Genomics 102 (2013) 500–506

Contents lists available at ScienceDirect

Genomics

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

Methods

AlienTrimmer: A tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads

Alexis Criscuolo *, Sylvain Brisse

Institut Pasteur, Genotyping of Pathogens and Public Health Platform (PF8), 28 rue du Dr Roux, 75724 Paris Cedex, France
Institut Pasteur, Microbial Evolutionary Genomics Unit, 28 rue du Dr Roux, 75724 Paris Cedex, France
CNRS, UMR3525, 75015 Paris, France

ARTICLE INFO

Article history:
Received 26 December 2012
Accepted 25 July 2013
Available online 1 August 2013

Keywords:
High-throughput sequencing
Raw read trimming
Short contaminant sequence
Adapter oligonucleotides
k-mer decomposition
Polynomial algorithm

ABSTRACT

Contaminant oligonucleotide sequences such as primers and adapters can occur in both ends of high-throughput sequencing (HTS) reads. AlienTrimmer was developed in order to detect and remove such contaminants. Based on the decomposition of specified alien nucleotide sequences into *k*-mers, AlienTrimmer is able to determine whether such alien *k*-mers are occurring in one or in both read ends by using a simple polynomial algorithm. Therefore, AlienTrimmer can process typical HTS single- or paired-end files with millions of reads in several minutes with very low computer resources. Based on the analysis of both simulated and real-case Illumina®, 454™ and Ion Torrent™ read data, we show that AlienTrimmer performs with excellent accuracy and speed in comparison with other trimming tools. The program is freely available at ftp://ftp.pasteur.fr/pub/gensoft/projects/AlienTrimmer/.

1. Introduction

High-throughput sequencing (HTS) technologies produce voluminous amounts of DNA sequence data, which are used in a wide array of biological applications including de novo genome sequencing, expression analysis and detection of sequence variants. As the raw sequence data often contain various types of errors and artifacts, processing of the sequence reads is necessary to remove e.g. low quality bases, read redundancy, and alien exogenous oligonucleotide sequences. Given the voluminous amount of primary data (typically millions of reads per sample), much effort must go into algorithm design to implement programs that are able to process such datasets with reasonable running times (e.g. few minutes).

This work focuses on the trimming of contaminant sequences that are often present in the ends of sequence reads generated by high-throughput sequencers. Trimming of contaminant sequences is analogous to vector removal that is applied on sequences obtained from cloned DNA fragments. Such alien sequences do not correspond to the biological material under study but result from the unwanted sequencing of specifically designed oligonucleotides that are used for DNA library preparation (e.g. primers, adapters and barcode indexes). As an example, this phenomenon happens when sequencing library fragments that are shorter than the read length. As the presence of such exogenous oligonucleotide sequences can negatively affect subsequent analyses such as mapping onto a reference sequence for polymorphism detection, or *de novo* assembly of genomic sequences (see Section 2.1), it is highly desirable to detect and remove contaminant sequences. Several software tools have been implemented to achieve this task, but many of them do not have desirable capabilities, including the identification of more than one possible alien oligonucleotide (sub)sequences in both ends, and the ability to process paired-end data simultaneously (see review in [1]). Moreover, current tools identify alien sequences by performing (semi)local alignments, which could incur long running times when searching many distinct alien sequences on very large datasets.

Here we present AlienTrimmer, a program that allows identifying and removing multiple alien oligonucleotide (sub)sequences in both 5′ and 3′ ends of single- or paired-end reads. Given a fixed integer value *k*, AlienTrimmer first performs *k*-mer decomposition of every specified alien sequences. Second, AlienTrimmer searches for the exact occurrence of each alien *k*-mer within reads, and all nucleotides covered by alien *k*-mers in 5′ and/or 3′ ends are trimmed (Section 4.2). Based on the analysis of different datasets (Section 4.1), AlienTrimmer was demonstrated to rapidly provide accurate results in comparison with three other trimming programs (Section 2).

2. Results and discussion

2.1. Negative impact of the presence of alien oligonucleotide sequences in sequence read ends

In order to illustrate the effect of alien sequences on downstream analysis of HTS datasets, we performed read mapping and

* Corresponding author at: Institut Pasteur, Genotyping of Pathogens and Public Health Platform (PF8), 28 rue du Dr Roux, 75724 Paris Cedex, France. Fax: +33 1 45 68 87 27.
E-mail address: alexis.criscuolo@pasteur.fr (A. Criscuolo).

0888-7543/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ygeno.2013.07.011

de novo assembly with the real-case read files BC.454 and PF.Ion containing ending-up exogenous (adapter) oligonucleotide residues (see Section 4.1 for details about these data files). Furthermore, in order to illustrate the benefit of performing alien oligonucleotide sequence trimming, we processed these two read files with AlienTrimmer, as well as three other recent alien sequence trimming implementations: cutadapt [2], flexbar [3], and btrim [4]. All programs were used with default options.

The mapping procedure was performed with the PF.Ion reads against the reference genome sequences of *Plasmodium falciparum* strain 3D7 [5] in order to call single nucleotide polymorphisms (SNPs). For each of the five read files (i.e. initial read file, and those returned by the four alien trimming programs), read alignments were performed with BWA (v. 0.5.9) [6], and SNPs were called by using the software package SAMtools (v. 0.1.18) [7], both with default options. Resulting SNP sets were intersected in order to only focus on the 312 common SNPs called from the five read files, and, for each resulting SNP subsets, the distribution of the coverage depths of mapped reads was graphically represented as a box plot in Fig. 1A. Knowing that the presence of an adapter oligonucleotide (sub)sequence in a read could result in strong dissimilarity with the reference genome, it was not surprising that 54.21% of the reads from the contaminated read file did not map successfully. As expected, a large proportion of the reads obtained after performing alien trimming mapped successfully, therefore lowering the levels of unmapped reads, i.e. AlienTrimmer: 37.24%; cutadapt: 35.86%, flexbar: 35.22%; btrim: 54.04%. Note that these important levels of unmapped reads are likely caused by the large number of low quality nucleotides within the PF.Ion reads (see Section 4.1). However, this analysis illustrates the benefit gained by trimming exogenous sequences for mapping approaches, as it leads to improved coverage depths (Fig. 1A) and would therefore result in a more reliable SNP calling, as compared to using reads without exogenous residue trimming step.

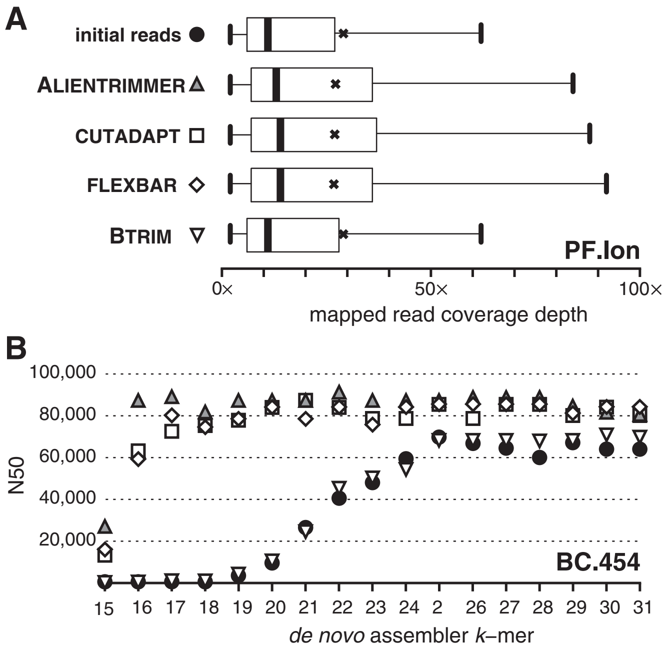

Fig. 1. Illustration of the negative impact on mapping and *de novo* assemblies of exogenous oligonucleotide sequences within HTS real-case reads. Five read files were used for these analyses: reads with alien sequences (black circles), and trimmed by AlienTrimmer (gray triangles), cutadapt (white squares), flexbar (white lozenges), and btrim (white reverse triangles), respectively. A: Box-plots representing the distribution of coverage depths of called SNPs from PF.Ion reads. For each box-plot, the left and right parts of each box represent the lower and upper quartiles, respectively; the bold vertical line inside the box represents the median, and the left and right whiskers represent the 9th and 91st percentiles, respectively. Average mapped read coverage depths (i.e. the number of read residues divided by the length of the reference genome) are represented by thick crosses. B: N50 contig lengths of *de novo* assemblies inferred from BC.454 reads with varying *k*-mer *de novo* assembler parameter values.

To evaluate the effect of eliminating contaminant sequences on *de novo* assembly, contig sequences were built from the BC.454 reads with the program tool clc_novo_assemble (v. 3.22.55708) from the CLC Genomics Workbench NGS analysis package [8]. This tool allows assembling contig sequences from raw reads by implementing methods related to *k*-mer graphs, often called de Bruijn graph *de novo* assembly methods (see e.g. [9]). These approaches consist in decomposing all read sequences into overlapping *k*-mer sequences of fixed length *k* (in a similar way as processed by AlienTrimmer; see Section 4.2), and building a graph where each vertex is a distinct read *k*-mer, whereas each edge connects two vertices when a $k - 1$ nucleotide overlap occurs between the two corresponding *k*-mers; based on this data structure, contig sequences are related to the longest paths within such graphs. For each of the five read sets (i.e. initial reads, and those returned by AlienTrimmer, cutadapt, flexbar, and btrim), we launched clc_novo_assemble with parameter *k* varying from 15 to 31 (its maximum allowed value). Assemblies were assessed by computing the N50 contig length (i.e. the length of the smallest element inside the minimum set of contigs whose lengths total to more than half the total number of nucleotides within all inferred contigs). This measure is inversely related to the fragmentation level of the assembly. For each read file, and for each value between 15 and 31 of the clc_novo_assemble parameter *k*, the N50 contig length was computed. The results are represented graphically in Fig. 1B. They show that although setting large *k*-mer lengths (e.g. $k > 24$) leads to higher N50 contig lengths (e.g. N50 > 60,000), the overall ability of clc_novo_assemble to infer large contigs was clearly affected by the presence of exogenous residues. Fig. 1B shows that trimming exogenous oligonucleotides clearly improves the overall *de novo* assemblies performed by clc_novo_assemble.

In conclusion, these experiments show that the presence of alien oligonucleotide (sub)sequences in read ends does affect negatively the quality of downstream analyses. In turn, filtering out alien residues improves the quality of read mapping or *de novo* assembly results. Clearly, the negative impact of contaminant sequences will depend on the ratio of alien versus non-alien residues within reads. However, in practice, this ratio is difficult to anticipate. In addition, other unexpected effects of contaminants might occur, such as assembly artifacts or quantification biases in gene expression analysis. A preliminary processing of sequence reads that trims off exogenous residues therefore appears advisable before engaging into the biological analysis of high-throughput sequencing data.

2.2. Comparing AlienTrimmer with other trimming tools

The accuracy and running time of AlienTrimmer were determined and compared with those of cutadapt (v. 0.9.4), flexbar (v. 2.33), and btrim (as of January 2013). These three tools were selected because they allow complete exogenous sequence trimming to be performed, i.e. 5′ and/or 3′ sequence trimming, and have the ability to use more than one possible alien oligonucleotide sequence as input (see [1]). For this purpose, we created three simulated read files related to data BC.454, KP.Illumina and PF.Ion (see Section 4.1). These simulated read files contained contaminant sequences with known positions (for more details, see Section 4.1), thus enabling to precisely assess the respective accuracy of the four tested programs.

Each artificial read file was processed using as alien sequences, the set of adapter sequences used for contamination (see Section 4.1). AlienTrimmer was run with different values of *k* (i.e. from 6 to 15). Cutadapt was run with varying error rates (i.e. from 0 to 0.4) to tolerate an increasing number of putative mismatches or indels between aligned alien and read sequence, as well as with the option that allows searching alien sequences in both read ends (see [2] for more details). Flexbar was used with the option that allows searching alien sequences in both read ends. Different cut-off parameter values, ranging from 0 to 40, were used. This parameter is related to the maximum number of mismatches allowed between aligned alien and read sequences (see [3] for more details). For a given read, BTRIM performs either 5′ trimming followed by 3′ trimming, when an alien sequence was detected, or only 3′ trimming (i.e. without seeking alien sequences at the 5′ end; see [4] for more details). Therefore, BTRIM was launched twice (once for 5′ and then 3′, and once for 3′ only), with different numbers of maximum allowed alien sequence match errors (i.e. from 0 to 19 for both ends). In all runs, options were set to prevent read discarding based on sequence length or Phred [10] quality criteria, in order to focus only on alien trimming approaches.

Each trimming process was evaluated for running time and accuracy. The latter was assessed by counting the true and false positives (*TP* and *FP*, i.e. number of alien and non-alien nucleotides that were trimmed, respectively), as well as the true and false negatives (*TN* and *FN*, i.e. number of non-alien and alien nucleotides that were not trimmed, respectively). For each program run, each observed false positive rate $FPR = FP / (FP + TN)$ was plotted against its corresponding true positive rate $TPR = TP / (TP + FN)$. These plots allow levels of sensitivity (i.e. *TPR*) and specificity (i.e. $1 - FPR$) to be observed, and were represented, for each of the three read data (i.e. BC.454, KP.Illumina, PF.Ion) and each of the four programs (i.e. AlienTrimmer, CUTADAPT, FLEXBAR, BTRIM), as receiving operating characteristic (ROC) curves (see e.g. [11]) in Fig. 2A. Down-left tails of ROC curves are related to the most conservative parameters (i.e. $k = 15$ for AlienTrimmer, and no mismatch or indel allowed for CUTADAPT, FLEXBAR and BTRIM), whereas top-right heads correspond to relaxed parameters. To quantify the respective accuracy of each trimming implementation, the area under curve (AUC; see e.g. [12]) was estimated from each ROC curve (Fig. 2A). In this approach, a method is considered as more able to discriminate exogenous residues from non-alien nucleotides as its AUC is closer to 1. To illustrate the respective speed of the four programs, we also processed the three read data with an increasing number of alien oligonucleotide sequences (i.e. from 1 to 15). The average numbers of processed reads per minute were measured on a 1.6-GHz Intel® Xeon® CPU computer (128 Gb RAM), and are represented in Fig. 2B.

When applied on the three read datasets, CUTADAPT and FLEXBAR led to the best accuracies (i.e. AUC > 0.99), whereas BTRIM showed less optimal results (i.e. AUC < 0.97; see Fig. 2A). Comparison of the processing speed of each program showed that BTRIM had the fastest running times, whereas CUTADAPT and FLEXBAR processed the lowest numbers of reads per minute (see Fig. 2B). Interestingly, AlienTrimmer showed similar accuracies as those observed with CUTADAPT and FLEXBAR, but with higher processing speed (i.e. ~1 M reads per minute; see Fig. 2). Moreover, as expected (see algorithm details in Section 4.2), running times of AlienTrimmer were largely insensitive to the number of input alien sequences, contrary to the three other programs (see Fig. 2B). Notably, high speed was also observed with data PF.Ion, even though it contains degenerated bases (i.e. ~0.01% PF.Ion simulated reads contain at least one nucleotide N), which are expected to have a negative impact on the running time of AlienTrimmer (see Section 4.2). Of note, FLEXBAR could be run by using *t* different threads, which allows increasing its speed by a factor of *t* [3]. However, a large thread number *t* must be used to equate the speed of AlienTrimmer (e.g. at least $t = 6$ with data BC.454), requiring large memory amounts [3].

It has to be stressed that algorithms based on *k*-mer decompositions (see Section 4.2), as implemented in AlienTrimmer, could decrease in accuracy when sequencing errors occur, or when dealing with short fragments of alien oligonucleotide sequences. In order to determine the respective performance of each tool for varying alien oligonucleotide sequence length and number of mismatches within alien residues, each artificial read data was stratified according to contaminant length and number of mismatches, and the corresponding AUC was determined

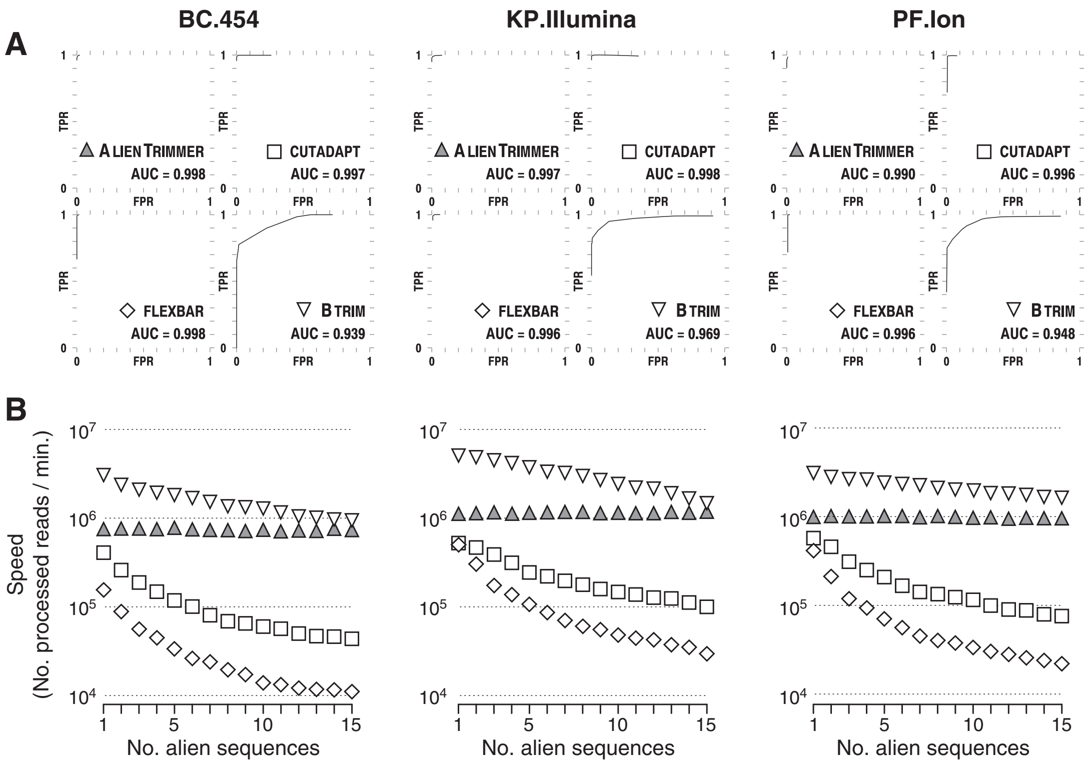

Fig. 2. Accuracy and processing speed of four alien sequence trimming tools on simulated reads containing adapter oligonucleotide sequences in 5′ or 3′ ends. Three sets of reads were processed: BC.454 (left), KP.Illumina (middle), and PF.Ion (right). A: ROC curves representing false positive rate (X-axis) against true positive rate (Y-axis) as observed after varying the trimming parameters of AlienTrimmer (parameter *k*), CUTADAPT (error rate parameter), FLEXBAR (mismatch parameter), and BTRIM (match error parameter in both read ends). B: Average number of processed reads per minute (on a logarithmic scale) measured with an increasing number of specified alien sequences as input. AlienTrimmer (gray triangles), CUTADAPT (white squares), and FLEXBAR (white lozenges) were used with default options. Note that the running times of BTRIM (white reverse triangles) correspond to the sum of two runs (trimming at 5′ then 3′, and 3′ only).

for each trimming program. Fig. 3 shows these different AUC values estimated from read subsets depending on the number of mismatches. More detailed graphical representations depending on both contaminant length and number of mismatches are provided in Supplementary materials (Supplementary Figs. S1, S2 and S3 for data BC.454, KP.Illumina and PF.Ion, respectively). Fig. 3 shows a slight decrease of AUC values when AlienTrimmer processed reads containing alien sequences with mismatches, leading to inaccurate results (e.g. AUC < 0.95) when at least 2 mismatches occur. However, due to the initial quality-based trimming (see Section 4.1), reads containing exogenous oligonucleotide sequences with mismatches were infrequent, e.g. representing 2.37% and 4.10% of the contaminated reads in BC.454 and KP.Illumina, respectively (Fig. 3). This aspect is illustrated in Fig. 4 that represents four examples of reads containing adapter oligonucleotide sequences picked from the real-case read data KP.Illumina. Although some reads could contain adapter sequences with mismatches (Fig. 4C), read regions with numerous sequencing errors are generally supported by low Phred quality scores [10,13] and therefore discarded by prior quality-based trimming (Fig. 4D). Accordingly, AlienTrimmer (as the three other programs) integrates an option to perform quality-based trimming together with alien trimming without significant impact on the overall running times. It should also be stressed that quality-based trimming can lead to very short fragments of exogenous oligonucleotides (e.g. less than 6 residue long) that are difficult to detect and trim off (see Supplementary Figs. S1, S2 and S3). However, reads with short remnants of alien sequences represent a small proportion of the reads after trimming (e.g. only 1.65%, 1.91%, and 3.36% of those outputted by AlienTrimmer when applied with default options on BC.454, KP.Illumina, and PF.Ion reads, respectively), and their presence is expected to have a limited impact on downstream analyses (see e.g. Section 2.1).

3. Conclusion

Exogenous oligonucleotide sequences can be incorporated in read ends during high-throughput sequencing procedures. We showed that their presence could negatively impact subsequent analyses based either on *de novo* assembly or mapping strategies. We developed and evaluated a new tool, AlienTrimmer, which is able to locate and trim off any specified alien (sub)sequence occurring in read ends, based on a simple polynomial algorithm performing *k*-mer decomposition. We showed that AlienTrimmer represents an excellent compromise between accuracy and speed. Although AlienTrimmer accuracy is impacted by nucleotide mismatches due to sequencing errors, this drawback can be greatly minimized by performing quality-based trimming (e.g. [14–17]), knowing that sequencing errors are often supported by low quality scores (e.g. [10,13]). Therefore, we recommend that AlienTrimmer be used subsequent to quality-based read trimming, or together with its Phred quality-based trimming option. Owing to an initial step consisting of *k*-mer decomposition of alien oligonucleotide sequences, the running time of AlienTrimmer is virtually not affected by the number of specified alien sequences, provided that reads contain few degenerated nucleotides. Therefore, AlienTrimmer analysis could be conveniently incorporated as a standard pre-processing step to detect and trim off every known putative alien sequences in a process. This could be particularly useful to automatically treat heterogeneous datasets from varied sources, and for public datasets for which the experimental details on the adapters used during library construction are unknown.

4. Material and methods

4.1. Real and simulated datasets

Real-case FASTQ [18] files were considered from three different organisms and three different HTS technologies. The first data file (SRA: SRR032593) contains 307,639 Roche (454™) GS-FLX™ single-end reads (~278 nucleotide long on average) from the α-proteobacteria *Brucella ceti* strain M644/93/1. The second data file consists in 19,646,070 Illumina® HiSeq™ 2000 100-nucleotide single-end reads obtained by sequencing the total DNA of *Klebsiella pneumoniae* strain MGH 78578. The third data file was obtained by concatenating four FASTQ files (SRA: ERR161538, ERR161539, ERR161540, and ERR161542; see [19]) resulted from the Ion Torrent™ PGM™ sequencing of the genomic DNA of the eukaryote parasite *Plasmodium falciparum* strain 3D7, leading to a total of 7,169,172 single-end reads (~142 nucleotide long on average). For the sake of simplicity, these three datasets are referred to as BC.454 (*B. ceti*), KP.Illumina (*K. pneumoniae*), and PF.Ion (*P. falciparum*) throughout this paper.

These three sets of reads were screened for different alien sequences. Two different alien sequences occurred within BC.454 reads: barcoded adapter A (CCATCTCATCCCTGCGTGTCTCCGACTCAG TCTCCGTC) and adapter B (CTGAGACACGCAACAGGGGATAGGCAAGGCACACAGGGGATA

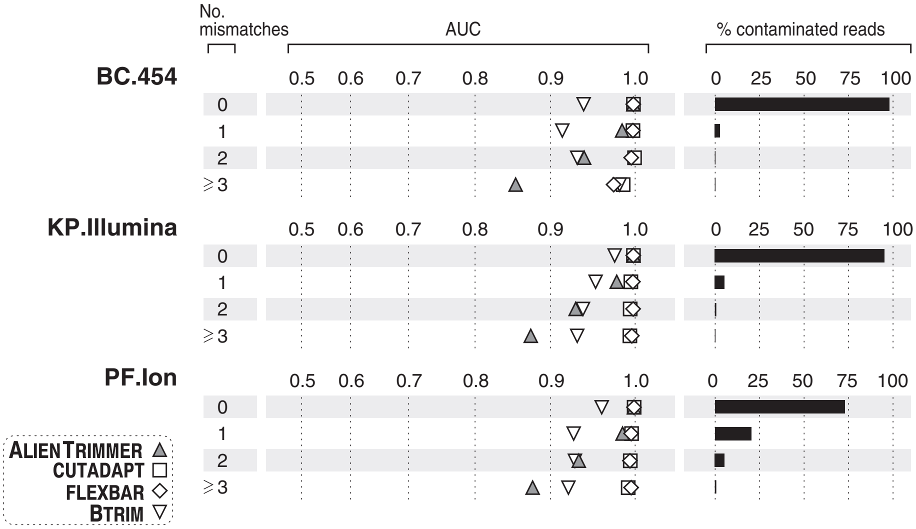

Fig. 3. Accuracy of four alien trimming tools depending on the number of mismatches occurring in alien oligonucleotide sequences within artificially generated read data. The first column gives the number of mismatches within alien residues. Histograms on the right represent the percentage of reads containing alien residues with specified mismatch(es). For each case, the central part represents on an exponential scale the AUC estimated from the ROC curve computed for each of the four alien trimming tools AlienTrimmer (gray triangles), cutadapt (white squares), flexbar (white lozenges), and btrim (white reverse triangles).

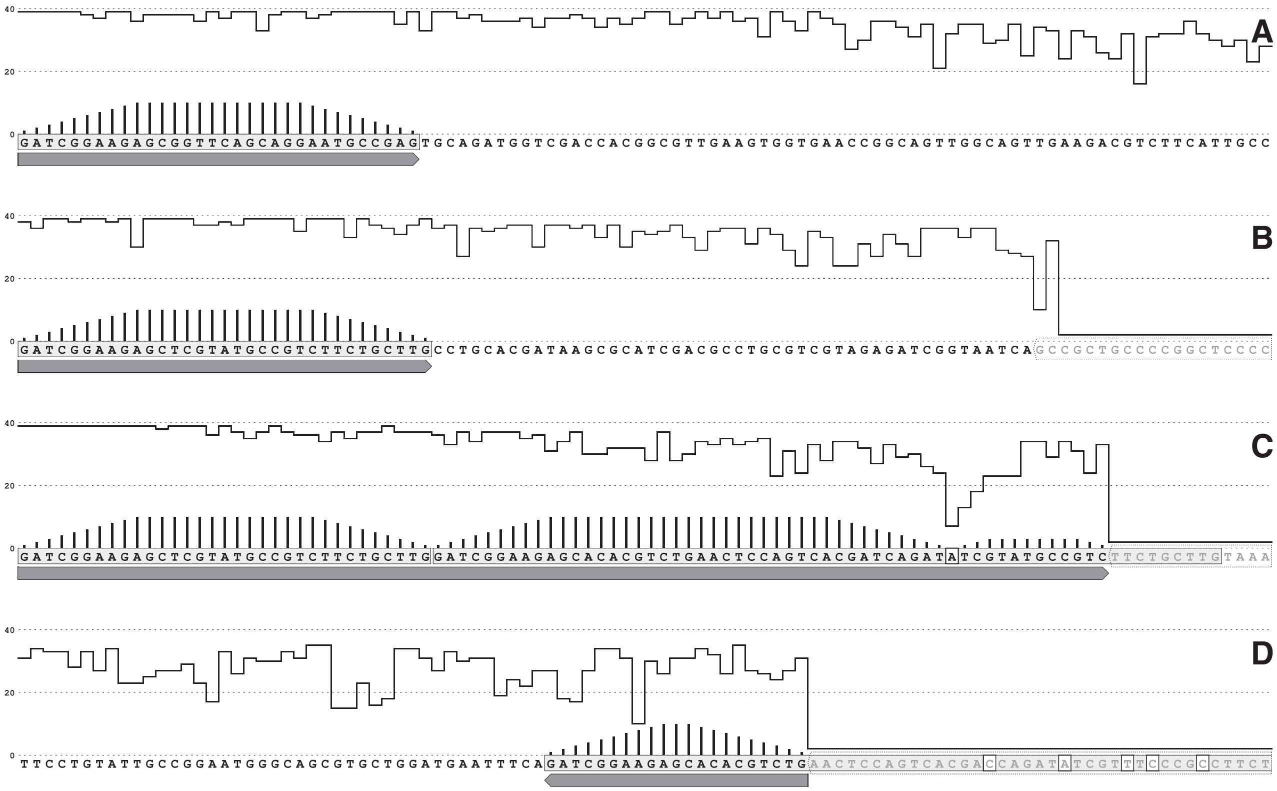

Fig. 4. Examples of exogenous oligonucleotide sequences in 5′ and 3′ ends of raw read data. For each raw read, the Phred quality score (up to 40) of each nucleotide is represented by skyline plot. For better reading, threshold quality scores of 0, 20 and 40 are represented by dashed horizontal lines. Low quality nucleotides (i.e. Phred quality score < 20) that have been trimmed using sickle are inside the dashed arrows (B, C, and D). Alien oligonucleotide sequences and alien residue mismatches are represented inside the boxes with light gray and white backgrounds, respectively. For each read, alien k-mers were searched by AlienTrimmer with $k = 10$, and the alien coverage is represented (up to 10) for each nucleotide by thick vertical lines using the same ordinate scale as for Phred quality score. Resulting read prefixes or suffixes trimmed by AlienTrimmer are indicated by a dark gray arrow under each read. A: Illumina paired-end adapter. B: Illumina adapter for genomic DNA. C: Illumina adapter for genomic DNA followed by TruSeq adapter (index 9) with one mismatch; note that this read is entirely filtered out by the trimming procedure. D: TruSeq adapter (index 9) with different mismatches. Oligonucleotide sequences© 2007–2013 Illumina, Inc. All rights reserved.

GG) in 5′ and 3′ ends, respectively. Three different adapters occurred within KP.Illumina reads: TruSeq® adapter index 9 (GATCGGAAGAGCACACGTCTGAACTCCAGTCAC GATCAG ATCTCGTATGCCGTCTTCTGCTTG), Illumina® adapter for genomic DNA (GATCGGAAGAGCTCGTATGCCGTCTTCTGCTTG), and Illumina® paired-end adapter (GATCGGAAGAGCGGTTCAGCAGGAATGCCGAG). Only one alien oligonucleotide sequence occurred within PF.Ion reads: the reverse-complement of the concatenated adapters P1 (CCTCTCTATGGGCAGTCGGTGAT) and B (CCTATCCCCTGTGTGCCTTGGCAGTCTCAG).

For quality-based trimming, reads from the three data files BC.454, KP.Illumina and PF.Ion were mined in order to discard low quality nucleotides, corresponding to putative sequencing errors [10,13]. We selected a Phred [10] quality score of 20 as quality cut-off for the two read files BC.454 and KP.Illumina. Knowing that PF.Ion reads were supported by low Phred quality scores on average (for more details, see [19]), we selected a quality cut-off of 13 for this dataset, in order to not discard too many reads. For the three datasets, we first discarded every read containing more than 70% low quality nucleotides. Next, we performed quality-based trimming with the dedicated program sickle [20] in order to remove low quality nucleotides that may occur in 5′ or 3′ ends. During this quality trimming step, every read with length lower than 30 nucleotides was discarded. Following these standard processes, the remaining data (BC.454: 306,315 reads, KP.Illumina: 11,885,438 reads, PF.Ion: 6,183,417 reads) were used for alien trimming and bioinformatics analyses (see Section 2.1).

For simulation analyses, the three original data files were used to generate FASTQ files containing simulated reads. Model genome sequences were selected for each dataset, i.e. BC.454: 17 genomic scaffolds from *B. ceti* strain M644/93/1 (GenBank accession no. NZ_ACBO00000000); KP.Illumina: the complete circular chromosome sequence of *K. pneumoniae* strain MGH 78578 (GenBank accession no. NC_009648); PF.Ion: 14 chromosomes, 1 mitochondrial DNA, and 2 apicoplastid sequences of *P. falciparum* strain 3D7 (version 2.1.5; available at [5]). For each initial (i.e. non-trimmed) read file BC.454, KP.Illumina and PF.Ion, we randomly extracted the same number of nucleotide substrings from the associated model genome sequences, each random substring having the same length as its corresponding raw read sequence. End replacement with alien sequences was performed with a 50% contamination rate. For each read selected for contamination, the 5′ or 3′ end was randomly chosen, and one adapter was randomly selected as the alien sequence (BC.454: adapters A and B; KP.Illumina: TruSeq® adapter index 9, Illumina® adapter for genomic DNA, and Illumina® paired-end adapter; PF.Ion: reverse-complement of concatenated P1 + B adapters, and non-barcoded adapter A). Each contamination was simulated by setting a substring of length 30 nucleotides of the picked alien sequence as a prefix or a suffix of the read for 5′ or 3′ end contamination, respectively. All simulated reads were associated with the initial Phred quality scores strings, and all nucleotide residues were randomly mutated based on their related base-calling error probabilities. Mutated residues were obtained by randomly picking among the three other nucleotides. We then applied a standard quality-based trimming step using sickle as described above. Following this procedure, we obtained three artificial FASTQ files containing the same number of reads and residues as the original data files BC.454, KP.Illumina and PF.Ion, but with known contaminant oligonucleotide frequencies (46.93%, 40.45%, and 39.48% contaminated reads for BC.454, KP.Illumina, and PF.Ion, respectively) and positions. These simulated FASTQ test files are available at ftp://ftp.pasteur.fr/pub/gensoft/projects/AlienTrimmer/.

4.2. The AlienTrimmer algorithm

Decomposing DNA sequence into overlapping k-mers (i.e. sequences of length k) is a standard task in bioinformatics. Given a non-zero positive integer value k, each of the 4^k k-mers can be stored with only nk bits by using an n-bit binary code b_n (provided $n > 1$). AlienTrimmer uses a binary coding b_2 defined as $b_2(\mathrm{A}) = 00$, $b_2(\mathrm{C}) = 01$, $b_2(\mathrm{G}) = 10$, and $b_2(\mathrm{T}) = 11$. In order to deal with degenerate nucleotides N, it also uses another binary coding b_4 defined as $b_4(\mathrm{A}) = 0001$, $b_4(\mathrm{C}) = 0010$, $b_4(\mathrm{G}) = 0100$, and $b_4(\mathrm{T}) = 1000$ (see below for details about b_4 coding of degenerate residues). Therefore, each k-mer is bijectively associated to a unique binary number of nk bits. Usual computer words being 32 or 64 bit long, such binary codes allow k-mers with $k \le 16$ to be easily computed and stored. Given an alien or read sequence, AlienTrimmer computes the binary representation of its k-mers by using three basic bit operations: bitwise OR (|), bitwise AND (&), and bit left shifting (<<). More formally, if w_i is the binary representation of the k-mer starting at position i and ending at position $i + k - 1$, then the binary representation w_{i+1} of the next k-mer is easily computed by the formula $w_{i+1} = ((w_i << n) \mid b_n(c_{i+k}))\ \&\ mask_n$, where c_{i+k} is the character state at position $i + k$, and $mask_n$ a constant binary representation of $n^k - 1$ that allows setting to zero all bits shifted beyond the nkth one. Following this approach, the list of every k-mer from a nucleotide sequence of length L is computed in time $O(L)$.

Given a fixed integer value k (=10 by default), AlienTrimmer first performs k-mer decomposition from each specified alien sequence, and the binary representation w of each extracted alien k-mer is computed following both binary codes b_2 and b_4 in order to deal with the presence of degenerate nucleotides (see below). Each binary representation of a k-mer encoded with b_2 being equivalent to an integer lying between 0 (i.e. poly-A of length k) and $4^k - 1$ (i.e. poly-T of length k), each distinct alien k-mer is stored inside a bitset B of size 4^k (i.e. if an alien k-mer is b_2-coded as w, the xth bit of B is set to 1 where x is the integer value equivalent to w). Each alien k-mer binary representation encoded with b_4 is stored in a sorted list Λ. Note that AlienTrimmer also stores the binary representations of the reverse-complement of each extracted alien k-mer, in order to be able to search for the reverse-complement of each specified alien sequence in read ends. This whole procedure allows computing and storing (in both bitset B and sorted list Λ) the $K = |\Lambda|$ distinct k-mers that can be extracted from the different specified alien sequences (see algorithm details in Supplementary materials S1.1). Denoting L_A as the sum of the different alien sequence lengths, we have $K \in O(L_A)$ even if K is much smaller than L_A in practice; therefore this first pre-computing step requires $O(L_A \log L_A)$ time complexity.

Given a read sequence of length L_R, AlienTrimmer proceeds its successive k-mer surveying following the same way as for alien sequences. However, for each read k-mer, AlienTrimmer searches whether it is present among the K alien ones. When a read k-mer only contains non-degenerate nucleotides (i.e. A, C, G and T), this search is directly performed by considering its b_2 coding, and looking in the pre-computed alien k-mer bitset B whether its corresponding bit is set to 1. In contrast, when a read k-mer contains at least one degenerate nucleotide, AlienTrimmer considers its b_4 coding, which is able to deal with such character state. By setting $b_4(\mathrm{N}) = 0000$, a k-mer containing degenerate nucleotides is compatible with another k-mer with no degenerate nucleotide if their respective binary coding w and w' verify the simple property $w \mid w' = w'$; indeed, $b_4(s) \mid b_4(s') = b_4(s')$ only when $s = s'$ or $s = \mathrm{N}$. When a read k-mer w contains at least one degenerate nucleotide, AlienTrimmer searches whether there exists one compatible alien k-mer w' among the K ones. This leads to a worst case $O(K)$ time complexity for each read k-mer with at least one degenerate nucleotide. However, as the K b_4-coded alien k-mers are sorted in Λ (see above), this step is accelerated by searching first the largest alien binary representation $w'_{\min}$ such that $w'_{\min} < w$. Recalling that $b_4(\mathrm{N}) = 0000$, the binary search algorithm (e.g. [21]) allows finding $w'_{\min}$ among the K alien k-mers in time $O(\log K)$. Reciprocally, by updating a second binary representation $\overline{w}$ from w with the coding $\overline{b}_4(\mathrm{N}) = 1111$, AlienTrimmer also searches for the lowest alien binary representation $w'_{\max}$ such that $\overline{w} < w'_{\max}$ with the binary search algorithm. Thanks to the two coding b_4 and $\overline{b}_4$ of N, if there exists inside Λ at least one alien binary representation w' compatible with the read one w containing at least one degenerate nucleotide, then one has $w'_{\min} \le w' \le w'_{\max}$, and AlienTrimmer searches for w' only between these two bounds. The two $O(\log K)$ binary searches of $w'_{\min}$ and $w'_{\max}$ do not modify the theoretical $O(K)$ computational complexity, but allow observing faster running times in practice than crudely testing compatibility of w with each of the K alien w' in Λ.

Using the previously described method, AlienTrimmer is able to determine every nucleotide indexes of a read where an exact alien k-mer match occurs. By using these nucleotide indexes, AlienTrimmer easily estimates the alien coverage within the read, i.e. the number of times each read nucleotide is covered by alien k-mers. The alien k-mer coverage can be directly computed together with the alien k-mer matching process without modifying the overall time complexity (see algorithm details in Supplementary materials S1.2). Moreover, this simple approach can be easily extended to quality-based trimming by incrementing by 1 the coverage value of every nucleotide supported by a low Phred score value. Finally, AlienTrimmer performs trimming by removing the prefix or suffix sub-sequences of the read when their alien k-mer coverage is higher than zero. Fig. 4 represents four examples of read contamination, with graphical representations of the alien coverage estimated for each nucleotide with $k = 10$, and the resulting trimmed alien residues.

Though fast and efficient when complete alien oligonucleotide sequences are present in at least one of both ends of a read (see Figs. 4A and B), this algorithm may not accurately perform its task when mismatches occur within the read ends to be trimmed. In practice, this situation corresponds to some single nucleotides with zero alien coverage occurring between nucleotides with non-zero alien coverage (see base 'A' with white background on Fig. 4C). Albeit infrequent when low quality nucleotides were trimmed off (see Section 2.2 and Fig. 3), such mismatches are easily accommodated by AlienTrimmer in the following manner. For 5′ end, AlienTrimmer first searches for the index i of the first nucleotide with non-zero alien coverage. Second, AlienTrimmer surveys the following alien covered nucleotides until a nucleotide of index j with zero alien coverage is reached. Given a specified number of mismatches m ($= \lceil k/2 \rceil$ by default), AlienTrimmer verifies whether the nucleotide at index $j + m$ is covered by an alien k-mer; if any, AlienTrimmer re-iterates its search of the next index j such that nucleotides at both indexes j and $j + m$ are not covered by any alien k-mer (see algorithm details in Supplementary materials S1.3). This approach allows identifying the sub-read defined between indexes i and j that is mainly covered by alien k-mers. If this sub-read is sufficiently close enough to the 5′ end (i.e. $j \ge 2i$), then AlienTrimmer removes the read prefix up to index j. The 3′ end trimming is performed following the same approach by starting from the last nucleotide index i with non-zero alien coverage, and performs backward searching of the index $j < i$ such that nucleotides at indexes j and $j - m$ are not covered by any alien k-mer.

Given a collection of alien sequences of total length L_A, decomposing and storing them into K distinct alien k-mers require $O(L_A \log L_A)$ time complexity (see above), which is negligible in comparison with the overall alien trimming process. Given a read of length L_R, AlienTrimmer is able to detect and remove alien sequences in 5′ and 3′ ends in time $\mathrm{O}(L_R)$ or $\mathrm{O}(L_R\ K)$ depending on the absence of degenerate nucleotide within the read or not, respectively. Knowing that $K \in O(L_A)$, the overall time complexity required by AlienTrimmer for processing each read of length L_R is $O(L_A \log L_A + L_R)$ in the best case scenario (i.e. no degenerate nucleotide), or $O(L_A(L_R + \log L_A))$ in the worst case scenario (i.e. presence of degenerate nucleotides within the read). Therefore, when dealing with standard HTS data with N reads with no degenerate nucleotide and N' reads with degenerate nucleotides, the overall time complexity required by AlienTrimmer is $O(L_A \log L_A + N\,L_R + N'\,L_A\,L_R)$.

Acknowledgments

We are grateful to the members of the platform Genotyping of Pathogens and Public Health (PF8) of Institut Pasteur (Paris) for fruitful discussions and extensive testing of the program AlienTrimmer. We also thank the anonymous reviewers for their comments, which led us to improve the manuscript. This work was supported by a grant from Région Ile-de-France.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.ygeno.2013.07.011.

References

[1] S. Lindgreen, AdapterRemoval: easy cleaning of next-generation sequencing reads, BMC Res. Notes 5 (2012) 337.
[2] M. Martin, Cutadapt removes adapter sequences from high-throughput sequencing reads, EMBnet J. 17 (2011) 10–12.
[3] M. Dodt, J.T. Roehr, R. Ahmed, C. Dieterich, FLEXBAR—flexible barcode and adapter processing for next-generation sequencing platforms, Biology 1 (2012) 895–905.
[4] Y. Kong, Btrim: a fast, lightweight adapter and quality trimming program for next-generation sequencing technologies, Genomics 98 (2011) 152–153.
[5] *Plasmodium falciparum* strain 3D7 reference genome version 2.1.5, ftp://ftp.sanger.ac.uk/pub/pathogens/Plasmodium/falciparum/3D7/3D7.version2.1.5/.
[6] H. Li, R. Durbin, Fast and accurate short read alignment with Burrows–Wheeler transform, Bioinformatics 25 (2009) 1754–1760.
[7] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, 1000 Genome Project Data Processing Subgroup, The sequence alignment/map (SAM) format and SAMtools, Bioinformatics 25 (2009) 2078–2079.
[8] CLC Genomics Workbench — a comprehensive and user-friendly analysis package for analyzing, comparing, and visualizing NGS data, http://www.clcbio.com/products/clc-genomics-workbench/.
[9] J.R. Miller, S. Koren, G. Sutton, Assembly algorithms for next-generation sequencing data, Genomics 95 (2010) 315–327.
[10] B. Ewing, P. Green, Base-calling of automated sequencer traces using phred. II. Error probabilities, Genome Res. 8 (1998) 186–194.
[11] J.A. Swets, R.M. Dawes, J. Monahan, Better decision through science, Sci. Am. 283 (2000) 82–87.
[12] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36.
[13] K. Nakamura, T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, S. Ishikawa, M.C. Linak, A. Hirai, H. Takahashi, Md Altaf-Ul-Amin, N. Ogasawara, S. Kanaya, Sequence-specific error profile of Illumina sequencers, Nucleic Acids Res. 39 (2011) e90.
[14] M.P. Cox, D.A. Peterson, P.J. Biggs, SolexaQA: at-a-glance quality assessment of Illumina second-generation sequencing data, BMC Bioinf. 11 (2010) 485.
[15] L. Smeds, A. Künstner, ConDeTri — a content dependent read trimmer for Illumina data, PLoS One 6 (2011) e26314.
[16] A.E. Minoche, J.C. Dohm, H. Himmelbauer, Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and Genome Analyzer systems, Genome Biol. 12 (2011) R112.
[17] X. Yu, K. Guda, J. Willis, M. Veigl, Z. Wang, S. Markowitz, M.D. Adams, S. Sun, How do alignment programs perform on sequencing data with varying qualities and from repetitive regions? BioData Min. 5 (2012) 6.
[18] P.J.A. Cock, C.J. Fields, N. Goto, M.L. Heuer, P.M. Rice, The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants, Nucleic Acids Res. 38 (2009) 1767–1771.
[19] M.A. Quail, M. Smith, P. Coupland, T.D. Otto, S.R. Harris, T.R. Connor, A. Bertoni, H.P. Swerdlow, Y. Gu, A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers, BMC Genomics 13 (2012) 341.
[20] Sickle — a windowed adaptive trimming tool for FASTQ files using quality, https://github.com/najoshi/sickle.
[21] D.E. Knuth, Sorting and searching, The Art of Computer Programming, vol. 3, Addison-Wesley Professional, Reading, MA, 1973.