JACUSA2 manual

Michael Piechotta
michael.piechotta@gmail.com

8th, 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 (*c*all-1, *c*all-2, and *p*ileup) are available in JACUSA2. The new release of JACUSA2 features great performance enhancements (3 faster) for existing methods and adds new methods (*r*t-arrestand *l*rt-arrest) that enable to identify read arrest events. *r*t-arrestallows to identify read arrest events by comparing read through

Figure 1: Schematic summarising JACUSA2 methods and underlying data

and read arrest counts. $l$rt-arrestis a combination of variant and arrest event detection that allows to identify linked events.

Some artefact filters from JACUSA have been moved to the R helper package called `https://github.com/dieterich-lab/JACUSAhelper2`JACUSA2helper. A new artefact filter has been added to JACUSA2 that allows to filter candidate variants or arrest events by providing a file that contains sites to be excluded.

Another new feature in JACUSA2, are optional data that can be added to the output. 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: reads containing the user specified base substitution and total reads.

JACUSA2 employs a window-based approach to traverse provided BAM[Li et al., 2009] files featuring highly parallel processing and utilizing the `https://github.com/samtools/htsjdk`htsjdk framework.

1.1 Variant calling

Robust identification of variants has proven to be a daunting task due to artefacts specific for NGS-data and employed mapping strategies. We implement various artefact filters that reduce the number of false positives (see section **??**).

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

1.2 Reverse transcriptase arrest events

[width=]figures/arrest_events

Figure 2: 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. 2). 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 method presented by Minka[1]. In order to represent arrest events in JACUAS2 output, the base vector colum “basesij” is split into two columns “arrest_basesij” and “through_basesij”. 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.

2 Download

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

[1]See `https://tminka.github.io/papers/dirichlet/minka-dirichlet.pdf`Minka-Dirichlet

`dieterich-lab/JACUSA2/releases`all 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.

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".

In JACUSA2 a ##" prefixed header line that contains command line options and used version is added to the default output file format. There is also a new version of `https://github.com/dieterich-lab/JACUSA2helper/`JACUSA2helper to support analysis of JACUSA2 output.

2.3 Sample *in silico* data

2.3.1 Variant calling

You can choose between different 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 frequency 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[2] and cDNA reads have been simulated with flux[3]. Read simulations have been restricted to the corresponding first chromosome of human. The archives have the following data:

***.bam** BAM files: gDNA.bam and cDNA OR cDNA_1.bam and cDNA_2.bam

variants.txt Coordinates of imputed variants and their target and sampled frequencies

(snps.txt) Only available for cDNA vs. cDNA. Coordinates of imputed SNPs. In both BAM files matching SNPs have the same target frequency but different effective or sampled frequency. The shape parameter determines

[2]`http://www.niehs.nih.gov/research/resources/software/biostatistics/art/`art
[3]`http://sammeth.net/confluence/display/SIM/Home`flux simulator

how much the sampled frequency will deviate from the target frequency in each BAM file. The suffixes: _cdna_1 and _cdna_2 correspond to the respective BAM file]

Available sample data:

-
- `https://data.dieterichlab.org/s/hg19_chr1_gDNA_VS_cDNA`hg19_chr1_gDNA_VS_cDNA
- `https://data.dieterichlab.org/s/hg19_chr1_cDNA_VS_cDNA`hg19_chr1_cDNA_VS_cDNA TODO

2.4 Reverse transcriptase arrest event

We have downloaded primary sequencing data from [Zhou et al., 2018] and mapped them according to . From the resulting set, we retained only uniquely mapping reads with mapping quality ≥ 20. `https://data.dieterichlab.org/s/data_Zhou2018`data_Zhou2018 [Zhou et al., 2018] map transcriptome-wide RNA modification of pseudouridine () by chemically modifying pseudouridines with carbodiimide (+CMC) and detecting arrest events that were induced by reverse transcription stops in high-throughput sequencing. In summary, +CMC and mock-treated HEK293T rRNA was reverse transcribed under 3 different conditions (HIVRT, SIIIMn, and SIIIMg) and seqeunced.

The archive `https://data.dieterichlab.org/s/data_Zhou2018`data_Zhou2018 consists of:

***.bam, *.bam.bai** BAM files and 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].

3 Input

All JACUSA2 methods require sorted and indexed `https://samtools.github.io/hts-specs/SAMv1.pdf`BAM files. BAM is a standardized file format for efficient storage of alignments. Furthermore, JACUSA2 requires that the reference sequence is available either through the "MD" `https://samtools.github.io/hts-specs/SAMtags.pdf`tag 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: `http://samtools.sourceforge.net/`SAMtools/BCFtools and/or `http://broadinstitute.github.io/picard/`picard tools 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 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" `https://samtools.github.io/hts-specs/SAMtags.pdf`tag if you want to provide reference sequence information via this tag. When your BAM files do not have the "MD" tag set correctly 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 duplicate 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 of arrest event identification.

Duplicated reads occur mostly due to PCR-artefacts. They are likely to harbour false variants and most statistical test require that reads are sampled independently. 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 that filter duplicates reads either remove duplicated reads from the final output or assign 1024 to the flag 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 duplicates.

3.1.2 Library type and strand information

JACUSA2 supports stranded paired end and single ends reads. Warning: the CLI option has changed (see Section 2.2). With the command line parameter "-P <LIBRARY-TYPE> | -P1 <LIBRARY-TYPE> -P2 <LIBRARY-TYPE>" the user can choose the underlying library type:

RF-FIRSTSTRAND STRANDED library - first strand sequenced,

FR-SECONDSTRAND STRANDED library - second strand sequenced, and

UNSTRANDED UNSTRANDED library.

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

Table 1: Example of BED-like traverse file

contig	**start**	**end**
1	1000	1100
2	10000	10000

3.2 Traverse BED-like file

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 this region(s) or site(s). Remaining region(s) of the BAM files will not be considered.

In the following traverse 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 coordinates 10,000: Many individual sites may impair running performance of JACUSA2. If possible, try to merge nearby sites into contiguous regions and extract specific sites from JACUSA2 output with `http://bedtools.readthedocs.org/en/latest/`bedtools “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 Output

JACUSA2 writes its output to a user speficied file. When using multiple threads, JACUSA2 will create a temporary file for each allocated thread in the temporary directory that is provided by the JAVA Virtual Machine. 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. Currently, JACUSA2 supports the following output formats, controlled by “-f”:

- Default (JACUSA2 output — varies between JACUSA2 methods)
- Variant Call Format (VCF)[4]

The default output format is based on BED6[5] with additional JACUSA2 methods specific columns. The actual number of columns depends on the JACUSA2 method and the number of provided BAM files.

(1, 2, 3) contig + start + end 0-based, genomic coordinates.

(4) name Currently, constant string that depends on each method. This dummy field is to ensure BED6 compatibility.

[4]`http://samtools.github.io/hts-specs/VCFv4.1.pdf]`VCF file format
[5]`http://genome.ucsc.edu/FAQ/FAQformat.html#format1`BED file format

Table 2: JACUSA2 default output format — core elements

Column:	1	2	3	4	5	6	...	N-1	N
	1	100	101	variant	8.07...	-	JACUSA2 method specific	*	*
			...				...		

(5) score 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.

(6) strand Possible values are: ".", "+", and "-" which correspond to "unstranded", "positive strand", and "negative strand", respectively. If strand is != ".", then the following b ase columns will be indicating base counts according to the strand - inverted base count if on the "negative strand".

(7-N-2) method specific The number of base columns depends on the JACUSA2 method — check method specific explanation.

(N-1) info Additional info for this specific site. Currently, details about the parameter estimation of the underlying distribution can be shown, insertion/deletion counts and statistics, and additional method specific data. If nothing provided, the empty field is equal to "*"

(N) filter_info Relevant, if feature filter(s) X have been provided with "-a X" on the command line. The column will contain a comma-separated list of feature filters that predict this site to be a potential artefact. Possible values depend on the utilized JACUSA2-method

4 Feature/Artefact filter

_filter_cropped.pdf _filter_cropped.png _filter_cropped.jpg _filter_cropped.mps _filter_cropped.jpeg _filter_cropped.jbig2 _filter_cropped.jb2 _filter_cropped.PDF _filter_cropped.PNG _filter_cropped.JPG _filter_cropped.JPEG _filter_cropped.JBIG2 _filter_cropped.JB2 _filter_cropped.eps

Value	Description of potential artefact
D	Variant call in the vicinity of Read Start/End, Intron, and/or INDEL position
B	Variant call in the vicinity of Read Start/End
I	Variant call in the vicinity of INDEL position
S	Variant call in the vicinity of Splice Site
Y	Variant call in the vicinity of homopolymer
M	Max allowed alleles exceeded
H	"Control" sample contains non-homozygous pileup

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 call-2 -r JACUSA.out -s -a H:1 gDNA.bam cDNA.bam
```

Option "-a H:1" ensures that potential polymorphisms in gDNA will be eliminated as artefacts. The number $x \in \{1, 2\}$ determines which sample has 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 cDNA1.bam cDNA2.bam
```

WARNING: If you want to identify RNA-RNA differences make sure NOT to use the filter "-a H:x"! Otherwise, potential valid variants will be filtered out.

6 Reverse transcriptase arrest events

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

8.2 Filtering artefacts

8.2.1 Configure feature filter

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

8.2.2 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

8.3 Input BED file

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

8.4 Reference fasta file

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

8.5 Library type

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

8.6 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

8.7 Show deletion score

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

8.8 General filtering

8.8.1 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

8.8.2 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

8.8.3 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

8.8.4 Limit maximal depth

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

8.9 Specific filtering

8.9.1 Filter by flag(s)

8.9.2 Retain by flag(s)

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

8.9.3 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

8.9.4 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

8.10 Thread related

8.10.1 Number of parallel threads

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

8.10.2 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

8.10.3 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

8.11 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

8.12 Filtering by Test-statistic threshold

8.13 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

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