Combining probabilistic alignments with read pair information improves accuracy of split-alignments

Naruki Yoshikawa[1], Anish Shrestha[2] and Kiyoshi Asai[2, 3]

[1] Department of Bioinformatics and Systems Biology, Faculty of Science, The University of Tokyo, Japan
[2] Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Japan
[3] Artificial Intelligence Research Center, AIST, Tokyo, Japan

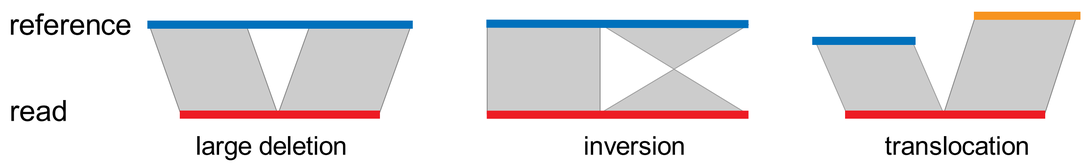
Introduction

What is split-alignment?

A split-alignment is a pairwise sequence alignment in which different parts of the query align to disjoint regions in the reference.

Why is it important?

They provide direct evidence of rearrangements such as large deletions, inversions, chromosomal translocations, etc.



What is difficult?

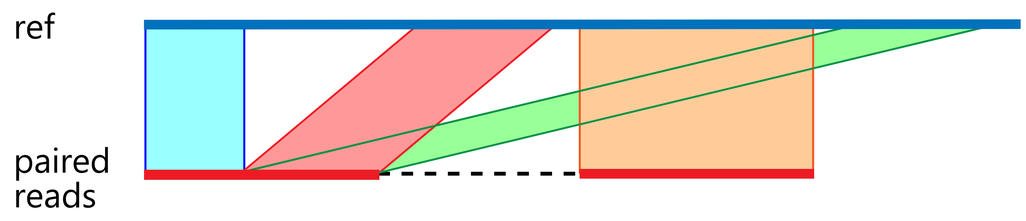
- Split-alignments cannot be found using conventional alignment methods (e.g. Smith-Waterman).
- In practice, they are computed by "stitching" parts of local alignments together.
- However, there might be many false high-scoring local alignments because:
  - reference is repetitive, reads are error-prone
  - rearrangements tend to occur in repeat-rich regions
- This leads to ambiguities in identifying the correct split-alignment.

Method

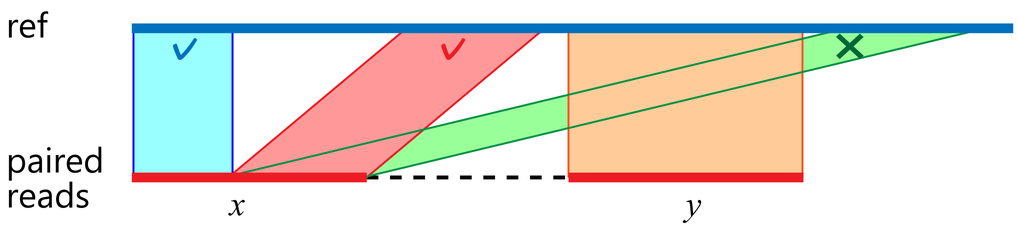
Incorporate paired-end reads information to combat split-alignment ambiguity

We combine probabilistic sequence alignment with Bayesian probability updating procedure to find accurate split-alignments.

Step 1. Compute local alignments and column probabilities



Step 2. Update column probabilities based on pairing information



Bayesian probability updating procedure

$$p(Y_k, I \mid H_j) \propto \begin{cases} p(l_f) \times e^{S(Y_k)} \times p(I=0) & \text{if conjoint} \\ \frac{1}{2l_g} \times e^{S(Y_k)} \times p(I=1) & \text{if disjoint} \end{cases}$$

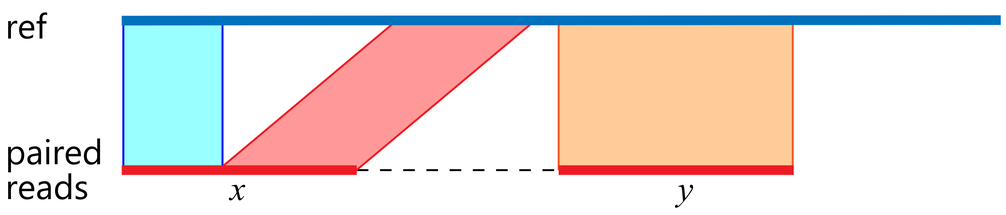
$$p(y \mid H_j) = \sum_k \sum_I p(Y_k, I \mid H_j)$$

$$p(H_j \mid y) = \frac{p(y \mid H_j) \times p(H_j)}{\sum_l p(y \mid H_l) \times p(H_l)}$$

H_j: hypothesis that base $x[i]$ is sequenced from position $g[i_j]$
$P(H_j)$: the column probability of $x[i]$ being aligned to $g[i_j]$
y: a set of local alignments $\{Y_i\}$
$S(Y_k)$: the alignment score of Y_k
ℓ_g: the length of reference
ℓ_f: the length of fragment from which paired-end reads are obtained
assume $p(\ell_f)$ follows a normal distribution
$p(I=0)$: probability that read y is informative about $x[i]$ ($p(I=1)$ is set to 0.01)

Step 3. Compute a final alignment

For each x[i], choose column with highest posterior probability

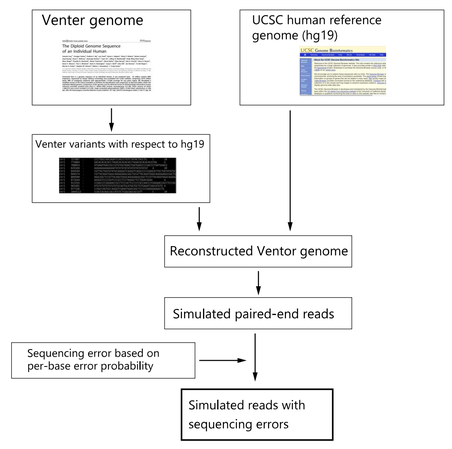


Our implementation is available at:
`https://bitbucket.org/splitpairedend/last-split-pe/`

Results

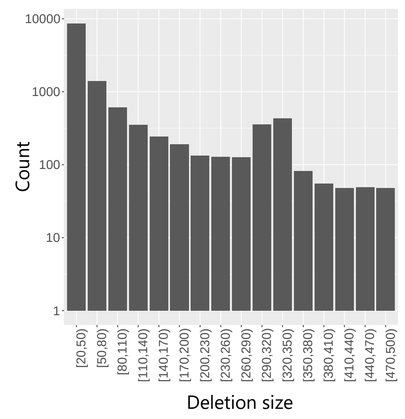
Simulated reads

How to generate



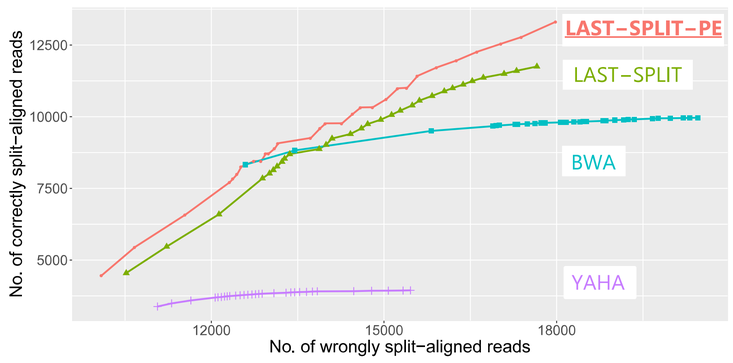
Deletion size distribution in Ventor Genome

Small deletions are more frequent



Split-alignment accuracy

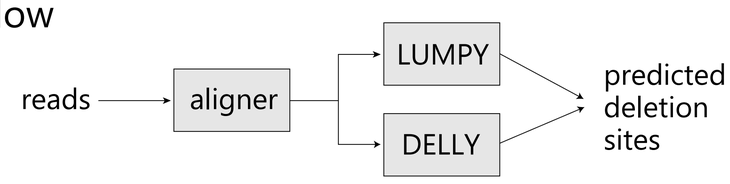
Evaluation of reads that come from sites of large deletion



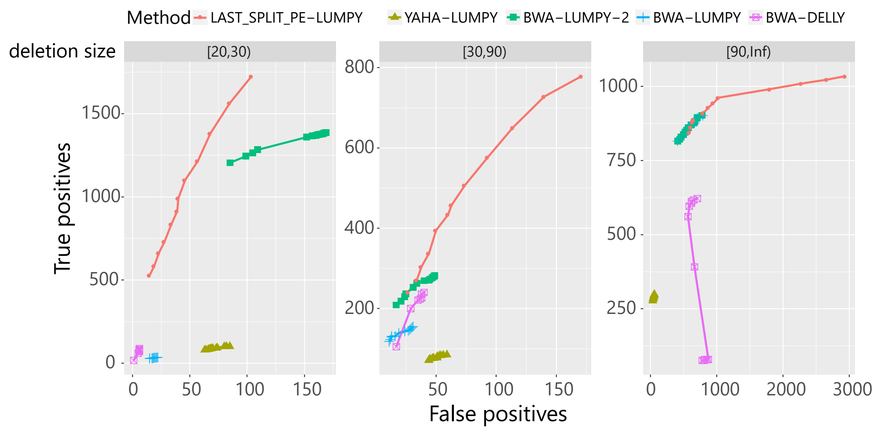
The curves are obtained by changing quality thresholds
A read flank is correctly aligned: at least one of bases in the flank is correctly aligned
A read is correctly aligned: both flanks are correctly aligned
A read is wrongly aligned: at least one flank is not correctly aligned

Effect on variant calling — Evaluation by deletion calls

Workflow



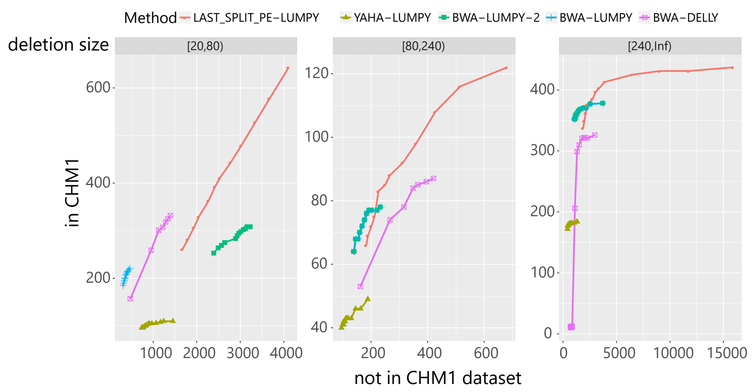
Result



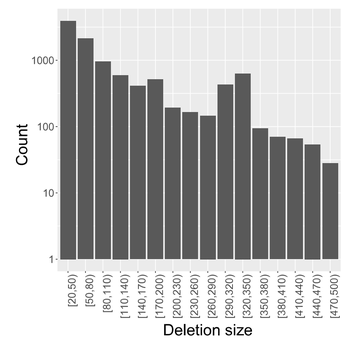
True positive: correctly identifies the start and end coordinates of true deletion
False positive: incorrectly identifies the start or end coordinates of true deletion

Real DNA reads (CHM1)

Evaluation of deletion calls made by aligning short reads from the CHM1 cell line by comparing with those made by aligning PacBio long reads



Size distribution of CHM1 deletions



in CHM1: call is present in the CHM1 PacBio-based dataset
not in CHM1: call is not present in the CHM1 PacBio-based dataset