Updating]

3 Current Approaches

There are four general approaches to detect structural variants each having specific advantages and limitations. We will review these methods quickly and list their Strengths and weaknesses.

Assembly-based methods theoretically can discover structural variants of all types. Having built the assembly of the targeted genome, these methods detect structural variants by aligning it to the reference genome. methods of these types are limited to read length, and are involved with other challenges of the assembly problem itself [25]. the computational cost is a case in point. A novel idea in this regard is to use this approach locally (local assembly). using such approach to discover the sequence of novel insertions is inevitable.

Assuming an expected distribution for sequencing depth, **Read-count** (RC or Read-Depth) methods search for SVs by checking the divergence of local estimated distributions from expected ones. The assumption itself is challenging due to the non-uniform behavior of read depth, and some factors found to affect read counts. In example, GC-content has been revealed to affect sequencing process [- -] and result in non-uniform read coverage which can somewhat be corrected and unbiased [- -]. Another challenge in this approach is to decide if a change caused due to noise or because of a variation; so theyre almost insensitive to small SVs. locating breakpoints with base pair resolution could be addressed hardly using RC. In addition, this approach is almost blind to balanced SVs, especially for inversions. However, theyre really powerful to discover deletions and Copy Number Variations (CNVs). Methods based on RC can estimate exact copy number while others are weak or at least are faced with many challenges in this regard.

Read-pair (RP) methods utilize the capability of paired-end reads (or mate-pair reads) to detect SVs. Paired-end reads expected to have a specific orientation and a known insert-size distribution with respect to the reference genome, the sequencing

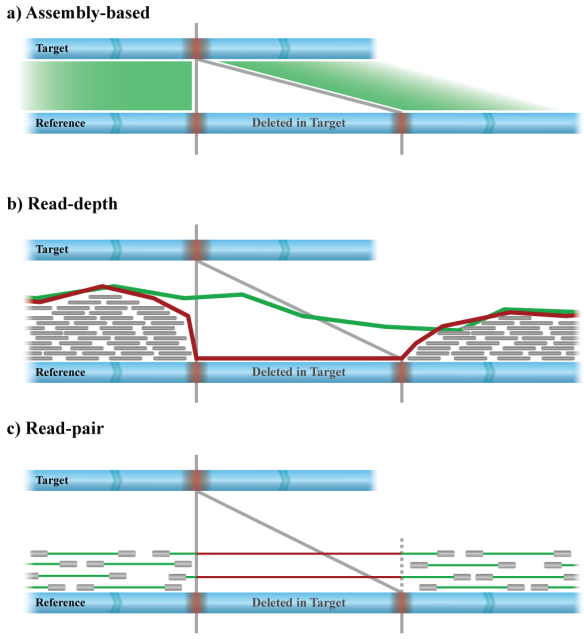

Figure 2: current approaches, [updating]

technology, and its parameters. Changes in orientation is a sign (an indication of) of inversion occurrence, while insertion and deletion events result in higher or lesser insert-size. Like RC, the breakpoint resolution is not exact and again it is difficult to decide if a change in insert-size distribution is caused by noise or by a variation event, especially in the case of small SVs. However, they are likely to have more information for such cases comparing with RC and also able to detect inversion. As a limitation, these methods are not sensitive to insertions longer than the insert-size and the detection needs more process steps on one-end aligned reads with using other methods like RC and SR. CNVs will be a big challenge for methods of this class. (smaller SVs in comparison with RC?)

Split-Read (SR) methods can result in breakpoints detection with base pair resolution (precision). A breakpoint-spanning read (an informative read) is unlikely to map, but split subsequences of that read tend to map to distant locations of the reference genome. [Incomplete] better than others for small SVs. Not good for CNVs and poor in repeat regions. Limited to read length. Applicable to the case of paired-end reads, checking the unmapped end of one-end anchored reads.

4 Proposed method

4.1 Method Overview

Suppose some reads from a target genome containing SVs. Assume that we have an Ideal read mapper meaning that it can assign each part of a single read to its corresponding location on the reference genome. With this attention, reads are divided into two categories (figure 3). The first group reads which span a breakpoint become split into (at least) two parts each of which maps to a distinct region on the reference. Every read from the second category maps entirely to a specific location without any division. We call the first group informative reads since they capture a breakpoint and thus they can signal a Structural Variation. The latter group reads are virtually non-informative reads. Why virtually? Leave it for now, but lets say they only contain useful information about the depth of their location.

In the first stage, which is the mapping step, our Idea is to remove non-informative

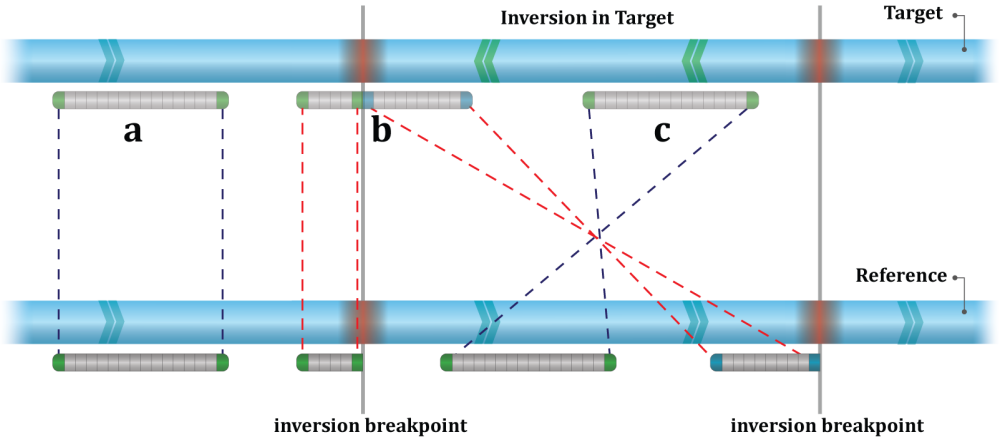

Figure 3: informative vs. non-informative reads from the target genome an their mappings via the assumed ideal aligner to their corresponding locations on the reference. a) a non-informative read in a normal region, b) an informative read which is split for mapping, and c) a non-informative read inside an inversion event

reads expeditiously, and to approach that Ideal read mapper for informative reads at the same time. For this purpose, we have exploited and revised an existing read aligner, Meta-aligner [26], so it can detect non-informative reads very quickly and efficiently without reporting any informative read as non-informative. In the same run, the revised version is able to split the informative reads into multiple segments and assign each segment into its original source on the reference genome. This way, almost all non-informative reads are identified and their contribution to local depths are considered. Then they are filtered, and the focus will be on split-mapped informative reads to detect SVs in the next stage. Indeed, informative reads pile up a very small fraction of the total number of reads, resulting in an elbowroom for applying algorithms of high complexity in the second stage without concerning about the computational cost.

As mentioned before, non-informative reads are not non-informative at all. Their contribution in local depths is useful, especially for detecting some types of CNVs (like RC methods). Thus, without any extra computation, they account for depth before being removed. Additionally, they are saved in buckets for further access in the second stage if needed.

In the second stage, overlapping informative reads which share the same direction

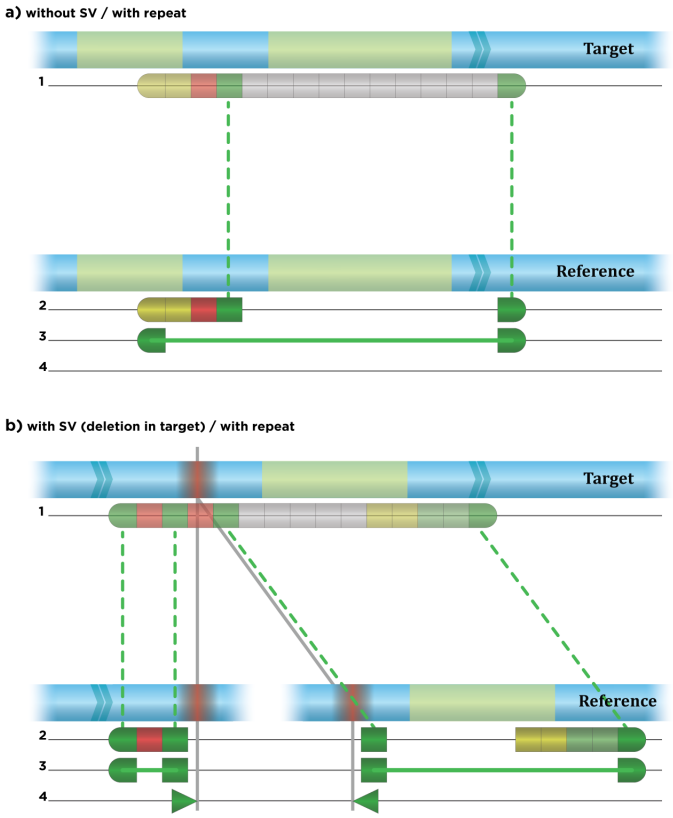

Figure 4: read chunking and mapping strategy a) non-informative read b) informative read [updating] / line 1: read sampled from the target genome; line2: mapping of the read chunks to reference genome; line3: after extension; line 4: re-alignment (optional) and breakpoint detection. / Green: unique true matches; Red: unmapped or wrong matches due to sequencing errors or small mutations; Yellow: non-unique chunks

(whether the breakpoint is right or left) contribute to piling up distinct events. Informative reads with more than one location (contributing to more than 1 event) connect some of these events together resulting in a sparse graph. In this graph, events are nodes and identical shared reads between events are edges. Multiple edges is equal to one weighted edge. For detection of SV types from subgraphs in the resulting sparse graph, a statistical approach measures the probability of each subgraph and maximize it given SV $type_i$ (equatuin 4). According to equations 4 and 5, it is done by likelihood maximization. There are five general types of SVs. The ith type which maximizes the likelihood is returned as the type of the SV for that subgraph.

[Updating:]

$$\underset{i}{\operatorname{argmax}}\, \mathbb{P}\left(type_i \mid subgraph\right) \tag{1}$$

$$= \underset{i}{\operatorname{argmax}} \sum_{Length} \mathbb{P}\left(type_i \mid subgraph, Length\right) \mathbb{P}\left(Length \mid subgraph\right) \tag{2}$$

$$= \underset{i}{\operatorname{argmax}} \sum_{Length} \frac{\mathbb{P}\left(subgraph, Length \mid type_i\right) \mathbb{P}\left(Length \mid subgraph\right)}{\mathbb{P}\left(subgraph\right) \mathbb{P}\left(Length \mid subgraph\right)} \tag{3}$$

$$= \underset{i}{\operatorname{argmax}} \sum_{Length} \frac{\mathbb{P}\left(subgraph \mid type_i, Length\right) \mathbb{P}\left(Length \mid type_i\right)}{\mathbb{P}\left(subgraph\right)} \tag{4}$$

In equation 4, $\mathbb{P}\left(subgraph\right)$ is not effective in maximization and thus:

$$= \underset{i}{\operatorname{argmax}} \sum_{Length} \mathbb{P}\left(subgraph \mid type_i, Length\right) \mathbb{P}\left(Length \mid type_i\right) \tag{5}$$

Here for a fixed Length "$Length$", $\mathbb{P}\left(Length \mid type_i\right)$ is considered as uniform since the distribution of the SV types are unknown. They are not achievable from the current discovered SVs as current methods are biased. Hence, it can be eliminated as well.

$$= \underset{i}{\operatorname{argmax}} \sum_{Length} \mathbb{P}\left(subgraph \mid type_i, Length\right) \tag{6}$$

The next step is to define $\mathbb{P}\left(subgraph \mid type_i, Length\right)$ which we will discuss in 4.3 in the incoming official preprint. Concisely, each subgraph suggests some fixed lengths and we exploit a term to penalize the difference between the proposed lengths by each subgraph and the $Length$. This penalty increases exponentially and dwindles the probability to zero for most of the lengths.

Another approach, exploited in the second stage, builds up local assemblies around locations anchored with informative reads. Resulting local assemblies being re-aligned to the reference genome and identify the type of each SV by rearrangement minimization. This approach helps in the detection of Complex SV with complicated breakpoints near each other.

4.2 Stage one: Qu-break

in section 4.2.1 and 4.2.2, we roughly discuss the idea and the algorithm. one can skip these sections and jump to section 4.2.3 for more detailed method and algorithm along with statistical analysis.

4.2.1 Meta-Aligner

The original version of the Meta-Aligner [26] uses the genome statistics and suggests a minimal size $2l$ for unique mapping of a subread, statistically sufficient to confidently assign the read to the location anchored by that subread. In the example of the human genome, the optimal size is approximately 50 bps. Excluding variants from consideration, It means that if we find a 50 bps long sub-read from the read uniquely in the reference, it is enough to be confident that this is the right location for that read. Another great idea of the Meta-Aligner is to divide this needed $2l$ bps unique mapping into two disjoint sub-reads of length l (2x25 for Human genome).

In a nutshell, the algorithm searches for two disjoint sub-reads of length 25, mapped uniquely to the genome and concordantly to each other. Finding these two sub-reads, the algorithm skips the remaining base pairs of the read and assigns the read to the location. Searching for a sequence of length 25 for unique hits in the reference can be done very quickly, making the algorithm extremely fast. Not only does this approach eliminate the cost of mapping of a large portion of many reads, but it also overcomes the challenge of repeat regions when mapping reads with a minimal mapping size. + an example of a run by Meta-Aligner and proportion of reads aligned after each iteration.

4.2.2 Revision: Qu-Break

When considering SVs, some reads (informative reads) are not related to a single location when aligned to the reference genome. Thus, two disjoint sub-reads cannot suggest the location of the read as a whole. However, two sub-reads of an acceptable distance of length L_{in} can decide about the original location of the proportion between the two sub-reads. L_{in} should be small enough that it is possible to distinguish between small scale-mutations and SVs. We will discuss calculating L_{in} in another research but let's assume that it is five times as big as the smallest SV which we want to detect. As a result, if the distance between two concordant sub-reads is less than L_{in}, the location for that proportion of the read is known. Relatively, if we find two pairs of concordant sub-reads, it means that this read captures a breakpoint (assume there is no errors nor noises).

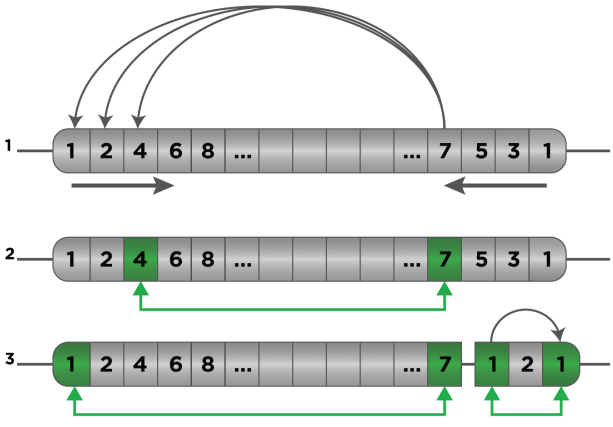

Figure 5: read chunking and mapping strategy - line1: the order of unique mapping and checking for concordance for disjoint sub-reads. line2: found concordant sub-reads. line3: first pair after extension and second pair after recursively checking the remaining chunks

With this in mind, the aim is to revise Meta-Aligner in order to detect non-informative reads quickly and with high accuracy. At the same time, it should find sub-read pairs in informative reads. For this purpose, we propose Qu-Break. This algorithm divides a read of length L_{read} into disjoint sub-reads (chunks) of length l_{sub} like what is depicted in figure 5. The algorithm starts by searching for unique hits for the first and last chunks and continue by searching for inner chunks one by one. Finding two concordantly-mapped sub-reads, the algorithm stops the unique search assuming the proportion between two concordant sub-reads as resolved. Thereafter, an extension phase starts to widen the resolved proportion. We will discuss the extension step later in 4.2.4. After extension, assuming the remaining unresolved basepairs as a new read, the algorithm search for concordant sub-reads in a recursive manner.

With the procedure discussed above, the algorithm find non-informative reads very quickly and by checking only a small proportion of the reads. Statistical analysis of the number of searched sub-reads is available in section 4.2.3. Additionally, practical results for human chr19 is available in section 5.1. In addition, a small proportion of reads, informative reads, are split into multiple segments each of which mapped to a specific location and threaded to each other (figure).

Qu-Break has two other optional procedures which are useful for further research in the second stage. Firstly, Qu-break can save a encoded mapping depth from non-informative reads which is crucial for investigation about the frequency of each SV in input genomes (in case of diploid or polyploid genomes like tumor genomes). Secondly, Qu-Break can save non-informative reads in buckets. Each bucket is for a fixed proportion of the genome. This way, it is possible to access the reads for a specific range very quickly.

4.2.3 Qu-Break: quick mapping of reads to locate breakpoints

[Updating:] figure 9 is related.

An ideal read mapper can map reads that span a SV breakpoint into the reference genome as shown in figure **??**. Classical methods consume a huge volume of process for mapping input reads to the reference genome and then call for SVs from the result of read mapping. The fundamental question is that, can SVs be resolved with lesser process? In this paper, we propose a method to resolve SVs only by anchoring reads in a very low-complexity way. As Figure **??** shows, the key point to resolve a SV is the detection of the breakpoints of that SV. This detection can be obtained by finding only ℓ-mers of reads which locate before and after the breakpoint, for some ℓ.

Assume that the ideal mapper exists which can map all ℓ-mers to their correct location. Consider a read r of length L comes from a SV region. Call all ℓ-mers of r as l_i's for $i \in \{1, \cdots, L-\ell+1\}$. Define *consistent sets* $\mathcal{C}_j(r)$'s (for some j, $1 \leq j \leq L-\ell$) such that each consistent set includes all ℓ-mers of r with consistent mapped locations. Normally, an anchored read has at least one consistent set. However, by bridging each SV (with at least ℓ-mer after SV) one consistent set is added. The first and the last ℓ-mers of each consistent set are called *informative ℓ-mers* if these ℓ-mers are different from l_1 and $l_{L-\ell+1}$. The informative ℓ-mers determine breakpoints of SVs and these are very important for calling SVs. Thus, the basic question alters to find these informative ℓ-mers efficiently.

For this purpose, we utilize the genomes model proposed by Meta-Aligner [26]. This model is based on two parameters, ℓ for fragment (sub-read) length and d for distance of mapping. Meta-Aligner proposes that a read can be anchored to the reference genome with only two uniquely and concordantly mapped disjoint random ℓ-mers, i.e. two disjoint ℓ-mers which are mapped to the genome uniquely and their reported locations are consistent. This model helps to anchor a read with only its $2 \times \ell$ bases (in ideal case)

without locally align other bases of reads. Also, using two ℓ-mers we can guess that a read has two consistent sets or not.

Using one ℓ-mer from one side and other ℓ-mer from the other side of read is a proper way for anchoring reads considering SVs, because these ℓ-mers provide information about two sides of the read. For evaluating this claim, assume that SVs occur with a Poisson distribution with rate of ψ ($\psi \ll 1$, i.e. ψ is very small) and the first step of the mapping algorithm. Therefore, if the first and last ℓ-mers of a read r confirm with each other, no SV exists within r with probability

$$p_{no} = e^{-\psi(L-\ell)(1+\mathcal{G})},$$

where $\mathcal{G}$ is a constant allows gap bases for confirmation ℓ-mers and it depends on the gap rate between the reference and target genomes and the gap error rate of the sequencing machine. For short-reads this probability is very close to one. However, for long-reads, we consider PacBio data set in NCBI GenBank SRX533609 (http://www.ncbi.nlm.nih.gov/sra?term=SRX533609) to determine this probability. Fig 6 shows the distribution of length of PacBio reads. Assume that $\mathcal{L}$ is a random variable shows read length. By setting $\ell = 25$, the average probability of no SV exist within PacBio reads when the first and last ℓ-mers confirm each other (i.e., $\mathbb{E}\{p_{no}\}$) is shown in Figure 7 for the range of $\psi \in [10^{-6}, 10^{-4}]$.

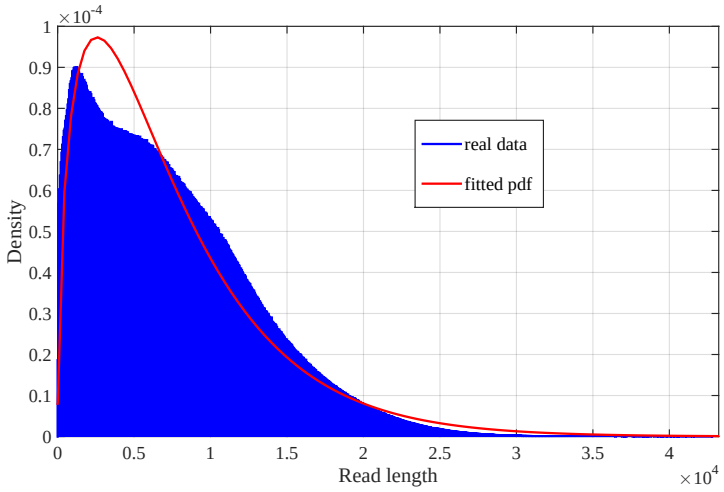

Figure 6: PacBio read length distribution and its fitted Negative-Binomial pdf.

Therefore, by anchoring reads by any two first confirmed mapped ℓ-mers from both sides, SVs are missed with very small probability. After mapping all ℓ-mers of a read, we check all ℓ-mers together to find if a read is anchored or not. Then, all consistent sets are extracted from mapped ℓ-mers. In an ideal case, one read with at least two consistent sets is enough to call a SV. However, error of sequencing and not ideal mapper cause some failures. In this case, we use information of all anchored reads which cover the SV breakpoint. By considering these reads, many informative ℓ-mers are located near each other before and after the breakpoint. Thus, we detect breakpoint of SVs by counting informative ℓ-mers.

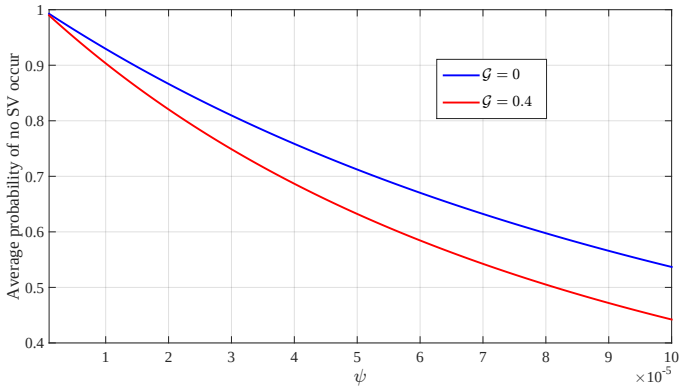

Figure 7: Probability of no SV occurs within the anchored reads for PacBio read set.

Before the detection step, we focus on two imperfectnesses: 1) mapping failures: due to sequencing error, variation between genomes and repeat regions, and 2) loosing SVs: some SVs with maximum length of $L-\ell$ that do not change the length of the genome or small SVs that change the length of the genome are discarded during anchoring reads. For the first problem, we propose the outer extension method such that the first (resp. second) informative ℓ-mer of the consistent set $\mathcal{C}_i(r)$ is extended toward the second (resp. first) informative ℓ-mer of the consistent set $\mathcal{C}_{i-1}(r)$ (resp. $\mathcal{C}_{i+1}(r)$) as Figure 8 shows. If $\mathcal{C}_{i-1}(r)$ does not exist, then the second informative ℓ-mer of the consistent set $\mathcal{C}_{i-1}(r)$ becomes the first ℓ-mer of read r. Also, if $\mathcal{C}_{i+1}(r)$ does not exist, then the first informative ℓ-mer of the consistent set $\mathcal{C}_{i+1}(r)$ becomes the last ℓ-mer of read r. To manage complexity of method, the extension of ℓ-mers are performed in an intelligent way: An ℓ-mer jumps and no extension occurs, if all ℓ bases within the reference at the corresponding location are covered by some extended ℓ-mers before. Extension is terminated if two corresponding two ℓ-mer meet each other within the reference genome. Choosing proper distance and threshold for mapping and extending ℓ-mers, respectively, are necessary.

For the second problem, we propose the same extension method but in other direction. In this way, the first and the second informative ℓ-mer of the consistent set $\mathcal{C}_i(r)$ is extended toward each other. Details of our procedure are presented in Method section.

The classical spliced read methods for detecting SVs fail to resolve copy number variations (CNVs). Reads comes from these variation are located within repeat regions, but we can handle CNVs by considering an interval of D bases (except $\approx \ell$ bases) at the extension step. The value of D can be 10,000.

In the following, we present the analysis of the proposed method for different types of SVs. We consider i.i.d. reference genome and two noiseless read and noisy read cases.

For an i.i.d. genome, we consider each base of the reference genome of length G is generated from $\{A, C, G, T\}$ alphabet with equal probability and independently. Also, the target genome is the same as the reference genome except that some SVs are added to it. Additionally, we assume that N reads of length L are extracted randomly from the target genome and for noisy reads, errors are added to reads with an i.i.d. model of

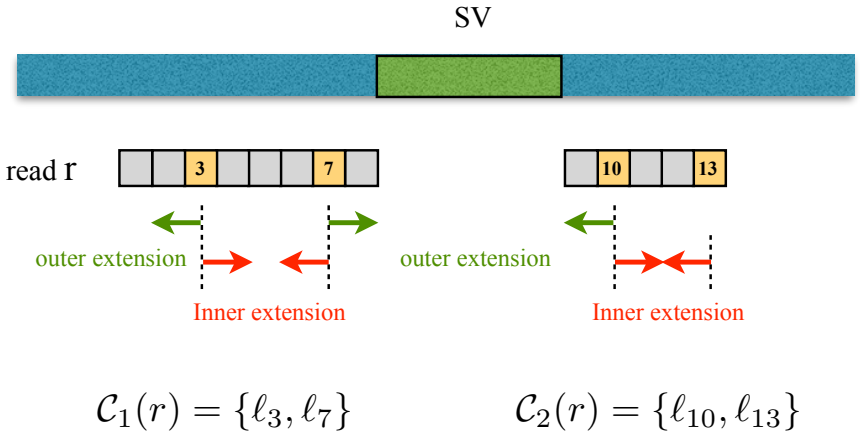

Figure 8: Extension of ℓ-mers. The anchored reads r has 13 disjoint ℓ-mers and two consistent sets $\mathcal{C}_1(r), \mathcal{C}_2(r)$. Directions of inner and outer extensions are shown in this figure.

rate ϵ. Start points of reads are assumed to come from a *Poisson* process with arrival rate of $\lambda = N/G$. Reads are fragmented to $K = L/\ell$ disjoint ℓ-mers. Assume that ℓ is large enough such that ℓ-mers do not include structural variated bases can be mapped uniquely to the reference genome (i.e. $\ell \approx \log G$ [27]). We classify SVs into two classes: 1) *Balanced* SV: If the SV does not change the length of the genome, e.g. inversion, and 2) *Unbalanced* SV: If the SV changes the length of the genome, e.g. insertion, deletion and copy number variation.

[Updating paragraph - contact for more details]

$$\mathbb{E}\{\mathcal{N}_S\} = (L - 2\ell)\,\lambda, \tag{7}$$

where $\mathcal{N}_S$ is the

[Updating paragraph - contact for more details]

$$\mathbb{E}\{\mathcal{N}_S\} = (L - \ell)\,\lambda,$$

because, small SVs hide at the anchoring step and expose at the extension step.

Now, consider noisy reads with error rate of ϵ. [Updating paragraph - contact for more details]

$$\mathbb{P}\{\mathcal{T}\} = \sum_{i \le d_{\max}} \binom{\ell}{i} \epsilon^i (1-\epsilon)^{\ell - i}. \tag{8}$$

We suppose that the desired set ℓ and d_{max} [Updating paragraph - contact for more details]

$$\mathbb{P}\{\mathcal{F}\} \le G \times P_w(\ell) \triangleq G \sum_{i \le d_{\max}} \binom{\ell}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{\ell - i} = G 4^{-\ell} \sum_{i \le d_{\max}} \binom{\ell}{i} 3^i, \tag{9}$$

where $P_w(\ell)$ represents the [Updating paragraph - contact for more details]

For $N = \Omega(G)$, if [Updating paragraph - contact for more details]

$$\mathbb{E}\left\{\mathcal{N}_S\right\} = \sum_{k=2}^{\lfloor \frac{L}{\ell} \rfloor - 1} \lambda\ell \sum_{\alpha=2}^{k} p_a \left(1-(1-p_a)^{\alpha-1}\right)(1-p_a)^{k-\alpha} p_{ext}^{k-\alpha}$$
$$= \lambda\ell \times p_a \sum_{k=2}^{\lfloor \frac{L}{\ell} \rfloor - 1} \left(\frac{1-\left((1-p_a)\,p_{ext}\right)^{k-1}}{1-(1-p_a)\,p_{ext}} - \frac{(1-p_a)^{k-1} - \left((1-p_a)\,p_{ext}\right)^{k-1}}{1-p_{ext}} \right). \tag{10}$$

A balanced SV S is studied in three cases: [Updating paragraph - contact for more details]

The first case in [Updating paragraph - contact for more details]

$$\mathbb{E}\left\{\mathcal{N}_S\right\} = \sum_{k=1}^{\lfloor \frac{L}{\ell} \rfloor - 1} \lambda\ell \sum_{\alpha=1}^{k} p_a \left(1-(1-p_a)^{\alpha-1} + (1-p_a)^{\alpha-1}\left(1-(1-p_a)^{\max\{0,\frac{L-L_S}{\ell}-k\}}\right)\right)(1-p_a)^{k-\alpha} p_{ext}^{k-\alpha}$$
$$= \lambda\ell \times p_a \sum_{k=2}^{\lfloor \frac{L}{\ell} \rfloor - 1} \left(\frac{1-\left((1-p_a)\,p_{ext}\right)^{k-1}}{1-(1-p_a)\,p_{ext}} - \frac{(1-p_a)^{k-1} - \left((1-p_a)\,p_{ext}\right)^{k-1}}{1-p_{ext}} \right)$$
$$+ \lambda\ell \times p_a \sum_{k=1}^{\lfloor \frac{L-L_S}{\ell} \rfloor - 1} \lambda\ell \times p_a\,(1-p_a)^{k-1} \sum_{\alpha=1}^{k} \left(1-(1-p_a)^{\frac{L-L_S}{\ell}-k}\right) p_{ext}^{k-\alpha}$$
$$= \lambda\ell \times p_a \sum_{k=2}^{\lfloor \frac{L}{\ell} \rfloor - 1} \left(\frac{1-\left((1-p_a)\,p_{ext}\right)^{k-1}}{1-(1-p_a)\,p_{ext}} - \frac{(1-p_a)^{k-1} - \left((1-p_a)\,p_{ext}\right)^{k-1}}{1-p_{ext}} \right)$$
$$+ \lambda\ell \times p_a \sum_{k=1}^{\lfloor \frac{L-L_S}{\ell} \rfloor - 1} \lambda\ell \times p_a \left((1-p_a)^{k-1} - (1-p_a)^{\frac{L-L_S}{\ell}-1}\right)\left(\frac{1-p_{ext}^{k}}{1-p_{ext}}\right). \tag{11}$$

4.2.4 Extension and informative l-mer

[updating] figure 9 is related.

4.3 Stage two in detail

4.3.1 Weights analysis

[Updating:] Statistical analysis - expected weights of the nodes and edges - figure 9

4.3.2 Detection method

There are two approaches for this step discussed in summery in section 4.1. The detailed version will be available on the official preprint.

5 Results

[updating - Results on Speed, Recall and Accuracy] Speed ! Recall ! Accuracy !

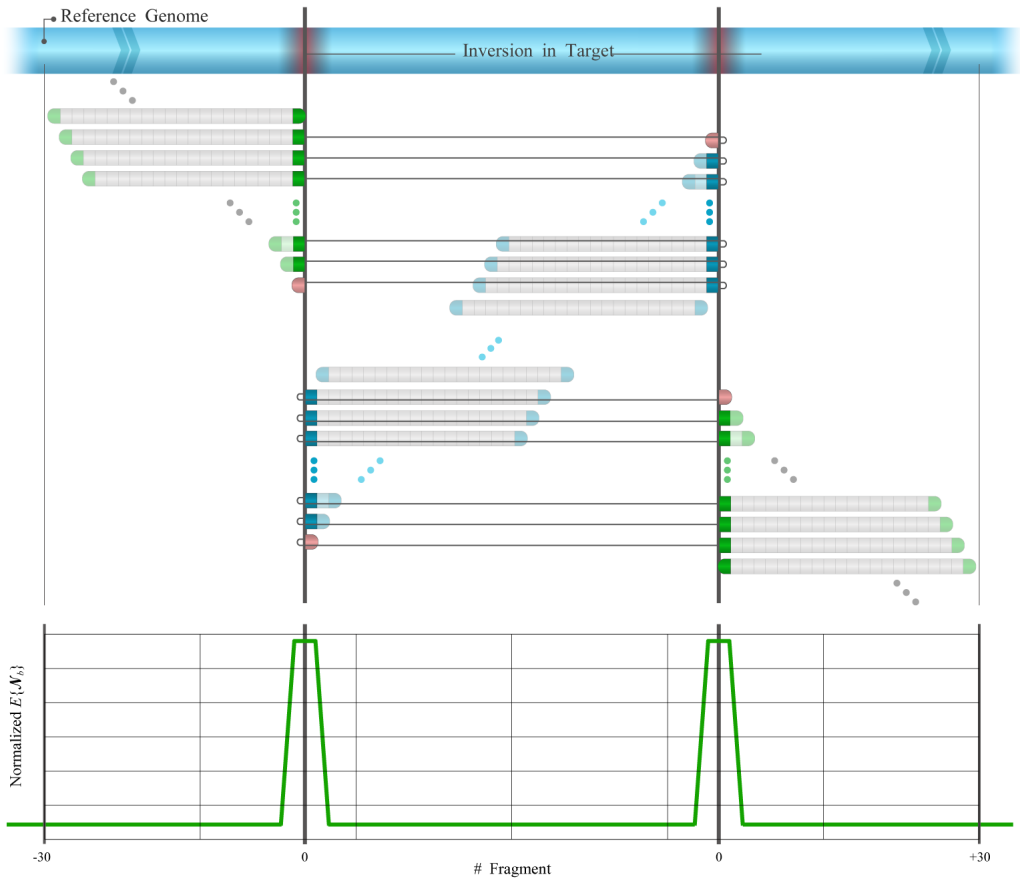

Figure 9: Informative chunks collaborate to signal for SVs - the expected weight for nodes in the sparse graph - Informative read chunks are visible around the breakpoints (green and blue with 100% opacity)

5.1 Mapping statistics for Chr19

5.2 Synthetic data

5.3 Real data

References

[1] C. M. Carvalho and J. R. Lupski. Mechanisms underlying structural variant formation in genomic disorders. *Nature Reviews Genetics*, 17(4):224–238, 2016.

[2] A. J. Iafrate, L. Feuk, M. L. Rivera, M. N.and Listewnik, P. K. Donahoe, Y. Qi, ..., and C. Lee. Detection of large-scale variation in the human genome. *Nature Genetics*, 36(9):949, 2004.

[3] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, ..., and N. Navin. Large-scale copy number polymorphism in the human genome. *Science*, 305(5683):525–528, 2004.

[4] D. F. Conrad, D. Pinto, R. Redon, L. Feuk, O. Gokcumen, Y. Zhang, ..., and T. Fitzgerald. Origins and functional impact of copy number variation in the human genome. *Nature*, 464(7289):704, 2010.

[5] C. Alkan, B. P. Coe, and E. E. Eichler. Genome structural variation discovery and genotyping. *Nature Reviews Genetics*, 12(5):363, 2011.

[6] D. F. Conrad and M. E. Hurles. The population genetics of structural variation. *Nature genetics*, 39:s30–s36, 2007.

[7] A. Sekar, A. R. Bialas, H. de Rivera, A. Davis, T. R. Hammond, N. Kamitaki, ..., and G. Genovese. Schizophrenia risk from complex variation of complement component 4. *Nature*, 530(7589):177–183, 2016.

[8] C. R. Marshall, A. Noor, J. B. Vincent, A. C. Lionel, L. Feuk, J. Skaug, ..., and B. Thiruvahindrapduram. Structural variation of chromosomes in autism spectrum disorder. *The American Journal of Human Genetics*, 82(2):477–488, 2008.

[9] R. K. Yuen, B. Thiruvahindrapuram, D. Merico, S. Walker, K. Tammimies, N. Hoang, ..., and M. J. Gazzellone. Whole-genome sequencing of quartet families with autism spectrum disorder. *Nature medicine*, 21(2):185–191, 2015.

[10] D. A. King, W. D. Jones, Y. J. Crow, A. F. Dominiczak, N. A. Foster, T. R. Gaunt, ..., and E. A. Jones. Mosaic structural variation in children with developmental disorders. *Human molecular genetics*, 24(10):2733–2745, 2015.

[11] L. M. Boettger, R. M. Salem, R. E. Handsaker, G. M. Peloso, S. Kathiresan, J. N. Hirschhorn, and S. A. McCarroll. Recurring exon deletions in the hp (haptoglobin) gene contribute to lower blood cholesterol levels. *Nature genetics*, -(-):–, 2016.

[12] M. Zanda, S. Onengut-Gumuscu, N. Walker, C. Shtir, D. Gallo, C. Wallace, ..., and S. S. Rich. A genome-wide assessment of the role of untagged copy number variants in type 1 diabetes. *PLoS Genet*, 10(5):e1004367, 2014.

[13] J. M. Tubio. Somatic structural variation and cancer. *Briefings in functional genomics*, -(-):elv016, 2015.

[14] E. Papaemmanuil, I. Rapado, Y. Li, N. E. Potter, D. C. Wedge, J. Tubio, ..., and I. Martincorena. Rag-mediated recombination is the predominant driver of oncogenic rearrangement in etv6-runx1 acute lymphoblastic leukemia. *Nature genetics*, 46(2):116–125, 2014.

[15] N. Waddell, M. Pajic, A. M. Patch, D. K. Chang, K. S. Kassahn, Bailey P., ..., and M. C. Quinn. Whole genomes redefine the mutational landscape of pancreatic cancer. *Nature*, 518(7540):495–501, 2015.

[16] E. R. Gamazon and B. E. Stranger. The impact of human copy number variation on gene expression. *Briefings in functional genomics*, 14(5):352–357, 2015.

[17] K. A. Frazer, S. S. Murray, N. J. Schork, and E. J. Topol. Human genetic variation and its contribution to complex traits. *Nature Reviews Genetics*, 10(4):241–251, 2009.

[18] A. Abyzov, S. Li, D. R. Kim, M. Mohiyuddin, A. M. Sttz, N. F. Parrish, ..., and J. O. Korbel. Analysis of deletion breakpoints from 1,092 humans reveals details of mutation mechanisms. *Nature communications*, 6, 2015.

[19] H. H. Kazazian. Mobile elements: drivers of genome evolution. *science*, 303(5664):1626–1632, 2004.

[20] P. H. Sudmant, T. Rausch, E. J. Gardner, R. E. Handsaker, A. Abyzov, J. Huddleston, ..., and M. K. Konkel. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75, 2015.

[21] J. Huddleston, M. J. Chaisson, K. M. Steinberg, W. Warren, K. Hoekzema, D. Gordon, ..., and P Peluso. Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome research*, 27(5):677–685, 2017.

[22] M. Puig, S. Casillas, S. Villatoro, and M. Cceres. Human inversions and their functional consequences. *Briefings in functional genomics*, 14(5):369–379, 2015.

[23] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, ..., and D. B. . Jaffe. Characterizing and measuring bias in sequence data. *Genome biology*, 14(5):R51, 2013.

[24] H. Lee and M. C. Schatz. Genomic dark matter: the reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics*, 28(16):2097–2105, 2012.

[25] C. Alkan, S. Sajjadian, and E. E. Eichler. Limitations of next-generation genome sequence assembly. *Nature methods*, 8(1):61–65, 2011.

[26] D. Nashta-ali, A. Aliyari, A. A. Moghadam, M. A. Edrisi, S. A. Motahari, and B. H. Khalaj. Meta-aligner: long-read alignment based on genome statistics. *BMC bioinformatics*, 18(1):126, 2017.

[27] A. S. Motahari, G. Bresler, and N. C. David. Information theory of dna shotgun sequencing. *IEEE Transactions on Information Theory*, 59(10):6273–6289, 2013.

[Updating]
UGALA: Ultrafast Genome Alignment with Limited Anchoring

A. Afshinfard, D. Nashta-ali, S. A. Motahari, H.R. Rabiee

March 18, 2018

Abstract

In this study, we explore the problem of pairwise and multiple whole-genome alignments. Firstly, we propose a novel framework by which one can quickly align whole genomes by optimized number of anchors. For doing so, we introduce two PAC bounds. Based on genome statistics and an expected rearrangement resolution, the PAC bounds calculate an acceptable distance between two short-length anchors. Considering the maximum acceptable distance, an algorithm searches for anchors between the input genomes and aligns them very quickly. It can find 85% of the homologous regions between an E.Coli and an E.Albertii by searching only less than 3% of the first genome in the second. This is performed 105x faster than running a simple BLAST between them. Running a low-cost extension step increases resolved regions into 94%. Secondly, we propose a direction to extend this anchoring criterion to multiple alignments which despite current multiple methods, is more sensitive than pairwise alignments. [See the results section for more info]

[updating - please contact a.afshinfard@gmail.com for slides and online presentation - the content of this paper is not reviewed and corrected for academic writing and grammatical mistakes yet - we are updating the results and figures]

1 Introduction

2 Current Approaches

There are generally two approaches for whole-genome alignment. The first approach starts by finding all the similarities between the two genomes. The next stage is to filter wrong similarities and to build larger syntenic regions by merging collinear segments. [a figure here]. The other approach tries to find anchors between the input genomes. Anchors are similar substrings between the genomes which are more likely to originate from an identical ancestral sequence or i.e. they are confirmed similarities.

3 Proposed method

3.1 The idea

[updating] Two questions to consider. First: do we need all the anchor between the input genomes? second: Could we consider disjoint similarities with some specific distance as anchors? answers to these two questions lead us to a new framework for Synteny search and Whole-genome alignment.

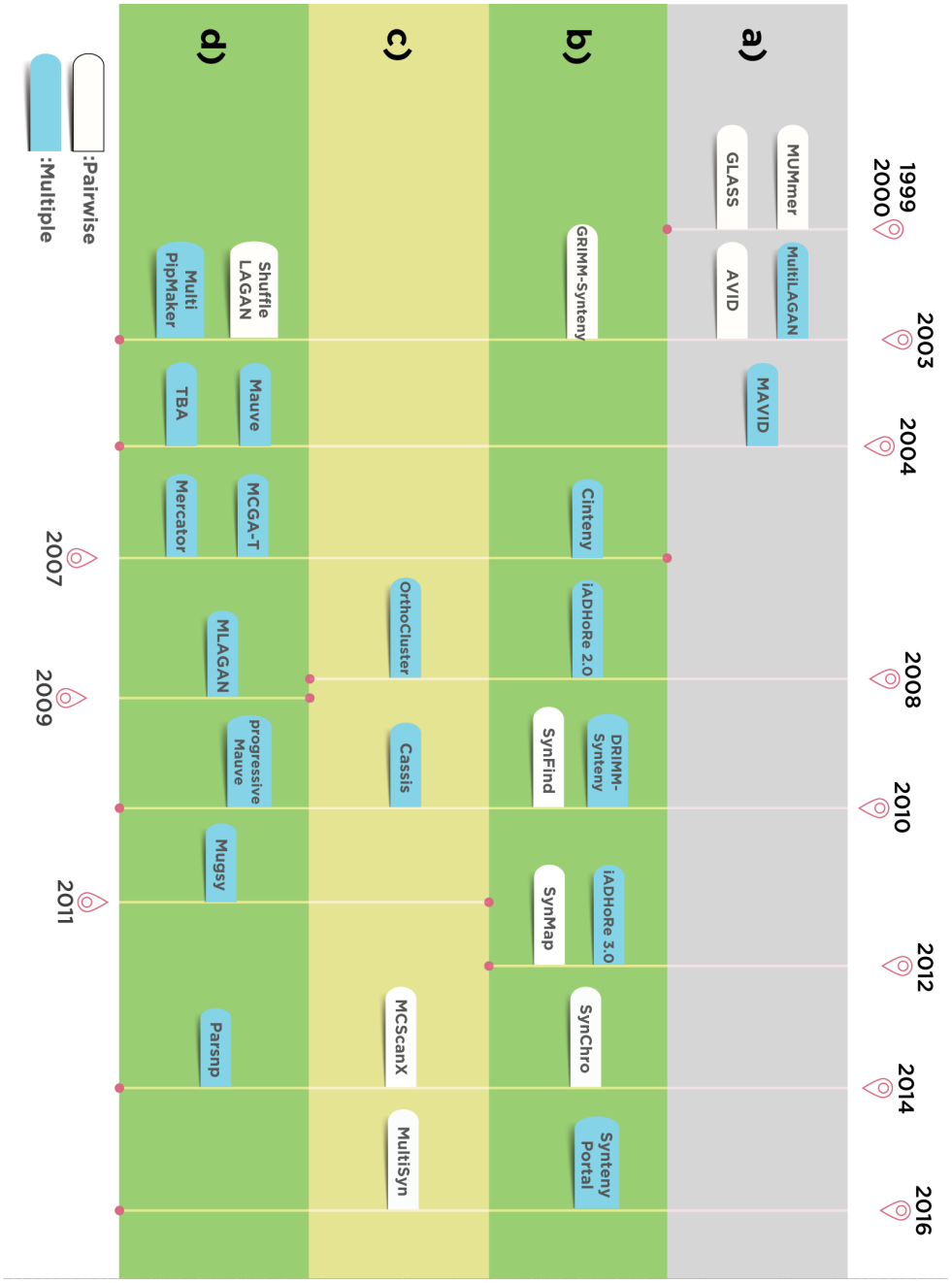

Figure 1: current approaches, [updating]

3.1.1 Concepts

3.1.2 PAC bounds

first PAC bound

Among small-scale mutations, substitutions do not change the length and only indels can alter the length of a sequence. Insertion increases the length and deletion reduces it

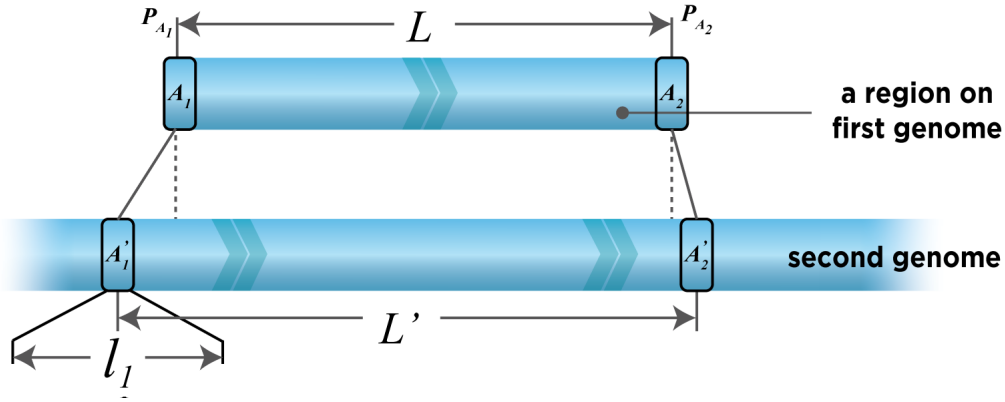

Figure 2: idea for PAC bound 1, [updating]

obviously. We can model their contribution to lengths with a joint distribution (e.g mixing two independent distribution) with some parameters λ_i for insertion and λ_d for deletion. The actual function will be released in the official preprint but for now it is trivial to show that for equal λ_i and λ_d the expected alteration in length is zero yet with some deviation in distribution around zero.

Assume a function f mentioned in equation 1 which receives the parameters and additionally a confidence parameter δ $(0 < \delta < 1)$ as input, and outputs a constant T_{indel}.

$$T_{indel} = f(\lambda_i,\ \lambda_d,\ \delta) \tag{1}$$

Function f calculates T_{indel} the way so it meets equation 2. i.e. in equation 2, ΔL is the final output.

$$\mathbb{P}[\ \Delta L \leqslant f(\ \lambda_i,\ \lambda_d,\ \delta) \times |L|\ ]\ \geqslant\ \delta \quad , \quad dir(A'_1) = dir(A'_2) \tag{2}$$

or equally:

$$\mathbb{P}[\ \frac{\Delta L}{|L|} \leqslant T_{indel}\ ]\ \geqslant\ \delta \quad , \quad dir(A'_1) = dir(A'_2) \tag{3}$$

For instance, $\delta = 0.95$ means 0.95 of the area under the distribution function is around zero with distance less than ΔL. Roughly, it means that in 0.95 of the cases, the alteration in length caused by indels in a substring of length L is less than ΔL. With this in mind, if an alteration is more than ΔL, it is a candidate for large-scale events (rearrangements).

next step: Fixing the bound in order to distinguish between small-scale mutations and large-scale events.

with minimum size of R_{max}, using a distance ($|L|$)

$$|L| \leqslant \frac{R_{max}}{T_{indel}} = L_{max} \tag{4}$$

and blah blah blah

Table 1: Bacteria Genomes considered

Genome	Length	GC content	non-coding Length	non-coding GC content
E.Albertii	4746590	49.34	953518	42.83
E.Coli	4686137	50.78	790246	45.05

$$\begin{cases} |L| \leqslant \frac{R_{max}}{T_{indel}} = L_{max} \\ \frac{\Delta L}{|L|} \leqslant T_{indel} \end{cases} , \quad dir(A'_1) = dir(A'_2) \tag{5}$$

Second PAC bound
Alignment probability

$$\mathbb{P}(A_{l_1}) = \sum_{i<d_{max}} \binom{l_1}{i} \upsilon^i (1-\upsilon)^{l_1-i} = p_a \tag{6}$$

where υ is blah

3.2 Local homology search

this is the algorithm:

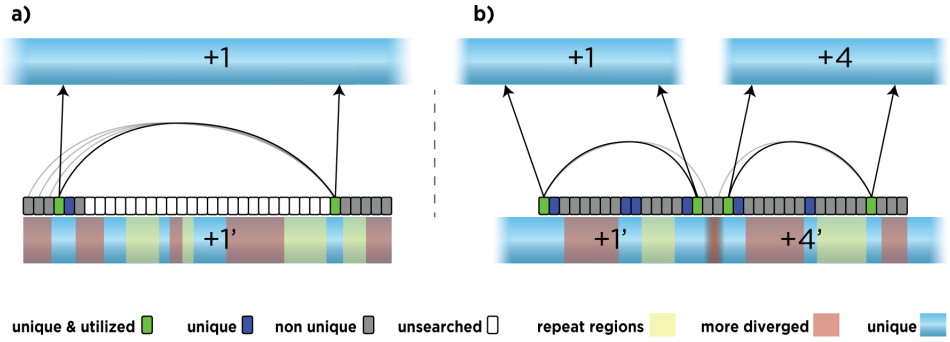

Figure 3: syteny search. a) inside a syntenic region b) in the boundary of two syntenic regions, [updating]

3.3 Whole-genome alignment

4 Results

Starting from figure 5 and table 1, the results are provided. Please check the tables and figures for now.[updating]

Table 2: Eukaryotic Genomes considered

Genome	Chromosome	Considered regions	Length	non-coding GC content
Homo Sapiens	X	Coding Regions	1325745	50.33
Mus musculus			1329361	48.15

Table 3: Simulated algorithm steps by $P_a = 0.5$ and $L_{max} = 4\ kb$

Step	Adjacent fragments	Resolved %	Searched %
1	2	25	0.63
2	4	51	1.09
3	8	81.8	
4	16	98.1	2.20
5	32	>99.5	2.30

Table 4: Simulated algorithm steps by $P_a = 0.2$ and $L_{max} = 4\ kb$

Steps	Adjacent fragments	Resolved %	Searched %
1	2	4	0.63
2	4	11.3	1.23
3	8	28.2	2.34
4	16	59.18	4.14
5	32	89.74	6.18
6	64	>99	7.21

Table 5: self-mapping of 25 bp long fragments with bowtie

Frag Length	-v 0		-v 1		-v 2		-v 3	
	unique	multiple	unique	multiple	unique	multiple	unique	multiple
25	92.3	7.8	91.7	8.3	91.3	8.7	90.4	9.6

Table 6: Mapping results for different lengths and different bowtie parameters - E.coli in E.Albertii

Frag Length	Frag count	-v 1		-v 2		-v 3	
		unique	multi	unique	multi	unique	multi
15	311404	32.7	38.8	0.6	98.3	0	99
20	233553	32.6	4.3	46.4	8	39.5	38.2
25	186842	26.4	3.7	40.2	4.5	51.3	5.5
30	155702	21.6	3.4	34	4.1	45.3	4.7
35	133458	17.9	3.2	28.9	3.8	39.7	4.3
40	116776	15.1	3.1	24.9	3.5	34.7	4
45	103801	12.8	2.9	21.6	3.3	30.4	3.7
50	93421	11.7	2.8	18.7	3.2	26.8	3.5

Algorithm 1 LoSy: Local Synteny search

function ANCHORING(F, K)
  ▷ Where F: chunks of fragment, K: number of chunks
  $loc[1] = uniqueSearch(F[1])$
  $loc[K] = uniqueSearch(F[K])$
  if $isConsistent(loc[1], loc[K], 1, K)$ **then**
    return $makeAnchor(loc[1], loc[K], 1, K)$
  end if
  for ($i = 2$ to $K/2$) **do**
    $loc[i] = uniqueSearch(F[i])$
    for ($j = K$ to $K - i + 2$) **do**
      if $isConsistent(loc[i], loc[j], i, j)$ **then**
        return $makeAnchor(loc[i], loc[j], i, j)$
      end if
    end for
    $loc[K - i] = uniqueSearch(F[K - i])$
    for ($j = 1$ to i) **do**
      if $isConsistent(loc[K - i], loc[j], K - i, j)$ **then**
        return $makeAnchor(loc[1], loc[K], 1, K)$
      end if
    end for
  end for
  return $recursivelyCheck(F, 1, K)$
  ▷ to search for anchors in intervals [1,K/2] and [K/2,K]
end function

Algorithm 2 pairwise genome alignment and synteny finding

function PAIRWISESYNTENYDETECTION($G_1, G_2, L_{max}, k_{orig}, l_1$)
  ▷ Where G_1, G_2: Input Genomes, L_{max}: Checking Distance,
  ▷ k_{orig}: Sufficient Steps, l_1: chunk length
  $Steps = \underset{i}{\operatorname{argmin}}[\ \log_2 k_{orig} < i + 1\ ]$
  $DG_1 = divider(G_1, l_1)$ ▷ Divide genome into chunks of length l_1
  for ($i = 1$ to $Steps$) **do**
    $uniqueSearch(L_{max}, i)$
    $checkUniques(L_{max}, i)$
    $checkConfidents(L_{max})$
  end for
  $extendUnsolveds(L_{max})$
  return $recursivelyCheck(F, 1, K)$
  ▷ to search for anchors in intervals [1,K/2] and [K/2,K]
end function

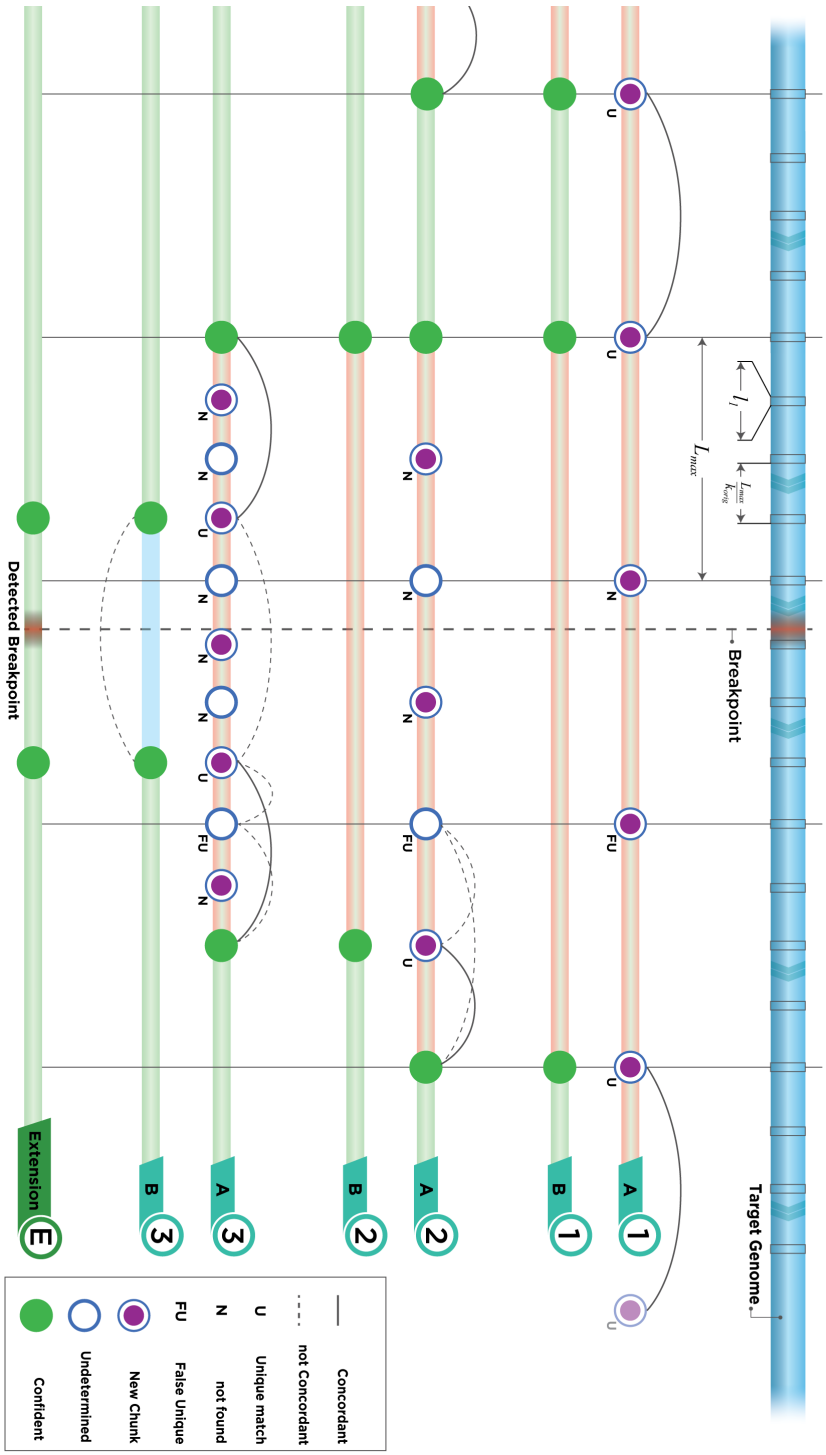

Figure 4: 3 example steps of the algorithm for Genome Alignment followed by an extension step. [updating]

Table 7: Algorithm speed compared with Blast runs. As highlighted by blue color, the algorithm is 105x faster than a simple Blast (until the 7th step and using -v 1 bowtie parameeters with 85% resolved regions)

Resolution	-v	step 5	step 7	step 9
25 kb	2	240	109	60
	3	81	36	16.7
5 kb	0	423	150	63
	1	260	105	41
	2	52	22	9.6
	3	13.8	6	2.7
1 kb	2	11.4	4.4	1.5
	3	3.2	1.2	0.4

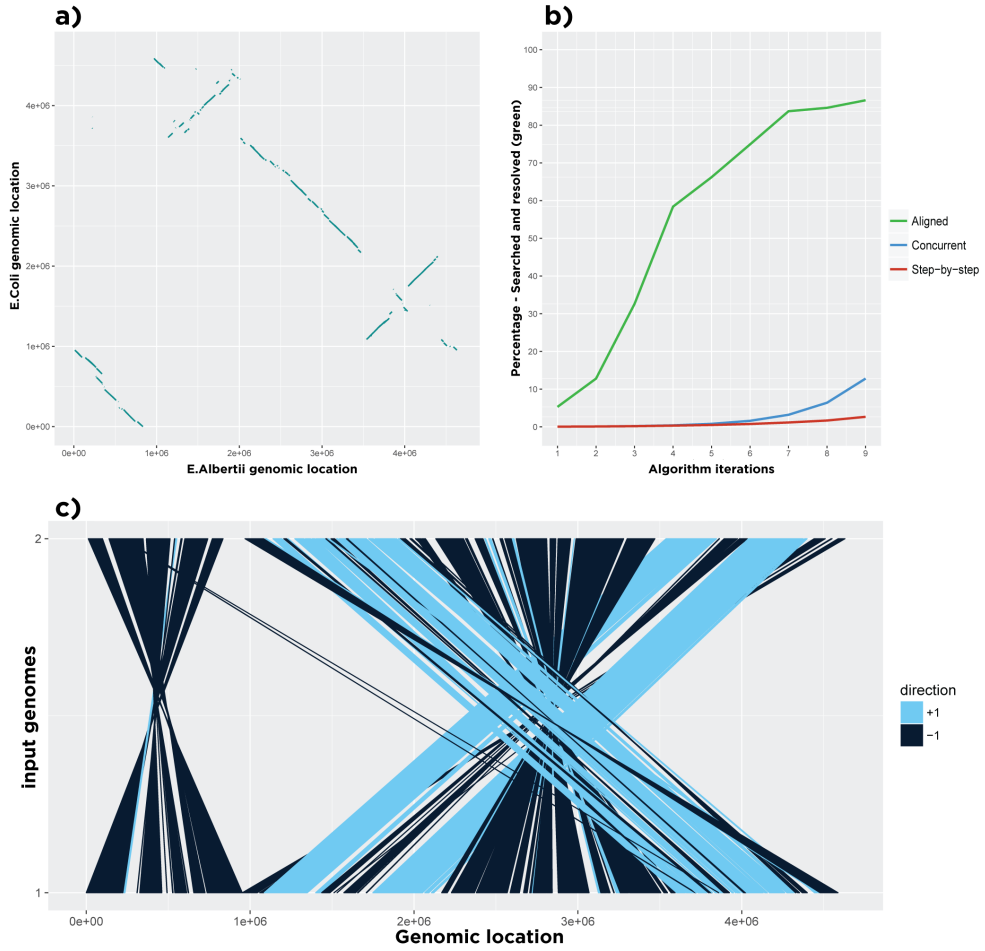

Figure 5: E.coli and E.Albertii, Resolution: 5kb. a) dot-matrix alignment of the two genomes. b) algorithm step-by-step. (Red: % searched regions. - Blue: % searched regions if the algorithm start from that step - Green: % resolved regions). E.g in the 7th step the algorithm finds ¿ 85% of the regions as homologous by searcing less tan 3% of the first genome in the second one. [updating]

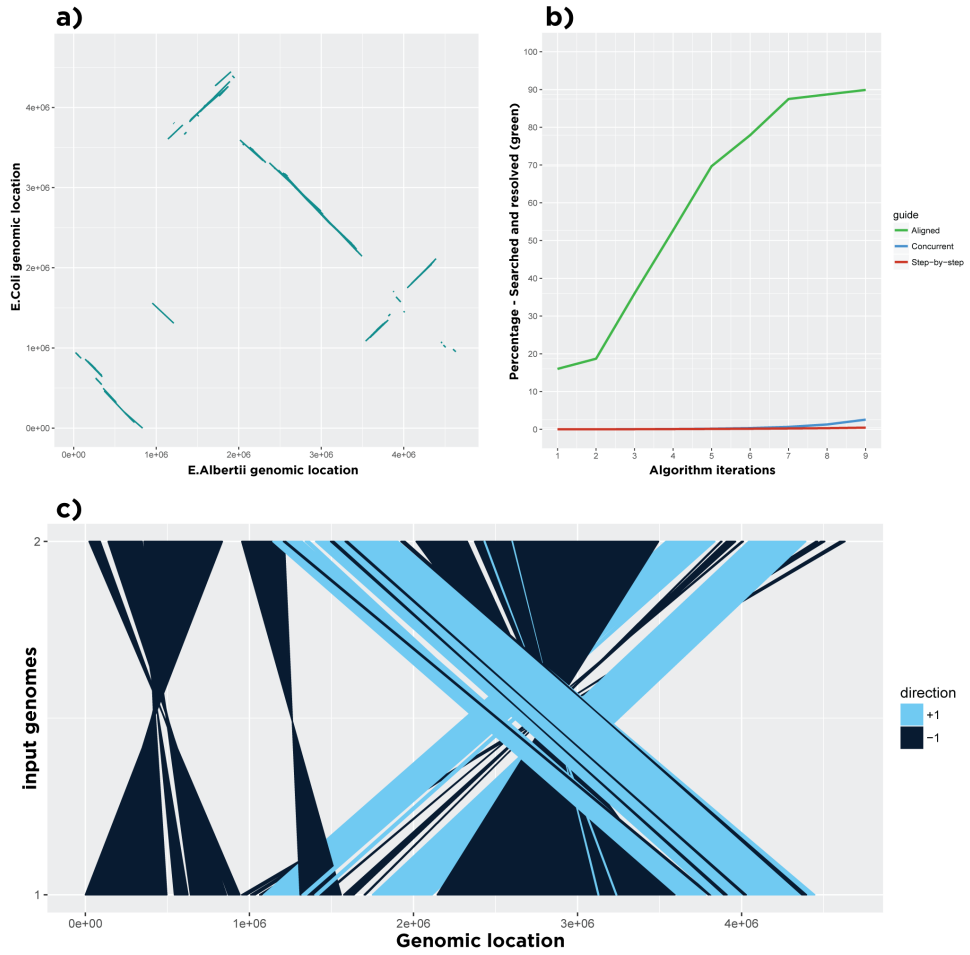

Figure 6: E.coli and E.Albertii, Resolution: 25kb. a) dot-matrix alignment of the two genomes. b) algorithm step-by-step. (Red: % searched regions. - Blue: % searched regions if the algorithm start from that step - Green: % resolved regions). E.g in the 9th step the algorithm finds 90% of the regions as homologous by searcing less tan 1% of the first genome in the second one. [updating]

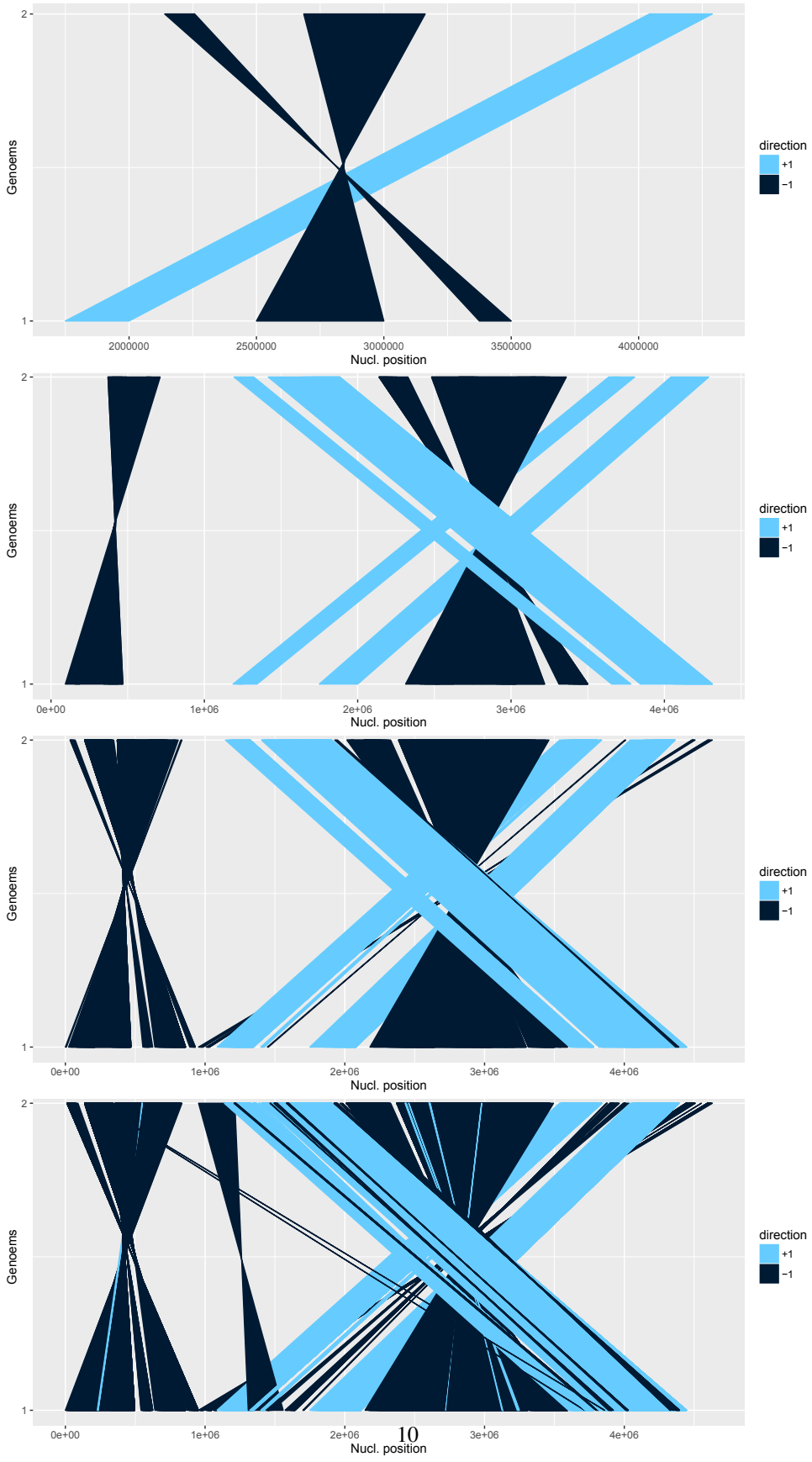

Figure 7: Steps of the algorithm for E.Coli vs. E.Albertii with resolution: 25kb [updating]

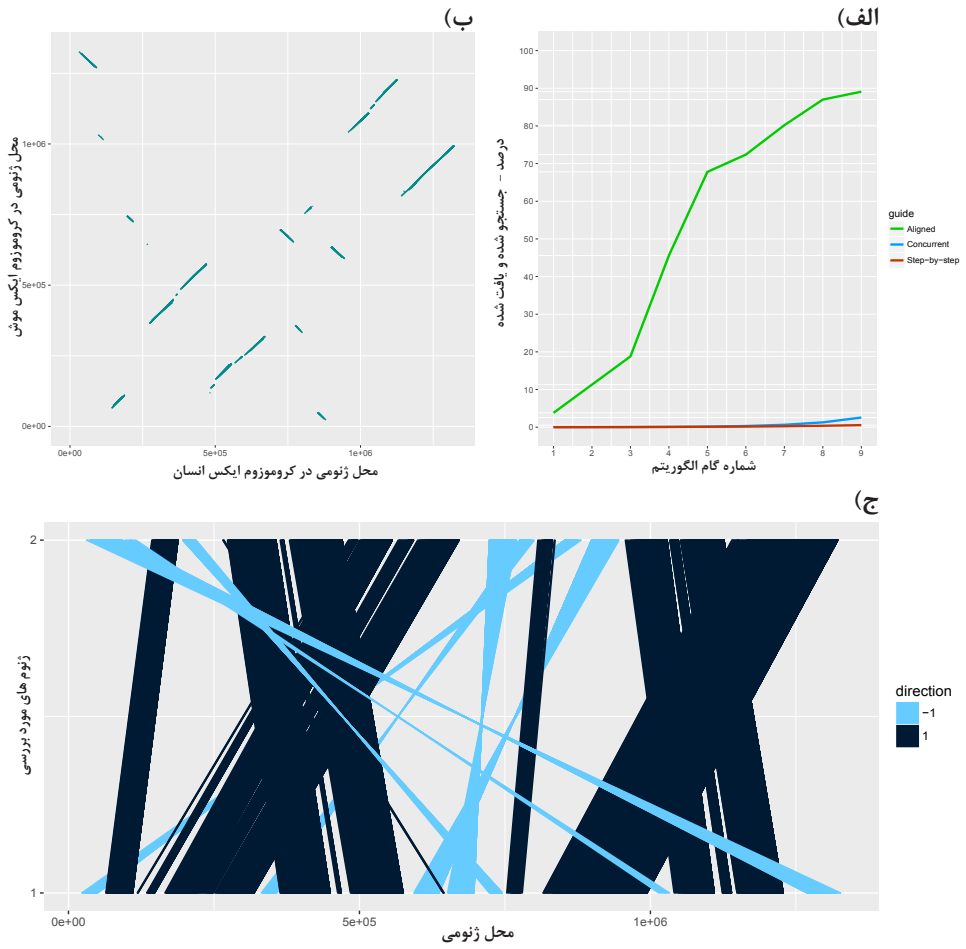

Figure 8: X chromosome, Human (1) vs. Mouse (2)., [updating]

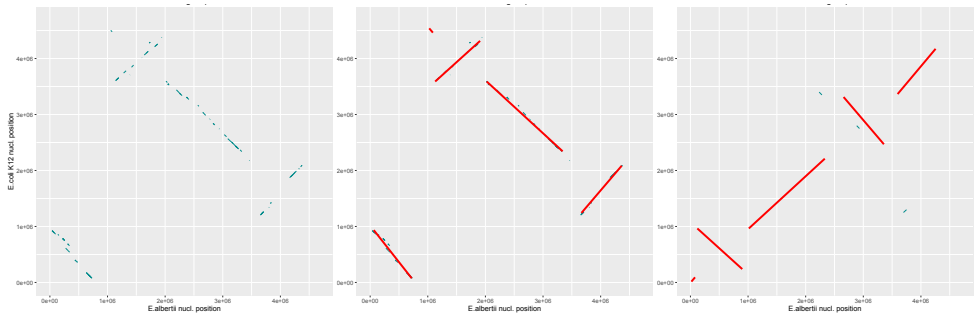

Figure 9: Unifying the circular genomes of E.Coli and E.Albertii, [updating]