JACUSA2 manual

Michael Piechotta
michael.piechotta@gmail.com

9th, October, 2019

Contents

1 Introduction

JAVA framework for accurate Variant assessment (JACUSA2) is a one-stop solution to detect single nucleotide variants (SNVs) and reverse transcriptase induced arrest events in Next-generation sequencing (NGS) data.

`https://github.com/dieterich-lab/JACUSA2/`JACUSA2 is a direct successor of `https://github.com/dieterich-lab/JACUSA/`JACUSA1 — JACUSA1 is hereby deprecated and won't be continued. All methods from JACUSA1 (*call-1*, *call-2*, and *pileup*) are available in JACUSA2.

The new release offers great performance enhancements (3 faster) for existing methods and adds new methods (*rt-arrest* and *lrt-arrest*) to identify read arrest events by means of comparing read through and read arrest counts. *lrt-arrest* is a combination of *call-2* and *rt-arrest* that allows to identify such linked variant and arrest events. Robust identification of variants has proven to be a daunting

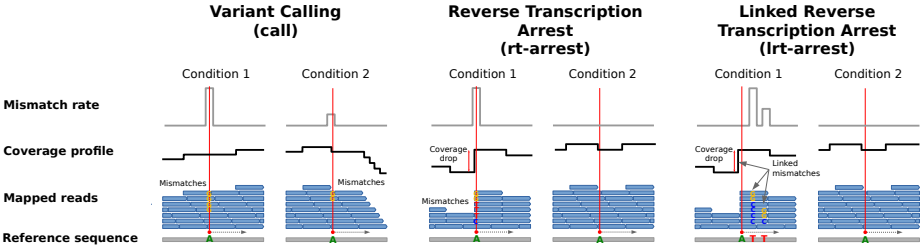


Figure 1: Schematic summary of JACUSA2 methods and underlying characteristics of data.

task due to artefacts specific for NGS-data and employed mapping strategies. We implement various artefact/feature filters that reduce the number of false positives (see section **??**). A new feature filter has been added to JACUSA2 to filter candidate variants or arrest events by providing a file with sites to be excluded. Some artefact filters have been moved from JACUSA1 to the R helper package called JACUSA**2**helper[1].

Another new feature, are optional data that can be added to the output of some methods. Such optional data are INDEL counts and associated differential statistics and read substitution information. A user defined base substitution can be used to partition reads into two sets .

JACUSA2 employs a window-based approach to traverse BAM [Li et al., 2009] files, featuring highly parallel processing and utilizing the htsjdk[2] framework.

1.1 Variant calling

JACUSA2 has been extensively evaluated and optimized to identify RNA editing sites in RNA-DNA and RNA-RNA sequencing samples (see Figure 2). Checkout the original publication and the supplementary material of JACUSA1 [Piechotta et al., 2017] if you are interested in details regarding the test-statistic.

[1] Check `https://github.com/dieterich-lab/JACUSA2helper`
[2] Check `https://github.com/samtools/htsjdk`

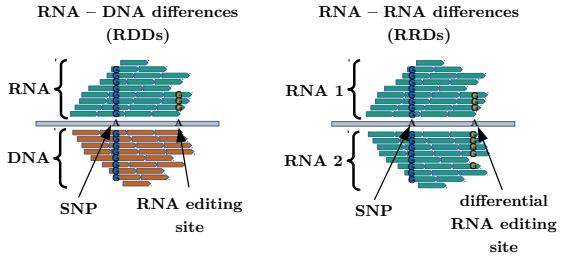


Figure 2: Identification of RNA-editing sites in JACUSA2

1.2 Reverse transcriptase arrest events

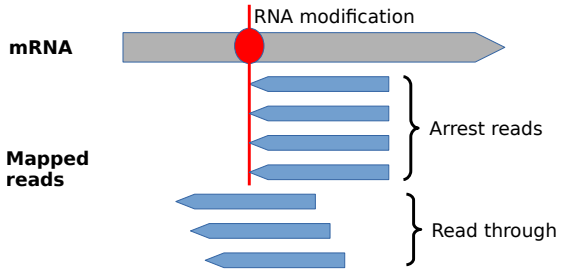


Figure 3: Schematic depiction of arrest events that have been induced by modifications that influence reverse transcription during library preparation

Reverse transcriptase arrest events can be induced during library preparation (see Figure 3). They are identified by reads that exhibit shorter than expected read length due to premature termination during first strand synthesis. For each site, a vector of read through and read arrest counts is constructed and modelled with a Beta-Binomial distribution. We estimate the parameters of the distribution with the method presented by Minka[3].

2 Download

The latest version of JACUSA2 can be obtained from `https://github.com/dieterich-lab/JACUSA2`.

Go to `https://github.com/dieterich-lab/JACUSA2/releases` and pick the lastest release.

2.1 Installation and requirements

JACUSA2 does not need any configuration but needs a correctly configured Java environment. We developed and tested JACUSA2 with Java v1.8. If you encounter any Java related problems please consider to change to Java v1.8.

[3]Check `https://tminka.github.io/papers/dirichlet/minka-dirichlet.pdf`

2.2 Migrating from JACUSA1 to JACUSA2

There are several important changes to the command line interface:

- ONLY single dash "-" options, e.g.: "-c 10" are supported. ALL two dash options "–option [...]" from JACUSA1 have been removed.
- Use "**-**filterNH" and "**-**filterNM" instead of "–filterNH" and "–filterNM".
- CLI option to provide library type has changed: JACUSA1: "-P Lib1,Lib2" → JACUSA2: "-P1 Lib1 -P2 Lib2".
- Artefact/feature filter has now named options, e.g.: JACUSA1: "-a H:1" → JACUSA2: "-a H:**condition=1**". Check Section 8.8 for details.
- Test-Statistic options "-u <STAT>" have named options.

A ##" prefixed header line has been added to the default output format of JACUSA2. The header line contains version and command line info. There is also a new version of JACUSA**2**helper[4] to support downstream analysis of JACUSA2 output. JACUSAhelper has been declared deprecated and won't be maintained anymore.

2.3 *In silico* and example data

2.3.1 Variant calling

You can choose between different *in silico* setups to detect variants.

The gDNA vs. cDNA represents the typical data setup that is encountered in detection of RNA editing sites via comparing genomic and transcriptomic sequencing reads. In this setup, variants have been only imputed to the cDNA BAM file.

The cDNA vs. cDNA data setup can be interpreted as representing allele specific expression of single variants or differential RNA editing. In this setup, variants with pairwise different base frequencies have been imputed into both cDNA BAM files. Additionally, to make the identification of variants more challenging, SNPs with pairwise similar base frequencies have been included to both BAM files. SNPs should not be identified as true positive sites.

gDNA data has been simulated with art[5] and cDNA reads have been simulated with flux simulator[6]. Read simulations have been restricted to the corresponding first chromosome of human. Simulated BAM files have been processed and only reads with mapping quality ≥ 20 have been retained. Check Table 1 for details about contents of available data.

:

[4]Check `https://github.com/dieterich-lab/JACUSA2helper/`
[5]Check `http://www.niehs.nih.gov/research/resources/software/biostatistics/art`
[6]Check `http://sammeth.net/confluence/display/SIM/Home`
[7]gDNA vs. cDNA: `https://data.dieterichlab.org/s/gDNA_VS_cDNA.tar.gz`
[8]cDNA vs. cDNA: `https://data.dieterichlab.org/s/cDNA_VS_cDNA.tar.gz`

Table 1: Detailed description of available *in silico* data for variant calling. Data has been simulated on human chromosome 1 using hg19.

Setup	File	Description
gDNA vs. cDNA [7]	gDNA.bam	Simulated genomic DNA
	cDNA.bam	Simulated RNA-Seq data
	variants.txt	Coordinates of imputed variants and their target and sampled frequencies.
gDNA vs. cDNA [8]	cDNA_1.bam	Simulated RNA-Seq data for 1st condition.
	cDNA_2.bam	Simulated RNA-Seq data for 2nd condition.
	variants.txt	Coordinates of imputed variants and their target and sampled frequencies.
	snps.txt	Coordinates of imputed SNPs. In both BAM files matching SNPs have the same target frequency but different effective or sampled frequencies. The shape parameter determines how much the sampled frequency will deviate from the target frequency. The suffixes: 1 and 2 correspond to the respective BAM file.

2.3.2 Reverse transcriptase arrest events

We have downloaded primary sequencing data from [Zhou et al., 2018] and mapped them according to . From the resulting read sets, we retained only read uniquely mapping to 18S and 28S rRNAs with mapping quality ≥ 20. [Zhou et al., 2018] map RNA modification of pseudouridine (Ψ) by chemically modifying pseudouridines with carbodiimide (+CMC) and detecting arrest events that are induced by reverse transcription stops in high-throughput sequencing under 3 different conditions (HIVRT, SIIIMn, and SIIIMg).

The archive `https://data.dieterichlab.org/s/arrest_events.tar.gz` consists of:

***.bam, *.bam.bai** BAM files and BAM indicices for different 3 conditions HIVRT, SIIIRTMg, and SIIIRTMn each treated with CMC(+CMC) or mock-treated(-CMC) (6 BAM + 6 BAI files).

fournier_db.txt Parsed and coordinate adjusted list of known modifications for human 18S and 28S according to Fournier lab's 3D rRNA modification maps database [Piekna-Przybylska et al., 2007].

README.txt Summary of referenced data and original data sources. Read for details.

3 Input/Output

All JACUSA2 methods require sorted and indexed BAM[9] files. BAM is a standardized file format for efficient storage of alignments. Furthermore, JACUSA2

[9] Check `https://samtools.github.io/hts-specs/SAMv1.pdf`

requires that the reference sequence is available either through the "MD"-tag [10] in BAM files or by providing the reference sequence in indexed FASTA format with the command line option "-R <reference.fasta>". The "MD" field contains mismatch information that allows to perform variant calling without providing the reference sequence.

Check the manuals of: SAMtools/BCFtools[11] and/or picard tools[12] for how to use the respective tool to convert your alignment files to valid JACUSA2 input BAM.

3.1 Processing BAM files

In the following, commands for SAMtools[13] are presented.

To sort and index your raw BAM files, perform the following sequence of commands:

SAM $\rightarrow BAM$ `samtools view -Sb mapping.sam > mapping.bam`

sort BAM `samtools sort mapping.bam mapping.sorted`

index BAM `samtools index mapping.sorted.bam`

Check your BAM file for the "MD"-tag if you want to provide reference sequence information via this tag. When your BAM files do not have the "MD" tag properly set, use SAMtools:

```
samtools calmd mapping.sorted.bam reference.fasta > mapping.sorted.MD.bam
```

3.1.1 Remove duplicates for variant calling

It is a recommended pre-processing step to remove duplicated reads when identifying variants - **omit this step for *rt-arrest* and *lrt-arrest***. Reads that are terminated prematurely during library preparation will falsely be identified as PCR duplicates and removed from the final output. This will dramatically degrade sensitivity for arrest event idetification.

Duplicated reads occur mostly due to PCR-artefacts. They are likely to harbour false variants and most statistical test require independently sampled reads. In the following, commands for picard tools are presented:

```
java -jar MarkDuplicates.jar \
 I=mapping.sorted.bam O=dedup_mapping.sorted.bam \
 M=duplication.info
```

Most tools either remove duplicated reads from the final output or assign 1024 to the flag[14] field of those reads. In the later case, invoke JACUSA2 with the additional command line option "-F 1024" to filter reads that have been marked as duplicated (See Section 8.7.5).

[10] Check `https://samtools.github.io/hts-specs/SAMtags.pdf`
[11] Check `http://samtools.sourceforge.net/`
[12] `http://broadinstitute.github.io/picard/`
[13] Check `http://www.htslib.org`
[14] Check `https://broadinstitute.github.io/picard/explain-flags.html`

Table 2: Definiton of an exemplary BED-like search region

contig	start	end
1	1000	1100
2	10000	10000

3.1.2 Library type and strand information

JACUSA2 supports stranded paired end (PE) and single ends (SE) reads. **Warning:** the CLI option has changed (see Section 2.2)!

With the command line parameter "-P <LIBRARY-TYPE> " or " -P1 <LIBRARY-TYPE> -P2 <LIBRARY-TYPE>" the user can choose from the following supported library types:

RF-FIRSTSTRAND STRANDED library - first strand sequenced,

FR-SECONDSTRAND STRANDED library - second strand sequenced, and

UNSTRANDED UNSTRANDED library.

The "UNSTRANDED" is not available for *rt-arrest* and *lrt-arrest* method because an arrest site can not be unambiguously defined for this library type. Read base substitution option "-B <BASE-SUB>" requires a stranded library type in order to corretly identify base substitutions based on strand information.

3.2 BED-like search region

Identification of interesting sites can be restricted to specific regions of the genome or transcriptome. Provide a minimalistic BED-like file to limit the search to some region(s) or site(s). Complementary region(s) of the BAM files will not be considered.

In the following file, the search is confined to a 100nt region on contig 1 starting at position 1,000 and a single site on contig 2 at position 10,000: Many individual sites may impair running performance of JACUSA2. Try to merge nearby sites to create contiguous regions and extract specific sites from JACUSA2 output with bedtools[15] "intersect":

merge sites
```
bedtools merge -d 500 singular_sites.bed > \
    contigous_regions.bed
```

run JACUSA2
```
java -jar JACUSA2.jar call-2 -b contigous_regions.bed \
    -r JACUSA2.out mapping_1.sorted.bam mapping_2.sorted.bam
```

extract sites
```
bedtools intersect -wa -a JACUSA2.out -b singular_sites.bed
```

3.3 General output format

JACUSA2 writes its output to a user specified file. When using multiple threads, JACUSA2 will create a temporary file for each allocated thread in the temp

[15] `http://bedtools.readthedocs.org/en/latest/`

directory that is provided by the JAVA Virtual Machine. Check the manual of your JAVA Virtual machine on how to change the temp directory.

Chosen command line parameters and current genomic position are printed to the command prompt and serve as a status guard. Furthermore, depending on the provided command line parameters, JACUSA2 will generate a file with sites that have been identified as potential artefacts when "-s" is provided.

Output format of JACUSA2 is controlled by the "-f <FORMAT>" command line option. Support for output formats depends on the used method. Check Table 3 for a summary of currently supported output formats. The default output

Table 3: Summary of available output formats for each method.

	Method			
Output format	*call-1,2*	*pileup*	*rt-arrest*	*lrt-arrest*
JACUSA2 BED-like format	x	x	x	x
Variant Call Format (VCF[16])	x	x		

format is combination of BED6[17] with JACUSA2 methods specific columns and common info column: "info", "filter_info", and "ref_base".

A "##" prefixed header that contain JACUSA2 runtime specific info such as version info and command line options is added to the default output format. The actual number of columns depends on the JACUSA2 method and the number of provided BAM files.

Check table 4 for a general description of JACUSA2 output format and check method or option specific adjustments in the following sections:

- *rt-arrest* method **??**,
- *lrt-arrest* method **??**,
- optional INDEL data fields **??**, or
- base substitution based read stratification **??**.

Table 4: General description of JACUSA2 default output format with exemplary *call-2* output — "N" corresponds to the total number of columns. Check detailed explanation of columns under the table.

	BED6 columns						**Method specific**			**Additional**	
Column:	1	2	3	4	5	6	7	...	N-2	N-1	N
Example:	1	100	101	call	8.07	-	0,0,0,10	...	0,2,0,10	*	*
				...				...			...

(1, 2, 3) contig + start + end 0-indexed coordinates $[start, end)$.

(4) name String that depends on method. Currently has no use except to ensure BED6 compatibility.

(5) score Score for this site. "*" if not available. Depends on actual method.

[16]Check `http://samtools.github.io/hts-specs/VCFv4.1.pdf`
[17]Check `http://genome.ucsc.edu/FAQ/FAQformat.html#format1`

(6) strand Possible values are: ".", "+", and "-" which correspond to "unstranded", "positive strand", and "negative strand", respectively. If strand is "= ".", then columns with base call information will be showing base counts **according** to the strand — inverted base count if on the "negative strand".

(7 — N-3) method specific The number of base columns depends on the JACUSA2 method and the number of BAM files — check method specific explanation.

(N-2) info Additional info for this site. Actual content depdents on method and command line parameters. Details about the parameter estimation of the underlying distribution can be shown, insertion/deletion counts, and additional method specific data. If nothing provided, set to "*". The "info" field is used to add optional site specific data without changing the total number of columns.

(N-1) filter_info Relevant, if artefact/feature filter(s) X have been provided with "-a X" on the command line. The column will contain a comma-separated list of artefact/feature filters that recognised this site. Will be "*" if empty.

(N) ref_base The reference base according to coordinats and (6) strand information — Inverted if on the "negative strand".

4 Artefact/feature filter

False positive variant calls can be related to mapping artefacts and sequencing technology. Short read mappers tend to produce incorrect alignments around INDEL positions that may be falsely identified as variant sites. Other false variant calls originate from uneven base call error distributions along short reads. We have implemented a panel of simple threshold based artefact filters. Beyond artefacts, we have added filters that identify some features of a pileup or a read and can be used to mark those sites.

When a site is identified by an artefact or feature filter X, X is added to the "filter_info" column and optionally this site can be moved to an other output file when command line "-s [FILTERED-OUTPUT]" has been used. This strategy presevers all sites and allows the user investigate the fraction of filtered vs. total calls. In the following, we will use artefact and feature filtered interchangeably and a site we will call a site removed by a feature filter albeit the site is marked by the ID of feature filter.

When identifying RDDs by comparing gDNA vs. cDNA, it is common practice to remove or mask sites that are homozygous in gDNA (feature filter H) and have more than three distinct base types (artefact filter M).

We have added the exclude site filter in JACUSA2 that allows to mark sites in the final output as overlapping with sites/regions that are contained in a file. The supported file type are: VCF, BED, or JACUSA2 output. Given a set of known SNPs, polymorphic sites can be directly marked and removed from the

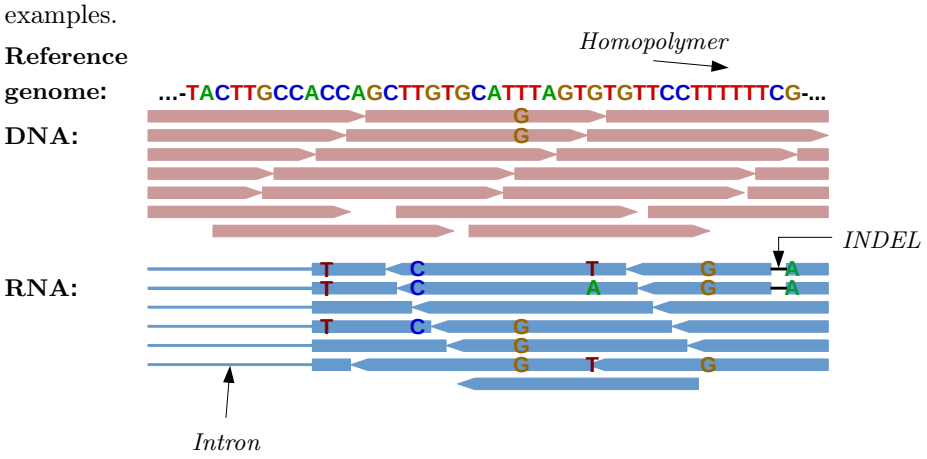
list of candidate RNA-editing sites are identified by JACUSA2 in gDNA vs. cDNA or cDNA vs. cDNA comparisons:

```
java -jar JACUSA2.jar -a E:file=snps.vcf:type:VCF -r JACUSA2.out cDNA-1.bam cDNA-2.bam
```

For performance reasons, the input file is required to be sorted by coordinate — **WARNING:** Sort order is not tested within JACUSA2.

Our filters (D,B,I,Y) monitor the distance d of a given candidate site to relevant read features such as start/end, INDEL positions, homopolymeric regions, and splice sites and remove the candidate site from further consideration if a proportion r of all reads falls below the given distance cutoff $\leq d$. In JACUSA2

Figure 4: Summary of available artefact/feature filters for each method and examples.



JACUSA filter: **D** **D, B** **H** **M** **Y** **I**

Value	Artefact/feature Filter Description	JACUSA2 method *call-1*	*call-2*	*pileup*	*rt-arrest*	*lrt-arrest*
	Filter potential false positive variants adjacent to:					
B	- read start/end	✓	✓	✓		
I	- INDEL position(s)	✓	✓	✓	✓	✓
S	- splice site(s)	✓	✓	✓	✓	✓
D	Combines Filters:					
	- I + B + S	✓	✓	✓		
	- I + S				✓	✓
Y	Filter wrong variant calls within homopolymers	✓	✓	✓	✓	✓
M	Max allowed alleles per site	✓	✓	✓	✓	✓
H	Filter non-homozygous sites in condition 1 or 2		✓	✓	✓	✓
E	Exclude sites contained in a file	✓	✓	✓	✓	✓

we have included structured sites meaning that linked data can be stored in the output across multiple lines. Filters are assigned to the top level site — on derieved lines the filter_info column will be equal to the top level site.

5 Variant detection

5.1 Identification of RNA editing sites

In order to identify RNA editing sites by comparing gDNA and *stranded* RNA-Seq (single or paired end) use:

first strand sequenced "-P1 UNSTRANDED -P2 RF-FIRSTSTRAND"

second strand sequenced "-P2 UNSTRANDED -P2 FR-SECONDSTRAND"

. When your RNA-Seq is unstranded use: "-P1 UNSTRANDED -P2 UNSTRANDED" and infer the correct orientation from annnotation.

Use the following command line to identify RNA-DNA differences in BAM files that might give rise to RNA editing sites:

```
java -jar JACUSA2.jar call-2 -r JACUSA2.out -s -a H:condition=1 gDNA.bam cDNA.bam
```

Option "-a H:condition=x" ensures that potential polymorphisms in gDNA will be eliminated as artefacts. The number $x \in \{1, 2\}$ determines which sample is required to be homomorph — in this case: gDNA.bam.

Use the following command line to identify RNA-DNA differences:

```
java -jar call-2 -r JACUSA2.out -s cDNA_1.bam cDNA_2.bam
```

WARNING: If you want to identify RNA-RNA differences make sure NOT to use the filter "-a H[...]"! Otherwise, potential valid variants will be filtered out. If you happen to have a list of known polymorphic sites in a file "snps.{vcf|bed|JACUSA2 output}" add the exclude site filter to the previous statemet:

```
java -jar call-2 -a E:file=snps.vcf:type=VCF -r JACUSA2.out -s cDNA_1.bam cDNA_2.bam
```

Polymorphic site defined in "snps.vcf" will be marked in the output file 'JACUSA2.out'.

5.2 *call-** output format

Test-statistic $z \in \mathbb{R}$ that indicates the likelihood that this is a true variant. Higher number indicates a higher likelihood for a variant.

5.3 *pileup* output format

The output format of *pileup* is derieved from the output format of *call-**. The only modification is that the "score" column corresponds to the total coverage of a site (see Section 5.2 for further details).

6 Reverse transcriptase arrest events

6.1 *rt-arrest* output format

In order to represent arrest events in JACUAS2 output, the base vector colum "basesij" is split into two columns "arrest_basesij" and "through_basesij" (i encodes the condition and j the replicate. The former column represents base

calls from reads that exhibit premature termination and the later corresponds to base calls from reads that show NO premature termination.

6.2 *lrt-arrest* output format

7 Usage

Calling JACUSA2 without any arguments will print the available tools which currently are:

```
java -jar JACUSA2.jar
  METHOD        DESCRIPTION
  call-1        Call variants - 1 condition
  call-2        Call variants - 2 conditions
  pileup        SAMtools like mpileup (2 conditions)
  rt-arrest     Reverse Transcription Arrest - 2 conditions
  lrt-arrest    Linkage arrest to base substitution - 2 conditions
Version:  [...]
Libraries:  [...]
```

7.1 call-1

Single sample (call-1) allows to call variants against a reference. Internally, an *in silico* sample is created from information that is provided by the "MD" field in BAM files.

The number of base columns depends on the number of BAM files. In basesIJ: I corresponds to sample and J to the respective replicate. Numbers indicate the base count of the following base vector: (A, C, G, T)

Sites that have a $>$ alleles are considered candidate variant sites and for this sites a test-score will be computed.

7.2 call-2

7.3 pileup

See "Call variant - two samples" for details.

7.4 rt-arrest - 2 conditions

In this method base call counts of arrest and read through reads are modelled by a Beta-Binomial distribution and differences between conditions are to be identified by means of a likelihood-ratio test. Subsequent approximiation with χ^2 distribution to compute a pvalue.

Sites are considered candidate arrest sites, if in all BAM files there is at least one read through AND one read arrest event. Furthermore, coverage filter and minBASQ of Base Call apply that will affect the output.

7.5 lrt-arrest - 2 conditions

lrt-arrest allows to link pileups to their arrest position. Output consists of read arrest and read through counts and a references to the associated arrest positions. There are cases, where currently an arrest position cannot be defined, e.g.: non properly paired reads. Output consits of at least one line. Each separate arrest position adds an additional row is The first row contains the unstratified data or total, the "arrest_pos" column is set to "*". Any following sites with identical coordinates (contig, start, end, strand) will have a different arrest position reference in the "arrest_pos" column.

This method supports partial artefact filtering. Currently, filters only apply to the unstratified data — sites with "*" in in "arrest_pos". Furthermore, coverage filter and minBASQ of Base Call apply that will affect the output.

8 Description of command line options

8.1 Input / Output

8.1.1 Input BAM files

8.1.2 Output file

-r RESULT-FILE	results are written to RESULT-FILE	call-1 call-2 pileup rt-arrest lrt-arrest

8.1.3 Output artefacts to separate file

-s	Store feature-filtered results in another file (= RESULT-FILE.filtered)	call-1 call-2 pileup rt-arrest lrt-arrest

8.2 Input BED file

-b BED	BED file to scan for variants	call-1 call-2 pileup rt-arrest lrt-arrest

8.3 Reference fasta file

-R REF-FASTA	use reference FASTA file (must be indexed)	call-1 call-2 pileup rt-arrest lrt-arrest

8.4 Library type

-P LIB-TYPE	multirow3 multirow2 lrt	call-1 call-2 pileup rt-arrest lrt-arrest

8.5 Read base changes

-B READ-SUB	Count non-reference base substitution per read and stratify. Requires stranded library type. (Format for T to C mismatch: T2C; use ',' to separate substitutions) Default: none	call-1 call-2 pileup rt-arrest

8.6 Show deletion score

-D	Show deletion score	call-1 call-2 pileup rt-arrest

8.7 General filtering

8.7.1 Filter by mapping quality

-m1 MIN-MAPQ1	filter positions with MAPQ < MIN-MAPQ1 for condition 1 default: 20	call-2 pileup rt-arrest lrt-arrest

8.7.2 Filter by base call quality

-q1 MIN-BASQ1	filter positions with base quality < MIN-BASQ1 for condition 1 default: 20	call-2 pileup rt-arrest lrt-arrest

8.7.3 Filter by minimal coverage

-c1 MIN-COVERAGE1	filter positions with coverage < MIN-COVERAGE1 for condition 1 default: 5	call-2 pileup rt-arrest lrt-arrest

8.7.4 Limit maximal depth

-d1 MAX-DEPTH1	max read depth for condition 1 default: -1	call-2 pileup lrt-arrest

8.7.5 Filter by flag(s)

-f OUTPUT-FORMAT	Choose output format: <*> B: BED6-extended format (VCF will be ignored) (VCF is unstranded)	call-1 call-2
	Choose output format: <*> B: Default < > M: samtools mpileup like format (base columns without: $ ^ < > *)	pileup

8.7.6 Retain by flag(s)

	filter reads with flags FLAG default: 0	call-1 call-2
-F FLAG	filter reads with flags FLAG for all conditions default: 0	pileup rt-arrest
	filter reads with flags FLAG for all conditions default: 0	lrt-arrest

8.7.7 Filter by number of hits

-filterNH_1 NH	Max NH-VALUE for SAM tag NH for condition 1	call-2 pileup rt-arrest lrt-arrest

8.7.8 Filter by number of mismatches

-filterNM_1 NM	Max NM-VALUE for SAM tag NM for condition 1	call-2 pileup rt-arrest lrt-arrest

8.8 Artefact/feature filter

-a FEATURE-FILTER	[...] Use -h to see extended help	call-1 call-2 pileup rt-arrest lrt-arrest

8.9 Thread related

8.9.1 Number of parallel threads

-p THREADS	use # THREADS default: 1	call-1 call-2 pileup rt-arrest lrt-arrest

8.9.2 Actual thread window size

-w WINDOW	size of the window used for caching. Make sure this is greater than the read size default: 10000	call-1 call-2 pileup
	size of the window used for caching. Make sure this is greater than the read size default: 10000	rt-arrest
	size of the window used for caching. Make sure this is greater than the read size default: 5000	lrt-arrest

8.9.3 Reserved thread window size

-W THREAD-WINDOW	size of the window used per thread default: 100000	call-1 call-2 pileup rt-arrest lrt-arrest

8.10 Test-statistic options

-T THRESHOLD	Filter positions based on test-statistic THRESHOLD default: DO NOT FILTER	call-1 call-2 rt-arrest lrt-arrest
-u MODE	[...] Use -h to see extended help	call-1 call-2 rt-arrest lrt-arrest

8.11 Filtering by Test-statistic threshold

8.12 Misc

-h	Print extended usage information	call-1 call-2 pileup rt-arrest lrt-arrest
-x	turn on Debug modus	call-1 call-2 pileup rt-arrest lrt-arrest

9 Used libraries

Library	Source
htsjdk 2.12.0	`https://github.com/samtools/htsjdk`
Apache:	
commons-cli 1.4	`https://commons.apache.org/proper/commons-cli`
commons-math3 3.6.1	`https://commons.apache.org/proper/commons-math`

References

[Li et al., 2009] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and and, R. D. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

[Piechotta et al., 2017] Piechotta, M., Wyler, E., Ohler, U., Landthaler, M., and Dieterich, C. (2017). JACUSA: site-specific identification of RNA editing events from replicate sequencing data. *BMC Bioinformatics*, 18(1).

[Piekna-Przybylska et al., 2007] Piekna-Przybylska, D., Decatur, W. A., and Fournier, M. J. (2007). The 3d rRNA modification maps database: with interactive tools for ribosome analysis. *Nucleic Acids Research*, 36(Database):D178–D183.

[Zhou et al., 2018] Zhou, K. I., Clark, W. C., Pan, D. W., Eckwahl, M. J., Dai, Q., and Pan, T. (2018). Pseudouridines have context-dependent mutation and stop rates in high-throughput sequencing. *RNA Biology*, 15(7):892–900.