SVseq: an approach for detecting exact breakpoints of deletions with low-coverage sequence data

Aleksander Sudakov

BI journal club 12.12.2011

BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 23 2011, pages 3228–3234
doi:10.1093/bioinformatics/btr563

Genome analysis

Advance Access publication October 12, 2011

SVseq: an approach for detecting exact breakpoints of deletions with low-coverage sequence data

Jin Zhang* and Yufeng Wu

Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA

Associate Editor: Alex Bateman

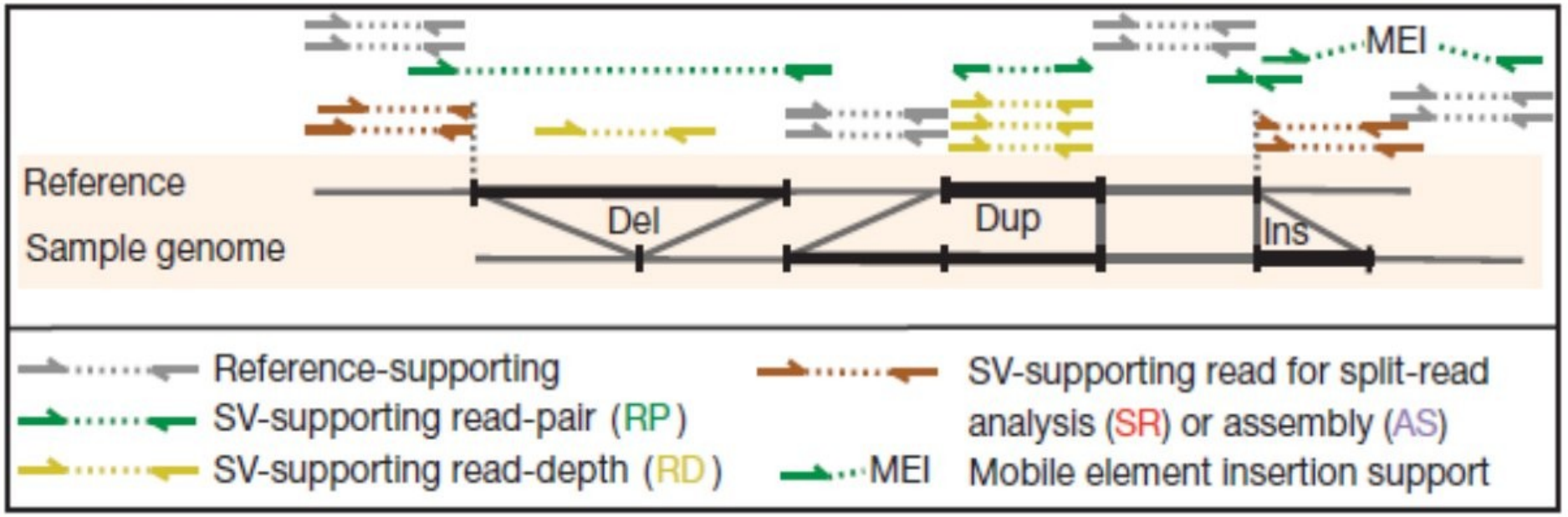
Zhang J , Wu Y Bioinformatics 2011;27:3228-3234

4 ALGORITHMS, 19 METHODS

- 6 methods using Read-Pair (RP)
- 4 methods using Read-Depth (RD)
- 4 methods using Split-Read (SR)
- 3 methods using local Sequence Assembly (AS)
- 2 methods using combination of RP and RD (PD)

Color-coding:

AS RP
SR RL Release set
PD RD

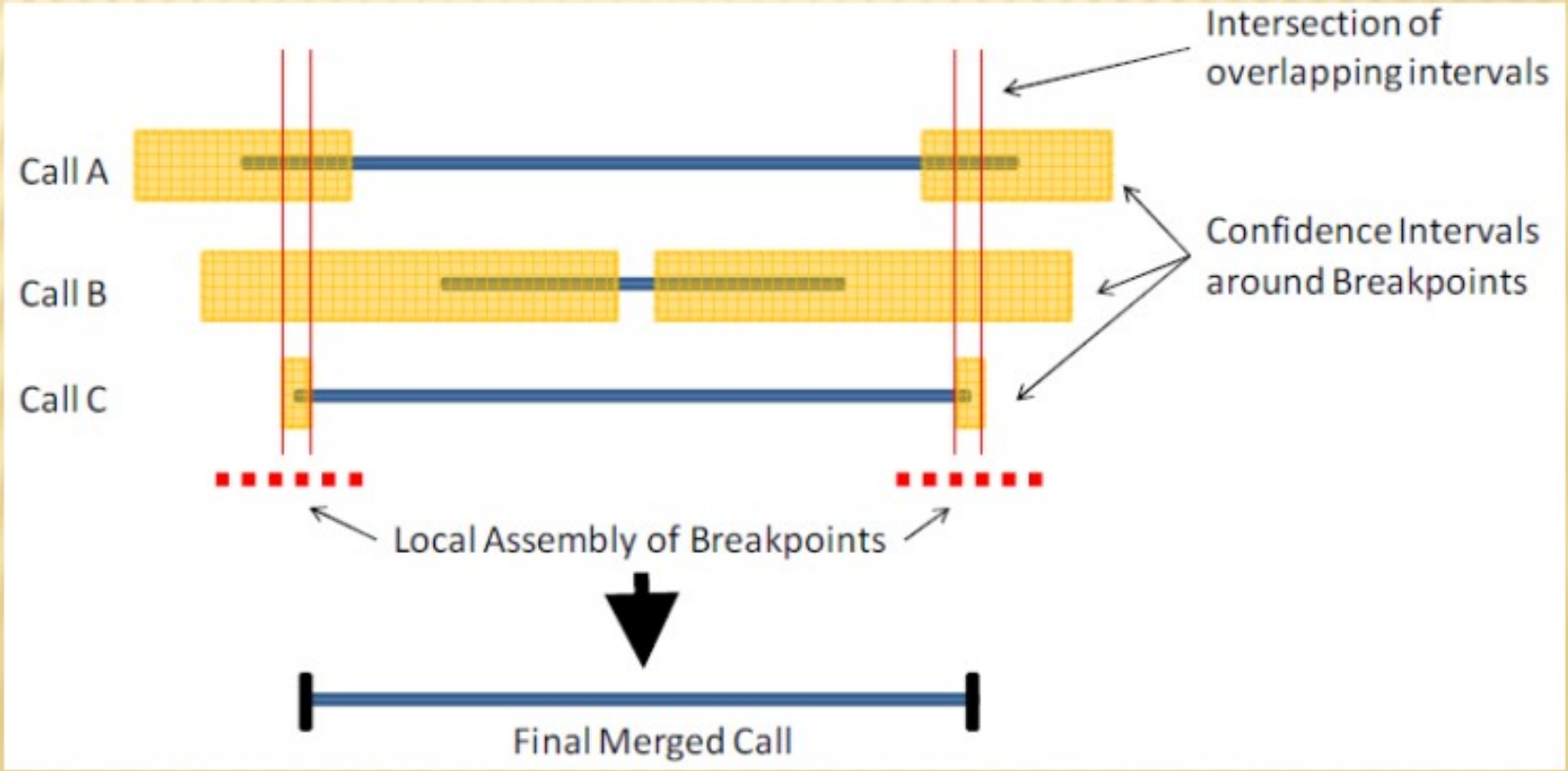


SVseq key features

- Finds deletions using low-coverage short sequence reads
- Exact breakpoints in resolution of 1 bp
- This facilitates the analysis of the origin and functional impact of the deletions
- Uses BWT read mapping for optimal speed and memory usage

MAPPING OF BREAKPOINTS:

- Sequence data allows mapping of breakpoints in single nucleotide precision. This was done for ca 15000 SVs.



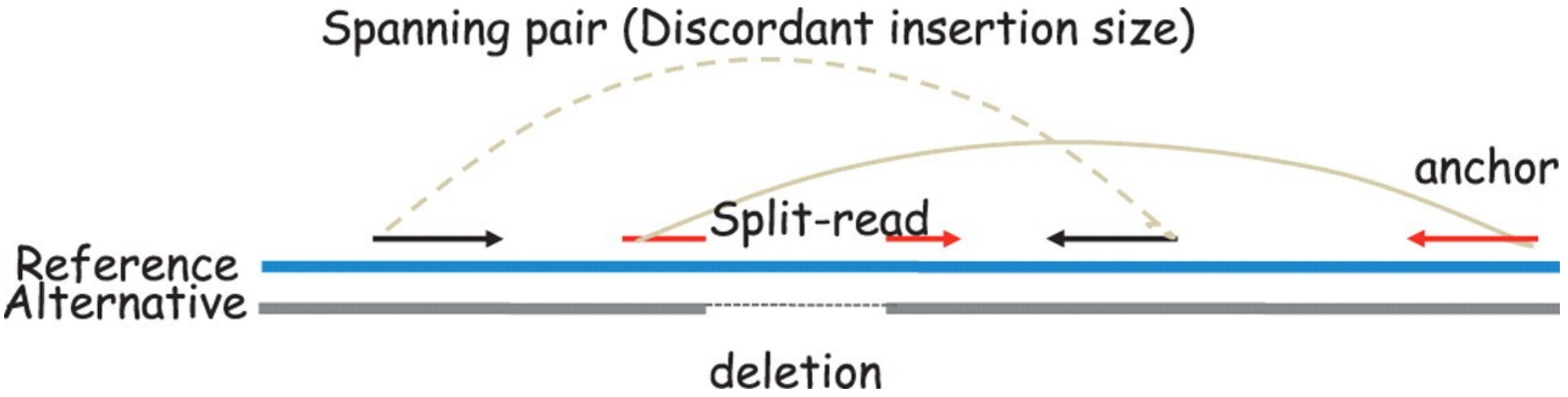
Other software

- Pindel (Ye et al., 2009)

  is currently one of the best performing methods for finding exact breakpoints

  v2 has added the ability of mapping split reads with mismatches and small indels. It is slower than Pindel v1 in finding larger deletions, and it also has a higher default cutoff value as 3.
- Dindel
- MoGUL (Mixture of Genotypes Variant Locator)
- ...

Split read



Our SVseq approach maps a **split read** with an anchor. The direction of the anchor of this read is on the reverse strand, so that the split read is on the forward strand. The two parts of the previously unmapped read are mapped on two different positions, which may indicate that there is a deletion. The **discordant distance** of a spanning paired-end read supports the deletion, since its abnormally large insert size can be explained by the presence of the deletion.

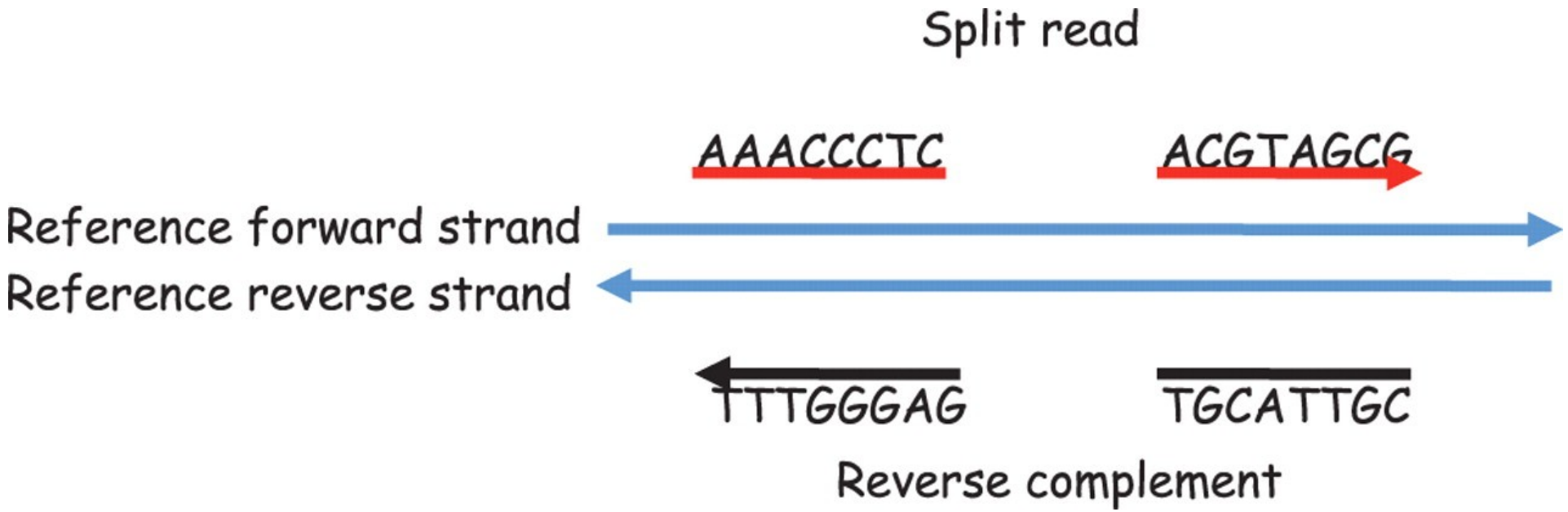
Algorithm

- BWA implements handling of mismatches and indels to BWT algorithm.
- SVseq is based on BWA but allows to map both portions of a split read
- Since BWT of a sequence has a direction, mapping of end of the read has to be done on BWT of reverse of reference

Algorithm

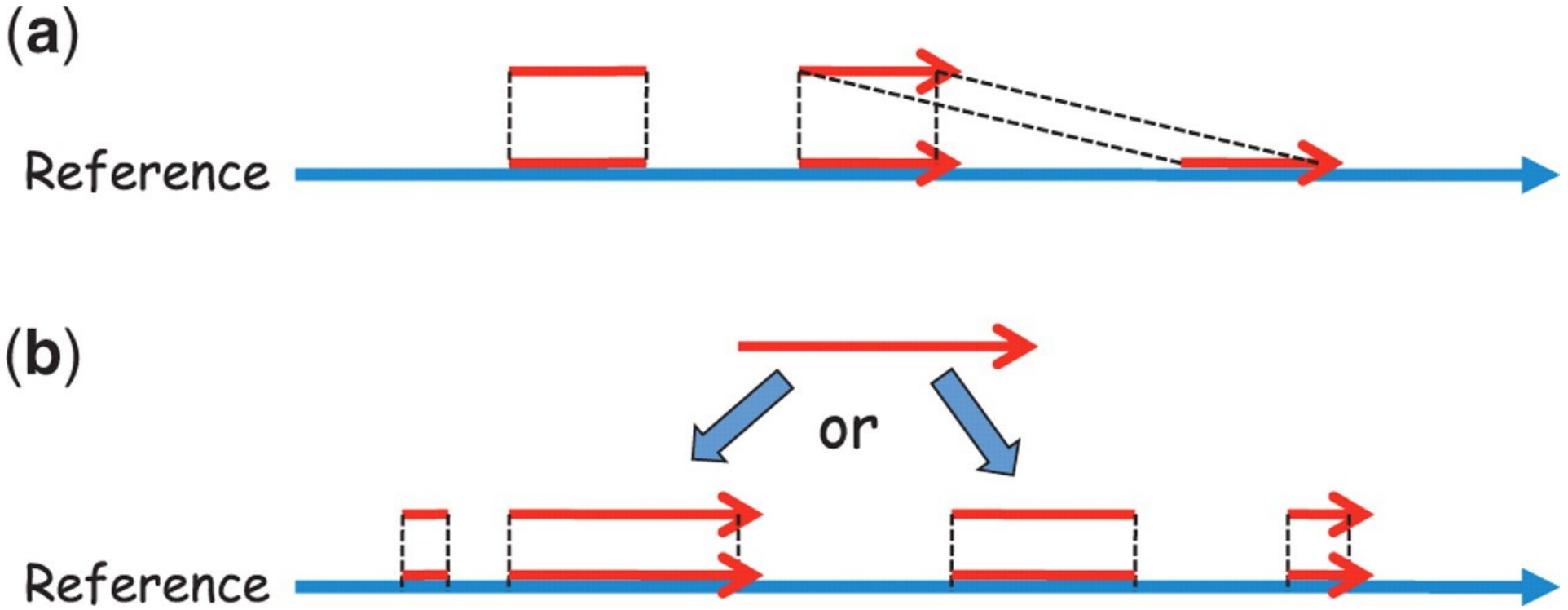
- A naive way for this purpose is to break the read into two portions, take each portion as a new read and run a reads mapping algorithm
  $2(n - 2m - 1)$ times, where m is the minimum allowed length of a portion.
- A faster approach is to store the mappings of the portions as the algorithm proceeds. From one end of a read, after the portion of length l is mapped and the mappings are stored, we then proceed to map the portion of l + 1 bps (BWT)
- After the read is mapped from both directions, if the two portions of the read meet each other (or the length of the two portions sums up to the length of the whole read) and are mapped with the least errors (mismatches and indels), the coordinates of the portions on the reference are computed

BWT needs separate mapping on reverse strand



Split reads mapping using BWT. Suppose the read in red color is from forward strand. Mapping it on the BWT of the forward strand reversely starts from GC to AA. Mapping from the other end is the same as mapping the reverse complement of the read on the other strand of BWT. So instead mapping from AA to CG on the forward strand, we map the reverse complement on the reverse strand from TT to GC.

Problems



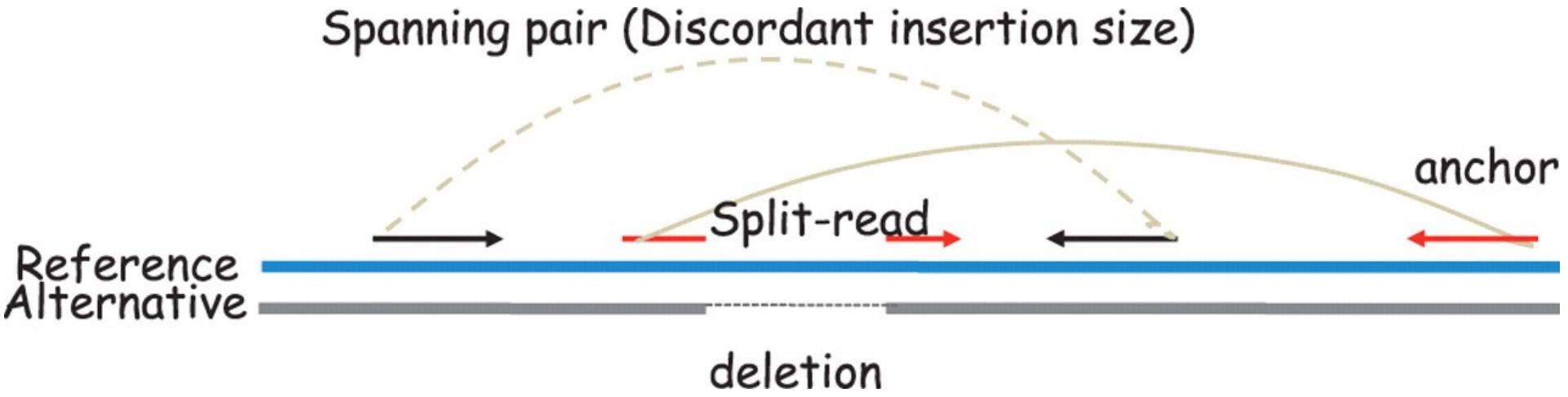
Due to repeats and errors, a read can be mapped at more than one pair of positions or the reads can have more than one split breakpoints
Method only keeps read mappings with the highest quality

Leftmost deletion breakpoints. Due to the occurrence of sequence ‘ACG’ at the two breakpoints of the deletion, if a split read from the alternative genome crosses the breakpoints of the deletion, splitting the read is not fully determined. Here, we only give the leftmost and rightmost way to break the read into two parts. SVseq uses the leftmost breakpoints to represent a deletion.

Calling deletions

- Candidate deletions found in first stage contain false positives
- Anchor reads: paired-end reads spanning candidate deletions (two ends maped to different sides of the candidate)
- Insert size of such deletion will be discordant (insert size + length of the deletion within 3 standard deviations of library insert size)
- When at least one anchor paired-end read is supporting the candidate we report a deletion

Split read



Our SVseq approach maps a **split read** with an anchor. The direction of the anchor of this read is on the reverse strand, so that the split read is on the forward strand. The two parts of the previously unmapped read are mapped on two different positions, which may indicate that there is a deletion. The **discordant distance** of a spanning paired-end read supports the deletion, since its abnormally large insert size can be explained by the presence of the deletion.

Simulation

- simulated sequence reads based on human chromosome 15, 100 338 915 bp
- true deletions from 1000 genomes project
- only deletions with exact breakpoints are used
- only deletions for 45 individuals in CEU population are used
- 132 such deletions (127 of which with length 8092 bp or less)

Simulation

- wgsim (https://github.com/lh3/wgsim) - generate paired-end reads
- SNP and small indels on each genomes are simulated using default parameters
- read length 50 bp – pair length 200 coverage 1.6×, 3.2× and 4.8×
- read length 100 bp – pair length 500 coverage 3.2×, 4.2× and 6.4×
- base error rate 2%

Workflow

- BWA is used to map simulated paired-end reads
- pairs are picked out as input with one end uniquely mapped as a whole read but the other end not mapped
- at 4.8× coverage with read length 50:
  ~ 216 million pairs of reads are generated
- ~ 208 million pairs are mapped in the right order on chromosome 15. These pairs are used to test discordant insert sizes

Comparison of SVseq and Pindel

Data	X	M	Findings	TP	M	Findings	TP
50	1.6×	SVseq	74	74	SVseq	72	72
		P v1	57	56	P v2	54	53
	3.2×	SVseq	95	94	SVseq	92	91
		P v1	76	74	P v2	68	66
	4.8×	SVseq	102	100	SVseq	100	96
		P v1	83	81	P v2	81	78
100	3.2×	SVseq	111	108	SVseq	108	105
		P v1	63	62	P v2	84	83
	4.2×	SVseq	117	109	SVseq	114	106
		P v1	69	68	P v2	87	86
	6.4×	SVseq	128	120	SVseq	124	116
		P v1	85	84	P v2	104	101

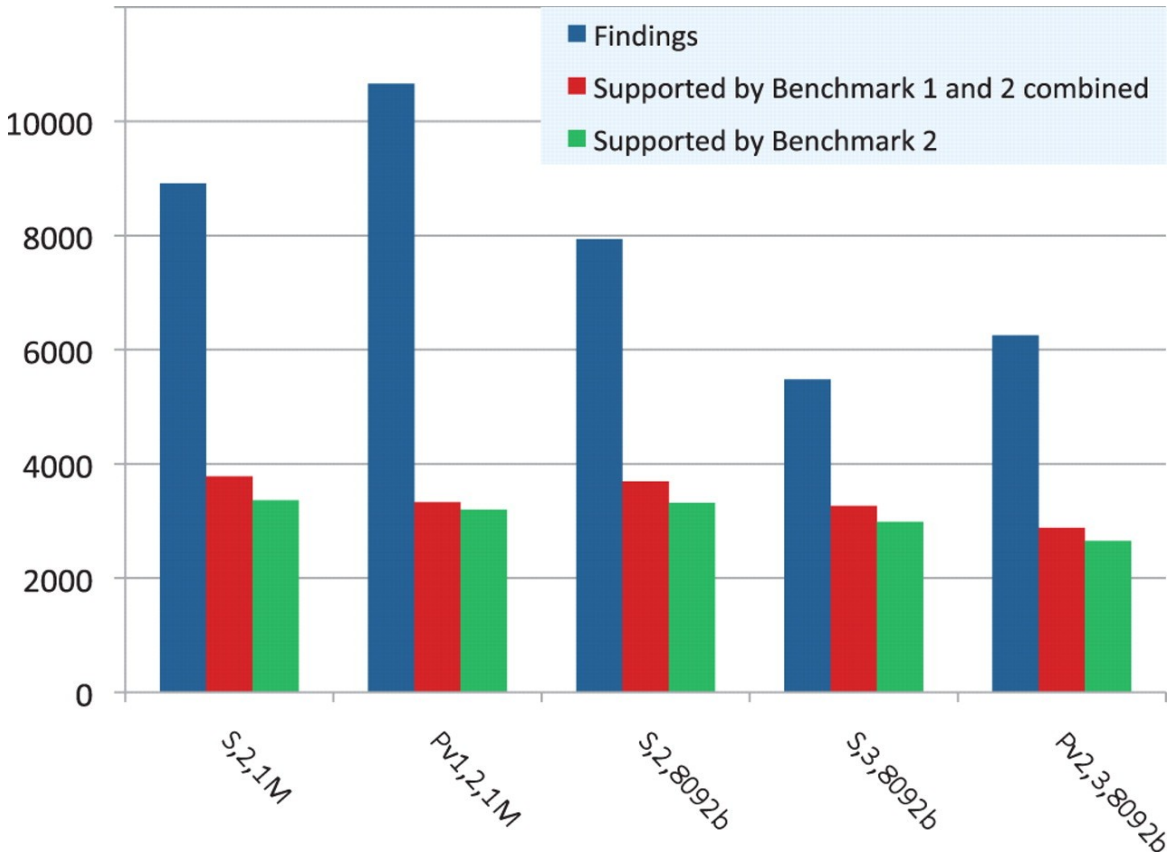
simulated reads of lengths 50 and 100 on chromosome 15 with 132 deletions, where 127 of them are less than 8092 bp. The maximum size of deletion events is 1 Mbps when comparing with Pindel v1 and 8092 bp when comparing with Pindel v2. The cutoff value is 2 for Pindel v1, and 3 for Pindel v2. SVseq uses cutoff value 2 for the data with read length 50, and 3 for the data with read length 100.
X - Coverage, M - Method, TP - True Positive, P v1 - Pindel v1, P v2 - Pindel v2.

Real data

- SVseq is tested using the 1000 genomes project pilot 1 low-coverage data (45 individuals) and pilot 2 high-coverage data (1 individual)
- results are compared with those of Pindel and the releases of the 1000 genomes project (called SVs in Mills et al. (2011))
- validated and assembled deletions are used as benchmarks
- Most methods in this table do not call deletions with exact breakpoints (estimated confidence intervals)

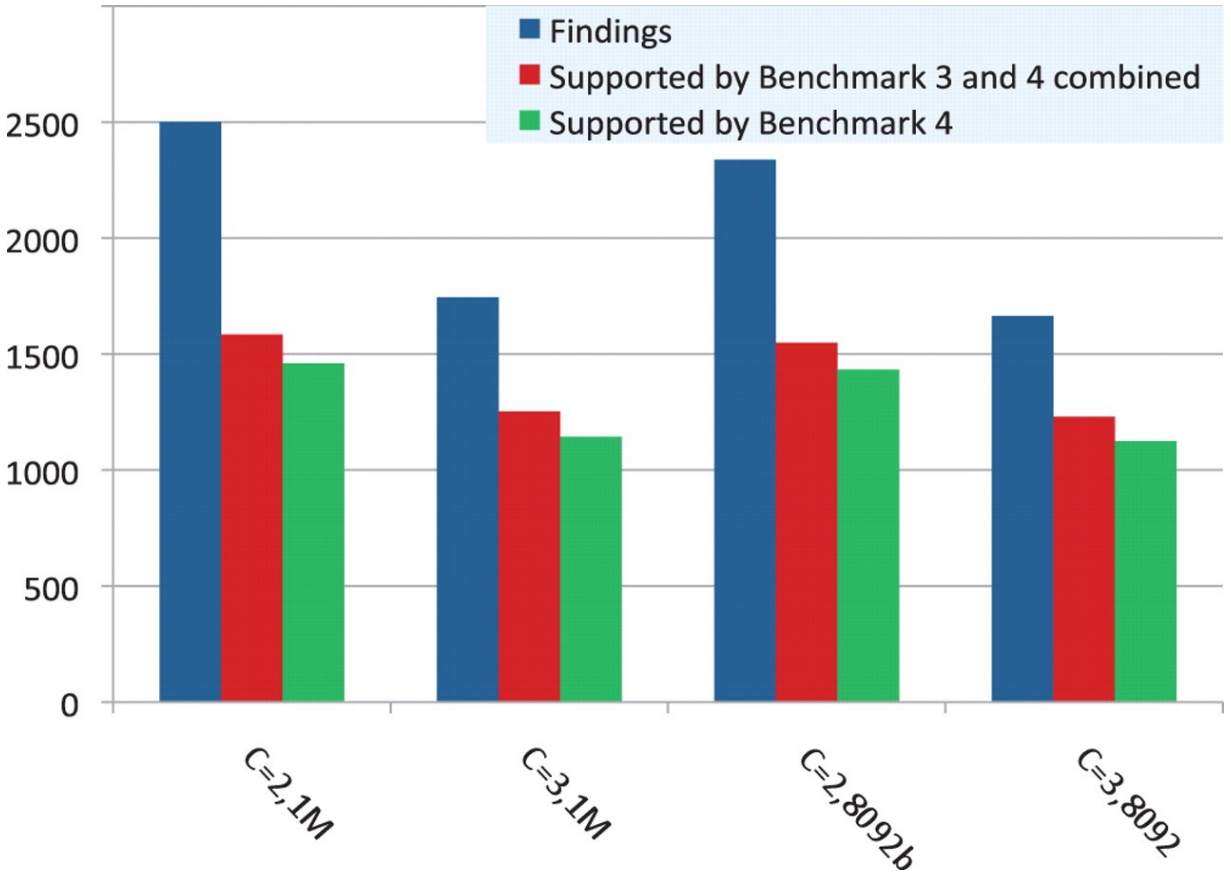
Benchmark

- Benchmark 1: validated deletions, low-coverage
- Benchmark 2: assembled deletions with exact breakpoint, low-coverage
- Benchmark 3: validated deletions, high-coverage
- Benchmark 4: assembled deletions with exact breakpoint, hihg-coverage



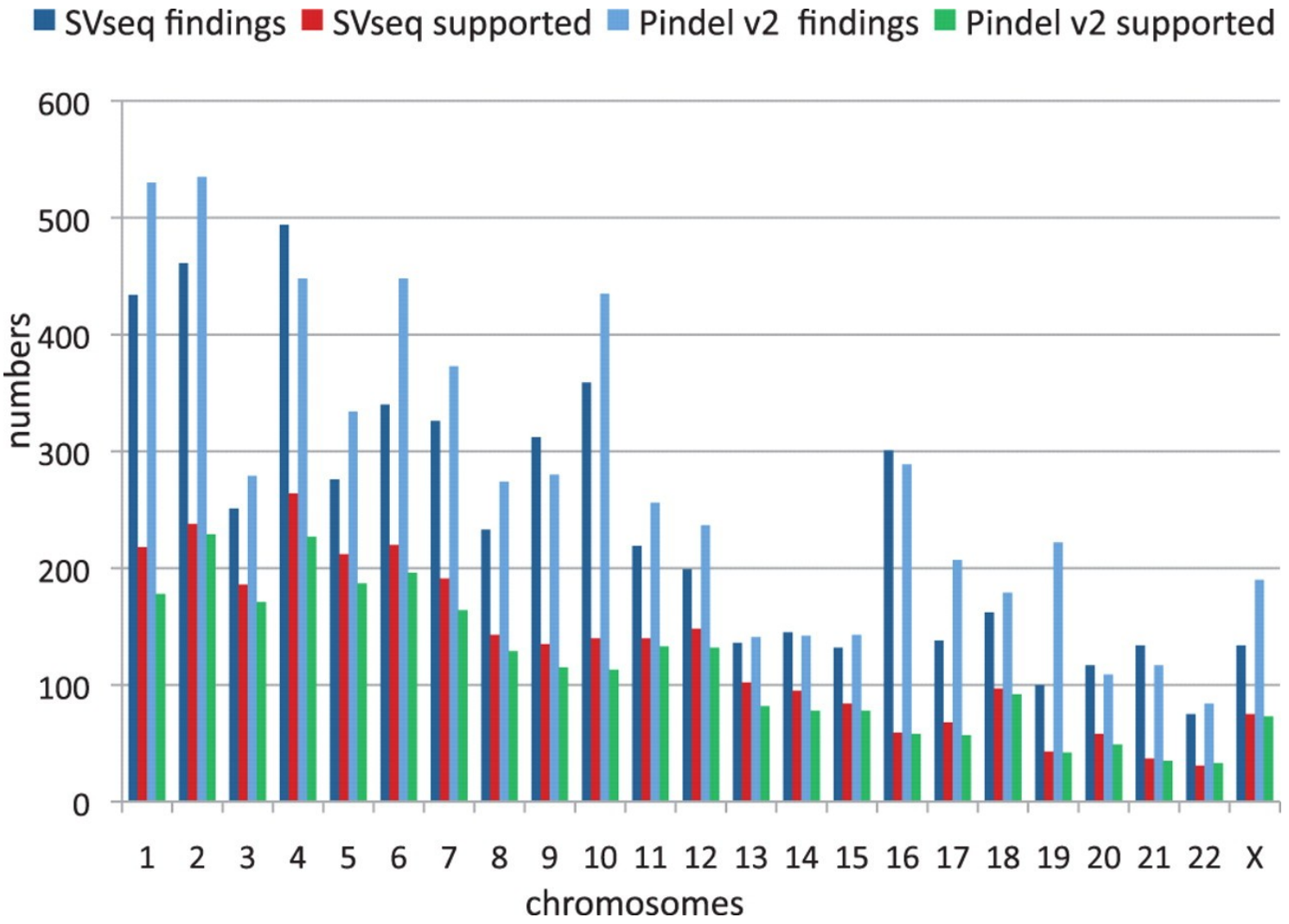
Benchmarks 1 and 2 combined: SVseq finds ~14% more deletions than Pindel v1
Benchmark 2: SVseq finds 5% more deletions than Pindel

Numbers of deletions found by SVseq, Pindel v1 and v2 using different parameters and numbers of deletions that are supported by the benchmarks are plotted in columns. S stands for SVseq. Pv1 stands for Pindel v1 and Pv2 stands for Pindel v2. Cutoff values are 2 or 3. Maximum event sizes are 1 Mbs or 8092 bp.

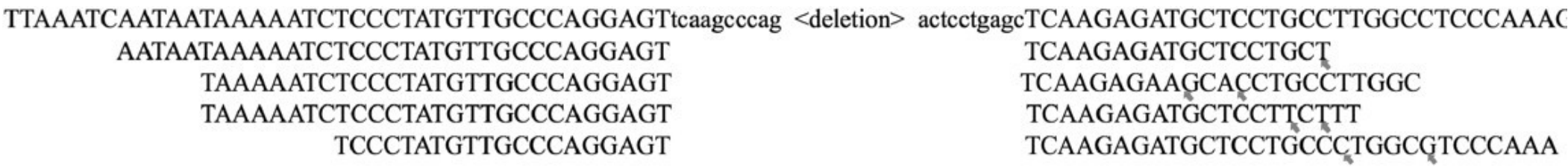


Using cutoff value 2 and maximum event size 1 Mb, SVseq finds 2500 deletions and 1585 (~63%) of them are supported by the validated deletions

Results of using high-coverage sequence data from the 1000 genomes project pilot 2 trio data. Numbers of deletions of individual NA12878 found by SVseq using different parameters and number of deletions that are supported by the benchmarks are plotted in columns. The cutoff value are 2 or 3 and the maximum even sizes are 1 Mb or 8092 bp.



Chromosome view of comparison of SVseq (cutoff 3) and Pindel v2 in finding deletions up to 8092 bp with low-coverage data using Benchmark 2. The horizontal axis is the name of chromosomes, and the vertical axis is the numbers of deletions.



Deletion P1_M_061510_1_922 is found by SVseq and Pindel v2, but not Pindel v1. Note that all four mapped split reads have one or more sequencing errors (pointed by arrows).

Running time

Comparing with Pindel v1, SVseq is generally faster. Pindel v1 does not allow inexact mapping while our method allows inexact mapping (i.e. SVseq considers a larger search space). Nonetheless our method is about **three times faster** than Pindel v1 for processing the same amount of sequence reads.

- Pindel v2 allows mismatches and runs with multi-threads. But Pindel v2 is still slow even with multi-threads. When using Pindel v2 to find SVs on chromosome 1 and setting parameter of Maximum Event Size Index to 9 (corresponding to 2 071 552 bp), we run it with 20 threads on our server that has 24 cores. Pindel v2 did not finish after more than 30 h on the server. When set the parameter to 8 (corresponding to 517 888 bp) with 20 threads, Pindel v2 runs 8 h on our server (Note there is time spent trying to find other SVs). SVseq handles the same amount of data in about 3.5 h with **one thread** on the same machine and finds deletions up to 1 Mbps.
- Note that efficiency is important since the sequence data size is very large and is growing rapidly.

Results

- Compared to Pindel SVseq improves number of deletions found, accuracy and running time in discovering larger deletions using low-coverage short sequence data
- Higher accuracy is due to the combination of split reads mapping and discordant insert size analysis
- SVseq tolerates errors in mapping and allows mismatches and small indels
- most deletion finding methods for low-coverage data do not give exact breakpoints