Bioinformatics, 32(1), 2016, 136–139
doi: 10.1093/bioinformatics/btv524
Advance Access Publication Date: 9 September 2015
Applications Note

Sequence analysis

RNF: a general framework to evaluate NGS read mappers

Karel Břinda[1,*], Valentina Boeva[2,3,4] and Gregory Kucherov[1]

[1]LIGM/CNRS, Université Paris-Est, 77454 Marne-la-Vallée, France, [2]Inserm, U900, Bioinformatics, Biostatistics, Epidemiology and Computational Systems Biology of Cancer, 75248 Paris, France, [3]Institut Curie, Centre de Recherche, 26 rue d'Ulm, 75248 Paris, France and [4]Mines ParisTech, 77300 Fontainebleau, France

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on April 20, 2015; revised on July 31, 2015; accepted on August 31, 2015

Abstract

Motivation: Read simulators combined with alignment evaluation tools provide the most straightforward way to evaluate and compare mappers. Simulation of reads is accompanied by information about their positions in the source genome. This information is then used to evaluate alignments produced by the mapper. Finally, reports containing statistics of successful read alignments are created.

In default of standards for encoding read origins, every evaluation tool has to be made explicitly compatible with the simulator used to generate reads.

Results: To solve this obstacle, we have created a generic format Read Naming Format (RNF) for assigning read names with encoded information about original positions. Futhermore, we have developed an associated software package RNFTools containing two principal components. MIShmash applies one of popular read simulating tools (among DwgSim, Art, Mason, CuReSim, etc.) and transforms the generated reads into RNF format. LAVEnder evaluates then a given read mapper using simulated reads in RNF format. A special attention is payed to mapping qualities that serve for parametrization of Roc curves, and to evaluation of the effect of read sample contamination.

Availability and implementation: RNFTools: http://karel-brinda.github.io/rnftools Spec. of RNF: http://karel-brinda.github.io/rnf-spec

Contact: karel.brinda@univ-mlv.fr

1 Introduction

The number of Next-Generation Sequencing (NGS) read mappers has been rapidly growing during the last years. Then, there is an increasing demand of methods for evaluation and comparison of mappers to select the most appropriate one for a specific task. The basic approach to compare mappers is based on simulating NGS reads, aligning them to the reference genome and assessing read mapping accuracy using a tool evaluating if each individual read has been aligned correctly.

There exist many read simulators [Art (Huang *et al.*, 2011), CuReSim (Caboche *et al.*, 2014), DnEmulator (Frith *et al.*, 2012), DwgSim (http://github.com/nh13/dwgsim), FastqSim (Shcherbina, 2014), FlowSim (Balzer *et al.*, 2010), GemSim (McElroy *et al.*, 2012), Mason (Holtgrewe, 2010), PbSim (Ono *et al.*, 2013), Pirs (Xu *et al.*, 2012), SINC (Pattnaik *et al.*, 2014), WgSim (http://github.com/lh3/wgsim), Xs (Pratas *et al.*, 2014)] as well as many evaluation tools [CuReSimEval, DwgSim_Eval, RABeMa (Holtgrewe *et al.*, 2011), Seg-Suite (http://cbrc3.cbrc.jp/~martin/seg-suite/), WgSim_Eval, etc.]. However, each read simulator encodes information about the origin of reads in its own manner. This makes combining tools complicated and makes writing ad-hoc conversion scripts inevitable.

Here we propose a standard for naming simulated NGS reads, called Read Naming Format (RNF), that makes evaluation tools for read mappers independent of the tool used for read simulation. Furthermore, we introduce RNFTools, an easily configurable software, to obtain simulated reads in RNF format using a wide class of existing read simulators, and also to evaluate NGS mappers.

1.1 Simulation of reads

A typical read simulator introduces mutations into a given reference genome (provided usually as a FASTA file) and generates reads as genomic substrings with randomly added sequencing errors. Different statistical models can be employed to simulate sequencing errors and artefacts observed in experimental reads. The models usually take into account CG-content, distributions of coverage, of sequencing errors in reads and of genomic mutations. Simulators can often learn their parameters from an experimental alignment file.

At the end, information about origin of every read is encoded in some way and the reads are saved into a FASTQ file.

1.2 Evaluation of mappers

When simulated reads are mapped back to the reference sequence and possibly processed by an independent post-processing tool (remapping around indels, etc.), an evaluation tool inputs the final alignments of all reads, extracts information about their origin and assesses if every single read has been aligned to a correct location (and possibly with correct edit operations). The whole procedure is finalized by creating a summarizing report.

Various evaluation strategies can be employed (see, e.g. introduction of Caboche *et al.*, 2014). Final statistics usually strongly depend on the definition of a correctly mapped read, mapper's approach to deal with multi-mapped reads and with mapping qualities.

1.3 Existing read naming approaches

Depending on the read simulator, information about the read's origin is either encoded in its name, or stored in a separate file, possibly augmented with information about the expected read alignment. While WGSIM encodes the first nucleotide of each end of the read in the read name, DWGSIM and CURESIM encode the leftmost nucleotide of each end. Unfortunately, these read naming schemes were specifically designed for particular sequencing technologies and single evaluation strategies, therefore they are not suitable as generic formats. ART produces SAM and ALN alignment files, MASON creates SAM files and PIRS makes text files in its own format.

2 Methods

We have created RNF, a standard for naming simulated reads. It is designed to be robust, easy to adopt by existing tools, extendable, and to provide human-readable read names. It respects a wide range of existing sequencing technologies as well as their possible future evolution (e.g. technologies producing several 'subreads'). We then developed a utility for generating RNF-compliant reads using existing simulators, and an associated mapper evaluation tool.

2.1 Read naming format (RNF)

2.1.1 Read tuples

Read tuple is a tuple of sequences (possibly overlapping) obtained from a sequencing machine from a single fragment of DNA. Elements of these tuples are called *reads*. For example, every 'paired-end read' is a *read tuple* and both of its 'ends' are individual *reads* in our notation.

To every *read tuple*, two strings are assigned: a short read name (SRN) and a long read name (LRN). SRN contains a hexadecimal unique ID of the *tuple* prefixed by '#'. LRN consists of four parts delimited by double-underscore: (i) a prefix (possibly containing expressive information for a user or a particular string for sorting or randomization of order of *read tuples*), (ii) a unique ID, (iii) information about origins of all segments (see below) that constitute *reads* of the *tuple*, (iv) a suffix containing arbitrary comments or extensions (for holding additional information). Preferred final read names are LRNs. If an LRN exceeds 255 (maximum allowed read length in SAM), SRNs are used instead and a SRN–LRN correspondence file must be created.

(a)

```
Coor        12345678901-2345678901234567890123456789

Source 1 - reference genome
chr 1       ATGTTAGATAA-GATAGCTGTGCTAGTAGGCAGTCAGCCC
chr 2       ttcttctggaa-gaccttctcctcctgcaaataaa

Source 2 - generator of random sequences

READS:
r001        ATG-TAGATA ->
r002/1         TTAGATAACGA ->
r002/2                                  <- TCAG-CGGG
r003/1                               tgcaaataa ->
r003/2              gaa-gacc-t ->
r004                       ATAGCT............TCAG ->
r005                                 GTAGG ->
             <- agacctt
                        <- TCGACACG
r006       ATATCACATCATTAGACACTA
```

Simulated reads

(b)

r. tuple	LRN	SRN
r001	sim__1__(1,1,F,01,10)__[single-end]	#1
r002	sim__2__(1,1,F,04,13),(1,1,R,31,39)__[paired-end]	#2
r003	sim__3__(1,2,F,09,17),(1,2,F,25,33)__[mate-pair]	#3
r004	sim__4__(1,1,F,15,36)__[RNA-seq-spliced],C:[6=12N4=]	#4
r005	sim__5__(1,1,R,15,22),(1,1,F,25,29),(1,2,R,05,11)__[chimeric]	#5
r006	rnd__6__(2,0,N,00,00)__[rand-contamin]	#6

Long and short read names

Fig. 1. Examples of simulated reads (in our definition *read tuples*) and their corresponding RNF names, which can be used as read names in the final FASTQ file: a single-end read (r001); a paired-end read (r002); a mate-pair read (r003); a spliced RNA-seq read (r004); a chimeric read (r005); and a random contaminating read with unspecified coordinates (r006)

2.1.2 Segments

Segments are substrings of a *read* which are spatially distinct in the reference and correspond to individual lines in a SAM file. Since spliced RNA-seq *reads* (Fig. 1, r004) are usually reported in single lines in SAM, we recommend to keep them in single RNF segments without splitting even though they might be considered spatially distinct. Thus, each *read* has an associated chain of *segments* and we associate a *read tuple* with *segments* of all its *reads*.

Within our definition, a 'single-end read' (Fig. 1, r001) consists of a single *read* with a single *segment* unless it comes from a region with genomic rearrangement. A 'paired-end read' or a 'mate-pair read' (Fig. 1, r002 and r003) consists of two *reads*, each with one *segment* (under the same condition). A 'strobe read' consists of several *reads*. Chimeric *reads* (i.e. reads corresponding to a genomic fusion, a long deletion, or a translocation; Fig. 1, r005) have at least two *segments*.

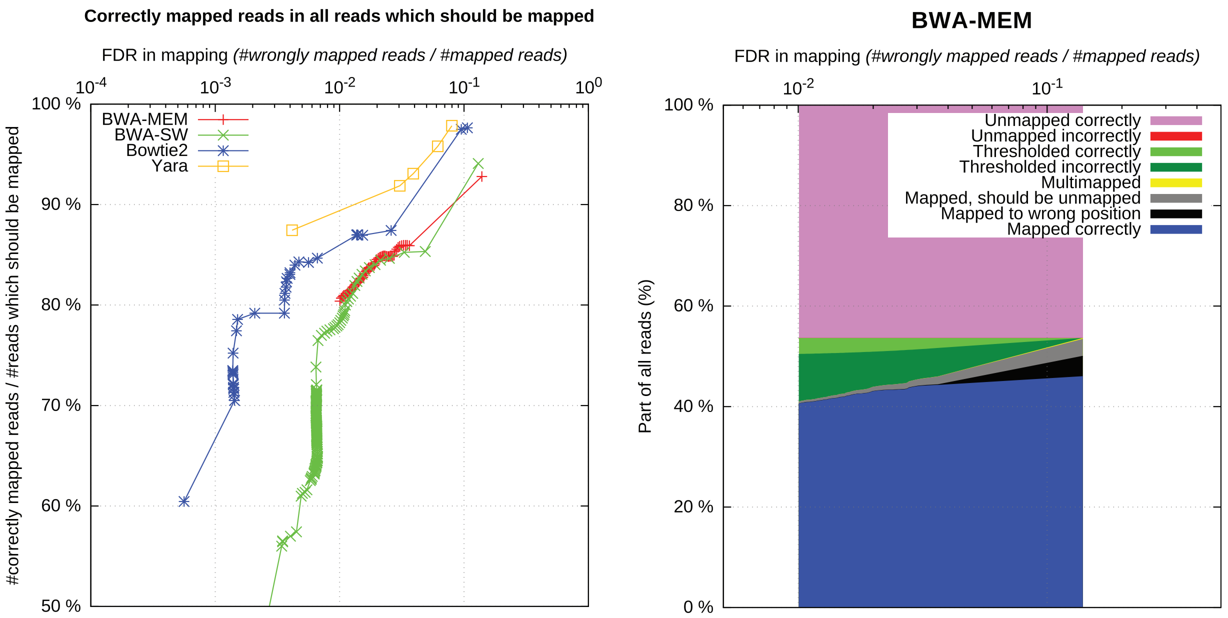

Fig. 2. Example of two graphs produced by LAVEnder as a part of comparison of mapper capabilities of contamination detection. 200.000 single-end reads were simulated from human and mouse genomes (100.000 from HG38, 100.000 from MM10) by DwgSim using MISHmash and mapped to HG38. All LAVEnder graphs have false discovery rate on *x*-axis and use mapping quality as the varying parameter for plotted curves. This experiment reveals that YARA copes with contamination better than Bowtie2, BWA-MEM and BWA-SW

For each *segment*, the following information is encoded: leftmost and rightmost 1-based coordinates in its reference, ID of its reference genome, ID of the chromosome and the direction ('F' or 'R'). The format is:

```
(genome_id,chromosome_id,direction, L_coor,R_coor).
```

Segments in LRN are recommended to be sorted with the following keys: `source`, `chromosome`, `L_coor`, `R_coor`, `direction`. When some information is not available (e.g. the rightmost coordinate), zero is used ('N' in case of direction; Fig. 1, r006).

2.1.3 Extensions

The basic standard can be extended for specific purposes by extensions. They are part of the suffix and encode supplementary information (e.g. information about CIGAR strings, sequencing errors, or mutations).

2.2 RNFtools

We also developed RnfTools, a software package associated with Rnf. It has two principal components: MISHmash for read simulation and LAVEnder for evaluation of NGS read mappers. RnfTools has been created using SnakeMake (Köster and Rahmann, 2012), a Python-based Make-like build system. All employed external programs are installed automatically when needed. The package also contains a lightweight console tool `rnftools` which can, in addition, be used for conversion of existing data and transformation of RNF coordinates using a LiftOver chain file.

MISHmash is a pipeline for simulating reads using existing simulators and combining obtained sets of reads together (e.g. to simulate contamination or metagenomic samples). Its output files respect Rnf format, therefore, any Rnf-compatible evaluation tool can be used for evaluation.

LAVEnder is a program for evaluating mappers. For a given set of Bam files, it creates an interactive Html report with several graphs. In practice, mapping qualities assigned by different mappers to a given read are not equal (although mappers tend to unify this). Moreover, even for a single mapper, mapping qualities are very data-specific. Therefore, results of mappers after the same thresholding on mapping quality are not comparable. To cope with this, we designed LAVEnder to use mapping qualities as parameterization of curves in 'sensitivity-precision' graphs (like it has been done in Li (2013)). Examples of output of LAVEnder can be found in Figure 2.

3 Conclusion

We designed Rnf format and propose it as a general standard for naming simulated Ngs reads. We developed RnfTools consisting of MISHmash, a pipeline for read simulation, and LAVEnder, an evaluation tool for mappers, both following the Rnf convention (thus inter-compatible). Currently, MISHmash has a built-in interface with the following existing read simulators: Art, CuReSim, DwgSim, Mason and WgSim.

We expect that authors of existing read simulators will adopt Rnf naming convention as it is technically simple and would allow them to extend the usability of their software. We also expect authors of evaluation tools to use Rnf to make their tools independent of the used read simulator.

Funding

This work was supported by ABS4NGS grant and by Labex Bézout of the French government (program *Investissement d'Avenir*).

Conflict of Interest: none declared.

References

Balzer,S. *et al.* (2010) Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim. *Bioinformatics*, **26**, i420–i425.

Caboche,S. *et al.* (2014) Comparison of mapping algorithms used in high-throughput sequencing: application to Ion Torrent data. *BMC Genomics*, **15**, 264.

Engström,P.G. *et al.* (2013) Systematic evaluation of spliced alignment programs for RNA-seq data. *Nat. Methods*, **10**, 1185–1191.

Frith,M.C. *et al.* (2012) A mostly traditional approach improves alignment of bisulfite-converted DNA. *Nucleic Acids Res*, **40**, e100.

Holtgrewe,M. (2010) Mason—a read simulator for second generation sequencing data. Technical Report TR-B-10-06, Institut für Mathematik und Informatik, Freie Universität Berlin.

Holtgrewe,M. *et al.* (2011) A novel and well-defined benchmarking method for second generation read mapping. *BMC Bioinformatics*, **12**, 210.

Huang,W. *et al.* (2011) ART: a next-generation sequencing read simulator. *Bioinformatics*, **28**, 593–594.

Hu,X. *et al.* (2012) pIRS: profile-based Illumina pair-end reads simulator. *Bioinformatics*, **28**, 1533–1535.

Köester,J. and Rahmann,S. (2012) Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, **28**, 2520–2522.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Li,H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv:1303.3997v2* [q-bio.GN].

McElroy,K.E. *et al.* (2012) GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics*, **13**, 74.

Ono,Y. *et al.* (2013) PBSIM: PacBio reads simulator–toward accurate genome assembly. *Bioinformatics*, **29**, 119–121.

Pattnaik,S. *et al.* (2014) SInC: an accurate and fast error-model based simulator. *BMC Bioinformatics*, **15**, 40.

Pratas,D. *et al.* (2014) XS: a FASTQ read simulator. *BMC Research Notes* **7**:40.

Shcherbina,A. (2014) FASTQSim: platform-independent data characterization and in silico read generation for NGS datasets. *BMC Res. Notes*, **7**, 533.