Sequence analysis

RNF: a general framework to evaluate NGS read mappers

Karel Brinda1,*, Valentina Boeva2,3,4 and Gregory Kucherov1

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


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 (Sherbina, 2014), FLOWSim (Balzer et al., 2010), GenSIM (McElroy et al., 2012), MASON (Holtgrewe, 2010), PaSim (Ono et al., 2013), PBRS (Xu et al., 2012), SINC (Pattnaik et al., 2014), WuSIM (http://github.com/lh3/wgsm), Xc (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/), WuSIM_ 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 GC-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 another 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 their origin and assessed 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.

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 and single evaluation strategies, therefore they are not suitable as generic formats. ART produces SAM and ALN alignment files, MASON creates SAM files and PARS makes text files in its own format.

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 reads tuples), (ii) a unique ID, (iii) information about origin 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 SRNs. If an LRN exceeds 253 (maximum allowed read length in SAM), SRNs are used instead and a SRN–LRN correspondence file must be created.

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 SRN 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.
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:

\[(\text{genome_id}, \text{chromosome_id}, \text{direction}, \text{L_coor}, \text{R_coor})\]

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 (Koster 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


