Journal of Biotechnology 250 (2017) 11–15

Contents lists available at ScienceDirect

Journal of Biotechnology

journal homepage: www.elsevier.com/locate/jbiotec

The challenge of detecting indels in bacterial genomes from short-read sequencing data

Matthias Steglich[a,*], Ulrich Nübel[a,b]

[a] Leibniz Institute DSMZ, Braunschweig, Germany
[b] German Center for Infection Research (DZIF), Partner Site, Hannover-Braunschweig, Germany

ARTICLE INFO

Article history:
Received 23 December 2016
Received in revised form 24 February 2017
Accepted 26 February 2017
Available online 4 March 2017

Keywords:
Indel calling
Short-read data
Bacterial genome
Structural variations
Mapping
de novo assembly

ABSTRACT

We tested the capabilities of four different software tools to detect insertions and deletions (indels) in a bacterial genome on the basis of short sequencing reads. We included tools applying the gapped-alignment (VarScan, FreeBayes) or split-read (Pindel) methods, respectively, and a combinatorial approach with local *de-novo* assembly (ScanIndel). Tests were performed with 151-basepair, paired-end sequencing reads simulated from a bacterial (*Clostridioides difficile* R20291) genome sequence with predefined indels (indel length, 1–2321 bp). Results achieved with the different tools varied widely, and the specific sensitivity and false-discovery rates strongly depended on indel size. All tools tested were able to detect short indels (≤29 basepairs) at sensitivities close to 100%, albeit Pindel reported up to 20% false calls. In contrast, gapped-alignment and split-read tools failed to recover large proportions of long indels (>29 bp) even at 120-fold coverage, and again, Pindel produced significant numbers of false-positive calls. Outstandingly, ScanIndel detected and reconstructed 97% of long indels on average (95% confidence intervals, 88%–99%) and, at the same time, produced negligible amounts of false calls. Hence, the combinatorial approach implemented in ScanIndel was able to recover the positions, types and sequences of indels with excellent sensitivity and false-discovery rate, by encompassing the full indel length spectrum present in the datasets.

1. Introduction

Thousands of bacterial genomes are currently being sequenced for tracking pathogen spatial spread and transmission (Roetzer et al., 2013; Nübel, 2016; Holden et al., 2013; He et al., 2013; Steglich et al., 2015). As a consequence, abundant genome sequence data is accumulating that allows for in-depth analyses of bacterial evolution, as it proceeds over periods of weeks to decades (McNally et al., 2016; Bentley and Parkhill, 2015; Biek et al., 2015). Increasingly, bioinformatic tools in combination with dedicated databases enable the prediction of bacterial phenotypes (e.g. antibiotic resistance) from sequence data alone, which may be very useful in diagnostic settings (Gordon et al., 2014; Bradley et al., 2015). All these analyses are based on detecting sequence variation from short-read sequencing data, usually provided by Illumina technology. Even though sequencing technology providing longer reads (i. e., >10,000 basepairs) is available, associated sequencing costs as yet impede its wider usage in epidemiology or diagnostics. As a consequence, short-read data from bacteria deposited in public sequence databases currently (as of December 2016) outweigh fully closed bacterial genome sequences by a factor of 500 (for example, see http://enterobase.warwick.ac.uk/). Sequence variation to be detected consists of single nucleotide polymorphisms (SNPs) and of structural variation, including insertions and deletions (indels) of variable size. While SNPs represent the most abundant type of variation, indels are more likely to cause measurable phenotypic effects.

The consistent, quantitative detection of single nucleotide polymorphisms (SNPs) in bacterial genomes commonly is achieved after aligning of sequencing reads to a related reference genome sequence. As a result, SNPs have been shown to accumulate in the genomes of several bacterial pathogens in a near clock-like fashion and at species-specific rates, aside from being shuffled by homologous recombination (Nübel, 2016; Biek et al., 2015).

The abundance and evolution of indels in bacterial genomes is less well documented, however, because the detection of indels is more challenging (Grimm et al., 2013; Abel and Duncavage, 2013). A variety of software tools for detecting indels are available, which are based on gapped alignment, split-read analysis, or *de-novo* assembly (Table 1). Both, the gapped alignment and

* Corresponding author.
E-mail address: mst15@dsmz.de (M. Steglich).

http://dx.doi.org/10.1016/j.jbiotec.2017.02.026
0168-1656/

Table 1
A selection of currently available implementations that provide variant detection on short-read sequencing data.

Application	Source	Publication	Type
VarScan/VarScan2	http://dkoboldt.github.io/varscan/	Koboldt et al. (2012)	gapped alignment
GATK toolkit	https://software.broadinstitute.org/gatk/	–	split read
FreeBayes	https://github.com/ekg/freebayes	Garrison and Marth (2012)	gapped alignment
Pindel	http://gmt.genome.wustl.edu/packages/pindel/	Ye et al. (2009)	split read
Dindel	https://sites.google.com/site/keesalbers/soft/dindel/	Albers et al. (2011)	split read
Breakdancer	http://gmt.genome.wustl.edu/packages/breakdancer/	Chen et al. (2009)	gapped alignment
ScanIndel	https://github.com/cauyrd/ScanIndel	Yang et al. (2015)	gapped alignment, split read, *de novo* assembly
ClC-Bio Workbench	https://www.qiagenbioinformatics.com/products/clc-main-workbench/	–	gapped alignment
Structural Variant Machine (SV-M)	https://www.bsse.ethz.ch/mlcb/research/bioinformatics-and-computational-biology/structural-variant-machine-sv-m-.html	Grimm et al. (2013)	split read
SVM2	http://nar.oxfordjournals.org/content/40/18/e145.full	Chiara et al. (2012)	split read
Modil	http://compbio.cs.toronto.edu/modil/	Lee et al. (2009)	mate pair
Platypus	http://www.well.ox.ac.uk/platypus	Rimmer et al. (2014)	assembly
Scalpel	http://www.nature.com/nmeth/journal/v11/n10/full/nmeth.3069.html	Narzisi et al. (2014)	assembly
FermiKit	https://github.com/lh3/fermikit	Li (2015)	assembly
Breseq	http://barricklab.org/twiki/bin/view/Lab/ToolsBacterialGenomeResequencing	Deatherage et al. (2014)	split read

split-read approaches rely on an alignment of reads to a reference sequence (similar to SNP detection). Based on the alignment, indel calling algorithms rate the evidence supporting each indel candidate and then report those indels with scores above a threshold. The influence of alignment tools on the capabilties of detecting indels was investigated previously (Yang et al., 2015). Gapped alignment methods require that indels are completely contained in individual sequencing reads and correctly reconstructed during the alignment step. This approach performs well for indels that are shorter than approximately 15% of the read length (Yang et al., 2015). In contrast, split-read methods identify indels on the basis of individual reads that map to the reference sequence discontinuously, such that the 5′-end of the read is aligned to one region and the 3′-end to another (Abel and Duncavage, 2013). While the split-read approach enables the detection of insertions that are longer than the reads, it is prone to false-positive calls due to alignment errors (Yang et al., 2015). *De-novo* assembly may also be used to identify large indels, but it does not as yet enable the reconstruction of full bacterial genomes from short reads due to repetitive structures. In addition, *de novo* assembly requires excessive computational resources and sequence reconstruction may be error-prone (Yang et al., 2015). Therefore combinations of several approaches have been implemented in software packages (Table 1).

Here, we have tested the performance of four software tools to detect indels in sequencing read datasets simulated from a bacterial (*Clostridioides difficile*) genome sequence with pre-defined mutations. In a first step, we aligned the reads to a reference sequence by using the alignment software BWA-MEM (Li, 2013). Subsequently, we included indel detection tools that employ gapped alignment (VarScan2 (Koboldt et al., 2012), FreeBayes (Garrison and Marth, 2012)), split reads (Pindel (Ye et al., 2009)), or a combination of these two methods with localized *de-novo* assembly (ScanIndel (Yang et al., 2015)).

2. Material and methods

2.1. Simulation of indel read data sets

We generated 50 simulated read data sets at each of two different coverage settings, i. e. 30-fold and 120-fold coverage, as follows. We used ten non-overlapping 300-kilobase sequence fragments from the genome sequence from *Clostridioides difficile* strain R20291 (sequence accession number, FN545816) as a basis, and introduced to each of these an individual set of 32 non-overlapping indels (16 insertions, 16 deletions) at random positions. Indel lengths were assigned arbitrarily, ensuring an exponential size distribution, ranging from 1 bp to 2321 bp length resulting in 44% indels with a length ≤29 bp. Insertion sequences were copied from the genome sequence from the myxobacterium *Sorangium cellulosum* So0157-2 (accession number, CP003969). Subsequently, we used the software WGSIM (https://github.com/lh3/wgsim) to simulate paired-end reads from each indel-carrying sequence fragment of the simulated data sets at both coverage settings, in total 100 different read files. The read length was set to 151 bp, the base error rate was set to 0.001 and the rate of mutations was set to 0.00001. No additional indel mutations were allowed (fraction of indels = 0). The amount of read pairs was adjusted to generate 30-fold and 120-fold coverage for 50 datasets each.

2.2. Indel detection with VarScan, Freebayes, Pindel and ScanIndel

We used the alignment tool BWA-MEM (Li, 2013) in accordance with manuals for indel detection tools ScanIndel, Varscan, Pindel and FreeBayes. By using BWA-MEM at default settings, simulated sequencing reads were aligned to their respective template sequences from which the original indel-carrying sequence was derived from. Results (BAM files) were analysed with VarScan2 mpileup2cns, setting the minimum coverage to 10, the minimum reads2 parameter to 6, the minimum var-frequency to 0.6, the p-value threshold for calling variants to 0.01 and the minimum average base quality to 20. The analyses with Pindel were performed setting the parameter for the maximum size of structural variations to be detected to 4 (≤8092 bp). Analyses with FreeBayes and ScanIndel were performed at default settings. Eventually, output in VCF format (VCFv4.1) from all four applications was analysed. In the case of Pindel, the pindel2vcf tool from the Pindel project (https://github.com/genome/pindel) was used to convert the raw format of all Pindel outputs.