resolwe-bio

Release 0.0.0

Dec 13, 2019

Contents

Bioinformatics pipelines for the Resolwe dataflow package for Django framework.

CHAPTER 1

Contents

1.1 Writing processes

A tutorial about writing bioinformatics pipelines (process is a step in the pipeline) is in Resolwe SDK for Python documentation.

1.1.1 Tools

Frequently, it is very useful to write a custom script in Python or R to perform a certain task in process' algorithm. For an example, see the tutorial in Resolwe SDK for Python documentation.

Custom scripts needed by processes included with Resolwe Bioinformatics are located in the resolwe_bio/tools directory.

Note: A Resolwe's `Flow Executor` searches for tools in a Django application's `tools` directory or directories specified in the `RESOLWE_CUSTOM_TOOLS_PATHS` Django setting.

1.2 Process catalog

Resolwe Bioinformatics includes over 100 processes. They are organized in categories. The type tree will help process developers with pipeline design. For process details browse process definitions.

1.2.1 Processes by category

Align

- WALT

- BWA MEM
- BWA SW
- BWA ALN
- Bowtie (Dicty)
- Bowtie2
- Subread
- HISAT2
- STAR

Chip-seq

Call peaks

- ChIP-seq (MACS2)
- MACS 2.0
- ChIP-seq (MACS2-ROSE2)
- MACS 1.4

Post process

- ROSE2

Qc report

- Pre-peakcall QC

Differential expression

- Cuffdiff 2.2
- edgeR
- DESeq2

Import

- GAF file
- VCF file
- Genome
- Expression data
- Expression data (Cuffnorm)
- Expression data (STAR)
- SAM header

- Upload Picard CollectTargetedPcrMetrics
- Metabolic pathway file
- BAM file
- BAM file and index
- Secondary hybrid BAM file
- Cuffquant results
- BED file
- SRA data
- SRA data (single-end)
- SRA data (paired-end)
- FASTA file
- BaseSpace file
- Differential Expression (table)
- OBO file
- Custom master file
- GFF3 file
- GTF file
- Mappability info
- Reads (QSEQ multiplexed, single)
- Reads (QSEQ multiplexed, paired)
- snpEff
- FASTQ file (single-end)
- FASTQ file (paired-end)
- Convert files to reads (single-end)
- Convert files to reads (paired-end)
- Gene set
- Gene set (create)
- Gene set (create from Venn diagram)
- Expression time course

Other

- Amplicon report
- Archive and make multi-sample report for amplicon data
- PCA
- Hierarchical clustering of samples
- Hierarchical clustering of genes

Pipeline

- ATAC-Seq
- BBDuk - STAR - HTSeq-count (single-end)
- BBDuk - STAR - HTSeq-count (paired-end)
- WGBS
- Cutadapt - STAR - RSEM (Diagenode CATS, single-end)
- Cutadapt - STAR - RSEM (Diagenode CATS, paired-end)
- BBDuk - STAR - featureCounts - QC (single-end)
- BBDuk - STAR - featureCounts - QC (paired-end)
- RNA-Seq (Cuffquant)
- BBDuk - STAR - FeatureCounts (3' mRNA-Seq, single-end)
- BBDuk - STAR - FeatureCounts (3' mRNA-Seq, paired-end)
- Cutadapt - STAR - HTSeq-count (single-end)
- Cutadapt - STAR - HTSeq-count (paired-end)
- Chemical Mutagenesis
- Accel Amplicon Pipeline
- Trim, align and quantify using a library as a reference.
- Whole exome sequencing (WES) analysis
- Trimmomatic - HISAT2 - HTSeq-count (single-end)
- Trimmomatic - HISAT2 - HTSeq-count (paired-end)
- MACS2 - ROSE2
- miRNA pipeline

Plot

- Bamplot

Quantify

- Cuffmerge
- Cuffnorm
- Cufflinks 2.2
- Cuffquant 2.2
- Quantify shRNA species using bowtie2
- HTSeq-count (TPM)
- HTSeq-count (CPM)
- featureCounts
- RSEM

Splice junctions

- Annotate novel splice junctions (regtools)

Trim

- Cutadapt (single-end)
- Cutadapt (paired-end)
- Trimmomatic (single-end)
- Trimmomatic (paired-end)
- BBDuk (single-end)
- BBDuk (paired-end)

Wgbs

- methcounts
- HMR

Abstract

- Abstract bed process
- Abstract differential expression process
- Abstract alignment process
- Abstract annotation process
- Abstract expression process

Uncategorized

- Test sleep progress
- Test disabled inputs
- Test basic fields
- Test hidden inputs
- Test select controler
- Detect library strandedness
- Salmon Index

1.2.2 Type tree

Process types are listed alphabetically. Next to each type is a list of processes of that type. Types are hierarchical, with levels of hierarchy separated by colon ":". The hierarchy defines what is accepted on inputs. For instance, Expression (Cuffnorm) process' input is `data:alignment:bam`. This means it also accepts all subtypes (*e.g.,* `data:alignment:bam:bwasw`, `data:alignment:bam:bowtie1` and `data:alignment:bam:tophat`). We encourage the use of existing types in custom processes.

- `data:aggregator:expression` - Expression aggregator
- `data:alignment` - Abstract alignment process
- `data:alignment:bam:bowtie1` - Bowtie (Dicty)
- `data:alignment:bam:bowtie2` - Bowtie2
- `data:alignment:bam:bwaaln` - BWA ALN
- `data:alignment:bam:bwamem` - BWA MEM
- `data:alignment:bam:bwasw` - BWA SW
- `data:alignment:bam:bwatrim` - Align (BWA) and trim adapters
- `data:alignment:bam:hisat2` - HISAT2
- `data:alignment:bam:primary` - Bam split
- `data:alignment:bam:secondary` - Secondary hybrid BAM file
- `data:alignment:bam:star` - STAR
- `data:alignment:bam:subread` - Subread
- `data:alignment:bam:upload` - BAM file, BAM file and index
- `data:alignment:bam:vc` - Indel Realignment and Base Recalibration
- `data:alignment:mr:walt` - WALT
- `data:annotation` - Abstract annotation process
- `data:annotation:cuffmerge` - Cuffmerge
- `data:annotation:gff3` - GFF3 file
- `data:annotation:gtf` - Convert GFF3 to GTF, GTF file
- `data:archive:samples` - Archive samples
- `data:archive:samples:amplicon` - Archive and make multi-sample report for amplicon data
- `data:bam:plot:bamliquidator` - Bamliquidator
- `data:bam:plot:bamplot` - Bamplot
- `data:bed` - Abstract bed process, BED file
- `data:chipseq:batch:macs2` - ChIP-seq (MACS2), ChIP-seq (MACS2-ROSE2)
- `data:chipseq:callpeak:macs14` - MACS 1.4
- `data:chipseq:callpeak:macs2` - MACS 2.0
- `data:chipseq:genescore` - ChIP-Seq (Gene Score)
- `data:chipseq:peakscore` - ChIP-Seq (Peak Score)
- `data:chipseq:rose2` - ROSE2

- data:clustering:hierarchical:gene - Hierarchical clustering of genes
- data:clustering:hierarchical:sample - Hierarchical clustering of samples
- data:coverage - coverageBed
- data:cufflinks:cufflinks - Cufflinks 2.2
- data:cufflinks:cuffquant - Cuffquant 2.2, Cuffquant results
- data:cuffnorm - Cuffnorm
- data:differentialexpression - Abstract differential expression process
- data:differentialexpression:cuffdiff - Cuffdiff 2.2
- data:differentialexpression:deseq2 - DESeq2
- data:differentialexpression:edger - edgeR
- data:differentialexpression:upload - Differential Expression (table)
- data:etc - Expression Time Course, Expression time course
- data:expression - Abstract expression process, Expression data, Expression data (Cuffnorm)
- data:expression:featurecounts - featureCounts
- data:expression:htseq:cpm - HTSeq-count (CPM)
- data:expression:htseq:normalized - HTSeq-count (TPM)
- data:expression:polya - Dictyostelium expressions
- data:expression:rsem - RSEM
- data:expression:shrna2quant - Quantify shRNA species using bowtie2
- data:expression:star - Expression data (STAR)
- data:expressionset - Expression matrix
- data:expressionset:etc - Merge Expressions (ETC)
- data:file - BaseSpace file
- data:gaf:2:0 - GAF file
- data:geneset - Gene set, Gene set (create)
- data:geneset:venn - Gene set (create from Venn diagram)
- data:genome:fasta - Genome
- data:genomeindex:star - STAR genome index
- data:goea - GO Enrichment analysis
- data:index:expression - Gene expression indices
- data:index:salmon - Salmon Index
- data:junctions:regtools - Annotate novel splice junctions (regtools)
- data:mappability:bcm - Mappability, Mappability info
- data:masterfile:amplicon - Custom master file
- data:metabolicpathway - Metabolic pathway file
- data:multiplexed:qseq:paired - Reads (QSEQ multiplexed, paired)

- `data:multiplexed:qseq:single` - Reads (QSEQ multiplexed, single)
- `data:multiqc` - MultiQC
- `data:ontology:obo` - OBO file
- `data:other:geo:chipseq` - Prepare GEO - ChIP-Seq
- `data:other:geo:rnaseq` - Prepare GEO - RNA-Seq
- `data:pca` - PCA
- `data:picard:coverage` - Picard CollectTargetedPcrMetrics
- `data:picard:coverage:upload` - Upload Picard CollectTargetedPcrMetrics
- `data:prepeakqc` - Pre-peakcall QC
- `data:reads:fastq:paired` - Convert files to reads (paired-end), FASTQ file (paired-end), SRA data (paired-end)
- `data:reads:fastq:paired:bbduk` - BBDuk (paired-end)
- `data:reads:fastq:paired:cutadapt` - Cutadapt (Diagenode CATS, paired-end), Cutadapt (paired-end)
- `data:reads:fastq:paired:seqtk` - Subsample FASTQ (paired-end)
- `data:reads:fastq:paired:trimmomatic` - Trimmomatic (paired-end)
- `data:reads:fastq:single` - Convert files to reads (single-end), FASTQ file (single-end), SRA data (single-end)
- `data:reads:fastq:single:bbduk` - BBDuk (single-end)
- `data:reads:fastq:single:cutadapt` - Cutadapt (Diagenode CATS, single-end), Cutadapt (single-end)
- `data:reads:fastq:single:seqtk` - Subsample FASTQ (single-end)
- `data:reads:fastq:single:trimmomatic` - Trimmomatic (single-end)
- `data:report:amplicon` - Amplicon report
- `data:sam:header` - SAM header
- `data:seq:nucleotide` - FASTA file
- `data:snpeff` - snpEff
- `data:snpeff:upload` - snpEff
- `data:spikeins` - Spike-ins quality control
- `data:sra` - SRA data
- `data:strandedness` - Detect library strandedness
- `data:test:disabled` - Test disabled inputs
- `data:test:fields` - Test basic fields
- `data:test:hidden` - Test hidden inputs
- `data:test:result` - Test select controler, Test sleep progress
- `data:variants:vcf` - VCF file
- `data:variants:vcf:chemut` - Variant calling (CheMut)

- `data:variants:vcf:filtering` - Variant filtering (CheMut)
- `data:variants:vcf:gatk:hc` - GATK3 (HaplotypeCaller), GATK4 (HaplotypeCaller)
- `data:variants:vcf:lofreq` - LoFreq (call)
- `data:varianttable:amplicon` - Amplicon table
- `data:wgbs:hmr` - HMR
- `data:wgbs:methcounts` - methcounts
- `data:workflow:amplicon` - Accel Amplicon Pipeline
- `data:workflow:atacseq` - ATAC-Seq
- `data:workflow:chemut` - Chemical Mutagenesis
- `data:workflow:chipseq:macs2rose2` - MACS2 - ROSE2
- `data:workflow:mirna` - miRNA pipeline
- `data:workflow:quant:featurecounts:paired` - BBDuk - STAR - FeatureCounts (3' mRNA-Seq, paired-end)
- `data:workflow:quant:featurecounts:single` - BBDuk - STAR - FeatureCounts (3' mRNA-Seq, single-end)
- `data:workflow:rnaseq:cuffquant` - RNA-Seq (Cuffquant)
- `data:workflow:rnaseq:featurecounts:qc` - BBDuk - STAR - featureCounts - QC (paired-end), BBDuk - STAR - featureCounts - QC (single-end)
- `data:workflow:rnaseq:htseq` - Cutadapt - STAR - HTSeq-count (paired-end), Cutadapt - STAR - HTSeq-count (single-end), Trimmomatic - HISAT2 - HTSeq-count (paired-end), Trimmomatic - HISAT2 - HTSeq-count (single-end)
- `data:workflow:rnaseq:htseq:paired` - BBDuk - STAR - HTSeq-count (paired-end)
- `data:workflow:rnaseq:htseq:single` - BBDuk - STAR - HTSeq-count (single-end)
- `data:workflow:rnaseq:rsem` - Cutadapt - STAR - RSEM (Diagenode CATS, paired-end), Cutadapt - STAR - RSEM (Diagenode CATS, single-end)
- `data:workflow:trimalquant` - Trim, align and quantify using a library as a reference.
- `data:workflow:wes` - Whole exome sequencing (WES) analysis
- `data:workflow:wgbs` - WGBS

1.2.3 Process definitions

ATAC-Seq

`data:workflow:atacseqworkflow-atac-seq` (*data:reads:fastq* **reads**, *data:genome:fasta* **genome**, *data:bed* **promoter**, *basic:string* **mode**, *basic:string* **speed**, *basic:boolean* **use_se**, *basic:boolean* **discordantly**, *basic:boolean* **rep_se**, *basic:integer* **minins**, *basic:integer* **maxins**, *basic:integer* **trim_5**, *basic:integer* **trim_3**, *basic:integer* **trim_iter**, *basic:integer* **trim_nucl**, *basic:string* **rep_mode**, *basic:integer* **k_reports**, *basic:integer* **q_threshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**, *basic:boolean* **tagalign**, *basic:string* **duplicates**, *basic:string* **duplicates_prepeak**, *basic:decimal* **qvalue**, *basic:decimal* **pvalue**, *basic:decimal* **pvalue_prepeak**, *basic:integer* **cap_num**, *basic:integer* **mfold_lower**, *basic:integer* **mfold_upper**, *basic:integer* **slocal**, *basic:integer* **llocal**, *basic:integer* **extsize**, *basic:integer* **shift**, *basic:integer* **band_width**, *basic:boolean* **nolambda**, *basic:boolean* **fix_bimodal**, *basic:boolean* **nomodel**, *basic:boolean* **nomodel_prepeak**, *basic:boolean* **down_sample**, *basic:boolean* **bedgraph**, *basic:boolean* **spmr**, *basic:boolean* **call_summits**, *basic:boolean* **broad**, *basic:decimal* **broad_cutoff**) [Source: v2.0.2]

This ATAC-seq pipeline closely follows the official ENCODE DCC pipeline. It is comprised of three steps; alignment, pre-peakcall QC, and calling peaks (with post-peakcall QC).

First, reads are aligned to a genome using [Bowtie2](http://bowtie-bio.sourceforge.net/index.shtml) aligner. Next, pre-peakcall QC metrics are calculated. QC report contains ENCODE 3 proposed QC metrics – [NRF](https://www.encodeproject.org/data-standards/terms/), [PBC bottlenecking coefficients, NSC, and RSC](https://genome.ucsc.edu/ENCODE/qualityMetrics.html#chipSeq). Finally, the peaks are called using [MACS2](https://github.com/taoliu/MACS/). The post-peakcall QC report includes additional QC metrics – number of peaks, fraction of reads in peaks (FRiP), number of reads in peaks, and if promoter regions BED file is provided, number of reads in promoter regions, fraction of reads in promoter regions, number of peaks in promoter regions, and fraction of reads in promoter regions.

Input arguments reads

> **label** Select sample(s)
>
> **type** `data:reads:fastq`

genome

> **label** Genome
>
> **type** `data:genome:fasta`

promoter

> **label** Promoter regions BED file
>
> **type** `data:bed`

description BED file containing promoter regions (TSS+-1000bp for example). Needed to get the number of peaks and reads mapped to promoter regions.

required False

alignment.mode

label Alignment mode

type `basic:string`

description End to end: Bowtie 2 requires that the entire read align from one end to the other, without any trimming (or "soft clipping") of characters from either end. Local: Bowtie 2 does not require that the entire read align from one end to the other. Rather, some characters may be omitted ("soft clipped") from the ends in order to achieve the greatest possible alignment score.

default `--local`

choices

- end to end mode: `--end-to-end`
- local: `--local`

alignment.speed

label Speed vs. Sensitivity

type `basic:string`

default `--sensitive`

choices

- Very fast: `--very-fast`
- Fast: `--fast`
- Sensitive: `--sensitive`
- Very sensitive: `--very-sensitive`

alignment.PE_options.use_se

label Map as single-ended (for paired-end reads only)

type `basic:boolean`

description If this option is selected paired-end reads will be mapped as single-ended and other paired-end options are ignored.

default `False`

alignment.PE_options.discordantly

label Report discordantly matched read

type `basic:boolean`

description If both mates have unique alignments, but the alignments do not match paired-end expectations (orientation and relative distance) then alignment will be reported. Useful for detecting structural variations.

default `True`

alignment.PE_options.rep_se

label Report single ended

type `basic:boolean`

description If paired alignment can not be found Bowtie2 tries to find alignments for the individual mates.

default `True`

alignment.PE_options.minins

label Minimal distance

type `basic:integer`

description The minimum fragment length for valid paired-end alignments. 0 imposes no minimum.

default `0`

alignment.PE_options.maxins

label Maximal distance

type `basic:integer`

description The maximum fragment length for valid paired-end alignments.

default `2000`

alignment.start_trimming.trim_5

label Bases to trim from 5’

type `basic:integer`

description Number of bases to trim from from 5’ (left) end of each read before alignment.

default `0`

alignment.start_trimming.trim_3

label Bases to trim from 3’

type `basic:integer`

description Number of bases to trim from from 3’ (right) end of each read before alignment

default `0`

alignment.trimming.trim_iter

label Iterations

type `basic:integer`

description Number of iterations.

default `0`

alignment.trimming.trim_nucl

label Bases to trim

type `basic:integer`

description Number of bases to trim from 3’ end in each iteration.

default `2`

alignment.reporting.rep_mode

label Report mode

type `basic:string`

description Default mode: search for multiple alignments, report the best one; -k mode: search for one or more alignments, report each; -a mode: search for and report all alignments

default `def`

choices

- Default mode: `def`
- -k mode: `k`
- -a mode (very slow): `a`

alignment.reporting.k_reports

label Number of reports (for -k mode only)

type `basic:integer`

description Searches for at most X distinct, valid alignments for each read. The search terminates when it can't find more distinct valid alignments, or when it finds X, whichever happens first.

default `5`

prepeakqc_settings.q_threshold

label Quality filtering threshold

type `basic:integer`

default `30`

prepeakqc_settings.n_sub

label Number of reads to subsample

type `basic:integer`

default `25000000`

prepeakqc_settings.tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on "+" strand by 4bp and reads on "-" strand by 5bp.

default `True`

prepeakqc_settings.shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

default `0`

settings.tagalign

label Use tagAlign files

type `basic:boolean`

description Use filtered tagAlign files as case (treatment) and control (background) samples. If extsize parameter is not set, run MACS using input's estimated fragment length.

default `True`

settings.duplicates

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden settings.tagalign

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.duplicates_prepeak

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden !settings.tagalign

default `all`

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.qvalue

label Q-value cutoff

type `basic:decimal`

description The q-value (minimum FDR) cutoff to call significant regions. Q-values are calculated from p-values using Benjamini-Hochberg procedure.

required False

disabled settings.pvalue && settings.pvalue_prepeak

settings.pvalue

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

required False

disabled settings.qvalue

hidden settings.tagalign

settings.pvalue_prepeak

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

disabled settings.qvalue

hidden !settings.tagalign || settings.qvalue

default `0.01`

settings.cap_num

label Cap number of peaks by taking top N peaks

type `basic:integer`

description To keep all peaks set value to 0.

disabled settings.broad

default `300000`

settings.mfold_lower

label MFOLD range (lower limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.mfold_upper

label MFOLD range (upper limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.slocal

label Small local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.llocal

label Large local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.extsize

label extsize

type `basic:integer`

description While '–nomodel' is set, MACS uses this parameter to extend reads in 5'->3' direction to fix-sized fragments. For example, if the size of binding region for your transcription factor is 200 bp, and you want to bypass the model building by MACS, this parameter can be set as 200. This option is only valid when –nomodel is set or when MACS fails to build model and –fix-bimodal is on.

default `150`

settings.shift

label Shift

type `basic:integer`

description Note, this is NOT the legacy –shiftsize option which is replaced by –extsize! You can set an arbitrary shift in bp here. Please Use discretion while setting it other than default value (0). When –nomodel is set, MACS will use this value to move cutting ends (5') then apply –extsize from 5' to 3' direction to extend them to fragments. When this value is negative, ends will be moved toward 3'->5' direction, otherwise 5'->3' direction. Recommended to keep it as default 0 for ChIP-Seq datasets, or -1 * half of EXTSIZE together with –extsize option for detecting enriched cutting loci such as certain DNAseI-Seq datasets. Note, you can't set values other than 0 if format is BAMPE for paired-end data. Default is 0.

default `-75`

settings.band_width

label Band width

type `basic:integer`

description The band width which is used to scan the genome ONLY for model building. You can set this parameter as the sonication fragment size expected from wet experiment. The previous side effect on the peak detection process has been removed. So this parameter only affects the model building.

required False

settings.nolambda

label Use backgroud lambda as local lambda

type `basic:boolean`

description With this flag on, MACS will use the background lambda as local lambda. This means MACS will not consider the local bias at peak candidate regions.

default `False`

settings.fix_bimodal

label Turn on the auto paired-peak model process

type `basic:boolean`

description Whether turn on the auto paired-peak model process. If it's set, when MACS failed to build paired model, it will use the nomodel settings, the '–extsize' parameter to extend each tags. If set, MACS will be terminated if paired-peak model is failed.

default `False`

settings.nomodel

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden settings.tagalign

default `False`

settings.nomodel_prepeak

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden !settings.tagalign

default `True`

settings.down_sample

label Down-sample

type `basic:boolean`

description When set, random sampling method will scale down the bigger sample. By default, MACS uses linear scaling. This option will make the results unstable and irreproducible since each time, random reads would be selected, especially the numbers (pileup, pvalue, qvalue) would change. Consider to use 'randsample' script before MACS2 runs instead.

default `False`

settings.bedgraph

label Save fragment pileup and control lambda

type `basic:boolean`

description If this flag is on, MACS will store the fragment pileup, control lambda, -log10pvalue and -log10qvalue scores in bedGraph files. The bedGraph files will be stored in current directory named NAME+'_treat_pileup.bdg' for treatment data, NAME+'_control_lambda.bdg' for local lambda values from control, NAME+'_treat_pvalue.bdg' for Poisson pvalue scores (in -log10(pvalue) form), and NAME+'_treat_qvalue.bdg' for q-value scores from Benjamini-Hochberg-Yekutieli procedure.

default `True`

settings.spmr

label Save signal per million reads for fragment pileup profiles

type `basic:boolean`

disabled settings.bedgraph === false

default `True`

settings.call_summits

label Call summits

type `basic:boolean`

description MACS will now reanalyze the shape of signal profile (p or q-score depending on cutoff setting) to deconvolve subpeaks within each peak called from general procedure. It's highly recommended to detect adjacent binding events. While used, the output subpeaks of a big peak region will have the same peak boundaries, and different scores and peak summit positions.

default `True`

settings.broad

label Composite broad regions

type `basic:boolean`

description When this flag is on, MACS will try to composite broad regions in BED12 (a gene-model-like format) by putting nearby highly enriched regions into a broad region with loose cutoff. The broad region is controlled by another cutoff through –broad-cutoff. The maximum length of broad region length is 4 times of d from MACS.

disabled settings.call_summits === true

default `False`

settings.broad_cutoff

label Broad cutoff

type `basic:decimal`

description Cutoff for broad region. This option is not available unless –broad is set. If -p is set, this is a p-value cutoff, otherwise, it's a q-value cutoff. DEFAULT = 0.1

required False

disabled settings.call_summits === true || settings.broad !== true

Output results

Abstract alignment process

data:alignmentabstract-alignment () [Source: v1.0.0]

Input arguments

Output results bam

label Alignment file

type `basic:file`

bai

label Alignment index BAI

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Abstract annotation process

data:annotationabstract-annotation () [Source: v1.0.0]

Input arguments

Output results annot

label Uploaded file

type `basic:file`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Abstract bed process

data:bedabstract-bed () [Source: v1.0.0]

Input arguments

Output results bed

label BED

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Abstract differential expression process

`data:differentialexpressionabstract-differentialexpression` `()` [Source: v1.0.0]

Input arguments

Output results raw

label Differential expression (gene level)

type `basic:file`

de_json

label Results table (JSON)

type `basic:json`

de_file

label Results table (file)

type `basic:file`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Abstract expression process

`data:expressionabstract-expression ()` [Source: v1.0.0]

Input arguments

Output results exp

label Normalized expression

type `basic:file`

rc

label Read counts

type `basic:file`

required False

exp_json

label Expression (json)

type `basic:json`

exp_type

label Expression type

type `basic:string`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Accel Amplicon Pipeline

data:workflow:ampliconworkflow-accel (*data:reads:fastq:paired* **reads**, *data:genome:fasta* **genome**, *data:masterfile:amplicon* **master_file**, *data:seq:nucleotide* **adapters**, *list:data:variants:vcf* **known_indels**, *list:data:variants:vcf* **known_vars**, *data:variants:vcf* **dbsnp**, *basic:integer* **mbq**, *basic:integer* **stand_call_conf**, *basic:integer* **min_bq**, *basic:integer* **min_alt_bq**, *list:data:variants:vcf* **known_vars_db**, *basic:decimal* **af_threshold**) [Source: v4.0.1]

Processing pipeline to analyse the Accel-Amplicon NGS panel data. The raw amplicon sequencing reads are quality trimmed using Trimmomatic. The quality of the raw and trimmed data is assesed using the FASTQC tool. Quality trimmed reads are aligned to a reference genome using BWA mem. Sequencing primers are removed from the aligned reads using Primerclip. Amplicon performance stats are calculated using Bedtools coveragebed and Picard CollectTargetedPcrMetrics programs. Prior to variant calling, the alignment file is preprocessed using the GATK IndelRealigner and BaseRecalibrator tools. GATK HaplotypeCaller and Lofreq tools are used to call germline variants. Called variants are annotated using the SnpEff tool. Finally, the amplicon performance metrics and identified variants data are used to generate the PDF analysis report.

Input arguments reads

label Input reads

type `data:reads:fastq:paired`

genome

label Genome

type `data:genome:fasta`

master_file

label Experiment Master file

type `data:masterfile:amplicon`

adapters

label Adapters

type `data:seq:nucleotide`

description Provide an Illumina sequencing adapters file (.fasta) with adapters to be removed by Trimmomatic.

preprocess_bam.known_indels

label Known indels

type `list:data:variants:vcf`

preprocess_bam.known_vars

label Known variants

type `list:data:variants:vcf`

gatk.dbsnp

label dbSNP

type `data:variants:vcf`

gatk.mbq

label Min Base Quality

type `basic:integer`

description Minimum base quality required to consider a base for calling.

default `20`

gatk.stand_call_conf

label Min call confidence threshold

type `basic:integer`

description The minimum phred-scaled confidence threshold at which variants should be called.

default `20`

lofreq.min_bq

label Min baseQ

type `basic:integer`

description Skip any base with baseQ smaller than the default value.

default `20`

lofreq.min_alt_bq

label Min alternate baseQ

type `basic:integer`

description Skip alternate bases with baseQ smaller than the default value.

default `20`

var_annot.known_vars_db

label Known variants

type `list:data:variants:vcf`

report.af_threshold

label Allele frequency threshold

type `basic:decimal`

default `0.01`

Output results

Align (BWA) and trim adapters

data:alignment:bam:bwatrimalign-bwa-trim (*data:masterfile:amplicon* **master_file**, *data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:integer* **seed_l**, *basic:integer* **band_w**, *basic:decimal* **re_seeding**, *basic:boolean* **m**, *basic:integer* **match**, *basic:integer* **missmatch**, *basic:integer* **gap_o**, *basic:integer* **gap_e**, *basic:integer* **clipping**, *basic:integer* **unpaired_p**, *basic:boolean* **report_all**, *basic:integer* **report_tr**) [Source: v1.2.2]

Align with BWA mem and trim the sam output. The process uses the memory-optimized Primertrim tool.

Input arguments master_file

label Master file

type `data:masterfile:amplicon`

description Amplicon experiment design file that holds the information about the primers to be removed.

genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

seed_l

label Minimum seed length

type `basic:integer`

description Minimum seed length. Matches shorter than minimum seed length will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20.

default `19`

band_w

label Band width

type `basic:integer`

description Gaps longer than this will not be found.

default `100`

re_seeding

label Re-seeding factor

type `basic:decimal`

description Trigger re-seeding for a MEM longer than minSeedLen*FACTOR. This is a key heuristic parameter for tuning the performance. Larger value yields fewer seeds, which leads to faster alignment speed but lower accuracy.

default `1.5`

m

label Mark shorter split hits as secondary

type `basic:boolean`

description Mark shorter split hits as secondary (for Picard compatibility)

default `False`

scoring.match

label Score of a match

type `basic:integer`

default `1`

scoring.missmatch

label Mismatch penalty

type `basic:integer`

default `4`

scoring.gap_o

label Gap open penalty

type `basic:integer`

default `6`

scoring.gap_e

label Gap extension penalty

type `basic:integer`

default `1`

scoring.clipping

label Clipping penalty

type `basic:integer`

description Clipping is applied if final alignment score is smaller than (best score reaching the end of query) - (Clipping penalty)

default `5`

scoring.unpaired_p

label Penalty for an unpaired read pair

type `basic:integer`

description Affinity to force pair. Score: scoreRead1+scoreRead2-Penalty

default `9`

reporting.report_all

label Report all found alignments

type `basic:boolean`

description Output all found alignments for single-end or unpaired paired-end reads. These alignments will be flagged as secondary alignments.

default `False`

reporting.report_tr

label Report threshold score

type `basic:integer`

description Don't output alignment with score lower than defined number. This option only affects output.

default `30`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Amplicon report

`data:report:ampliconamplicon-report` (*data:picard:coverage* **pcr_metrics**, *data:coverage* **coverage**, *data:masterfile:amplicon* **master_file**, *list:data:snpeff* **annot_vars**, *basic:decimal* **af_threshold**) [Source: v1.0.4]

Create amplicon report.

Input arguments pcr_metrics

label Picard TargetedPcrMetrics

type `data:picard:coverage`

coverage

label Coverage

type `data:coverage`

master_file

label Amplicon master file

type `data:masterfile:amplicon`

annot_vars

label Annotated variants (snpEff)

type `list:data:snpeff`

af_threshold

label Allele frequency threshold

type `basic:decimal`

default `0.01`

Output results report

label Report

type `basic:file`

panel_name

label Panel name

type `basic:string`

stats

label File with sample statistics

type `basic:file`

amplicon_cov

label Amplicon coverage file (nomergebed)

type `basic:file`

variant_tables

label Variant tabels (snpEff)

type `list:basic:file`

Amplicon table

`data:varianttable:ampliconamplicon-table` (*data:masterfile:amplicon* **master_file**, *data:coverage* **coverage**, *list:data:snpeff* **annot_vars**, *basic:boolean* **all_amplicons**, *basic:string* **table_name**) [Source: v1.0.1]

Create variant table for use together with the genome browser.

Input arguments master_file

label Master file

type `data:masterfile:amplicon`

coverage

label Amplicon coverage

type `data:coverage`

annot_vars

label Annotated variants

type `list:data:snpeff`

all_amplicons

label Report all amplicons

type `basic:boolean`

default `False`

table_name

label Amplicon table name

type `basic:string`

default `Amplicons containing variants`

Output results variant_table

label Variant table

type `basic:json`

Annotate novel splice junctions (regtools)

`data:junctions:regtoolsregtools-junctions-annotate` (*data:genome:fasta* **genome**, *data:annotation:gtf* **annotation**, *data:alignment:bam:star* **alignment_star**, *data:alignment:bam* **alignment**, *data:bed* **input_bed_junctions**) [Source: v0.2.1]

Identify novel splice junctions by using regtools to annotate against a reference. The process accepts reference genome, reference genome annotation (GTF), and input with reads information (STAR aligment or reads aligned by any other aligner or junctions in BED12 format). If STAR aligner data is given as input, the process calculates BED12 file from STAR 'SJ.out.tab' file, and annotates all junctions with 'regtools junctions annotate' command. When reads are aligned by other aligner, junctions are extracted with 'regtools junctions extract' tool and then annotated with 'junction annotate' command. Third option allows user to provide directly BED12 file with junctions, which are then annotated. Finnally, annotated novel junctions are filtered in a separate output file. More information can be found in the [regtools manual](https://regtools.readthedocs.io/en/latest/).

Input arguments genome

label Reference genome

type `data:genome:fasta`

annotation

label Reference genome annotation (GTF)

type `data:annotation:gtf`

alignment_star

label STAR alignment

type `data:alignment:bam:star`

description Splice junctions detected by STAR aligner (SJ.out.tab STAR output file). Please provide one input ‘STAR alignment’ or ‘Alignment’ by any aligner or directly ‘Junctions in BED12 format’.

required False

alignment

label Alignment

type `data:alignment:bam`

description Aligned reads from which splice junctions are going to be extracted. Please provide one input ‘STAR alignment’ or ‘Alignment’ by any aligner or directly ‘Junctions in BED12 format’.

required False

input_bed_junctions

label Junctions in BED12 format

type `data:bed`

description Splice junctions in BED12 format. Please provide one input ‘STAR alignment’ or ‘Alignment’ by any aligner or directly ‘Junctions in BED12 format’.

required False

Output results novel_splice_junctions

label Table of annotated novel splice junctions

type `basic:file`

splice_junctions

label Table of annotated splice junctions

type `basic:file`

novel_sj_bed

label Novel splice junctions in BED format

type `basic:file`

bed

label Splice junctions in BED format

type `basic:file`

novel_sj_bigbed_igv_ucsc

label Novel splice junctions in BigBed format

type `basic:file`

required False

bigbed_igv_ucsc

label Splice junctions in BigBed format

type `basic:file`

required False

novel_sj_tbi_jbrowse

label Novel splice junctions bed tbi index for JBrowse

type `basic:file`

tbi_jbrowse

label Bed tbi index for JBrowse

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Archive and make multi-sample report for amplicon data

`data:archive:samples:ampliconamplicon-archive-multi-report` (*list:data* **data**, *list:basic:string* **fields**, *basic:boolean* **j**) [Source: v0.2.5]

Create an archive of output files. The ouput folder structure is organized by sample slug and data object's output-field names. Additionally, create multi-sample report for selected samples.

Input arguments data

label Data list

type `list:data`

fields

label Output file fields

type `list:basic:string`

j

label Junk paths

type `basic:boolean`

description Store just names of saved files (junk the path)

default `False`

Output results archive

label Archive of selected samples and a heatmap comparing them

type `basic:file`

Archive samples

data:archive:samplesarchive-samples (*list:data* **data**, *list:basic:string* **fields**, *basic:boolean* **j**) [Source: v0.2.3]

Create an archive of output files. The ouput folder structure is organized by sample slug and data object's output-field names.

Input arguments data

label Data list

type `list:data`

fields

label Output file fields

type `list:basic:string`

j

label Junk paths

type `basic:boolean`

description Store just names of saved files (junk the path)

default `False`

Output results archive

label Archive

type `basic:file`

BAM file

data:alignment:bam:uploadupload-bam (*basic:file* **src**, *basic:string* **species**, *basic:string* **build**) [Source: v1.4.1]

Import a BAM file (.bam), which is the binary format for storing sequence alignment data. This format is described on the [SAM Tools web site](http://samtools.github.io/hts-specs/).

Input arguments src

label Mapping (BAM)

type `basic:file`

description A mapping file in BAM format. The file will be indexed on upload, so additional BAI files are not required.

validate_regex `\.(bam)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`

- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

Output results bam

label Uploaded file

type `basic:file`

bai

label Index BAI

type `basic:file`

stats

label Alignment statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

BAM file and index

`data:alignment:bam:uploadupload-bam-indexed` (*basic:file* **src**, *basic:file* **src2**, *basic:string* **species**, *basic:string* **build**) [Source: v1.4.1]

Import a BAM file (.bam) and BAM index (.bam.bai). BAM file is the binary format for storing sequence alignment data. This format is described on the [SAM Tools web site](http://samtools.github.io/hts-specs/).

Input arguments src

label Mapping (BAM)

type `basic:file`

description A mapping file in BAM format.

validate_regex `\.(bam)$`

src2

label bam index (*.bam.bai file)

type `basic:file`

description An index file of a BAM mapping file (ending with bam.bai).

validate_regex `\.(bam.bai)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

Output results bam

label Uploaded file

type `basic:file`

bai

label Index BAI

type `basic:file`

stats

label Alignment statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

BBDuk (paired-end)

`data:reads:fastq:paired:bbdukbbduk-paired` (*data:reads:fastq:paired* **reads**, *basic:integer* **min_length**, *basic:boolean* **show_advanced**, *list:data:seq:nucleotide* **sequences**, *list:basic:string* **literal_sequences**, *basic:integer* **kmer_length**, *basic:boolean* **check_reverse_complements**, *basic:boolean* **mask_middle_base**, *basic:integer* **min_kmer_hits**, *basic:decimal* **min_kmer_fraction**, *basic:decimal* **min_coverage_fraction**, *basic:integer* **hamming_distance**, *basic:integer* **query_hamming_distance**, *basic:integer* **edit_distance**, *basic:integer* **hamming_distance2**, *basic:integer* **query_hamming_distance2**, *basic:integer* **edit_distance2**, *basic:boolean* **forbid_N**, *basic:boolean* **remove_if_either_bad**, *basic:boolean* **find_best_match**, *basic:boolean* **perform_error_correction**, *basic:string* **k_trim**, *basic:string* **k_mask**, *basic:boolean* **mask_fully_covered**, *basic:integer* **min_k**, *basic:string* **quality_trim**, *basic:integer* **trim_quality**, *basic:integer* **trim_poly_A**, *basic:decimal* **min_length_fraction**, *basic:integer* **max_length**, *basic:integer* **min_average_quality**, *basic:integer* **min_average_quality_bases**, *basic:integer* **min_base_quality**, *basic:integer* **min_consecutive_bases**, *basic:integer* **trim_pad**, *basic:boolean* **trim_by_overlap**, *basic:boolean* **strict_overlap**, *basic:integer* **min_overlap**, *basic:integer* **min_insert**, *basic:boolean* **trim_pairs_evenly**, *basic:integer* **force_trim_left**, *basic:integer* **force_trim_right**, *basic:integer* **force_trim_right2**, *basic:integer* **force_trim_mod**, *basic:integer* **restrict_left**, *basic:integer* **restrict_right**, *basic:decimal* **min_GC**, *basic:decimal* **max_GC**, *basic:integer* **maxns**, *basic:boolean* **toss_junk**, *basic:boolean* **chastity_filter**, *basic:boolean* **barcode_filter**, *list:data:seq:nucleotide* **barcode_files**, *list:basic:string* **barcode_sequences**, *basic:integer* **x_min**, *basic:integer* **y_min**, *basic:integer* **x_max**, *basic:integer* **y_max**, *basic:decimal* **entropy**, *basic:integer* **entropy_window**, *basic:integer* **entropy_k**, *basic:boolean* **entropy_mask**, *basic:integer* **min_base_frequency**, *basic:boolean* **nogroup**) [Source: v2.2.2]

BBDuk combines the most common data-quality-related trimming, filtering, and masking operations into a single high-performance tool. It is capable of quality-trimming and filtering, adapter-trimming, contaminant-filtering via kmer matching, sequence masking, GC-filtering, length filtering, entropy-filtering, format conversion, histogram generation, subsampling, quality-score recalibration, kmer cardinality estimation, and various other operations in a single pass. See [here](https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbduk-guide/) for more information.

Input arguments reads

label Reads

type `data:reads:fastq:paired`

min_length

label Minimum length [minlength=10]

type `basic:integer`

description Reads shorter than the minimum length will be discarded after trimming.

default `10`

show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

reference.sequences

label Sequences [ref]

type `list:data:seq:nucleotide`

description Reference sequences include adapters, contaminants, and degenerate sequences. They can be provided in a multi-sequence FASTA file or as a set of literal sequences below.

required False

reference.literal_sequences

label Literal sequences [literal]

type `list:basic:string`

description Literal sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

default `[]`

processing.kmer_length

label Kmer length [k=27]

type `basic:integer`

description Kmer length used for finding contaminants. Contaminants shorter than kmer length will not be found. Kmer length must be at least 1.

default `27`

processing.check_reverse_complements

label Look for reverse complements of kmers in addition to forward kmers [rcomp=t]

type `basic:boolean`

default `True`

processing.mask_middle_base

label Treat the middle base of a kmer as a wildcard to increase sensitivity in the presence of errors [maskmiddle=t]

type `basic:boolean`

default `True`

processing.min_kmer_hits

label Minimum number of kmer hits [minkmerhits=1]

type `basic:integer`

description Reads need at least this many matching kmers to be considered as matching the reference.

default `1`

processing.min_kmer_fraction

label Minimum kmer fraction [minkmerfraction=0.0]

type `basic:decimal`

description A read needs at least this fraction of its total kmers to hit a reference in order to be considered a match. If this and 'Minimum number of kmer hits' are set, the greater is used.

default `0.0`

processing.min_coverage_fraction

label Minimum coverage fraction [mincovfraction=0.0]

type `basic:decimal`

description A read needs at least this fraction of its total bases to be covered by reference kmers to be considered a match. If specified, 'Minimum coverage fraction' overrides 'Minimum number of kmer hits' and 'Minimum kmer fraction'.

default `0.0`

processing.hamming_distance

label Maximum Hamming distance for kmers (substitutions only) [hammingdistance=0]

type `basic:integer`

default `0`

processing.query_hamming_distance

label Hamming distance for query kmers [qhdist=0]

type `basic:integer`

default `0`

processing.edit_distance

label Maximum edit distance from reference kmers (substitutions and indels) [editdistance=0]

type `basic:integer`

default `0`

processing.hamming_distance2

label Hamming distance for short kmers when looking for shorter kmers [hammingdistance2=0]

type `basic:integer`

default `0`

processing.query_hamming_distance2

label Hamming distance for short query kmers when looking for shorter kmers [qhdist2=0]

type `basic:integer`

default `0`

processing.edit_distance2

label Maximum edit distance from short reference kmers (substitutions and indels) when looking for shorter kmers [editdistance2=0]

type `basic:integer`

default `0`

processing.forbid_N

label Forbid matching of read kmers containing N [forbidn=f]

type `basic:boolean`

description By default, these will match a reference 'A' if 'Maximum Hamming distance for kmers' > 0 or 'Maximum edit distance from reference kmers' > 0, to increase sensitivity.

default `False`

processing.remove_if_either_bad

label Remove both sequences of a paired-end read, if either of them is to be removed [removeifeitherbad=t]

type `basic:boolean`

default `True`

processing.find_best_match

label If multiple matches, associate read with sequence sharing most kmers [findbestmatch=t]

type `basic:boolean`

default `True`

processing.perform_error_correction

label Perform error correction with BBMerge prior to kmer operations [ecco=f]

type `basic:boolean`

default `False`

operations.k_trim

label Trimming protocol to remove bases matching reference kmers from reads [ktrim=f]

type `basic:string`

default `f`

choices

- Don't trim: `f`
- Trim to the right: `r`
- Trim to the left: `l`

operations.k_mask

label Symbol to replace bases matching reference kmers [kmask=f]

type `basic:string`

description Allows any non-whitespace character other than t or f. Processes short kmers on both ends.

default `f`

operations.mask_fully_covered

label Only mask bases that are fully covered by kmers [maskfullycovered=f]

type `basic:boolean`

default `False`

operations.min_k

label Look for shorter kmers at read tips down to this length when k-trimming or masking [mink=0]

type `basic:integer`

description -1 means disabled. Enabling this will disable treating the middle base of a kmer as a wildcard to increase sensitivity in the presence of errors.

default `-1`

operations.quality_trim

label Trimming protocol to remove bases with quality below the minimum average region quality from read ends [qtrim=f]

type `basic:string`

description Performed after looking for kmers. If enabled, set also 'Average quality below which to trim region'.

default `f`

choices

- Trim neither end: `f`
- Trim both ends: `rl`
- Trim only right end: `r`
- Trim only left end: `l`
- Use sliding window: `w`

operations.trim_quality

label Average quality below which to trim region [trimq=6]

type `basic:integer`

description Set trimming protocol to enable this parameter.

disabled operations.quality_trim == 'f'

default `6`

operations.trim_poly_A

label Minimum length of poly-A or poly-T tails to trim on either end of reads [trimpolya=0]

type `basic:integer`

default `0`

operations.min_length_fraction

label Minimum length fraction [mlf=0.0]

type `basic:decimal`

description Reads shorter than this fraction of original length after trimming will be discarded.

default `0.0`

operations.max_length

label Maximum length [maxlength]

type `basic:integer`

description Reads longer than this after trimming will be discarded.

required False

operations.min_average_quality

label Minimum average quality [minavgquality=0]

type `basic:integer`

description Reads with average quality (after trimming) below this will be discarded.

default `0`

operations.min_average_quality_bases

label Number of initial bases to calculate minimum average quality from [maqb=0]

type `basic:integer`

description Used only if positive.

default `0`

operations.min_base_quality

label Minimum base quality below which reads are discarded after trimming [minbasequality=0]

type `basic:integer`

default `0`

operations.min_consecutive_bases

label Minimum number of consecutive called bases [mcb=0]

type `basic:integer`

default `0`

operations.trim_pad

label Number of bases to trim around matching kmers [tp=0]

type `basic:integer`

default `0`

operations.trim_by_overlap

label Trim adapters based on where paired-end reads overlap [tbo=f]

type `basic:boolean`

default `False`

operations.strict_overlap

label Adjust sensitivity in 'Trim adapters based on where paired-end reads overlap' mode [strictoverlap=t]

type `basic:boolean`

default `True`

operations.min_overlap

label Minimum number of overlapping bases [minoverlap=14]

type `basic:integer`

description Require this many bases of overlap for detection.

default `14`

operations.min_insert

label Minimum insert size [mininsert=40]

type `basic:integer`

description Require insert size of at least this for overlap. Should be reduced to 16 for small RNA sequencing.

default `40`

operations.trim_pairs_evenly

label Trim both sequences of paired-end reads to the minimum length of either sequence [tpe=f]

type `basic:boolean`

default `False`

operations.force_trim_left

label Position from which to trim bases to the left [forcetrimleft=0]

type `basic:integer`

default `0`

operations.force_trim_right

label Position from which to trim bases to the right [forcetrimright=0]

type `basic:integer`

default `0`

operations.force_trim_right2

label Number of bases to trim from the right end [forcetrimright2=0]

type `basic:integer`

default `0`

operations.force_trim_mod

label Modulo to right-trim reads [forcetrimmod=0]

type `basic:integer`

description Trim reads to the largest multiple of modulo.

default `0`

operations.restrict_left

label Number of leftmost bases to look in for kmer matches [restrictleft=0]

type `basic:integer`

default `0`

operations.restrict_right

label Number of rightmosot bases to look in for kmer matches [restrictright=0]

type `basic:integer`

default `0`

operations.min_GC

label Minimum GC content [mingc=0.0]

type `basic:decimal`

description Discard reads with lower GC content.

default `0.0`

operations.max_GC

label Maximum GC content [maxgc=1.0]

type `basic:decimal`

description Discard reads with higher GC content.

default `1.0`

operations.maxns

label Max Ns after trimming [maxns=-1]

type `basic:integer`

description If non-negative, reads with more Ns than this (after trimming) will be discarded.

default `-1`

operations.toss_junk

label Discard reads with invalid characters as bases [tossjunk=f]

type `basic:boolean`

default `False`

header_parsing.chastity_filter

label Discard reads that fail Illumina chastity filtering [chastityfilter=f]

type `basic:boolean`

description Discard reads with id containing ‘ 1:Y:’ or ‘ 2:Y:’.

default `False`

header_parsing.barcode_filter

label Remove reads with unexpected barcodes if barcodes are set, or barcodes containing ‘N’ otherwise [barcodefilter=f]

type `basic:boolean`

description A barcode must be the last part of the read header.

default `False`

header_parsing.barcode_files

label Barcode sequences [barcodes]

type `list:data:seq:nucleotide`

required False

header_parsing.barcode_sequences

label Literal barcode sequences [barcodes]

type `list:basic:string`

description Literal barcode sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

default `[]`

header_parsing.x_min

label Minimum X coordinate [xmin=-1]

type `basic:integer`

description If positive, discard reads with a smaller X coordinate.

default `-1`

header_parsing.y_min

label Minimum Y coordinate [ymin=-1]

type `basic:integer`

description If positive, discard reads with a smaller Y coordinate.

default `-1`

header_parsing.x_max

label Maximum X coordinate [xmax=-1]

type `basic:integer`

description If positive, discard reads with a larger X coordinate.

default `-1`

header_parsing.y_max

label Maximum Y coordinate [ymax=-1]

type `basic:integer`

description If positive, discard reads with a larger Y coordinate.

default `-1`

complexity.entropy

label Minimum entropy [entropy=-1.0]

type `basic:decimal`

description Set between 0 and 1 to filter reads with entropy below that value. Higher is more stringent.

default `-1.0`

complexity.entropy_window

label Length of sliding window used to calculate entropy [entropywindow=50]

type `basic:integer`

description To use the sliding window set minimum entropy in range between 0.0 and 1.0.

default `50`

complexity.entropy_k

label Length of kmers used to calcuate entropy [entropyk=5]

type `basic:integer`

default `5`

complexity.entropy_mask

label Mask low-entropy parts of sequences with N instead of discarding [entropymask=f]

type `basic:boolean`

default `False`

complexity.min_base_frequency

label Minimum base frequency [minbasefrequency=0]

type `basic:integer`

default `0`

fastqc.nogroup

label Disable grouping of bases for reads >50bp [nogroup]

type `basic:boolean`

description All reports will show data for every base in the read. Using this option will cause fastqc to crash and burn if you use it on really long reads.

default `False`

Output results fastq

label Remaining upstream reads

type `list:basic:file`

fastq2

label Remaining downstream reads

type `list:basic:file`

statistics

label Statistics

type `list:basic:file`

fastqc_url

label Upstream quality control with FastQC

type `list:basic:file:html`

fastqc_url2

label Downstream quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download upstream FastQC archive

type `list:basic:file`

fastqc_archive2

label Download downstream FastQC archive

type `list:basic:file`

BBDuk (single-end)

`data:reads:fastq:single:bbdukbbduk-single` (*data:reads:fastq:single* **reads**, *basic:integer* **min_length**, *basic:boolean* **show_advanced**, *list:data:seq:nucleotide* **sequences**, *list:basic:string* **literal_sequences**, *basic:integer* **kmer_length**, *basic:boolean* **check_reverse_complements**, *basic:boolean* **mask_middle_base**, *basic:integer* **min_kmer_hits**, *basic:decimal* **min_kmer_fraction**, *basic:decimal* **min_coverage_fraction**, *basic:integer* **hamming_distance**, *basic:integer* **query_hamming_distance**, *basic:integer* **edit_distance**, *basic:integer* **hamming_distance2**, *basic:integer* **query_hamming_distance2**, *basic:integer* **edit_distance2**, *basic:boolean* **forbid_N**, *basic:boolean* **find_best_match**, *basic:string* **k_trim**, *basic:string* **k_mask**, *basic:boolean* **mask_fully_covered**, *basic:integer* **min_k**, *basic:string* **quality_trim**, *basic:integer* **trim_quality**, *basic:integer* **trim_poly_A**, *basic:decimal* **min_length_fraction**, *basic:integer* **max_length**, *basic:integer* **min_average_quality**, *basic:integer* **min_average_quality_bases**, *basic:integer* **min_base_quality**, *basic:integer* **min_consecutive_bases**, *basic:integer* **trim_pad**, *basic:integer* **min_overlap**, *basic:integer* **min_insert**, *basic:integer* **force_trim_left**, *basic:integer* **force_trim_right**, *basic:integer* **force_trim_right2**, *basic:integer* **force_trim_mod**, *basic:integer* **restrict_left**, *basic:integer* **restrict_right**, *basic:decimal* **min_GC**, *basic:decimal* **max_GC**, *basic:integer* **maxns**, *basic:boolean* **toss_junk**, *basic:boolean* **chastity_filter**, *basic:boolean* **barcode_filter**, *list:data:seq:nucleotide* **barcode_files**, *list:basic:string* **barcode_sequences**, *basic:integer* **x_min**, *basic:integer* **y_min**, *basic:integer* **x_max**, *basic:integer* **y_max**, *basic:decimal* **entropy**, *basic:integer* **entropy_window**, *basic:integer* **entropy_k**, *basic:boolean* **entropy_mask**, *basic:integer* **min_base_frequency**, *basic:boolean* **nogroup**) [Source: v2.2.2]

BBDuk combines the most common data-quality-related trimming, filtering, and masking operations into a single high-performance tool. It is capable of quality-trimming and filtering, adapter-trimming, contaminant-filtering via kmer matching, sequence masking, GC-filtering, length filtering, entropy-filtering, format conversion, histogram generation, subsampling, quality-score recalibration, kmer cardinality estimation, and various other operations in a single pass. See [here](https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbduk-guide/) for more information.

Input arguments reads

label Reads

type `data:reads:fastq:single`

min_length

label Minimum length [minlength=10]

type `basic:integer`

description Reads shorter than the minimum length will be discarded after trimming.

default `10`

show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

reference.sequences

label Sequences [ref]

type `list:data:seq:nucleotide`

description Reference sequences include adapters, contaminants, and degenerate sequences. They can be provided in a multi-sequence FASTA file or as a set of literal sequences below.

required False

reference.literal_sequences

label Literal sequences [literal]

type `list:basic:string`

description Literal sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

default `[]`

processing.kmer_length

label Kmer length [k=27]

type `basic:integer`

description Kmer length used for finding contaminants. Contaminants shorter than Kmer length will not be found. Kmer length must be at least 1.

default `27`

processing.check_reverse_complements

label Look for reverse complements of kmers in addition to forward kmers [rcomp=t]

type `basic:boolean`

default `True`

processing.mask_middle_base

label Treat the middle base of a kmer as a wildcard to increase sensitivity in the presence of errors [maskmiddle=t]

type `basic:boolean`

default `True`

processing.min_kmer_hits

label Minimum number of kmer hits [minkmerhits=1]

type `basic:integer`

description Reads need at least this many matching kmers to be considered matching the reference.

default `1`

processing.min_kmer_fraction

label Minimum kmer fraction [minkmerfraction=0.0]

type `basic:decimal`

description A read needs at least this fraction of its total kmers to hit a reference in order to be considered a match. If this and 'Minimum number of kmer hits' are set, the greater is used.

default `0.0`

processing.min_coverage_fraction

label Minimum coverage fraction [mincovfraction=0.0]

type `basic:decimal`

description A read needs at least this fraction of its total bases to be covered by reference kmers to be considered a match. If specified, 'Minimum coverage fraction' overrides 'Minimum number of kmer hits' and 'Minimum kmer fraction'.

default `0.0`

processing.hamming_distance

label Maximum Hamming distance for kmers (substitutions only) [hammingdistance=0]

type `basic:integer`

default `0`

processing.query_hamming_distance

label Hamming distance for query kmers [qhdist=0]

type `basic:integer`

default `0`

processing.edit_distance

label Maximum edit distance from reference kmers (substitutions and indels) [editdistance=0]

type `basic:integer`

default `0`

processing.hamming_distance2

label Hamming distance for short kmers when looking for shorter kmers [hammingdistance2=0]

type `basic:integer`

default `0`

processing.query_hamming_distance2

label Hamming distance for short query kmers when looking for shorter kmers [qhdist2=0]

type `basic:integer`

default `0`

processing.edit_distance2

label Maximum edit distance from short reference kmers (substitutions and indels) when looking for shorter kmers [editdistance2=0]

type `basic:integer`

default `0`

processing.forbid_N

label Forbid matching of read kmers containing N [forbidn=f]

type `basic:boolean`

description By default, these will match a reference 'A' if 'Maximum Hamming distance for kmers' > 0 or 'Maximum edit distance from reference kmers' > 0, to increase sensitivity.

default `False`

processing.find_best_match

label If multiple matches, associate read with sequence sharing most kmers [findbestmatch=f]

type `basic:boolean`

default `True`

operations.k_trim

label Trimming protocol to remove bases matching reference kmers from reads [ktrim=f]

type `basic:string`

default `f`

choices

- Don't trim: `f`
- Trim to the right: `r`
- Trim to the left: `l`

operations.k_mask

label Symbol to replace bases matching reference kmers [kmask=f]

type `basic:string`

description Allows any non-whitespace character other than t or f. Processes short kmers on both ends.

default `f`

operations.mask_fully_covered

label Only mask bases that are fully covered by kmers [maskfullycovered=f]

type `basic:boolean`

default `False`

operations.min_k

label Look for shorter kmers at read tips down to this length when k-trimming or masking [mink=0]

type `basic:integer`

description -1 means disabled. Enabling this will disable treating the middle base of a kmer as a wildcard to increase sensitivity in the presence of errors.

default `-1`

operations.quality_trim

label Trimming protocol to remove bases with quality below the minimum average region quality from read ends [qtrim=f]

type `basic:string`

description Performed after looking for kmers. If enabled, set also 'Average quality below which to trim region'.

default `f`

choices

- Trim neither end: `f`
- Trim both ends: `rl`
- Trim only right end: `r`
- Trim only left end: `l`
- Use sliding window: `w`

operations.trim_quality

label Average quality below which to trim region [trimq=6]

type `basic:integer`

description Set trimming protocol to enable this parameter.

disabled operations.quality_trim == 'f'

default `6`

operations.trim_poly_A

label Minimum length of poly-A or poly-T tails to trim on either end of reads [trimpolya=0]

type `basic:integer`

default `0`

operations.min_length_fraction

label Minimum length fraction [mlf=0]

type `basic:decimal`

description Reads shorter than this fraction of original length after trimming will be discarded.

default `0.0`

operations.max_length

label Maximum length [maxlength]

type `basic:integer`

description Reads longer than this after trimming will be discarded.

required False

operations.min_average_quality

label Minimum average quality [minavgquality=0]

type `basic:integer`

description Reads with average quality (after trimming) below this will be discarded.

default `0`

operations.min_average_quality_bases

label Number of initial bases to calculate minimum average quality from [maqb=0]

type `basic:integer`

description Used only if positive.

default `0`

operations.min_base_quality

label Minimum base quality below which reads are discarded after trimming [minbasequality=0]

type `basic:integer`

default `0`

operations.min_consecutive_bases

label Minimum number of consecutive called bases [mcb=0]

type `basic:integer`

default `0`

operations.trim_pad

label Number of bases to trim around matching kmers [tp=0]

type `basic:integer`

default `0`

operations.min_overlap

label Minimum number of overlapping bases [minoverlap=14]

type `basic:integer`

description Require this many bases of overlap for detection.

default `14`

operations.min_insert

label Minimum insert size [mininsert=40]

type `basic:integer`

description Require insert size of at least this for overlap. Should be reduced to 16 for small RNA sequencing.

default `40`

operations.force_trim_left

label Position from which to trim bases to the left [forcetrimleft=0]

type `basic:integer`

default `0`

operations.force_trim_right

label Position from which to trim bases to the right [forcetrimright=0]

type `basic:integer`

default `0`

operations.force_trim_right2

label Number of bases to trim from the right end [forcetrimright2=0]

type `basic:integer`

default `0`

operations.force_trim_mod

label Modulo to right-trim reads [forcetrimmod=0]

type `basic:integer`

description Trim reads to the largest multiple of modulo.

default `0`

operations.restrict_left

label Number of leftmost bases to look in for kmer matches [restrictleft=0]

type `basic:integer`

default `0`

operations.restrict_right

label Number of rightmosot bases to look in for kmer matches [restricright=0]

type `basic:integer`

default `0`

operations.min_GC

label Minimum GC content [mingc=0.0]

type `basic:decimal`

description Discard reads with lower GC content.

default `0.0`

operations.max_GC

label Maximum GC content [maxgc=1.0]

type `basic:decimal`

description Discard reads with higher GC content.

default `1.0`

operations.maxns

label Max Ns after trimming [maxns=-1]

type `basic:integer`

description If non-negative, reads with more Ns than this (after trimming) will be discarded.

default `-1`

operations.toss_junk

label Discard reads with invalid characters as bases [tossjunk=f]

type `basic:boolean`

default `False`

header_parsing.chastity_filter

label Discard reads that fail Illumina chastity filtering [chastityfilter=f]

type `basic:boolean`

description Discard reads with id containing ‘ 1:Y:’ or ‘ 2:Y:’.

default `False`

header_parsing.barcode_filter

label Remove reads with unexpected barcodes if barcodes are set, or barcodes containing ‘N’ otherwise [barcodefilter=f]

type `basic:boolean`

description A barcode must be the last part of the read header.

default `False`

header_parsing.barcode_files

label Barcode sequences [barcodes]

type `list:data:seq:nucleotide`

required False

header_parsing.barcode_sequences

label Literal barcode sequences [barcodes]

type `list:basic:string`

description Literal barcode sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

default `[]`

header_parsing.x_min

label Minimum X coordinate [xmin=-1]

type `basic:integer`

description If positive, discard reads with a smaller X coordinate.

default `-1`

header_parsing.y_min

label Minimum Y coordinate [ymin=-1]

type `basic:integer`

description If positive, discard reads with a smaller Y coordinate.

default `-1`

header_parsing.x_max

label Maximum X coordinate [xmax=-1]

type `basic:integer`

description If positive, discard reads with a larger X coordinate.

default `-1`

header_parsing.y_max

label Maximum Y coordinate [ymax=-1]

type `basic:integer`

description If positive, discard reads with a larger Y coordinate.

default `-1`

complexity.entropy

label Minimum entropy [entropy=-1]

type `basic:decimal`

description Set between 0 and 1 to filter reads with entropy below that value. Higher is more stringent.

default `-1.0`

complexity.entropy_window

label Length of sliding window used to calculate entropy [entropywindow=50]

type `basic:integer`

description To use the sliding window set minimum entropy in range between 0.0 and 1.0.

default `50`

complexity.entropy_k

label Length of kmers used to calcuate entropy [entropyk=5]

type `basic:integer`

default `5`

complexity.entropy_mask

label Mask low-entropy parts of sequences with N instead of discarding [entropymask=f]

type `basic:boolean`

default `False`

complexity.min_base_frequency

label Minimum base frequency [minbasefrequency=0]

type `basic:integer`

default `0`

fastqc.nogroup

label Disable grouping of bases for reads >50bp [nogroup]

type `basic:boolean`

description All reports will show data for every base in the read. Using this option will cause fastqc to crash and burn if you use it on really long reads.

default `False`

Output results fastq

label Remaining reads

type `list:basic:file`

statistics

label Statistics

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

BBDuk - STAR - FeatureCounts (3' mRNA-Seq, paired-end)

`data:workflow:quant:featurecounts:pairedworkflow-bbduk-star-fc-quant-paired` (*data:reads:fastq:pa* *data:genomeindex:s* *list:data:seq:nucleo* *data:annotation* **annotation**, *basic:string* **stranded**, *basic:integer* **n_reads**, *basic:integer* **seed**, *basic:decimal* **fraction**, *basic:boolean* **two_pa** *data:genomeindex:s* *data:genomeindex:s* v1.1.0]

This 3' mRNA-Seq pipeline is comprised of QC, preprocessing, alignment and quantification steps.

Reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Preprocessed reads are aligned by __STAR__ aligner. For read-count quantification, the __FeatureCounts__ tool is used.

QC steps include downsampling, QoRTs QC analysis and alignment of input reads to the rRNA/globin reference sequences. The reported alignment rate is used to asses the rRNA/globin sequence depletion rate.

Input arguments reads

label Paired-end reads

type `data:reads:fastq:paired`

star_index

label Star index

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

adapters

label Adapters

type `list:data:seq:nucleotide`

description Provide a list of sequencing adapters files (.fasta) to be removed by BBDuk.

required False

annotation

label Annotation

type `data:annotation`

stranded

label Select the type of kit used for library preparation.

type `basic:string`

choices

- Strand-specific forward: `forward`
- Strand-specific reverse: `reverse`

downsampling.n_reads

label Number of reads

type `basic:integer`

default `1000000`

downsampling.advanced.seed

label Seed

type `basic:integer`

default `11`

downsampling.advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads in range [0.0, 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the "Number of reads" input parameter.

required False

downsampling.advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

qc.rrna_reference

label Indexed rRNA reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

qc.globin_reference

label Indexed Globin reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

Output results

BBDuk - STAR - FeatureCounts (3' mRNA-Seq, single-end)

`data:workflow:quant:featurecounts:singleworkflow-bbduk-star-fc-quant-single` (*data:reads:fastq:sin* *data:genomeindex:s* *list:data:seq:nucleo* *data:annotation* **annotation**, *basic:string* **stranded**, *basic:integer* **n_reads**, *basic:integer* **seed**, *basic:decimal* **fraction**, *basic:boolean* **two_pa** *data:genomeindex:s* *data:genomeindex:s* v1.1.0]

This 3' mRNA-Seq pipeline is comprised of QC, preprocessing, alignment and quantification steps.

Reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Preprocessed reads are aligned by __STAR__ aligner. For read-count quantification, the __FeatureCounts__ tool is used.

QC steps include downsampling, QoRTs QC analysis and alignment of input reads to the rRNA/globin reference sequences. The reported alignment rate is used to asses the rRNA/globin sequence depletion rate.

Input arguments reads

label Input single-end reads

type `data:reads:fastq:single`

star_index

label Star index

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

adapters

label Adapters

type `list:data:seq:nucleotide`

description Provide a list of sequencing adapters files (.fasta) to be removed by BBDuk.

required False

annotation

label Annotation

type `data:annotation`

stranded

label Select the type of kit used for library preparation.

type `basic:string`

choices

- Strand-specific forward: `forward`
- Strand-specific reverse: `reverse`

downsampling.n_reads

label Number of reads

type `basic:integer`

default `1000000`

downsampling.advanced.seed

label Seed

type `basic:integer`

default `11`

downsampling.advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads in range [0.0, 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the "Number of reads" input parameter.

required False

downsampling.advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

qc.rrna_reference

label Indexed rRNA reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

qc.globin_reference

label Indexed Globin reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

Output results

BBDuk - STAR - HTSeq-count (paired-end)

`data:workflow:rnaseq:htseq:pairedworkflow-bbduk-star-htseq-paired` (*data:reads:fastq:paired* **reads**, *data:genomeindex:star* **star_index**, *list:data:seq:nucleotide* **adapters**, *data:annotation* **annotation**, *basic:string* **stranded**) [Source: v1.0.1]

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Compared to similar tools, BBDuk is regarded for its computational efficiency. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.nature.com/articles/nmeth.4106). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label Paired-end reads

type `data:reads:fastq:paired`

star_index

label Star index

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

adapters

label Adapters

type `list:data:seq:nucleotide`

description Provide a list of sequencing adapters files (.fasta) to be removed by BBDuk.

required False

annotation

label Annotation

type `data:annotation`

stranded

label Select the QuantSeq kit used for library preparation.

type `basic:string`

choices

- QuantSeq FWD: `yes`
- QuantSeq REV: `reverse`

Output results

BBDuk - STAR - HTSeq-count (single-end)

`data:workflow:rnaseq:htseq:singleworkflow-bbduk-star-htseq` (*data:reads:fastq:single* **reads**, *data:genomeindex:star* **star_index**, *list:data:seq:nucleotide* **adapters**, *data:annotation* **annotation**, *basic:string* **stranded**) [Source: v1.0.1]

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Compared to similar tools, BBDuk is regarded for its computational efficiency. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.nature.com/articles/nmeth.4106). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label Input single-end reads

type `data:reads:fastq:single`

star_index

label Star index

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

adapters

label Adapters

type `list:data:seq:nucleotide`

description Provide a list of sequencing adapters files (.fasta) to be removed by BBDuk.

required False

annotation

label annotation

type `data:annotation`

stranded

label Select the QuantSeq kit used for library preparation.

type `basic:string`

choices

- QuantSeq FWD: `yes`
- QuantSeq REV: `reverse`

Output results

BBDuk - STAR - featureCounts - QC (paired-end)

data:workflow:rnaseq:featurecounts:qcworkflow-bbduk-star-featurecounts-qc-paired (*data:reads:f*
list:data:seq
basic:boolean
list:basic:str
tom_adapte
basic:integer **k**
basic:integer **n**
basic:integer **h**
ming_distan
basic:integer **n**
basic:integer **t**
basic:integer **n**
data:genome
basic:boolean
basic:boolean
stranded,
basic:boolean
cannonical,
basic:boolean
basic:integer **c**
SegmentMin,
basic:boolean
mode,
basic:boolean
gleend,
basic:boolean
basic:string **ou**
FilterType,
basic:integer **o**
FilterMultimapN-

This RNA-seq pipeline is comprised of three steps preprocessing, alignment, and quantification.

First, reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Compared to similar tools, BBDuk is regarded for its computational efficiency. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __featureCounts__. Gaining wide adoption among the bioinformatics community, featureCounts yields expressions in a computationally efficient manner. All three tools in this workflow support parallelization to accelerate the analysis.

rRNA contamination rate in the sample is determined using the STAR aligner. Quality-trimmed reads are down-sampled (using Seqtk tool) and aligned to the rRNA reference sequences. The alignment rate indicates the percentage of the reads in the sample that are derived from the rRNA sequences.

Input arguments preprocessing.reads

label Reads

type `data:reads:fastq:paired`

preprocessing.adapters

label Adapters

type `list:data:seq:nucleotide`

required False

preprocessing.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

preprocessing.custom_adapter_sequences

label Custom adapter sequences [literal]

type `list:basic:string`

description Custom adapter sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

hidden !preprocessing.show_advanced

default `[]`

preprocessing.kmer_length

label K-mer length

type `basic:integer`

description K-mer length must be smaller or equal to the length of adapters.

hidden !preprocessing.show_advanced

default `23`

preprocessing.min_k

label Minimum k-mer length at right end of reads used for trimming

type `basic:integer`

disabled preprocessing.adapters.length === 0 && preprocessing.custom_adapter_sequences.length === 0

hidden !preprocessing.show_advanced

default `11`

preprocessing.hamming_distance

label Maximum Hamming distance for k-mers

type `basic:integer`

hidden !preprocessing.show_advanced

default `1`

preprocessing.maxns

label Max Ns after trimming [maxns=-1]

type `basic:integer`

description If non-negative, reads with more Ns than this (after trimming) will be discarded.

hidden !preprocessing.show_advanced

default `-1`

preprocessing.trim_quality

label Quality below which to trim reads from the right end

type `basic:integer`

description Phred algorithm is used, which is more accurate than naive trimming.

hidden !preprocessing.show_advanced

default `10`

preprocessing.min_length

label Minimum read length

type `basic:integer`

description Reads shorter than minimum read length after trimming are discarded.

hidden !preprocessing.show_advanced

default `20`

alignment.genome

label Indexed reference genome

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

alignment.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

alignment.unstranded

label The data is unstranded

type `basic:boolean`

description For unstranded RNA-seq data, Cufflinks/Cuffdiff require spliced alignments with XS strand attribute, which STAR will generate with –outSAMstrandField intronMotif option. As required, the XS strand attribute will be generated for all alignments that contain splice junctions. The spliced alignments that have undefined strand (i.e. containing only non-canonical unannotated junctions) will be suppressed. If you have stranded RNA-seq data, you do not need to use any specific STAR options. Instead, you need to run Cufflinks with the library option –library-type options. For example, c ufflinks –library-type fr-firststrand should be used for the standard dUTP protocol, including Illumina's stranded Tru-Seq. This option has to be used only for Cufflinks runs and not for STAR runs.

hidden !alignment.show_advanced

default `False`

alignment.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

hidden !alignment.show_advanced

default `False`

alignment.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

alignment.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled detect_chimeric.chimeric != true

default `20`

alignment.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `False`

alignment.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled t_coordinates.quantmode != true

default `False`

alignment.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled t_coordinates.quantmode != true

default `False`

alignment.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

alignment.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

alignment.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

alignment.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

alignment.filtering.outFilterScoreMin

label –outFilterScoreMin

type `basic:integer`

description Alignment will be output only if its score is higher than or equal to this value (default: 0).

required False

alignment.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

alignment.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

alignment.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

alignment.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignment.alignEndsType

label –alignEndsType

type `basic:string`

description Type of read ends alignment (default: Local).

required False

default `Local`

choices

- Local: `Local`
- EndToEnd: `EndToEnd`
- Extend5pOfRead1: `Extend5pOfRead1`
- Extend5pOfReads12: `Extend5pOfReads12`

alignment.output_sam_bam.outSAMunmapped

label –outSAMunmapped

type `basic:string`

description Output of unmapped reads in the SAM format.

required False

default `None`

choices

- None: `None`
- Within: `Within`

alignment.output_sam_bam.outSAMattributes

label –outSAMattributes

type `basic:string`

description a string of desired SAM attributes, in the order desired for the output SAM.

required False

default `Standard`

choices

- None: `None`
- Standard: `Standard`

- All: `All`

alignment.output_sam_bam.outSAMattrRGline

label –outSAMattrRGline

type `basic:string`

description SAM/BAM read group line. The first word contains the read group identifier and must start with “ID:”, e.g. –outSAMattrRGline ID:xxx CN:yy “DS:z z z”

required False

quantification.annotation

label Annotation

type `data:annotation`

quantification.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

quantification.assay_type

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

hidden !quantification.show_advanced

default `non_specific`

choices

- Strand non-specific: `non_specific`
- Strand-specific forward: `forward`
- Strand-specific reverse: `reverse`
- Detect automatically: `auto`

quantification.cdna_index

label cDNA index file

type `data:index:salmon`

description Transcriptome index file created using the Salmon indexing tool. cDNA (transcriptome) sequences used for index file creation must be derived from the same species as the input sequencing reads to obtain the reliable analysis results.

required False

hidden quantification.assay_type != ‘auto’

quantification.n_reads

label Number of reads in subsampled alignment file

type `basic:integer`

description Alignment (.bam) file subsample size. Increase the number of reads to make automatic detection more reliable. Decrease the number of reads to make automatic detection run faster.

hidden quantification.assay_type != 'auto'

default `5000000`

quantification.feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GTF/GFF3 file) to be used. All other features will be ignored.

hidden !quantification.show_advanced

default `exon`

quantification.feature_type

label Feature type

type `basic:string`

description The type of feature the quantification program summarizes over (e.g. gene or transcript-level analysis). The value of this parameter needs to be chosen in line with 'ID attribute' below.

hidden !quantification.show_advanced

default `gene`

choices

- gene: `gene`
- transcript: `transcript`

quantification.id_attribute

label ID attribute

type `basic:string`

description GTF/GFF3 attribute to be used as feature ID. Several GTF/GFF3 lines with the same feature ID are considered as parts of the same feature. The feature ID is used to identify the counts in the output table. In GTF files this is usually 'gene_id', in GFF3 files this is often 'ID', and 'transcript_id' is frequently a valid choice for both annotation formats.

hidden !quantification.show_advanced

default `gene_id`

choices

- gene_id: `gene_id`
- transcript_id: `transcript_id`
- ID: `ID`
- geneid: `geneid`

downsampling.n_reads

label Number of reads

type `basic:integer`

default `1000000`

downsampling.advanced.seed

label Seed

type `basic:integer`

default `11`

downsampling.advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads [0 - 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the “Number of reads” input parameter.

required False

downsampling.advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

qc.rrna_reference

label Indexed rRNA reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

qc.globin_reference

label Indexed Globin reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

Output results

BBDuk - STAR - featureCounts - QC (single-end)

data:workflow:rnaseq:featurecounts:qcworkflow-bbduk-star-featurecounts-qc-single (*data:reads:f* *list:data:seq* *basic:boolean* *list:basic:str* **tom_adapte** *basic:integer* **k** *basic:integer* **n** *basic:integer* **h** **ming_distan** *basic:integer* **n** *basic:integer* **t** *basic:integer* **n** *data:genome* *basic:boolean* *basic:boolean* **stranded**, *basic:boolean* **cannonical**, *basic:boolean* *basic:integer* **c** **SegmentMin**, *basic:boolean* **mode**, *basic:boolean* **gleend**, *basic:boolean* *basic:string* **ou** **FilterType**, *basic:integer* **o** **FilterMultimapN-**

This RNA-seq pipeline is comprised of three steps preprocessing, alignment, and quantification.

First, reads are preprocessed by __BBDuk__ which removes adapters, trims reads for quality from the 3'-end, and discards reads that are too short after trimming. Compared to similar tools, BBDuk is regarded for its computational efficiency. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __featureCounts__. Gaining wide adoption among the bioinformatics community, featureCounts yields expressions in a computationally efficient manner. All three tools in this workflow support parallelization to accelerate the analysis.

rRNA contamination rate in the sample is determined using the STAR aligner. Quality-trimmed reads are down-sampled (using Seqtk tool) and aligned to the rRNA reference sequences. The alignment rate indicates the percentage of the reads in the sample that are derived from the rRNA sequences.

Input arguments preprocessing.reads

label Reads

type `data:reads:fastq:single`

preprocessing.adapters

label Adapters

type `list:data:seq:nucleotide`

required False

preprocessing.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

preprocessing.custom_adapter_sequences

label Custom adapter sequences [literal]

type `list:basic:string`

description Custom adapter sequences can be specified by inputting them one by one and pressing Enter after each sequence.

required False

hidden !preprocessing.show_advanced

default `[]`

preprocessing.kmer_length

label K-mer length

type `basic:integer`

description K-mer length must be smaller or equal to the length of adapters.

hidden !preprocessing.show_advanced

default `23`

preprocessing.min_k

label Minimum k-mer length at right end of reads used for trimming

type `basic:integer`

disabled preprocessing.adapters.length === 0 && preprocessing.custom_adapter_sequences.length === 0

hidden !preprocessing.show_advanced

default `11`

preprocessing.hamming_distance

label Maximum Hamming distance for k-mers

type `basic:integer`

hidden !preprocessing.show_advanced

default `1`

preprocessing.maxns

label Max Ns after trimming [maxns=-1]

type `basic:integer`

description If non-negative, reads with more Ns than this (after trimming) will be discarded.

hidden !preprocessing.show_advanced

default `-1`

preprocessing.trim_quality

label Quality below which to trim reads from the right end

type `basic:integer`

description Phred algorithm is used, which is more accurate than naive trimming.

hidden !preprocessing.show_advanced

default `10`

preprocessing.min_length

label Minimum read length

type `basic:integer`

description Reads shorter than minimum read length after trimming are discarded.

hidden !preprocessing.show_advanced

default `20`

alignment.genome

label Indexed reference genome

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

alignment.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

alignment.unstranded

label The data is unstranded

type `basic:boolean`

description For unstranded RNA-seq data, Cufflinks/Cuffdiff require spliced alignments with XS strand attribute, which STAR will generate with –outSAMstrandField intronMotif option. As required, the XS strand attribute will be generated for all alignments that contain splice junctions. The spliced alignments that have undefined strand (i.e. containing only non-canonical unannotated junctions) will be suppressed. If you have stranded RNA-seq data, you do not need to use any specific STAR options. Instead, you need to run Cufflinks with the library option –library-type options. For example, c ufflinks –library-type fr-firststrand should be used for the standard dUTP protocol, including Illumina's stranded Tru-Seq. This option has to be used only for Cufflinks runs and not for STAR runs.

hidden !alignment.show_advanced

default `False`

alignment.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

hidden !alignment.show_advanced

default `False`

alignment.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

alignment.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled detect_chimeric.chimeric != true

default `20`

alignment.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `False`

alignment.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled t_coordinates.quantmode != true

default `False`

alignment.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled t_coordinates.quantmode != true

default `False`

alignment.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

alignment.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

alignment.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

alignment.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

alignment.filtering.outFilterScoreMin

label –outFilterScoreMin

type `basic:integer`

description Alignment will be output only if its score is higher than or equal to this value (default: 0).

required False

alignment.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

alignment.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

alignment.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

alignment.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignment.alignEndsType

label –alignEndsType

type `basic:string`

description Type of read ends alignment (default: Local).

required False

default `Local`

choices

- Local: `Local`
- EndToEnd: `EndToEnd`
- Extend5pOfRead1: `Extend5pOfRead1`
- Extend5pOfReads12: `Extend5pOfReads12`

alignment.output_sam_bam.outSAMunmapped

label –outSAMunmapped

type `basic:string`

description Output of unmapped reads in the SAM format.

required False

default `None`

choices

- None: `None`
- Within: `Within`

alignment.output_sam_bam.outSAMattributes

label –outSAMattributes

type `basic:string`

description a string of desired SAM attributes, in the order desired for the output SAM.

required False

default `Standard`

choices

- None: `None`
- Standard: `Standard`

- All: `All`

alignment.output_sam_bam.outSAMattrRGline

label –outSAMattrRGline

type `basic:string`

description SAM/BAM read group line. The first word contains the read group identifier and must start with "ID:", e.g. –outSAMattrRGline ID:xxx CN:yy "DS:z z z"

required False

quantification.annotation

label Annotation

type `data:annotation`

quantification.show_advanced

label Show advanced parameters

type `basic:boolean`

default `False`

quantification.assay_type

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

hidden !quantification.show_advanced

default `non_specific`

choices

- Strand non-specific: `non_specific`
- Strand-specific forward: `forward`
- Strand-specific reverse: `reverse`
- Detect automatically: `auto`

quantification.cdna_index

label cDNA index file

type `data:index:salmon`

description Transcriptome index file created using the Salmon indexing tool. cDNA (transcriptome) sequences used for index file creation must be derived from the same species as the input sequencing reads to obtain the reliable analysis results.

required False

hidden quantification.assay_type != 'auto'

quantification.n_reads

label Number of reads in subsampled alignment file

type `basic:integer`

description Alignment (.bam) file subsample size. Increase the number of reads to make automatic detection more reliable. Decrease the number of reads to make automatic detection run faster.

hidden quantification.assay_type != ‘auto’

default `5000000`

quantification.feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GTF/GFF3 file) to be used. All other features will be ignored.

hidden !quantification.show_advanced

default `exon`

quantification.feature_type

label Feature type

type `basic:string`

description The type of feature the quantification program summarizes over (e.g. gene or transcript-level analysis). The value of this parameter needs to be chosen in line with ‘ID attribute’ below.

hidden !quantification.show_advanced

default `gene`

choices

- gene: `gene`
- transcript: `transcript`

quantification.id_attribute

label ID attribute

type `basic:string`

description GTF/GFF3 attribute to be used as feature ID. Several GTF/GFF3 lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identify the counts in the output table. In GTF files this is usually ‘gene_id’, in GFF3 files this is often ‘ID’, and ‘transcript_id’ is frequently a valid choice for both annotation formats.

hidden !quantification.show_advanced

default `gene_id`

choices

- gene_id: `gene_id`
- transcript_id: `transcript_id`
- ID: `ID`
- geneid: `geneid`

downsampling.n_reads

label Number of reads

type `basic:integer`

default `1000000`

downsampling.advanced.seed

label Seed

type `basic:integer`

default `11`

downsampling.advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads [0 - 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the “Number of reads” input parameter.

required False

downsampling.advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

qc.rrna_reference

label Indexed rRNA reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

qc.globin_reference

label Indexed Globin reference sequence

type `data:genomeindex:star`

description Reference sequence index prepared by STAR aligner indexing tool.

Output results

BED file

`data:bedupload-bed` (*basic:file* **src**, *basic:string* **species**, *basic:string* **build**) [Source: v1.3.1]

Import a BED file (.bed) which is a tab-delimited text file that defines a feature track. It can have any file extension, but .bed is recommended. The BED file format is described on the [UCSC Genome Bioinformatics web site](http://genome.ucsc.edu/FAQ/FAQformat#format1).

Input arguments src

label BED file

type `basic:file`

description Upload BED file annotation track. The first three required BED fields are chrom, chromStart and chromEnd.

required True

validate_regex `\.(bed|narrowPeak)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Genome build

type `basic:string`

Output results bed

label BED file

type `basic:file`

bed_jbrowse

label Bgzip bed file for JBrowse

type `basic:file`

tbi_jbrowse

label Bed file index for Jbrowse

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

BWA ALN

data:alignment:bam:bwaalnalignment-bwa-aln (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:integer* **q**, *basic:boolean* **use_edit**, *basic:integer* **edit_value**, *basic:decimal* **fraction**, *basic:boolean* **seeds**, *basic:integer* **seed_length**, *basic:integer* **seed_dist**) [Source: v1.4.2]

Read aligner for mapping low-divergent sequences against a large reference genome. Designed for Illumina sequence reads up to 100bp.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

q

label Quality threshold

type `basic:integer`

description Parameter for dynamic read trimming.

default `0`

use_edit

label Use maximum edit distance (excludes fraction of missing alignments)

type `basic:boolean`

default `False`

edit_value

label Maximum edit distance

type `basic:integer`

hidden !use_edit

default `5`

fraction

label Fraction of missing alignments

type `basic:decimal`

description The fraction of missing alignments given 2% uniform base error rate. The maximum edit distance is automatically chosen for different read lengths.

hidden use_edit

default `0.04`

seeds

label Use seeds

type `basic:boolean`

default `False`

seed_length

label Seed length

type `basic:integer`

description Take the first X subsequence as seed. If X is larger than the query sequence, seeding will be disabled. For long reads, this option is typically ranged from 25 to 35 for value 2 in seed maximum edit distance.

hidden !seeds

default `35`

seed_dist

label Seed maximum edit distance

type `basic:integer`

hidden !seeds

default `2`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

BWA MEM

data:alignment:bam:bwamemalignment-bwa-mem (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:integer* **seed_l**, *basic:integer* **band_w**, *basic:decimal* **re_seeding**, *basic:boolean* **m**, *basic:integer* **match**, *basic:integer* **missmatch**, *basic:integer* **gap_o**, *basic:integer* **gap_e**, *basic:integer* **clipping**, *basic:integer* **unpaired_p**, *basic:boolean* **report_all**, *basic:integer* **report_tr**) [Source: v2.2.2]

BWA MEM is a read aligner for mapping low-divergent sequences against a large reference genome. Designed for longer sequences ranged from 70bp to 1Mbp. The algorithm works by seeding alignments with maximal exact matches (MEMs) and then extending seeds with the affine-gap Smith-Waterman algorithm (SW). See [here](http://bio-bwa.sourceforge.net/) for more information.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

seed_l

label Minimum seed length

type `basic:integer`

description Minimum seed length. Matches shorter than minimum seed length will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20.

default `19`

band_w

label Band width

type `basic:integer`

description Gaps longer than this will not be found.

default `100`

re_seeding

label Re-seeding factor

type `basic:decimal`

description Trigger re-seeding for a MEM longer than minSeedLen*FACTOR. This is a key heuristic parameter for tuning the performance. Larger value yields fewer seeds, which leads to faster alignment speed but lower accuracy.

default `1.5`

m

label Mark shorter split hits as secondary

type `basic:boolean`

description Mark shorter split hits as secondary (for Picard compatibility)

default `False`

scoring.match

label Score of a match

type `basic:integer`

default `1`

scoring.missmatch

label Mismatch penalty

type `basic:integer`

default `4`

scoring.gap_o

label Gap open penalty

type `basic:integer`

default `6`

scoring.gap_e

label Gap extension penalty

type `basic:integer`

default `1`

scoring.clipping

label Clipping penalty

type `basic:integer`

description Clipping is applied if final alignment score is smaller than (best score reaching the end of query) - (Clipping penalty)

default `5`

scoring.unpaired_p

label Penalty for an unpaired read pair

type `basic:integer`

description Affinity to force pair. Score: scoreRead1+scoreRead2-Penalty

default `9`

reporting.report_all

label Report all found alignments

type `basic:boolean`

description Output all found alignments for single-end or unpaired paired-end reads. These alignments will be flagged as secondary alignments.

default `False`

reporting.report_tr

label Report threshold score

type `basic:integer`

description Don't output alignment with score lower than defined number. This option only affects output.

default `30`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

BWA SW

`data:alignment:bam:bwaswalignment-bwa-sw` (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:integer* **match**, *basic:integer* **missmatch**, *basic:integer* **gap_o**, *basic:integer* **gap_e**) [Source: v1.3.2]

Read aligner for mapping low-divergent sequences against a large reference genome. Designed for longer sequences ranged from 70bp to 1Mbp. The paired-end mode only works for reads Illumina short-insert libraries.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

match

label Score of a match

type `basic:integer`

default `1`

missmatch

label Mismatch penalty

type `basic:integer`

default `3`

gap_o

label Gap open penalty

type `basic:integer`

default `5`

gap_e

label Gap extension penalty

type `basic:integer`

default `2`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Bam split

`data:alignment:bam:primarybam-split` (*data:alignment:bam* **bam**, *data:sam:header* **header**, *data:sam:header* **header2**) [Source: v0.4.0]

Split hybrid bam file into two bam files.

Input arguments bam

label Hybrid alignment bam

type `data:alignment:bam`

header

label Primary header sam file (optional)

type `data:sam:header`

description If no header file is provided, the headers will be extracted from the hybrid alignment bam file.

required False

header2

label Secondary header sam file (optional)

type `data:sam:header`

description If no header file is provided, the headers will be extracted from the hybrid alignment bam file.

required False

Output results bam

label Uploaded file

type `basic:file`

bai

label Index BAI

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Bamliquidator

`data:bam:plot:bamliquidatorbamliquidator` (*basic:string* **analysis_type**, *list:data:alignment:bam* **bam**, *basic:string* **cell_type**, *basic:integer* **bin_size**, *data:annotation:gtf* **regions_gtf**, *data:bed* **regions_bed**, *basic:integer* **extension**, *basic:string* **sense**, *basic:boolean* **skip_plot**, *list:basic:string* **black_list**, *basic:integer* **threads**) [Source: v0.2.1]

Set of tools for analyzing the density of short DNA sequence read alignments in the BAM file format.

Input arguments analysis_type

label Analysis type

type `basic:string`

default `bin`

choices

- Bin mode: `bin`
- Region mode: `region`
- BED mode: `bed`

bam

label BAM File

type `list:data:alignment:bam`

cell_type

label Cell type

type basic:string

default cell_type

bin_size

label Bin size

type basic:integer

description Number of base pairs in each bin. The smaller the bin size the longer the runtime and the larger the data files. Default is 100000.

required False

hidden analysis_type != 'bin'

regions_gtf

label Region gff file / Annotation file (.gffl.gtf)

type data:annotation:gtf

required False

hidden analysis_type != 'region'

regions_bed

label Region bed file / Annotation file (.bed)

type data:bed

required False

hidden analysis_type != 'bed'

extension

label Extension

type basic:integer

description Extends reads by number of bp

default 200

sense

label Mapping strand to gff file

type basic:string

default .

choices

- Forward: +
- Reverse: -
- Both: .

skip_plot

label Skip plot

type basic:boolean

required False

black_list

label Black list

type `list:basic:string`

description One or more chromosome patterns to skip during bin liquidation. Default is to skip any chromosomes that contain any of the following substrings chrUn _random Zv9_ _hap.

required False

threads

label Threads

type `basic:integer`

description Number of threads to run concurrently during liquidation.

default `1`

Output results analysis_type

label Analysis type

type `basic:string`

hidden True

output_dir

label Output directory

type `basic:file`

counts

label Counts HDF5 file

type `basic:file`

matrix

label Matrix file

type `basic:file`

required False

hidden analysis_type != 'region'

summary

label Summary file

type `basic:file:html`

required False

hidden analysis_type != 'bin'

Bamplot

data:bam:plot:bamplotbamplot (*basic:string* **genome**, *data:annotation:gtf* **input_gff**, *basic:string* **input_region**, *list:data:alignment:bam* **bam**, *basic:integer* **stretch_input**, *basic:string* **color**, *basic:string* **sense**, *basic:integer* **extension**, *basic:boolean* **rpm**, *basic:string* **yscale**, *list:basic:string* **names**, *basic:string* **plot**, *basic:string* **title**, *basic:string* **scale**, *list:data:bed* **bed**, *basic:boolean* **multi_page**) [Source: v1.3.1]

Plot a single locus from a bam.

Input arguments genome

label Genome

type `basic:string`

choices

- HG19: `HG19`
- HG18: `HG18`
- MM8: `MM8`
- MM9: `MM9`
- MM10: `MM10`
- RN6: `RN6`
- RN4: `RN4`

input_gff

label Region string

type `data:annotation:gtf`

description Enter .gff file.

required False

input_region

label Region string

type `basic:string`

description Enter genomic region e.g. chr1:+:1-1000.

required False

bam

label Bam

type `list:data:alignment:bam`

description bam to plot from

required False

stretch_input

label Stretch-input

type `basic:integer`

description Stretch the input regions to a minimum length in bp, e.g. 10000 (for 10kb).

required False

color

label Color

type `basic:string`

description Enter a colon separated list of colors e.g. 255,0,0:255,125,0, default samples the rainbow.

default `255,0,0:255,125,0`

sense

label Sense

type `basic:string`

description Map to forward, reverse or'both strands. Default maps to both.

default `both`

choices

- Forward: `forward`
- Reverse: `reverse`
- Both: `both`

extension

label Extension

type `basic:integer`

description Extends reads by n bp. Default value is 200bp.

default `200`

rpm

label rpm

type `basic:boolean`

description Normalizes density to reads per million (rpm) Default is False.

required False

yscale

label y scale

type `basic:string`

description Choose either relative or uniform y axis scaling. Default is relative scaling.

default `relative`

choices

- relative: `relative`
- uniform: `uniform`

names

label Names

type `list:basic:string`

description Enter a comma separated list of names for your bams.

required False

plot

label Single or multiple polt

type `basic:string`

description Choose either all lines on a single plot or multiple plots.

default `merge`

choices

- single: `single`
- multiple: `multiple`
- merge: `merge`

title

label Title

type `basic:string`

description Specify a title for the output plot(s), default will be the coordinate region.

default `output`

scale

label Scale

type `basic:string`

description Enter a comma separated list of multiplicative scaling factors for your bams. Default is none.

required False

bed

label Bed

type `list:data:bed`

description Add a space-delimited list of bed files to plot.

required False

multi_page

label Multi page

type `basic:boolean`

description If flagged will create a new pdf for each region.

default `False`

Output results plot

label region plot

type `basic:file`

BaseSpace file

data:filebasespace-file-import (*basic:string* **file_id**, *basic:secret* **access_token_secret**) [Source: v1.0.3]

Import a file from Illumina BaseSpace.

Input arguments file_id

label BaseSpace file ID

type `basic:string`

access_token_secret

label BaseSpace access token

type `basic:secret`

description BaseSpace access token secret handle needed to download the file.

Output results file

label File

type `basic:file`

Bowtie (Dicty)

data:alignment:bam:bowtie1alignment-bowtie (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:string* **mode**, *basic:integer* **m**, *basic:integer* **l**, *basic:boolean* **use_se**, *basic:integer* **trim_5**, *basic:integer* **trim_3**, *basic:integer* **trim_nucl**, *basic:integer* **trim_iter**, *basic:string* **r**) [Source: v1.4.1]

An ultrafast memory-efficient short read aligner.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

mode

label Alignment mode

type `basic:string`

description When the -n option is specified (which is the default), bowtie determines which alignments are valid according to the following policy, which is similar to Maq's default policy. 1. Alignments may have no more than N mismatches (where N is a number 0-3, set with -n) in the first L bases (where L is a number 5 or greater, set with -l) on the high-quality (left) end of the read. The first L bases are called the "seed". 2. The sum of the Phred quality values at all mismatched positions (not just in the seed) may not exceed E (set with -e). Where qualities are unavailable (e.g. if the reads are from a FASTA file), the Phred quality defaults to 40. In -v mode, alignments may have no more than V mismatches, where V may be a number from 0 through 3 set using the -v option. Quality values are ignored. The -v option is mutually exclusive with the -n option.

default `-n`

choices

- Use qualities (-n): `-n`
- Use mismatches (-v): `-v`

m

label Allowed mismatches

type `basic:integer`

description When used with "Use qualities (-n)" it is the maximum number of mismatches permitted in the "seed", i.e. the first L base pairs of the read (where L is set with -l/–seedlen). This may be 0, 1, 2 or 3 and the default is 2 When used with "Use mismatches (-v)" report alignments with at most <int> mismatches.

default `2`

l

label Seed length (for -n only)

type `basic:integer`

description Only for "Use qualities (-n)". Seed length (-l) is the number of bases on the high-quality end of the read to which the -n ceiling applies. The lowest permitted setting is 5 and the default is 28. bowtie is faster for larger values of -l.

default `28`

use_se

label Map as single-ended (for paired end reads only)

type `basic:boolean`

description If this option is selected paired-end reads will be mapped as single-ended.

default `False`

start_trimming.trim_5

label Bases to trim from 5'

type `basic:integer`

description Number of bases to trim from from 5' (left) end of each read before alignment

default `0`

start_trimming.trim_3

label Bases to trim from 3'

type `basic:integer`

description Number of bases to trim from from 3' (right) end of each read before alignment

default `0`

trimming.trim_nucl

label Bases to trim

type `basic:integer`

description Number of bases to trim from 3' end in each iteration.

default `2`

trimming.trim_iter

label Iterations

type `basic:integer`

description Number of iterations.

default `0`

reporting.r

label Reporting mode

type `basic:string`

description Report up to <int> valid alignments per read or pair (-k) (default: 1). Validity of alignments is determined by the alignment policy (combined effects of -n, -v, -l, and -e). If more than one valid alignment exists and the –best and –strata options are specified, then only those alignments belonging to the best alignment "stratum" will be reported. Bowtie is designed to be very fast for small -k but bowtie can become significantly slower as -k increases. If you would like to use Bowtie for larger values of -k, consider building an index with a denser suffix-array sample, i.e. specify a smaller -o/–offrate when invoking bowtie-build for the relevant index (see the Performance tuning section for details).

default `-a -m 1 --best --strata`

choices

- Report unique alignments: `-a -m 1 --best --strata`
- Report all alignments: `-a --best`
- Report all alignments in the best stratum: `-a --best --strata`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Bowtie2

`data:alignment:bam:bowtie2alignment-bowtie2` (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:string* **mode**, *basic:string* **speed**, *basic:boolean* **use_se**, *basic:boolean* **discordantly**, *basic:boolean* **rep_se**, *basic:integer* **minins**, *basic:integer* **maxins**, *basic:integer* **N**, *basic:integer* **L**, *basic:integer* **gbar**, *basic:string* **mp**, *basic:string* **rdg**, *basic:string* **rfg**, *basic:string* **score_min**, *basic:integer* **trim_5**, *basic:integer* **trim_3**, *basic:integer* **trim_iter**, *basic:integer* **trim_nucl**, *basic:string* **rep_mode**, *basic:integer* **k_reports**) [Source: v1.5.0]

Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small–typically about 2.2 GB for the human genome (2.9 GB for paired-end). See [here](http://bowtie-bio.sourceforge.net/index.shtml) for more information.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

mode

label Alignment mode

type `basic:string`

description End to end: Bowtie 2 requires that the entire read align from one end to the other, without any trimming (or "soft clipping") of characters from either end. local: Bowtie 2 does not require that the entire read align from one end to the other. Rather, some characters may be omitted ("soft clipped") from the ends in order to achieve the greatest possible alignment score.

default `--end-to-end`

choices

- end to end mode: `--end-to-end`
- local: `--local`

speed

label Speed vs. Sensitivity

type `basic:string`

description A quick setting for aligning fast or accurately. This option is a shortcut for parameters as follows:

For –end-to-end: –very-fast -D 5 -R 1 -N 0 -L 22 -i S,0,2.50 –fast -D 10 -R 2 -N 0 -L 22 -i S,0,2.50 –sensitive -D 15 -R 2 -N 0 -L 22 -i S,1,1.15 (default) –very-sensitive -D 20 -R 3 -N 0 -L 20 -i S,1,0.50

For –local: –very-fast-local -D 5 -R 1 -N 0 -L 25 -i S,1,2.00 –fast-local -D 10 -R 2 -N 0 -L 22 -i S,1,1.75 –sensitive-local -D 15 -R 2 -N 0 -L 20 -i S,1,0.75 (default) –very-sensitive-local -D 20 -R 3 -N 0 -L 20 -i S,1,0.50

required False

choices

- Very fast: `--very-fast`
- Fast: `--fast`
- Sensitive: `--sensitive`
- Very sensitive: `--very-sensitive`

PE_options.use_se

label Map as single-ended (for paired-end reads only)

type `basic:boolean`

description If this option is selected paired-end reads will be mapped as single-ended and other paired-end options are ignored.

default `False`

PE_options.discordantly

label Report discordantly matched read

type `basic:boolean`

description If both mates have unique alignments, but the alignments do not match paired-end expectations (orientation and relative distance) then alignment will be reported. Useful for detecting structural variations.

default `True`

PE_options.rep_se

label Report single ended

type `basic:boolean`

description If paired alignment can not be found Bowtie2 tries to find alignments for the individual mates.

default `True`

PE_options.minins

label Minimal distance

type `basic:integer`

description The minimum fragment length for valid paired-end alignments. 0 imposes no minimum.

default `0`

PE_options.maxins

label Maximal distance

type `basic:integer`

description The maximum fragment length for valid paired-end alignments.

default `500`

alignment_options.N

label Number of mismatches allowed in seed alignment (N)

type `basic:integer`

description Sets the number of mismatches to allowed in a seed alignment during multiseed alignment. Can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity. Default: 0.

required False

alignment_options.L

label Length of seed substrings (L)

type `basic:integer`

description Sets the length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more sensitive. Default: the –sensitive preset is used by default for end-to-end alignment and –sensitive-local for local alignment. See documentation for details.

required False

alignment_options.gbar

label Disallow gaps within positions (gbar)

type `basic:integer`

description Disallow gaps within <int> positions of the beginning or end of the read. Default: 4.

required False

alignment_options.mp

label Maximal and minimal mismatch penalty (mp)

type `basic:string`

description Sets the maximum (MX) and minimum (MN) mismatch penalties, both integers. A number less than or equal to MX and greater than or equal to MN is subtracted from the alignment score for each position where a read character aligns to a reference character, the characters do not match, and neither is an N. If –ignore-quals is specified, the number subtracted quals MX. Otherwise, the number subtracted is MN + floor((MX-MN)(MIN(Q, 40.0)/40.0)) where Q is the Phred quality value. Default for MX, MN: 6,2.

required False

alignment_options.rdg

label Set read gap open and extend penalties (rdg)

type `basic:string`

description Sets the read gap open (<int1>) and extend (<int2>) penalties. A read gap of length N gets a penalty of <int1> + N * <int2>. Default: 5,3.

required False

alignment_options.rfg

label Set reference gap open and close penalties (rfg)

type `basic:string`

description Sets the reference gap open (<int1>) and extend (<int2>) penalties. A reference gap of length N gets a penalty of <int1> + N * <int2>. Default: 5,3.

required False

alignment_options.score_min

label Minimum alignment score needed for "valid" alignment (score_min)

type `basic:string`

description Sets a function governing the minimum alignment score needed for an alignment to be considered "valid" (i.e. good enough to report). This is a function of read length. For instance, specifying L,0,-0.6 sets the minimum-score function to f(x) = 0 + -0.6 * x, where x is the read length. The default in –end-to-end mode is L,-0.6,-0.6 and the default in –local mode is G,20,8.

required False

start_trimming.trim_5

label Bases to trim from 5'

type `basic:integer`

description Number of bases to trim from from 5' (left) end of each read before alignment

default `0`

start_trimming.trim_3

label Bases to trim from 3'

type `basic:integer`

description Number of bases to trim from from 3' (right) end of each read before alignment

default `0`

trimming.trim_iter

label Iterations

type `basic:integer`

description Number of iterations.

default `0`

trimming.trim_nucl

label Bases to trim

type `basic:integer`

description Number of bases to trim from 3' end in each iteration.

default `2`

reporting.rep_mode

label Report mode

type `basic:string`

description Default mode: search for multiple alignments, report the best one; -k mode: search for one or more alignments, report each; -a mode: search for and report all alignments

default `def`

choices

- Default mode: `def`
- -k mode: `k`
- -a mode (very slow): `a`

reporting.k_reports

label Number of reports (for -k mode only)

type `basic:integer`

description Searches for at most X distinct, valid alignments for each read. The search terminates when it can't find more distinct valid alignments, or when it finds X, whichever happens first. default: 5

default `5`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

ChIP-Seq (Gene Score)

`data:chipseq:genescorechipseq-genescore` (*data:chipseq:peakscore* **peakscore**, *basic:decimal* **fdr**, *basic:decimal* **pval**, *basic:decimal* **logratio**) [Source: v1.1.1]

Chip-Seq analysis - Gene Score (BCM)

Input arguments peakscore

label PeakScore file

type `data:chipseq:peakscore`

description PeakScore file

fdr

label FDR threshold

type `basic:decimal`

description FDR threshold value (default = 0.00005).

default `5e-05`

pval

label Pval threshold

type `basic:decimal`

description Pval threshold value (default = 0.00005).

default `5e-05`

logratio

label Log-ratio threshold

type `basic:decimal`

description Log-ratio threshold value (default = 2).

default `2.0`

Output results genescore

label Gene Score

type `basic:file`

ChIP-Seq (Peak Score)

data:chipseq:peakscorechipseq-peakscore (*data:chipseq:callpeak:macs2* **peaks**, *data:bed* **bed**) [Source: v2.1.0]

Chip-Seq analysis - Peak Score (BCM)

Input arguments peaks

label MACS2 results

type `data:chipseq:callpeak:macs2`

description MACS2 results file (NarrowPeak)

bed

label BED file

type `data:bed`

Output results peak_score

label Peak Score

type `basic:file`

ChIP-seq (MACS2)

data:chipseq:batch:macs2macs2-batch (*list:data:alignment:bam* **alignments**, *basic:boolean* **advanced**, *data:bed* **promoter**, *basic:boolean* **tagalign**, *basic:integer* **q_threshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**, *basic:string* **duplicates**, *basic:string* **duplicates_prepeak**, *basic:decimal* **qvalue**, *basic:decimal* **pvalue**, *basic:decimal* **pvalue_prepeak**, *basic:integer* **cap_num**, *basic:integer* **mfold_lower**, *basic:integer* **mfold_upper**, *basic:integer* **slocal**, *basic:integer* **llocal**, *basic:integer* **extsize**, *basic:integer* **shift**, *basic:integer* **band_width**, *basic:boolean* **nolambda**, *basic:boolean* **fix_bimodal**, *basic:boolean* **nomodel**, *basic:boolean* **nomodel_prepeak**, *basic:boolean* **down_sample**, *basic:boolean* **bedgraph**, *basic:boolean* **spmr**, *basic:boolean* **call_summits**, *basic:boolean* **broad**, *basic:decimal* **broad_cutoff**) [Source: v1.0.3]

This process runs MACS2 in batch mode. MACS2 analysis is triggered for pairs of samples as defined using treatment-background sample relations. If there are no sample relations defined, each sample is treated individually for the MACS analysis.

Model-based Analysis of ChIP-Seq (MACS 2.0), is used to identify transcript factor binding sites. MACS 2.0 captures the influence of genome complexity to evaluate the significance of enriched ChIP regions, and MACS improves the spatial resolution of binding sites through combining the information of both sequencing tag position and orientation. It has also an option to link nearby peaks together in order to call broad peaks. See [here](https://github.com/taoliu/MACS/) for more information.

In addition to peak-calling, this process computes ChIP-Seq and ATAC-Seq QC metrics. Process returns a QC metrics report, fragment length estimation, and a deduplicated tagAlign file. QC report contains ENCODE 3 proposed QC metrics – [NRF](https://www.encodeproject.org/data-standards/terms/), [PBC bottlenecking coefficients, NSC, and RSC](https://genome.ucsc.edu/ENCODE/qualityMetrics.html#chipSeq).

Input arguments alignments

label Aligned reads

type `list:data:alignment:bam`

description Select multiple treatment/background samples.

advanced

label Show advanced options

type `basic:boolean`

description Inspect and modify parameters.

default `False`

promoter

label Promoter regions BED file

type `data:bed`

description BED file containing promoter regions (TSS+-1000bp for example). Needed to get the number of peaks and reads mapped to promoter regions.

required False

hidden !advanced

tagalign

label Use tagAlign files

type `basic:boolean`

description Use filtered tagAlign files as case (treatment) and control (background) samples. If extsize parameter is not set, run MACS using input's estimated fragment length.

hidden !advanced

default `False`

prepeakqc_settings.q_threshold

label Quality filtering threshold

type `basic:integer`

default `30`

prepeakqc_settings.n_sub

label Number of reads to subsample

type `basic:integer`

default `15000000`

prepeakqc_settings.tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on "+" strand by 4bp and reads on "-" strand by 5bp.

default `False`

prepeakqc_settings.shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

required False

settings.duplicates

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden tagalign

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.duplicates_prepeak

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden !tagalign

default `all`

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.qvalue

label Q-value cutoff

type `basic:decimal`

description The q-value (minimum FDR) cutoff to call significant regions. Q-values are calculated from p-values using Benjamini-Hochberg procedure.

required False

disabled settings.pvalue && settings.pvalue_prepeak

settings.pvalue

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

required False

disabled settings.qvalue

hidden tagalign

settings.pvalue_prepeak

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

disabled settings.qvalue

hidden !tagalign || settings.qvalue

default `1e-05`

settings.cap_num

label Cap number of peaks by taking top N peaks

type `basic:integer`

description To keep all peaks set value to 0.

disabled settings.broad

default `500000`

settings.mfold_lower

label MFOLD range (lower limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.mfold_upper

label MFOLD range (upper limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.slocal

label Small local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.llocal

label Large local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.extsize

label extsize

type `basic:integer`

description While '–nomodel' is set, MACS uses this parameter to extend reads in 5'->3' direction to fix-sized fragments. For example, if the size of binding region for your transcription factor is 200 bp, and you want to bypass the model building by MACS, this parameter can be set as 200. This option is only valid when –nomodel is set or when MACS fails to build model and –fix-bimodal is on.

required False

settings.shift

label Shift

type `basic:integer`

description Note, this is NOT the legacy –shiftsize option which is replaced by –extsize! You can set an arbitrary shift in bp here. Please Use discretion while setting it other than default value (0). When –nomodel is set, MACS will use this value to move cutting ends (5') then apply –extsize from 5' to 3' direction to extend them to fragments. When this value is negative, ends will be moved toward 3'->5' direction, otherwise 5'->3' direction. Recommended to keep it as default 0 for ChIP-Seq datasets, or -1 * half of EXTSIZE together with –extsize option for detecting enriched cutting loci such as certain DNAseI-Seq datasets. Note, you can't set values other than 0 if format is BAMPE for paired-end data. Default is 0.

required False

settings.band_width

label Band width

type `basic:integer`

description The band width which is used to scan the genome ONLY for model building. You can set this parameter as the sonication fragment size expected from wet experiment. The previous side effect on the peak detection process has been removed. So this parameter only affects the model building.

required False

settings.nolambda

label Use backgroud lambda as local lambda

type `basic:boolean`

description With this flag on, MACS will use the background lambda as local lambda. This means MACS will not consider the local bias at peak candidate regions.

default `False`

settings.fix_bimodal

label Turn on the auto paired-peak model process

type `basic:boolean`

description Whether turn on the auto paired-peak model process. If it's set, when MACS failed to build paired model, it will use the nomodel settings, the '–extsize' parameter to extend each tags. If set, MACS will be terminated if paired-peak model is failed.

default `False`

settings.nomodel

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden tagalign

default `False`

settings.nomodel_prepeak

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden !tagalign

default `True`

settings.down_sample

label Down-sample

type `basic:boolean`

description When set, random sampling method will scale down the bigger sample. By default, MACS uses linear scaling. This option will make the results unstable and irreproducible since each time, random reads would be selected, especially the numbers (pileup, pvalue, qvalue) would change. Consider to use 'randsample' script before MACS2 runs instead.

default `False`

settings.bedgraph

label Save fragment pileup and control lambda

type `basic:boolean`

description If this flag is on, MACS will store the fragment pileup, control lambda, -log10pvalue and -log10qvalue scores in bedGraph files. The bedGraph files will be stored in current directory named NAME+'_treat_pileup.bdg' for treatment data, NAME+'_control_lambda.bdg' for local lambda values from control, NAME+'_treat_pvalue.bdg' for Poisson pvalue scores (in -log10(pvalue) form), and NAME+'_treat_qvalue.bdg' for q-value scores from Benjamini-Hochberg-Yekutieli procedure.

default `True`

settings.spmr

label Save signal per million reads for fragment pileup profiles

type `basic:boolean`

disabled settings.bedgraph === false

default `True`

settings.call_summits

label Call summits

type `basic:boolean`

description MACS will now reanalyze the shape of signal profile (p or q-score depending on cutoff setting) to deconvolve subpeaks within each peak called from general procedure. It's highly recommended to detect adjacent binding events. While used, the output subpeaks of a big peak region will have the same peak boundaries, and different scores and peak summit positions.

default `False`

settings.broad

label Composite broad regions

type `basic:boolean`

description When this flag is on, MACS will try to composite broad regions in BED12 (a gene-model-like format) by putting nearby highly enriched regions into a broad region with loose cutoff. The broad region is controlled by another cutoff through –broad-cutoff. The maximum length of broad region length is 4 times of d from MACS.

disabled settings.call_summits === true

default `False`

settings.broad_cutoff

label Broad cutoff

type `basic:decimal`

description Cutoff for broad region. This option is not available unless –broad is set. If -p is set, this is a p-value cutoff, otherwise, it's a q-value cutoff. DEFAULT = 0.1

required False

disabled settings.call_summits === true || settings.broad !== true

Output results

ChIP-seq (MACS2-ROSE2)

`data:chipseq:batch:macs2macs2-rose2-batch` (*list:data:alignment:bam* **alignments**, *basic:boolean* **advanced**, *data:bed* **promoter**, *basic:boolean* **tagalign**, *basic:integer* **q_threshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**, *basic:string* **duplicates**, *basic:string* **duplicates_prepeak**, *basic:decimal* **qvalue**, *basic:decimal* **pvalue**, *basic:decimal* **pvalue_prepeak**, *basic:integer* **cap_num**, *basic:integer* **mfold_lower**, *basic:integer* **mfold_upper**, *basic:integer* **slocal**, *basic:integer* **llocal**, *basic:integer* **extsize**, *basic:integer* **shift**, *basic:integer* **band_width**, *basic:boolean* **nolambda**, *basic:boolean* **fix_bimodal**, *basic:boolean* **nomodel**, *basic:boolean* **nomodel_prepeak**, *basic:boolean* **down_sample**, *basic:boolean* **bedgraph**, *basic:boolean* **spmr**, *basic:boolean* **call_summits**, *basic:boolean* **broad**, *basic:decimal* **broad_cutoff**, *basic:integer* **tss**, *basic:integer* **stitch**, *data:bed* **mask**) [Source: v1.0.3]

This process runs MACS2 in batch mode. MACS2 analysis is triggered for pairs of samples as defined using treatment-background sample relations. If there are no sample relations defined, each sample is treated individually for the MACS analysis.

Model-based Analysis of ChIP-Seq (MACS 2.0), is used to identify transcript factor binding sites. MACS 2.0 captures the influence of genome complexity to evaluate the significance of enriched ChIP regions, and MACS improves the spatial resolution of binding sites through combining the information of both sequencing tag position and orientation. It has also an option to link nearby peaks together in order to call broad peaks. See [here](https://github.com/taoliu/MACS/) for more information.

In addition to peak-calling, this process computes ChIP-Seq and ATAC-Seq QC metrics. Process returns a QC metrics report, fragment length estimation, and a deduplicated tagAlign file. QC report contains ENCODE 3 proposed QC metrics – [NRF](https://www.encodeproject.org/data-standards/terms/), [PBC bottlenecking coefficients, NSC, and RSC](https://genome.ucsc.edu/ENCODE/qualityMetrics.html#chipSeq).

For identification of super enhancers R2 uses the Rank Ordering of Super-Enhancers algorithm (ROSE2). This takes the peaks called by RSEG for acetylation and calculates the distances in-between to judge whether they can be considered super-enhancers. The ranked values can be plotted and by locating the inflection point in the resulting graph, super-enhancers can be assigned. It can also be used with the MACS calculated data. See [here](http://younglab.wi.mit.edu/super_enhancer_code.html) for more information.

Input arguments alignments

label Aligned reads

type `list:data:alignment:bam`

description Select multiple treatment/background samples.

advanced

label Show advanced options

type `basic:boolean`

description Inspect and modify parameters.

default `False`

promoter

label Promoter regions BED file

type `data:bed`

description BED file containing promoter regions (TSS+-1000bp for example). Needed to get the number of peaks and reads mapped to promoter regions.

required False

hidden !advanced

tagalign

label Use tagAlign files

type `basic:boolean`

description Use filtered tagAlign files as case (treatment) and control (background) samples. If extsize parameter is not set, run MACS using input's estimated fragment length.

hidden !advanced

default `False`

prepeakqc_settings.q_threshold

label Quality filtering threshold

type `basic:integer`

default `30`

prepeakqc_settings.n_sub

label Number of reads to subsample

type `basic:integer`

default `15000000`

prepeakqc_settings.tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on "+" strand by 4bp and reads on "-" strand by 5bp.

default `False`

prepeakqc_settings.shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

required False

settings.duplicates

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden tagalign

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.duplicates_prepeak

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden !tagalign

default `all`

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.qvalue

label Q-value cutoff

type `basic:decimal`

description The q-value (minimum FDR) cutoff to call significant regions. Q-values are calculated from p-values using Benjamini-Hochberg procedure.

required False

disabled settings.pvalue && settings.pvalue_prepeak

settings.pvalue

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

required False

disabled settings.qvalue

hidden tagalign

settings.pvalue_prepeak

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

disabled settings.qvalue

hidden !tagalign || settings.qvalue

default `1e-05`

settings.cap_num

label Cap number of peaks by taking top N peaks

type `basic:integer`

description To keep all peaks set value to 0.

disabled settings.broad

default `500000`

settings.mfold_lower

label MFOLD range (lower limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.mfold_upper

label MFOLD range (upper limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.slocal

label Small local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.llocal

label Large local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.extsize

label extsize

type `basic:integer`

description While '–nomodel' is set, MACS uses this parameter to extend reads in 5'->3' direction to fix-sized fragments. For example, if the size of binding region for your transcription factor is 200 bp, and you want to bypass the model building by MACS, this parameter can be set as 200. This option is only valid when –nomodel is set or when MACS fails to build model and –fix-bimodal is on.

required False

settings.shift

label Shift

type `basic:integer`

description Note, this is NOT the legacy –shiftsize option which is replaced by –extsize! You can set an arbitrary shift in bp here. Please Use discretion while setting it other than default value (0). When –nomodel is set, MACS will use this value to move cutting ends (5') then apply –extsize from 5' to 3' direction to extend them to fragments. When this value is negative, ends will be moved toward 3'->5' direction, otherwise 5'->3' direction. Recommended to keep it as default 0 for ChIP-Seq datasets, or -1 * half of EXTSIZE together with –extsize option for detecting enriched cutting loci such as certain DNAseI-Seq datasets. Note, you can't set values other than 0 if format is BAMPE for paired-end data. Default is 0.

required False

settings.band_width

label Band width

type `basic:integer`

description The band width which is used to scan the genome ONLY for model building. You can set this parameter as the sonication fragment size expected from wet experiment. The previous side effect on the peak detection process has been removed. So this parameter only affects the model building.

required False

settings.nolambda

label Use backgroud lambda as local lambda

type `basic:boolean`

description With this flag on, MACS will use the background lambda as local lambda. This means MACS will not consider the local bias at peak candidate regions.

default `False`

settings.fix_bimodal

label Turn on the auto paired-peak model process

type `basic:boolean`

description Whether turn on the auto paired-peak model process. If it's set, when MACS failed to build paired model, it will use the nomodel settings, the '–extsize' parameter to extend each tags. If set, MACS will be terminated if paired-peak model is failed.

default `False`

settings.nomodel

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden tagalign

default `False`

settings.nomodel_prepeak

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden !tagalign

default `True`

settings.down_sample

label Down-sample

type `basic:boolean`

description When set, random sampling method will scale down the bigger sample. By default, MACS uses linear scaling. This option will make the results unstable and irreproducible since each time, random reads would be selected, especially the numbers (pileup, pvalue, qvalue) would change. Consider to use 'randsample' script before MACS2 runs instead.

default `False`

settings.bedgraph

label Save fragment pileup and control lambda

type `basic:boolean`

description If this flag is on, MACS will store the fragment pileup, control lambda, -log10pvalue and -log10qvalue scores in bedGraph files. The bedGraph files will be stored in current directory named NAME+'_treat_pileup.bdg' for treatment data, NAME+'_control_lambda.bdg' for local lambda values from control, NAME+'_treat_pvalue.bdg' for Poisson pvalue scores (in -log10(pvalue) form), and NAME+'_treat_qvalue.bdg' for q-value scores from Benjamini-Hochberg-Yekutieli procedure.

default `True`

settings.spmr

label Save signal per million reads for fragment pileup profiles

type `basic:boolean`

disabled settings.bedgraph === false

default `True`

settings.call_summits

label Call summits

type `basic:boolean`

description MACS will now reanalyze the shape of signal profile (p or q-score depending on cutoff setting) to deconvolve subpeaks within each peak called from general procedure. It's highly recommended to detect adjacent binding events. While used, the output subpeaks of a big peak region will have the same peak boundaries, and different scores and peak summit positions.

default `False`

settings.broad

label Composite broad regions

type `basic:boolean`

description When this flag is on, MACS will try to composite broad regions in BED12 (a gene-model-like format) by putting nearby highly enriched regions into a broad region with loose cutoff. The broad region is controlled by another cutoff through –broad-cutoff. The maximum length of broad region length is 4 times of d from MACS.

disabled settings.call_summits === true

default `False`

settings.broad_cutoff

label Broad cutoff

type `basic:decimal`

description Cutoff for broad region. This option is not available unless –broad is set. If -p is set, this is a p-value cutoff, otherwise, it's a q-value cutoff. DEFAULT = 0.1

required False

disabled settings.call_summits === true || settings.broad !== true

rose_settings.tss

label TSS exclusion

type `basic:integer`

description Enter a distance from TSS to exclude. 0 = no TSS exclusion

default `0`

rose_settings.stitch

label Stitch

type `basic:integer`

description Enter a max linking distance for stitching. If not given, optimal stitching parameter will be determined automatically.

required False

rose_settings.mask

label Masking BED file

type `data:bed`

description Mask a set of regions from analysis. Provide a BED of masking regions.

required False

Output results

Chemical Mutagenesis

`data:workflow:chemutworkflow-chemut` (*basic:string* **analysis_type**, *data:genome:fasta* **genome**, *list:data:alignment:bam* **parental_strains**, *list:data:alignment:bam* **mutant_strains**, *basic:boolean* **advanced**, *basic:boolean* **br_and_ind_ra**, *basic:boolean* **dbsnp**, *data:variants:vcf* **known_sites**, *list:data:variants:vcf* **known_indels**, *basic:integer* **stand_emit_conf**, *basic:integer* **stand_call_conf**, *basic:boolean* **rf**, *basic:boolean* **advanced**, *basic:integer* **read_depth**) [Source: v0.0.6]

Input arguments analysis_type

label Analysis type

type `basic:string`

description Choice of the analysis type. Use “SNV” or “INDEL” options to run the GATK analysis only on the haploid portion of the dicty genome. Choose options SNV_CHR2 or INDEL_CHR2 to run the analysis only on the diploid portion of CHR2 (-ploidy 2 -L chr2:2263132-3015703).

default `snv`

choices

- SNV: `snv`
- INDEL: `indel`
- SNV_CHR2: `snv_chr2`
- INDEL_CHR2: `indel_chr2`

genome

label Reference genome

type `data:genome:fasta`

parental_strains

label Parental strains

type `list:data:alignment:bam`

mutant_strains

label Mutant strains

type `list:data:alignment:bam`

Vc.advanced

label Advanced options

type `basic:boolean`

required False

default `False`

Vc.br_and_ind_ra

label Do variant base recalibration and indel realignment

type `basic:boolean`

required False

hidden Vc.advanced === false

default `False`

Vc.dbsnp

label Use dbSNP file

type `basic:boolean`

description rsIDs from this file are used to populate the ID column of the output. Also, the DB INFO flag will be set when appropriate. dbSNP is not used in any way for the calculations themselves.

required False

hidden Vc.advanced === false

default `False`

Vc.known_sites

label Known sites (dbSNP)

type `data:variants:vcf`

required False

hidden Vc.advanced === false || Vc.br_and_ind_ra === false && Vc.dbsnp === false

Vc.known_indels

label Known indels

type `list:data:variants:vcf`

required False

hidden Vc.advanced === false || Vc.br_and_ind_ra === false

default `[]`

Vc.stand_emit_conf

label Emission confidence threshold

type `basic:integer`

description The minimum confidence threshold (phred-scaled) at which the program should emit sites that appear to be possibly variant.

required False

hidden Vc.advanced === false

default `10`

Vc.stand_call_conf

label Calling confidence threshold

type `basic:integer`

description The minimum confidence threshold (phred-scaled) at which the program should emit variant sites as called. If a site's associated genotype has a confidence score lower than the calling threshold, the program will emit the site as filtered and will annotate it as LowQual. This threshold separates high confidence calls from low confidence calls.

required False

hidden Vc.advanced === false

default `30`

Vc.rf

label ReasignOneMappingQuality Filter

type `basic:boolean`

description This read transformer will change a certain read mapping quality to a different value without affecting reads that have other mapping qualities. This is intended primarily for users of RNA-Seq data handling programs such as TopHat, which use MAPQ = 255 to designate uniquely aligned reads. According to convention, 255 normally designates "unknown" quality, and most GATK tools automatically ignore such reads. By reassigning a different mapping quality to those specific reads, users of TopHat and other tools can circumvent this problem without affecting the rest of their dataset.

required False

hidden Vc.advanced === false

default `False`

Vf.advanced

label Advanced options

type `basic:boolean`

required False

default `False`

Vf.read_depth

label Read depth cutoff

type `basic:integer`

description The minimum number of replicate reads required for a variant site to be included.

required False

hidden Vf.advanced === false

default `5`

Output results

Convert GFF3 to GTF

`data:annotation:gtfgff-to-gtf` (*data:annotation:gff3* **annotation**) [Source: v0.3.1]

Convert GFF3 file to GTF format.

Input arguments annotation

label Annotation (GFF3)

type `data:annotation:gff3`

description Annotation in GFF3 format.

Output results annot

label Converted GTF file

type `basic:file`

annot_sorted

label Sorted GTF file

type `basic:file`

annot_sorted_idx_igv

label Igv index for sorted GTF file

type `basic:file`

annot_sorted_track_jbrowse

label Jbrowse track for sorted GTF

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Convert files to reads (paired-end)

`data:reads:fastq:pairedfiles-to-fastq-paired` (*list:data:file* **src1**, *list:data:file* **src2**) [Source: v1.2.1]

Convert FASTQ files to paired-end reads.

Input arguments src1

label Mate1

type `list:data:file`

src2

label Mate2

type `list:data:file`

Output results fastq

label Reads file (mate 1)

type `list:basic:file`

fastq2

label Reads file (mate 2)

type `list:basic:file`

fastqc_url

label Quality control with FastQC (Upstream)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (Downstream)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (Upstream)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (Downstream)

type `list:basic:file`

Convert files to reads (single-end)

data:reads:fastq:singlefiles-to-fastq-single (*list:data:file* **src**) [Source: v1.2.1]

Convert FASTQ files to single-end reads.

Input arguments src

label Reads

type `list:data:file`

description Sequencing reads in FASTQ format

Output results fastq

label Reads file

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

Cuffdiff 2.2

data:differentialexpression:cuffdiffcuffdiff (*list:data:cufflinks:cuffquant* **case**, *list:data:cufflinks:cuffquant* **control**, *list:basic:string* **labels**, *data:annotation* **annotation**, *data:genome:fasta* **genome**, *basic:boolean* **multi_read_correct**, *basic:decimal* **fdr**, *basic:string* **library_type**, *basic:string* **library_normalization**, *basic:string* **dispersion_method**) [Source: v2.2.1]

Cuffdiff finds significant changes in transcript expression, splicing, and promoter use. You can use it to find differentially expressed genes and transcripts, as well as genes that are being differentially regulated at the transcriptional and post-transcriptional level. See [here](http://cole-trapnell-lab.github.io/cufflinks/cuffdiff/) and [here](https://software.broadinstitute.org/cancer/software/genepattern/modules/docs/Cuffdiff/7) for more information.

Input arguments case

label Case samples

type `list:data:cufflinks:cuffquant`

control

label Control samples

type `list:data:cufflinks:cuffquant`

labels

label Group labels

type `list:basic:string`

description Define labels for each sample group.

default `['control', 'case']`

annotation

label Annotation (GTF/GFF3)

type `data:annotation`

description A transcript annotation file produced by cufflinks, cuffcompare, or other tool.

genome

label Run bias detection and correction algorithm

type `data:genome:fasta`

description Provide Cufflinks with a multifasta file (genome file) via this option to instruct it to run a bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates.

required False

multi_read_correct

label Do initial estimation procedure to more accurately weight reads with multiple genome mappings

type `basic:boolean`

default `False`

fdr

label Allowed FDR

type `basic:decimal`

description The allowed false discovery rate. The default is 0.05.

default `0.05`

library_type

label Library type

type `basic:string`

description In cases where Cufflinks cannot determine the platform and protocol used to generate input reads, you can supply this information manually, which will allow Cufflinks to infer source strand information with certain protocols. The available options are listed below. For paired-end data, we currently only support protocols where reads are point towards each other: fr-unstranded - Reads from the left-most end of the fragment (in transcript coordinates) map to the transcript strand, and the right-most end maps to the opposite strand; fr-firststrand - Same as above except we enforce the rule that the right-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during first strand synthesis is sequenced; fr-secondstrand - Same as above except we enforce the rule that the left-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during second strand synthesis is sequenced.

default `fr-unstranded`

choices

- fr-unstranded: `fr-unstranded`
- fr-firststrand: `fr-firststrand`
- fr-secondstrand: `fr-secondstrand`

library_normalization

label Library normalization method

type `basic:string`

description You can control how library sizes (i.e. sequencing depths) are normalized in Cufflinks and Cuffdiff. Cuffdiff has several methods that require multiple libraries in order to work. Library normalization methods supported by Cufflinks work on one library at a time.

default `geometric`

choices

- geometric: `geometric`
- classic-fpkm: `classic-fpkm`
- quartile: `quartile`

dispersion_method

label Dispersion method

type `basic:string`

description Cuffdiff works by modeling the variance in fragment counts across replicates as a function of the mean fragment count across replicates. Strictly speaking, models a quantitity called dispersion - the variance present in a group of samples beyond what is expected from a simple Poisson model of RNA_Seq. You can control how Cuffdiff constructs its model of dispersion in locus fragment counts. Each condition that has replicates can receive its own model, or Cuffdiff can use a global model for all conditions. All of these policies are identical to those used by DESeq (Anders and Huber, Genome Biology, 2010).

default `pooled`

choices

- pooled: `pooled`
- per-condition: `per-condition`
- blind: `blind`
- poisson: `poisson`

Output results raw

label Differential expression (gene level)

type `basic:file`

de_json

label Results table (JSON)

type `basic:json`

de_file

label Results table (file)

type `basic:file`

transcript_diff_exp

label Differential expression (transcript level)

type `basic:file`

tss_group_diff_exp

label Differential expression (primary transcript)

type `basic:file`

cds_diff_exp

label Differential expression (coding sequence)

type `basic:file`

cuffdiff_output

label Cuffdiff output

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Cufflinks 2.2

`data:cufflinks:cufflinkscufflinks` (*data:alignment:bam* **alignment**, *data:annotation* **annotation**, *data:genome:fasta* **genome**, *data:annotation:gtf* **mask_file**, *basic:string* **library_type**, *basic:string* **annotation_usage**, *basic:boolean* **multi_read_correct**) [Source: v2.1.1]

Cufflinks assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols. See [here](http://cole-trapnell-lab.github.io/cufflinks/) for more information.

Input arguments alignment

label Aligned reads

type `data:alignment:bam`

annotation

label Annotation (GTF/GFF3)

type `data:annotation`

required False

genome

label Run bias detection and correction algorithm

type `data:genome:fasta`

description Provide Cufflinks with a multifasta file (genome file) via this option to instruct it to run a bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates.

required False

mask_file

label Mask file

type `data:annotation:gtf`

description Ignore all reads that could have come from transcripts in this GTF file. We recommend including any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates.

required False

library_type

label Library type

type `basic:string`

description In cases where Cufflinks cannot determine the platform and protocol used to generate input reads, you can supply this information manually, which will allow Cufflinks to infer source strand information with certain protocols. The available options are listed below. For paired-end data, we currently only support protocols where reads are point towards each other: fr-unstranded - Reads from the left-most end of the fragment (in transcript coordinates) map to the transcript strand, and the right-most end maps to the opposite strand; fr-firststrand - Same as above except we enforce the rule that the right-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during first strand synthesis is sequenced; fr-secondstrand - Same as above except we enforce the rule that the left-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during second strand synthesis is sequenced.

default `fr-unstranded`

choices

- fr-unstranded: `fr-unstranded`
- fr-firststrand: `fr-firststrand`
- fr-secondstrand: `fr-secondstrand`

annotation_usage

label Instruct Cufflinks how to use the provided annotation (GFF/GTF) file

type `basic:string`

description GTF-guide - tells Cufflinks to use the supplied reference annotation (GFF) to guide RABT assembly. Reference transcripts will be tiled with faux-reads to provide additional information in assembly. Output will include all reference transcripts as well as any novel genes and isoforms that are assembled. –GTF - tells Cufflinks to use the supplied reference annotation (a GFF file) to estimate isoform expression. It will not assemble novel transcripts, and the program will ignore alignments not structurally compatible with any reference transcript.

default `--GTF-guide`

choices

- Use supplied reference annotation to guide RABT assembly (–GTF-guide): `--GTF-guide`
- Use supplied reference annotation to estimate isoform expression (–GTF): `--GTF`

multi_read_correct

label Do initial estimation procedure to more accurately weight reads with multiple genome mappings

type `basic:boolean`

description Run an initial estimation procedure that weights reads mapping to multiple locations more accurately.

default `False`

Output results transcripts

label Assembled transcript isoforms

type `basic:file`

isoforms_fpkm_tracking

label Isoforms FPKM tracking

type `basic:file`

genes_fpkm_tracking

label Genes FPKM tracking

type `basic:file`

skipped_loci

label Skipped loci

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Cuffmerge

`data:annotation:cuffmergecuffmerge` (*list:data:cufflinks:cufflinks* **expressions**, *list:data:annotation:gtf* **gtf**, *data:annotation* **gff**, *data:genome:fasta* **genome**, *basic:integer* **threads**) [Source: v1.3.1]

Cufflinks includes a script called Cuffmerge that you can use to merge together several Cufflinks assemblies. It also handles running Cuffcompare for you, and automatically filters a number of transfrags that are probably artfifacts. The main purpose of Cuffmerge is to make it easier to make an assembly GTF file suitable for use with Cuffdiff. See [here](http://cole-trapnell-lab.github.io/cufflinks/cuffmerge/) for more information.

Input arguments expressions

label Cufflinks transcripts (GTF)

type `list:data:cufflinks:cufflinks`

required False

gtf

label Annotation files (GTF)

type `list:data:annotation:gtf`

description Annotation files you wish to merge together with Cufflinks produced annotation files (e.g. upload Cufflinks annotation GTF file)

required False

gff

label Reference annotation (GTF/GFF3)

type `data:annotation`

description An optional "reference" annotation GTF. The input assemblies are merged together with the reference GTF and included in the final output.

required False

genome

label Reference genome

type `data:genome:fasta`

description This argument should point to the genomic DNA sequences for the reference. If a directory, it should contain one fasta file per contig. If a multifasta file, all contigs should be present. The merge script will pass this option to cuffcompare, which will use the sequences to assist in classifying transfrags and excluding artifacts (e.g. repeats). For example, Cufflinks transcripts consisting mostly of lower-case bases are classified as repeats. Note that <seq_dir> must contain one fasta file per reference chromosome, and each file must be named after the chromosome, and have a .fa or .fasta extension

required False

threads

label Use this many processor threads

type `basic:integer`

description Use this many threads to align reads. The default is 1.

default `1`

Output results annot

  label Merged GTF file

  type `basic:file`

source

  label Gene ID database

  type `basic:string`

species

  label Species

  type `basic:string`

build

  label Build

  type `basic:string`

Cuffnorm

`data:cuffnormcuffnorm` (*list:data:cufflinks:cuffquant* **cuffquant**, *data:annotation* **annotation**, *basic:boolean* **useERCC**) [Source: v2.1.3]

Cufflinks includes a program, Cuffnorm, that you can use to generate tables of expression values that are properly normalized for library size. Cuffnorm takes a GTF2/GFF3 file of transcripts as input, along with two or more SAM, BAM, or CXB files for two or more samples. See [here](http://cole-trapnell-lab.github.io/cufflinks/cuffnorm/) for more information.

Replicate relation needs to be defined for Cuffnorm to account for replicates. If the replicate relation is not defined, each sample will be treated individually.

Input arguments cuffquant

  label Cuffquant expression file

  type `list:data:cufflinks:cuffquant`

annotation

  label Annotation (GTF/GFF3)

  type `data:annotation`

  description A transcript annotation file produced by cufflinks, cuffcompare, or other source.

useERCC

  label ERCC spike-in normalization

  type `basic:boolean`

  description Use ERRCC spike-in controls for normalization.

  default `False`

Output results genes_count

  label Genes count

  type `basic:file`

genes_fpkm

label Genes FPKM

type `basic:file`

genes_attr

label Genes attr table

type `basic:file`

isoform_count

label Isoform count

type `basic:file`

isoform_fpkm

label Isoform FPKM

type `basic:file`

isoform_attr

label Isoform attr table

type `basic:file`

cds_count

label CDS count

type `basic:file`

cds_fpkm

label CDS FPKM

type `basic:file`

cds_attr

label CDS attr table

type `basic:file`

tss_groups_count

label TSS groups count

type `basic:file`

tss_groups_fpkm

label TSS groups FPKM

type `basic:file`

tss_attr

label TSS attr table

type `basic:file`

run_info

label Run info

type `basic:file`

raw_scatter

label FPKM exp scatter plot

type `basic:file`

boxplot

label Boxplot

type `basic:file`

fpkm_exp_raw

label FPKM exp raw

type `basic:file`

replicate_correlations

label Replicate correlatios plot

type `basic:file`

fpkm_means

label FPKM means

type `basic:file`

exp_fpkm_means

label Exp FPKM means

type `basic:file`

norm_scatter

label FKPM exp scatter normalized plot

type `basic:file`

required False

fpkm_exp_norm

label FPKM exp normalized

type `basic:file`

required False

spike_raw

label Spike raw

type `basic:file`

required False

spike_norm

label Spike normalized

type `basic:file`

required False

R_data

label All R normalization data

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Cuffquant 2.2

`data:cufflinks:cuffquantcuffquant` (*data:alignment:bam* **alignment**, *data:annotation* **annotation**, *data:genome:fasta* **genome**, *data:annotation:gtf* **mask_file**, *basic:string* **library_type**, *basic:boolean* **multi_read_correct**) [Source: v1.3.1]

Cuffquant allows you to compute the gene and transcript expression profiles and save these profiles to files that you can analyze later with Cuffdiff or Cuffnorm. See [here](http://cole-trapnell-lab.github.io/cufflinks/manual/) for more information.

Input arguments alignment

label Aligned reads

type `data:alignment:bam`

annotation

label Annotation (GTF/GFF3)

type `data:annotation`

genome

label Run bias detection and correction algorithm

type `data:genome:fasta`

description Provide Cufflinks with a multifasta file (genome file) via this option to instruct it to run a bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates.

required False

mask_file

label Mask file

type `data:annotation:gtf`

description Ignore all reads that could have come from transcripts in this GTF file. We recommend including any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates.

required False

library_type

label Library type

type `basic:string`

description In cases where Cufflinks cannot determine the platform and protocol used to generate input reads, you can supply this information manually, which will allow Cufflinks to infer source strand information with certain protocols. The available options are listed below. For paired-end data, we currently only support protocols where reads are point towards each other: fr-unstranded - Reads from the left-most end of the fragment (in transcript coordinates) map to the transcript strand, and the right-most end maps to the opposite strand; fr-firststrand - Same as above except we enforce the rule that the right-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during first strand synthesis is sequenced; fr-secondstrand - Same as above except we enforce the rule that the left-most end of the fragment (in transcript coordinates) is the first sequenced (or only sequenced for single-end reads). Equivalently, it is assumed that only the strand generated during second strand synthesis is sequenced.

default `fr-unstranded`

choices

- fr-unstranded: `fr-unstranded`
- fr-firststrand: `fr-firststrand`
- fr-secondstrand: `fr-secondstrand`

multi_read_correct

label Do initial estimation procedure to more accurately weight reads with multiple genome mappings

type `basic:boolean`

description Run an initial estimation procedure that weights reads mapping to multiple locations more accurately.

default `False`

Output results cxb

label Abundances (.cxb)

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Cuffquant results

data:cufflinks:cuffquantupload-cxb (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**, *basic:string* **feature_type**) [Source: v1.2.1]

Upload Cuffquant results file (.cxb)

Input arguments src

label Cuffquant file

type `basic:file`

description Upload Cuffquant results file. Supported extention: *.cxb

required True

validate_regex `\.(cxb)$`

source

label Gene ID database

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

default `gene`

choices

- gene: `gene`
- transcript: `transcript`
- exon: `exon`

Output results cxb

label Cuffquant results

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Custom master file

`data:masterfile:ampliconupload-master-file` (*basic:file* **src**, *basic:string* **panel_name**) [Source: v1.1.1]

This should be a tab delimited file (*.txt). Please check the [example](http://genial.is/amplicon-masterfile) file for details.

Input arguments src

label Master file

type `basic:file`

validate_regex `\.txt(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

panel_name

label Panel name

type `basic:string`

Output results master_file

label Master file

type `basic:file`

bedfile

label BED file (merged targets)

type `basic:file`

nomergebed

label BED file (nonmerged targets)

type `basic:file`

olapfreebed

label BED file (overlap-free targets)

type `basic:file`

primers

label Primers

type `basic:file`

panel_name

label Panel name

type `basic:string`

Cutadapt (Diagenode CATS, paired-end)

`data:reads:fastq:paired:cutadaptcutadapt-custom-paired` (*data:reads:fastq:paired* **reads**) [Source: v1.1.2]

Cutadapt process configured to be used with the Diagenode CATS kits.

Input arguments reads

label NGS reads

type `data:reads:fastq:paired`

Output results fastq

label Reads file (forward)

type `list:basic:file`

fastq2

label Reads file (reverse)

type `list:basic:file`

report

label Cutadapt report

type `basic:file`

fastqc_url

label Quality control with FastQC (forward)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (reverse)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (forward)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (reverse)

type `list:basic:file`

Cutadapt (Diagenode CATS, single-end)

`data:reads:fastq:single:cutadaptcutadapt-custom-single` (*data:reads:fastq:single* **reads**) [Source: v1.1.2]

Cutadapt process configured to be used with the Diagenode CATS kits.

Input arguments reads

label NGS reads

type `data:reads:fastq:single`

Output results fastq

label Reads file

type `list:basic:file`

report

label Cutadapt report

type `basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

Cutadapt (paired-end)

data:reads:fastq:paired:cutadaptcutadapt-paired (*data:reads:fastq:paired* **reads**, *data:seq:nucleotide* **mate1_5prime_file**, *data:seq:nucleotide* **mate1_3prime_file**, *data:seq:nucleotide* **mate2_5prime_file**, *data:seq:nucleotide* **mate2_3prime_file**, *list:basic:string* **mate1_5prime_seq**, *list:basic:string* **mate1_3prime_seq**, *list:basic:string* **mate2_5prime_seq**, *list:basic:string* **mate2_3prime_seq**, *basic:integer* **times**, *basic:decimal* **error_rate**, *basic:integer* **min_overlap**, *basic:boolean* **match_read_wildcards**, *basic:integer* **leading**, *basic:integer* **trailing**, *basic:integer* **crop**, *basic:integer* **headcrop**, *basic:integer* **minlen**, *basic:integer* **max_n**, *basic:string* **pair_filter**) [Source: v2.1.2]

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. More information about Cutadapt can be found [here](http://cutadapt.readthedocs.io/en/stable/).

Input arguments reads

label NGS reads

type `data:reads:fastq:paired`

adapters.mate1_5prime_file

label 5 prime adapter file for Mate 1

type `data:seq:nucleotide`

required False

adapters.mate1_3prime_file

label 3 prime adapter file for Mate 1

type `data:seq:nucleotide`

required False

adapters.mate2_5prime_file

label 5 prime adapter file for Mate 2

type `data:seq:nucleotide`

required False

adapters.mate2_3prime_file

label 3 prime adapter file for Mate 2

type `data:seq:nucleotide`

required False

adapters.mate1_5prime_seq

label 5 prime adapter sequence for Mate 1

type `list:basic:string`

required False

adapters.mate1_3prime_seq

label 3 prime adapter sequence for Mate 1

type `list:basic:string`

required False

adapters.mate2_5prime_seq

label 5 prime adapter sequence for Mate 2

type `list:basic:string`

required False

adapters.mate2_3prime_seq

label 3 prime adapter sequence for Mate 2

type `list:basic:string`

required False

adapters.times

label Times

type `basic:integer`

description Remove up to COUNT adapters from each read.

default `1`

adapters.error_rate

label Error rate

type `basic:decimal`

description Maximum allowed error rate (no. of errors divided by the length of the matching region).

default `0.1`

adapters.min_overlap

label Minimal overlap

type `basic:integer`

description Minimum overlap for an adapter match.

default `3`

adapters.match_read_wildcards

label Match read wildcards

type `basic:boolean`

description Interpret IUPAC wildcards in reads.

default `False`

modify_reads.leading

label Quality on 5 prime

type `basic:integer`

description Remove low quality bases from 5 prime. Specifies the minimum quality required to keep a base.

required False

modify_reads.trailing

label Quality on 3 prime

type `basic:integer`

description Remove low quality bases from the 3 prime. Specifies the minimum quality required to keep a base.

required False

modify_reads.crop

label Crop

type `basic:integer`

description Cut the specified number of bases from the end of the reads.

required False

modify_reads.headcrop

label Headcrop

type `basic:integer`

description Cut the specified number of bases from the start of the reads.

required False

filtering.minlen

label Min length

type `basic:integer`

description Drop the read if it is below a specified.

required False

filtering.max_n

label Max numebr of N-s

type `basic:integer`

description Discard reads having more 'N' bases than specified.

required False

filtering.pair_filter

label Which of the reads have to match the filtering criterion

type `basic:string`

description Which of the reads in a paired-end read have to match the filtering criterion in order for the pair to be filtered.

default `any`

choices

- Any of the reads in a paired-end read have to match the filtering criterion: `any`
- Both of the reads in a paired-end read have to match the filtering criterion: `both`

Output results fastq

label Reads file (forward)

type `list:basic:file`

fastq2

label Reads file (reverse)

type `list:basic:file`

report

label Cutadapt report

type `basic:file`

fastqc_url

label Quality control with FastQC (forward)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (reverse)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (forward)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (reverse)

type `list:basic:file`

Cutadapt (single-end)

data:reads:fastq:single:cutadaptcutadapt-single (*data:reads:fastq:single* **reads**, *data:seq:nucleotide* **up_primers_file**, *data:seq:nucleotide* **down_primers_file**, *list:basic:string* **up_primers_seq**, *list:basic:string* **down_primers_seq**, *basic:integer* **polya_tail**, *basic:integer* **leading**, *basic:integer* **trailing**, *basic:integer* **crop**, *basic:integer* **headcrop**, *basic:integer* **min_overlap**, *basic:integer* **minlen**, *basic:integer* **max_n**, *basic:boolean* **match_read_wildcards**, *basic:integer* **times**, *basic:decimal* **error_rate**) [Source: v1.2.2]

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. More information about Cutadapt can be found [here](http://cutadapt.readthedocs.io/en/stable/).

Input arguments reads

label NGS reads

type `data:reads:fastq:single`

up_primers_file

label 5 prime adapter file

type `data:seq:nucleotide`

required False

down_primers_file

label 3 prime adapter file

type `data:seq:nucleotide`

required False

up_primers_seq

label 5 prime adapter sequence

type `list:basic:string`

required False

down_primers_seq

label 3 prime adapter sequence

type `list:basic:string`

required False

polya_tail

label Poly-A tail

type `basic:integer`

description Length of poly-A tail, example - AAAN -> 3, AAAAAN -> 5

required False

leading

label Quality on 5 prime

type `basic:integer`

description Remove low quality bases from 5 prime. Specifies the minimum quality required to keep a base.

required False

trailing

label Quality on 3 prime

type `basic:integer`

description Remove low quality bases from the 3 prime. Specifies the minimum quality required to keep a base.

required False

crop

label Crop

type `basic:integer`

description Cut the read to a specified length by removing bases from the end

required False

headcrop

label Headcrop

type `basic:integer`

description Cut the specified number of bases from the start of the read

required False

min_overlap

label Minimal overlap

type `basic:integer`

description Minimum overlap for an adapter match

default `3`

minlen

label Min length

type `basic:integer`

description Drop the read if it is below a specified length

required False

max_n

label Max numebr of N-s

type `basic:integer`

description Discard reads having more 'N' bases than specified.

required False

match_read_wildcards

label Match read wildcards

type `basic:boolean`

description Interpret IUPAC wildcards in reads.

required False

default `False`

times

label Times

type `basic:integer`

description Remove up to COUNT adapters from each read.

required False

default `1`

error_rate

label Error rate

type `basic:decimal`

description Maximum allowed error rate (no. of errors divided by the length of the matching region).

required False

default `0.1`

Output results fastq

label Reads file

type `list:basic:file`

report

label Cutadapt report

type `basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

Cutadapt - STAR - HTSeq-count (paired-end)

`data:workflow:rnaseq:htseqworkflow-custom-cutadapt-star-htseq-paired` (*data:reads:fastq:paired* **reads** *data:genomeindex:star* **genom** *data:annotation:gtf* **gff**, *basic:string* **stranded**, *basic:boolean* **advanced**, *basic:boolean* **noncannonical**, *basic:boolean* **chimeric**, *basic:integer* **chimSegmentMin**, *basic:boolean* **quantmode**, *basic:boolean* **singleend**, *basic:boolean* **gene_counts**, *basic:string* **outFilterType**, *basic:integer* **outFilterMultimapNmax**, *basic:integer* **outFilterMismatchNmax**, *basic:decimal* **outFilterMismatchNoverLmax**, *ba-*

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __cutadapt__ which finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label NGS reads

type `data:reads:fastq:paired`

genome

label Indexed reference genome

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool

gff

label Annotation (GFF)

type `data:annotation:gtf`

stranded

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

advanced

label Advanced

type `basic:boolean`

default `False`

star.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

default `False`

star.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

star.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled !star.detect_chimeric.chimeric

default `20`

star.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `False`

star.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled !star.t_coordinates.quantmode

default `False`

star.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled !star.t_coordinates.quantmode

default `False`

star.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

star.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

star.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

star.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

star.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

star.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

star.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

star.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

star.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

htseq.mode

label Mode

type `basic:string`

description Mode to handle reads overlapping more than one feature. Possible values for <mode> are union, intersection-strict and intersection-nonempty

default `union`

choices

- union: `union`
- intersection-strict: `intersection-strict`
- intersection-nonempty: `intersection-nonempty`

htseq.feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GFF file) to be used. All other features will be ignored.

default `exon`

htseq.id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GFF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

htseq.name_ordered

label Use name-ordered BAM file for counting reads

type `basic:boolean`

description Use name-sorted BAM file for reads quantification. Improves compatibility with larger BAM files, but requires more computational time.

required False

default `False`

Output results

Cutadapt - STAR - HTSeq-count (single-end)

`data:workflow:rnaseq:htseqworkflow-custom-cutadapt-star-htseq-single` (*data:reads:fastq:single* **reads**, *data:genomeindex:star* **genom** *data:annotation:gtf* **gff**, *basic:string* **stranded**, *basic:boolean* **advanced**, *basic:boolean* **noncannonical**, *basic:boolean* **chimeric**, *basic:integer* **chimSegmentMin**, *basic:boolean* **quantmode**, *basic:boolean* **singleend**, *basic:boolean* **gene_counts**, *basic:string* **outFilterType**, *basic:integer* **outFilterMultimapNmax**, *basic:integer* **outFilterMismatchNmax**, *basic:decimal* **outFilterMismatchNoverLmax**, *ba-*

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __cutadapt__ which finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. Next, preprocessed reads are aligned by __STAR__ aligner. At the time of implementation, STAR is considered a state-of-the-art tool that consistently produces accurate results from diverse sets of reads, and performs well even with default settings. For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label NGS reads

type `data:reads:fastq:single`

genome

label Indexed reference genome

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool

gff

label Annotation (GFF)

type `data:annotation:gtf`

stranded

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

advanced

label Advanced

type `basic:boolean`

default `False`

star.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

default `False`

star.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

star.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled !star.detect_chimeric.chimeric

default `20`

star.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `False`

star.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled !star.t_coordinates.quantmode

default `False`

star.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled !star.t_coordinates.quantmode

default `False`

star.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

star.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

star.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

star.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

star.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

star.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

star.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

star.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

star.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

htseq.mode

label Mode

type `basic:string`

description Mode to handle reads overlapping more than one feature. Possible values for <mode> are union, intersection-strict and intersection-nonempty

default `union`

choices

- union: `union`
- intersection-strict: `intersection-strict`
- intersection-nonempty: `intersection-nonempty`

htseq.feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GFF file) to be used. All other features will be ignored.

default `exon`

htseq.id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GFF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

htseq.name_ordered

label Use name-ordered BAM file for counting reads

type `basic:boolean`

description Use name-sorted BAM file for reads quantification. Improves compatibility with larger BAM files, but requires more computational time.

required False

default `False`

Output results

Cutadapt - STAR - RSEM (Diagenode CATS, paired-end)

`data:workflow:rnaseq:rsemworkflow-custom-cutadapt-star-rsem-paired` (*data:reads:fastq:paired* **reads**, *data:genomeindex:star* **star_inde** *data:index:expression* **expression_index**, *basic:string* **stranded**, *basic:boolean* **advanced**, *basic:boolean* **noncannonical**, *basic:boolean* **chimeric**, *basic:integer* **chimSegmentMin**, *basic:boolean* **quantmode**, *basic:boolean* **singleend**, *basic:boolean* **gene_counts**, *basic:string* **outFilterType**, *basic:integer* **outFilterMultimapNmax**, *basic:integer* **outFilterMismatchNmax**, *basic:decimal* **outFilterMismatchNoverLmax**, *basic:integer* **alignSJoverhangMin**, *basic:integer* **alignSJDBoverhangMin**, *ba-*

This RNA-seq pipeline is configured to be used with the Diagenode CATS RNA-seq kits. It is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by cutadapt which finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. Next, preprocessed reads are aligned by STAR aligner. Finally, RSEM estimates gene and isoform expression levels from the aligned reads.

Input arguments reads

label NGS reads

type `data:reads:fastq:paired`

star_index

label STAR genome index

type `data:genomeindex:star`

expression_index

label Gene expression indices

type `data:index:expression`

stranded

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

advanced

label Advanced

type `basic:boolean`

default `False`

star.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

default `False`

star.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

star.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled !star.detect_chimeric.chimeric

default `20`

star.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `True`

star.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled !star.t_coordinates.quantmode

default `False`

star.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled !star.t_coordinates.quantmode

default `False`

star.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

star.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

star.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

star.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

star.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

star.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

star.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

star.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

star.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

Output results

Cutadapt - STAR - RSEM (Diagenode CATS, single-end)

`data:workflow:rnaseq:rsemworkflow-custom-cutadapt-star-rsem-single` (*data:reads:fastq:single* **reads**, *data:genomeindex:star* **star_inde** *data:index:expression* **expression_index**, *basic:string* **stranded**, *basic:boolean* **advanced**, *basic:boolean* **noncannonical**, *basic:boolean* **chimeric**, *basic:integer* **chimSegmentMin**, *basic:boolean* **quantmode**, *basic:boolean* **singleend**, *basic:boolean* **gene_counts**, *basic:string* **outFilterType**, *basic:integer* **outFilterMultimapNmax**, *basic:integer* **outFilterMismatchNmax**, *basic:decimal* **outFilterMismatchNoverLmax**, *basic:integer* **alignSJoverhangMin**, *basic:integer* **alignSJDBoverhangMin**, *ba-*

This RNA-seq pipeline is configured to be used with the Diagenode CATS RNA-seq kits. It is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by cutadapt which finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads. Next, preprocessed reads are aligned by STAR aligner. Finally, RSEM estimates gene and isoform expression levels from the aligned reads.

Input arguments reads

label NGS reads

type `data:reads:fastq:single`

star_index

label STAR genome index

type `data:genomeindex:star`

expression_index

label Gene expression indices

type `data:index:expression`

stranded

label Assay type

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

advanced

label Advanced

type `basic:boolean`

default `False`

star.noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

default `False`

star.detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates. –chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

star.detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled !star.detect_chimeric.chimeric

default `20`

star.t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `True`

star.t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled !star.t_coordinates.quantmode

default `False`

star.t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled !star.t_coordinates.quantmode

default `False`

star.filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

star.filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

star.filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

star.filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

star.alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

star.alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

star.alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

star.alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

star.alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

Output results

DESeq2

`data:differentialexpression:deseq2differentialexpression-deseq2` (*list:data:expression* **case**, *list:data:expression* **control**, *basic:boolean* **count**, *basic:integer* **min_count_sum**, *basic:boolean* **cook**, *basic:decimal* **cooks_cutoff**, *basic:boolean* **independent**, *basic:decimal* **alpha**) [Source: v2.2.0]

The DESeq2 package estimates variance-mean dependence in count data from high-throughput sequencing assays and tests for differential expression based on a model using the negative binomial distribution. See [here](https://www.bioconductor.org/packages/release/bioc/manuals/DESeq2/man/DESeq2.pdf) and [here](http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html) for more information.

Input arguments case

label Case

type `list:data:expression`

description Case samples (replicates)

control

label Control

type `list:data:expression`

description Control samples (replicates)

filter.count

label Filter genes based on expression count

type `basic:boolean`

default `True`

filter.min_count_sum

label Minimum raw gene expression count summed over all samples

type `basic:integer`

description Filter genes in the expression matrix input. Remove genes where the expression count sum over all samples is below the threshold.

hidden !filter.count

default `10`

filter.cook

label Filter genes based on Cook's distance

type `basic:boolean`

default `False`

filter.cooks_cutoff

label Threshold on Cook's distance

type `basic:decimal`

description If one or more samples have Cook's distance larger than the threshold set here, the p-value for the row is set to NA. If left empty, the default threshold of 0.99 quantile of the F(p, m-p) distribution is used, where p is the number of coefficients being fitted and m is the number of samples. This test excludes Cook's distance of samples belonging to experimental groups with only two samples.

required False

hidden !filter.cook

filter.independent

label Apply independent gene filtering

type `basic:boolean`

default `False`

filter.alpha

label Significance cut-off used for optimizing independent gene filtering

type `basic:decimal`

description The value should be set to adjusted p-value cut-off (FDR).

hidden !filter.independent

default `0.1`

Output results raw

label Differential expression

type `basic:file`

de_json

label Results table (JSON)

type `basic:json`

de_file

label Results table (file)

type `basic:file`

count_matrix

label Count matrix

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Detect library strandedness

`data:strandednesslibrary-strandedness` (*data:reads:fastq* **reads**, *basic:integer* **read_number**, *data:index:salmon* **salmon_index**) [Source: v0.1.2]

This process uses the Salmon transcript quantification tool to automatically infer the NGS library strandedness. For more details, please see the Salmon [documentation](https://salmon.readthedocs.io/en/latest/library_type.html)

Input arguments reads

label Sequencing reads

type `data:reads:fastq`

description Sequencing reads in .fastq format. Both single and paired-end libraries are supported

read_number

label Number of input reads

type `basic:integer`

description Number of sequencing reads that are subsampled from each of the original .fastq files before library strand detection

default `50000`

salmon_index

label Transcriptome index file

type `data:index:salmon`

description Transcriptome index file created using the Salmon indexing tool. cDNA (transcriptome) sequences used for index file creation must be derived from the same species as the input sequencing reads to obtain the reliable analysis results

Output results strandedness

label Library strandedness type

type `basic:string`

description The predicted library strandedness type. The codes U and IU indicate 'strand non-specific' library for single or paired-end reads, respectively. Codes SF and ISF correspond to the 'strand-specific forward' library, for the single or paired-end reads, respectively. For 'strand-specific reverse' library, the corresponding codes are SR and ISR. For more details, please see the Salmon [documentation](https://salmon.readthedocs.io/en/latest/library_type.html)

fragment_ratio

label Compatible fragment ratio

type `basic:decimal`

description The ratio of fragments that support the predicted library strandedness type

log

label Log file

type `basic:file`

description Analysis log file.

Dictyostelium expressions

`data:expression:polyaexpression-dicty` (*data:alignment:bam* **alignment**, *data:annotation:gff3* **gff**, *data:mappability:bcm* **mappable**) [Source: v1.3.1]

Dictyostelium-specific pipeline. Developed by Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia and Shaulsky Lab, Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX, USA.

Input arguments alignment

label Aligned sequence

type `data:alignment:bam`

gff

label Features (GFF3)

type `data:annotation:gff3`

mappable

label Mappability

type `data:mappability:bcm`

Output results exp

label Expression RPKUM (polyA)

type `basic:file`

description mRNA reads scaled by uniquely mappable part of exons.

rpkmpolya

label Expression RPKM (polyA)

type `basic:file`

description mRNA reads scaled by exon length.

rc

label Read counts (polyA)

type `basic:file`

description mRNA reads uniquely mapped to gene exons.

rpkum

label Expression RPKUM

type `basic:file`

description Reads scaled by uniquely mappable part of exons.

rpkm

label Expression RPKM

type `basic:file`

description Reads scaled by exon length.

rc_raw

label Read counts (raw)

type `basic:file`

description Reads uniquely mapped to gene exons.

exp_json

label Expression RPKUM (polyA) (json)

type `basic:json`

exp_type

label Expression Type (default output)

type `basic:string`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Differential Expression (table)

`data:differentialexpression:uploadupload-diffexp` (*basic:file* **src**, *basic:string* **gene_id**, *basic:string* **logfc**, *basic:string* **fdr**, *basic:string* **logodds**, *basic:string* **fwer**, *basic:string* **pvalue**, *basic:string* **stat**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**, *basic:string* **feature_type**, *list:data:expression* **case**, *list:data:expression* **control**) [Source: v1.2.1]

Upload Differential Expression table.

Input arguments src

label Differential expression file

type `basic:file`

description Differential expression file. Supported file types: *.xls, *.xlsx, *.tab (tab-delimited file), *.diff. DE file must include columns with log2(fold change) and FDR or pval information. DE file must contain header row with column names. Accepts DESeq, DESeq2, edgeR and CuffDiff output files.

validate_regex `\.(xls|xlsx|tab|tab.gz|diff|diff.gz)$`

gene_id

label Gene ID label

type `basic:string`

logfc

label LogFC label

type `basic:string`

fdr

label FDR label

type `basic:string`

required False

logodds

label LogOdds label

type `basic:string`

required False

fwer

label FWER label

type `basic:string`

required False

pvalue

label Pvalue label

type `basic:string`

required False

stat

label Statistics label

type `basic:string`

required False

source

label Gene ID database

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

description Genome build or annotation version.

feature_type

label Feature type

type `basic:string`

default `gene`

choices

- gene: `gene`
- transcript: `transcript`
- exon: `exon`

case

label Case

type `list:data:expression`

description Case samples (replicates)

required False

control

label Control

type `list:data:expression`

description Control samples (replicates)

required False

Output results raw

label Differential expression

type `basic:file`

de_json

label Results table (JSON)

type `basic:json`

de_file

label Results table (file)

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Expression Time Course

`data:etcetc-bcm` (*list:data:expression* **expressions**, *basic:boolean* **avg**) [Source: v1.1.1]

Select gene expression data and form a time course.

Input arguments expressions

label RPKM expression profile

type `list:data:expression`

required True

avg

label Average by time

type `basic:boolean`

default `True`

Output results etcfile

label Expression time course file

type `basic:file`

etc

label Expression time course

type `basic:json`

Expression aggregator

`data:aggregator:expressionexpression-aggregator` (*list:data:expression* **exps**, *basic:string* **group_by**, *data:aggregator:expression* **expr_aggregator**) [Source: v0.2.2]

Collect expression data from samples grouped by sample descriptor field. The Expression aggregator process should not be run in Batch Mode, as this will create redundant outputs. Rather, select multiple samples below for which you wish to aggregate the expression matrix.

Input arguments exps

label Expressions

type `list:data:expression`

group_by

label Sample descriptor field

type `basic:string`

expr_aggregator

label Expression aggregator

type `data:aggregator:expression`

required False

Output results exp_matrix

label Expression matrix

type `basic:file`

box_plot

label Box plot

type `basic:json`

log_box_plot

label Log box plot

type `basic:json`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

exp_type

label Expression type

type `basic:string`

Expression data

`data:expressionupload-expression` (*basic:file* **rc**, *basic:file* **exp**, *basic:string* **exp_name**, *basic:string* **exp_type**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**, *basic:string* **feature_type**) [Source: v2.2.1]

Upload expression data by providing raw expression data (read counts) and/or normalized expression data together with the associated data normalization type.

Input arguments rc

label Read counts (raw expression)

type `basic:file`

description Reads mapped to genomic features (raw count data). Supported extensions: .txt.gz (preferred), .tab.* or .txt.*

required False

validate_regex `\.(txt|tab|gz)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

exp

label Normalized expression

type `basic:file`

description Normalized expression data. Supported extensions: .tab.gz (preferred) or .tab.*

required False

validate_regex `\.(tab|gz)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

exp_name

label Expression name

type `basic:string`

exp_type

label Normalization type

type `basic:string`

description Normalization type

required False

source

label Gene ID source

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

description Genome build or annotation version.

feature_type

label Feature type

type `basic:string`

default `gene`

choices

- gene: `gene`
- transcript: `transcript`
- exon: `exon`

Output results exp

label Normalized expression

type `basic:file`

description Normalized expression

rc

label Read counts

type `basic:file`

description Reads mapped to genomic features.

required False

exp_json

label Expression (json)

type `basic:json`

exp_type

label Expression type

type `basic:string`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Expression data (Cuffnorm)

`data:expressionupload-expression-cuffnorm` (*basic:file* **exp**, *data:cufflinks:cuffquant* **cxb**, *basic:string* **exp_type**) [Source: v1.4.1]

Upload expression data by providing Cuffnorm results.

Input arguments exp

label Normalized expression

type `basic:file`

cxb

label Cuffquant analysis

type `data:cufflinks:cuffquant`

description Cuffquant analysis.

exp_type

label Normalization type

type `basic:string`

default `Cuffnorm`

Output results exp

label Normalized expression

type `basic:file`

description Normalized expression

rc

label Read counts

type `basic:file`

description Reads mapped to genomic features.

required False

exp_json

label Expression (json)

type `basic:json`

exp_type

label Expression type

type `basic:string`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Expression data (STAR)

`data:expression:starupload-expression-star` (*basic:file* **rc**, *basic:string* **stranded**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**, *basic:string* **feature_type**) [Source: v1.3.1]

Upload expression data by providing STAR aligner results.

Input arguments rc

label Read counts (raw expression)

type `basic:file`

description Reads mapped to genomic features (raw count data). Supported extensions: .txt.gz (preferred), .tab.* or .txt.*

validate_regex `\.(txt|tab|gz)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

stranded

label Is data from a strand specific assay?

type `basic:string`

description For stranded=no, a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. For stranded=yes and single-end reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. For stranded=reverse, these rules are reversed.

default `yes`

choices

- yes: `yes`
- no: `no`
- reverse: `reverse`

source

label Gene ID source

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`

- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

description Genome build or annotation version.

feature_type

label Feature type

type `basic:string`

default `gene`

choices

- gene: `gene`
- transcript: `transcript`
- exon: `exon`

Output results rc

label Read counts (raw data)

type `basic:file`

description Reads mapped to genomic features.

exp

label Expression data

type `basic:file`

exp_json

label Expression (json)

type `basic:json`

exp_type

label Expression type

type `basic:string`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Expression matrix

data:expressionsetmergeexpressions (*list:data:expression* **exps**, *list:basic:string* **genes**) [Source: v1.1.1]

Merge expression data to create an expression matrix where each column represents all the gene expression levels from a single experiment, and each row represents the expression of a gene across all experiments.

Input arguments exps

label Gene expressions

type `list:data:expression`

genes

label Filter genes

type `list:basic:string`

required False

Output results expset

label Expression set

type `basic:file`

expset_type

label Expression set type

type `basic:string`

Expression time course

data:etcupload-etc (*basic:file* **src**) [Source: v1.1.1]

Upload Expression time course.

Input arguments src

label Expression time course file (xls or tab)

type `basic:file`

description Expression time course

required True

validate_regex `\.(xls|xlsx|tab)$`

Output results etcfile

label Expression time course file

type `basic:file`

etc

label Expression time course

type `basic:json`

FASTA file

`data:seq:nucleotideupload-fasta-nucl` (*basic:file* **src**, *basic:string* **species**, *basic:string* **build**, *basic:string* **source**) [Source: v2.0.0]

Import a FASTA file, which is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes.

Input arguments src

label Sequence file (FASTA)

type `basic:file`

description Sequence file (containing single or multiple sequences) in FASTA format. Supported extensions: .fasta.gz (preferred), .fa.*, .fna.* or .fasta.*

validate_regex `\.(fasta|fa|fna)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

species

label Species

type `basic:string`

description Species latin name.

required False

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`

build

label Genome build

type `basic:string`

required False

source

label Database source

type `basic:string`

required False

Output results fastagz

label FASTA file (compressed)

type `basic:file`

fasta

label FASTA file

type `basic:file`

fai

label FASTA file index

type `basic:file`

number

label Number of sequences

type `basic:integer`

species

label Species

type `basic:string`

required False

source

label Database source

type `basic:string`

required False

build

label Build

type `basic:string`

required False

FASTQ file (paired-end)

`data:reads:fastq:paired:upload-fastq-paired` (*list:basic:file* **src1**, *list:basic:file* **src2**) [Source: v2.2.1]

Import paired-end reads in FASTQ format, which is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores.

Input arguments src1

label Mate1

type `list:basic:file`

description Sequencing reads in FASTQ format. Supported extensions: .fastq.gz (preferred), .fq.* or .fastq.*

validate_regex `(\.(fastq|fq)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

src2

label Mate2

type `list:basic:file`

description Sequencing reads in FASTQ format. Supported extensions: .fastq.gz (preferred), .fq.* or .fastq.*

validate_regex `(\.(fastq|fq)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

Output results fastq

label Reads file (mate 1)

type `list:basic:file`

fastq2

label Reads file (mate 2)

type `list:basic:file`

fastqc_url

label Quality control with FastQC (Upstream)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (Downstream)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (Upstream)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (Downstream)

type `list:basic:file`

FASTQ file (single-end)

`data:reads:fastq:singleupload-fastq-single` (*list:basic:file* **src**) [Source: v2.2.1]

Import single-end reads in FASTQ format, which is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores.

Input arguments src

label Reads

type `list:basic:file`

description Sequencing reads in FASTQ format. Supported extensions: .fastq.gz (preferred), .fq.* or .fastq.*

validate_regex `(\.(fastq|fq)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

Output results fastq

label Reads file

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

GAF file

`data:gaf:2:0upload-gaf` (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**) [Source: v1.1.1]

GO annotation file (GAF v2.0) relating gene ID and associated GO terms

Input arguments src

label GO annotation file (GAF v2.0)

type `basic:file`

description Upload GO annotation file (GAF v2.0) relating gene ID and associated GO terms

source

label Gene ID database

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- MGI: `MGI`
- NCBI: `NCBI`
- UCSC: `UCSC`
- UniProtKB: `UniProtKB`

species

label Species

type `basic:string`

Output results gaf

label GO annotation file (GAF v2.0)

type `basic:file`

gaf_obj

label GAF object

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

GATK3 (HaplotypeCaller)

`data:variants:vcf:gatk:hcvc-gatk-hc` (*data:alignment:bam* **alignment**, *data:genome:fasta* **genome**, *data:masterfile:amplicon* **intervals**, *data:bed* **intervals_bed**, *data:variants:vcf* **dbsnp**, *basic:integer* **stand_call_conf**, *basic:integer* **stand_emit_conf**, *basic:integer* **mbq**) [Source: v0.4.0]

GATK HaplotypeCaller Variant Calling

Input arguments alignment

label Alignment file (BAM)

type `data:alignment:bam`

genome

label Genome

type `data:genome:fasta`

intervals

label Intervals (from master file)

type `data:masterfile:amplicon`

description Use this option to perform the analysis over only part of the genome. This option is not compatible with ‘‘intervals_bed‘‘ option.

required False

intervals_bed

label Intervals (from BED file)

type `data:bed`

description Use this option to perform the analysis over only part of the genome. This options is not compatible with ‘‘intervals‘‘ option.

required False

dbsnp

label dbSNP file

type `data:variants:vcf`

stand_call_conf

label Min call confidence threshold

type `basic:integer`

description The minimum phred-scaled confidence threshold at which variants should be called.

default `20`

stand_emit_conf

label Emission confidence threshold

type `basic:integer`

description The minimum confidence threshold (phred-scaled) at which the program should emit sites that appear to be possibly variant.

default `20`

mbq

label Min Base Quality

type `basic:integer`

description Minimum base quality required to consider a base for calling.

default `20`

Output results vcf

label Variants

type `basic:file`

tbi

label Tabix index

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

GATK4 (HaplotypeCaller)

`data:variants:vcf:gatk:hcvc-gatk4-hc` (*data:alignment:bam* **alignment**, *data:genome:fasta* **genome**, *data:masterfile:amplicon* **intervals**, *data:bed* **intervals_bed**, *data:variants:vcf* **dbsnp**, *basic:integer* **stand_call_conf**, *basic:integer* **mbq**, *basic:integer* **max_reads**) [Source: v0.2.0]

GATK HaplotypeCaller Variant Calling

Input arguments alignment

label Alignment file (BAM)

type `data:alignment:bam`

genome

label Genome

type `data:genome:fasta`

intervals

label Intervals (from master file)

type `data:masterfile:amplicon`

description Use this option to perform the analysis over only part of the genome. This option is not compatible with ‘‘intervals_bed‘‘ option.

required False

intervals_bed

label Intervals (from BED file)

type `data:bed`

description Use this option to perform the analysis over only part of the genome. This options is not compatible with ‘‘intervals‘‘ option.

required False

dbsnp

label dbSNP file

type `data:variants:vcf`

stand_call_conf

label Min call confidence threshold

type `basic:integer`

description The minimum phred-scaled confidence threshold at which variants should be called.

default `20`

mbq

label Min Base Quality

type `basic:integer`

description Minimum base quality required to consider a base for calling.

default `20`

max_reads

label Max reads per aligment start site

type `basic:integer`

description Maximum number of reads to retain per alignment start position. Reads above this threshold will be downsampled. Set to 0 to disable.

default `50`

Output results vcf

label Variants

type `basic:file`

tbi

label Tabix index

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

GFF3 file

`data:annotation:gff3upload-gff3` (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**) [Source: v3.2.1]

Import a General Feature Format (GFF) file which is a file format used for describing genes and other features of DNA, RNA and protein sequences. See [here](https://useast.ensembl.org/info/website/upload/gff3.html) and [here](https://en.wikipedia.org/wiki/General_feature_format) for more information.

Input arguments src

label Annotation (GFF3)

type `basic:file`

description Annotation in GFF3 format. Supported extensions are: .gff, .gff3 and .gtf

validate_regex `\.(gff|gff3|gtf)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

source

label Gene ID database

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

Output results annot

label Uploaded GFF3 file

type `basic:file`

annot_sorted

label Sorted GFF3 file

type `basic:file`

annot_sorted_idx_igv

label IGV index for sorted GFF3

type `basic:file`

annot_sorted_track_jbrowse

label Jbrowse track for sorted GFF3

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

GO Enrichment analysis

data:goeagoenrichment (*data:ontology:obo* **ontology**, *data:gaf* **gaf**, *list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**, *basic:decimal* **pval_threshold**, *basic:integer* **min_genes**) [Source: v3.2.1]

Identify significantly enriched Gene Ontology terms for given genes.

Input arguments ontology

label Gene Ontology

type `data:ontology:obo`

gaf

label GO annotation file (GAF v2.0)

type `data:gaf`

genes

label List of genes

type `list:basic:string`

placeholder `new gene id`

source

label Source

type `basic:string`

species

label Species

type `basic:string`

description Species latin name. This field is required if gene subset is set.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

pval_threshold

label P-value threshold

type `basic:decimal`

required False

default `0.1`

min_genes

label Minimum number of genes

type `basic:integer`

description Minimum number of genes on a GO term.

required False

default `1`

Output results terms

label Enriched terms

type `basic:json`

source

label Source

type `basic:string`

species

label Species

type `basic:string`

GTF file

data:annotation:gtfupload-gtf (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**) [Source: v3.2.1]

Import a Gene Transfer Format (GTF) file. It is a file format used to hold information about gene structure. It is a tab-delimited text format based on the general feature format (GFF), but contains some additional conventions specific to gene information. See [here](https://en.wikipedia.org/wiki/General_feature_format) for differences between GFF and GTF files.

Input arguments src

label Annotation (GTF)

type `basic:file`

description Annotation in GTF format.

validate_regex `\.(gtf|gff)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

source

label Gene ID database

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Build

type `basic:string`

Output results annot

label Uploaded GTF file

type `basic:file`

annot_sorted

label Sorted GTF file

type `basic:file`

annot_sorted_idx_igv

label IGV index for sorted GTF file

type `basic:file`

required False

annot_sorted_track_jbrowse

label Jbrowse track for sorted GTF

type `basic:file`

required False

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Gene expression indices

data:index:expressionindex-fasta-nucl (*data:seq:nucleotide* **nucl**, *basic:string* **nucl_genome**, *data:genome:fasta* **genome**, *data:annotation:gtf* **annotation**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**) [Source: v0.3.2]

Generate gene expression indices.

Input arguments nucl

label Nucleotide sequence

type `data:seq:nucleotide`

required False

hidden genome

nucl_genome

label Type of nucleotide sequence

type `basic:string`

hidden !nucl

default `gs`

choices

- Genome sequence: `gs`
- Transcript sequences: `ts`

genome

label Genome sequence

type `data:genome:fasta`

required False

hidden nucl

annotation

label Annotation

type `data:annotation:gtf`

required False

hidden nucl && nucl_genome == ‘ts’

source

label Gene ID database

type `basic:string`

required False

hidden !(nucl && nucl_genome == ‘ts’)

choices

- AFFY: `AFFY`

- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

required False

hidden !(nucl && nucl_genome == 'ts')

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Genome build

type `basic:string`

required False

hidden !(nucl && nucl_genome == 'ts')

Output results rsem_index

label RSEM index

type `basic:dir`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Gene set

`data:genesetupload-geneset` (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**) [Source: v1.1.2]

Import a set of genes. Provide one gene ID per line in a .tab, .tab.gz, or .txt file format.

Input arguments src

label Gene set

type `basic:file`

description List of genes (.tab/.txt, one Gene ID per line. Supported extensions: .tab, .tab.gz (preferred), tab.*

validate_regex `(\.(tab|txt)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

source

label Gene ID source

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

Output results geneset

label Gene set

type `basic:file`

geneset_json

label Gene set (JSON)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

Gene set (create from Venn diagram)

`data:geneset:venncreate-geneset-venn` (*list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**, *basic:file* **venn**) [Source: v1.1.2]

Create a gene set from a Venn diagram.

Input arguments genes

label Genes

type `list:basic:string`

description List of genes.

source

label Gene ID source

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

venn

label Venn diagram

type `basic:file`

description JSON file. Supported extensions: .json.gz

validate_regex `(\.json)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

Output results geneset

label Gene set

type `basic:file`

geneset_json

label Gene set (JSON)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

venn

label Venn diagram

type `basic:json`

Gene set (create)

`data:genesetcreate-geneset` (*list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**) [Source: v1.1.2]

Create a gene set from a list of genes.

Input arguments genes

label Genes

type `list:basic:string`

description List of genes.

source

label Gene ID source

type `basic:string`

choices

- AFFY: `AFFY`
- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`

- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

Output results geneset

label Gene set

type `basic:file`

geneset_json

label Gene set (JSON)

type `basic:json`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

Genome

`data:genome:fastaupload-genome` (*basic:file* **src**, *basic:string* **species**, *basic:string* **build**, *basic:file* **bowtie_index**, *basic:file* **bowtie2_index**, *basic:file* **bwa_index**, *basic:file* **hisat2_index**, *basic:file* **subread_index**, *basic:file* **walt_index**) [Source: v3.3.2]

Import genome sequence in FASTA format which includes .fasta.gz (preferred), .fa., .fna., or .fasta extensions.

Input arguments src

label Genome sequence (FASTA)

type `basic:file`

description Genome sequence in FASTA format. Supported extensions: .fasta.gz (preferred), .fa.*, .fna.* or .fasta.*

validate_regex `\.(fasta|fa|fna|fsa)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Genome build

type `basic:string`

advanced.bowtie_index

label Bowtie index files

type `basic:file`

description Bowtie index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

advanced.bowtie2_index

label Bowtie2 index files

type `basic:file`

description Bowtie2 index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

advanced.bwa_index

label BWA index files

type `basic:file`

description BWA index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

advanced.hisat2_index

label HISAT2 index files

type `basic:file`

description HISAT2 index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

advanced.subread_index

label subread index files

type `basic:file`

description Subread index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

advanced.walt_index

label WALT index files

type `basic:file`

description WALT index files. Supported extensions (*.tar.gz).

required False

validate_regex `(\.tar\.gz)$`

Output results fastagz

label Genome FASTA file (compressed)

type `basic:file`

fasta

label Genome FASTA file

type `basic:file`

index_bt

label Bowtie index

type `basic:dir`

index_bt2

label Bowtie2 index

type `basic:dir`

index_bwa

label BWA index

type `basic:dir`

index_hisat2

label HISAT2 index

type `basic:dir`

index_subread

label subread index

type `basic:dir`

index_walt

label WALT index

type `basic:dir`

fai

label Fasta index

type `basic:file`

dict

label Fasta dict

type `basic:file`

fasta_track_jbrowse

label Jbrowse track

type `basic:file`

hidden True

species

label Species

type `basic:string`

build

label Build

type `basic:string`

HISAT2

`data:alignment:bam:hisat2alignment-hisat2` (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:boolean* **softclip**, *basic:integer* **noncansplice**, *basic:boolean* **cufflinks**) [Source: v1.6.1]

HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (both DNA and RNA) to a population of genomes (as well as to a single reference genome). See [here](https://ccb.jhu.edu/software/hisat2/index.shtml) for more information.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

softclip

label Disallow soft clipping

type `basic:boolean`

default `False`

spliced_alignments.noncansplice

label Non-canonical splice sites penalty (optional)

type `basic:integer`

description Sets the penalty for each pair of non-canonical splice sites (e.g. non-GT/AG).

required False

spliced_alignments.cufflinks

label Report alignments tailored specifically for Cufflinks

type `basic:boolean`

description With this option, HISAT2 looks for novel splice sites with three signals (GT/AG, GC/AG, AT/AC), but all user-provided splice sites are used irrespective of their signals. HISAT2 produces an optional field, XS:A:[+-], for every spliced alignment.

default `False`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

stats

label Statistics

type `basic:file`

splice_junctions

label Splice junctions

type `basic:file`

unmapped_f

label Unmapped reads (mate 1)

type `basic:file`

required False

unmapped_r

label Unmapped reads (mate 2)

type `basic:file`

required False

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

HMR

`data:wgbs:hmrhmr` (*data:wgbs:methcounts* **methcounts**) [Source: v1.1.0]

Identify hypo-methylated regions.

Input arguments methcounts

label Methylation levels

type `data:wgbs:methcounts`

description Methylation levels data calculated using methcounts.

Output results hmr

label Hypo-methylated regions

type `basic:file`

tbi_jbrowse

label Bed file index for Jbrowse

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

HTSeq-count (CPM)

`data:expression:htseq:cpmhtseq-count-raw` (*data:alignment:bam* **alignments**, *data:annotation:gtf* **gtf**, *basic:string* **mode**, *basic:string* **stranded**, *basic:string* **feature_class**, *basic:string* **id_attribute**, *basic:string* **feature_type**, *basic:boolean* **name_ordered**) [Source: v1.5.1]

HTSeq-count is useful for preprocessing RNA-Seq alignments for differential expression calling. It counts the number of reads that map to a genomic feature (e.g. gene). For computationally efficient quantification consider using featureCounts instead of HTSeq-count.

The expressions with raw counts, produced by HTSeq are then normalized by computing CPM. See [the official website](https://htseq.readthedocs.io/en/release_0.9.1) and [the introductory paper](https://academic.oup.com/bioinformatics/article/31/2/166/2366196) for more information.

For computationally efficient quantification consider using featureCounts instead of HTSeq-count.

Input arguments alignments

label Aligned reads

type `data:alignment:bam`

gtf

label Annotation (GTF)

type `data:annotation:gtf`

mode

label Mode

type `basic:string`

description Mode to handle reads overlapping more than one feature. Possible values for <mode> are union, intersection-strict and intersection-nonempty

default `union`

choices

- union: `union`
- intersection-strict: `intersection-strict`
- intersection-nonempty: `intersection-nonempty`

stranded

label Is data from a strand specific assay?

type `basic:string`

description For stranded=no, a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. For stranded=yes and single-end reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. For stranded=reverse, these rules are reversed

default `yes`

choices

- yes: `yes`
- no: `no`
- reverse: `reverse`

feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GTF file) to be used. All other features will be ignored.

default `exon`

id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GTF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

feature_type

label Feature type

type `basic:string`

description The type of feature the quantification program summarizes over (e.g. gene or transcript-level analysis).

default `gene`

choices

- gene: `gene`
- transcript: `transcript`

name_ordered

label Use name-ordered BAM file for counting reads

type `basic:boolean`

description Use name-sorted BAM file for reads quantification. Improves compatibility with larger BAM files, but requires more computational time. Setting this to false may cause the process to fail for large BAM files due to buffer overflow.

default `True`

Output results htseq_output

label HTseq-count output

type `basic:file`

rc

label Read count

type `basic:file`

exp

label CPM (Counts per million)

type `basic:file`

exp_json

label CPM (json)

type `basic:json`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

exp_type

label Expression Type (default output)

type `basic:string`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

HTSeq-count (TPM)

`data:expression:htseq:normalizedhtseq-count` (*data:alignment:bam* **alignments**, *data:annotation:gtf* **gff**, *basic:string* **mode**, *basic:string* **stranded**, *basic:string* **feature_class**, *basic:string* **id_attribute**, *basic:string* **feature_type**, *basic:boolean* **name_ordered**) [Source: v1.4.1]

HTSeq-count is useful for preprocessing RNA-Seq alignments for differential expression calling. It counts the number of reads that map to a genomic feature (e.g. gene).

The expressions with raw counts, produced by HTSeq are then normalized by computing FPKM and TPM.

For computationally efficient quantification consider using featureCounts instead of HTSeq-count.

Input arguments alignments

label Aligned reads

type `data:alignment:bam`

gff

label Annotation (GFF)

type `data:annotation:gtf`

mode

label Mode

type `basic:string`

description Mode to handle reads overlapping more than one feature. Possible values for <mode> are union, intersection-strict and intersection-nonempty

default `union`

choices

- union: `union`
- intersection-strict: `intersection-strict`
- intersection-nonempty: `intersection-nonempty`

stranded

label Is data from a strand specific assay?

type `basic:string`

description For stranded=no, a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. For stranded=yes and single-end reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. For stranded=reverse, these rules are reversed

default `yes`

choices

- yes: `yes`
- no: `no`
- reverse: `reverse`

feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GFF file) to be used. All other features will be ignored.

default `exon`

id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GFF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

feature_type

label Feature type

type `basic:string`

description The type of feature the quantification program summarizes over (e.g. gene or transcript-level analysis).

default `gene`

choices

- gene: `gene`
- transcript: `transcript`

name_ordered

label Use name-ordered BAM file for counting reads

type `basic:boolean`

description Use name-sorted BAM file for reads quantification. Improves compatibility with larger BAM files, but requires more computational time. Setting this to false may cause the process to fail for large BAM files due to buffer overflow.

default `True`

Output results htseq_output

label HTseq-count output

type `basic:file`

rc

label Read counts

type `basic:file`

fpkm

label FPKM

type `basic:file`

exp

label TPM (Transcripts Per Million)

type `basic:file`

exp_json

label TPM (json)

type `basic:json`

exp_type

label Expression Type (default output)

type `basic:string`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Hierarchical clustering of genes

`data:clustering:hierarchical:geneclustering-hierarchical-genes` (*list:data:expression* **exps**, *basic:boolean* **advanced**, *list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**, *basic:boolean* **log2**, *basic:boolean* **z_score**, *basic:string* **distance_metric**, *basic:string* **linkage_method**, *basic:boolean* **order**) [Source: v3.0.1]

Hierarchical clustering of genes.

Input arguments exps

label Expressions

type `list:data:expression`

description Select at least two data objects.

advanced

label Show advanced options

type `basic:boolean`

default `False`

preprocessing.genes

label Gene subset

type `list:basic:string`

description Select at least two genes or leave this field empty.

required False

placeholder `new gene id`

preprocessing.source

label Gene ID database of selected genes

type `basic:string`

description This field is required if gene subset is set.

required False

hidden !preprocessing.genes

preprocessing.species

label Species

type `basic:string`

description Species latin name. This field is required if gene subset is set.

required False

hidden !preprocessing.genes

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

preprocessing.log2

label Log-transform expressions

type `basic:boolean`

description Transform expressions with log2(x + 1) before clustering.

default `True`

preprocessing.z_score

label Z-score normalization

type `basic:boolean`

description Use Z-score normalization of gene expressions before clustering.

default `True`

processing.distance_metric

label Distance metric

type `basic:string`

default `pearson`

choices

- Euclidean: `euclidean`
- Pearson: `pearson`
- Spearman: `spearman`

processing.linkage_method

label Linkage method

type `basic:string`

default `average`

choices

- single: `single`
- average: `average`
- complete: `complete`

postprocessing.order

label Order samples optimally

type `basic:boolean`

default `True`

Output results cluster

label Hierarchical clustering

type `basic:json`

required False

Hierarchical clustering of samples

data:clustering:hierarchical:sampleclustering-hierarchical-samples (*list:data:expression* **exps**, *basic:boolean* **advanced**, *list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**, *basic:boolean* **log2**, *basic:boolean* **z_score**, *basic:string* **distance_metric**, *basic:string* **linkage_method**, *basic:boolean* **order**) [Source: v3.0.1]

Hierarchical clustering of samples.

Input arguments exps

  label Expressions

  type `list:data:expression`

  description Select at least two data objects.

advanced

  label Show advanced options

  type `basic:boolean`

  default `False`

preprocessing.genes

  label Gene subset

  type `list:basic:string`

  description Select at least two genes or leave this field empty.

  required False

  placeholder `new gene id`

preprocessing.source

  label Gene ID database of selected genes

  type `basic:string`

  description This field is required if gene subset is set.

required False

hidden !preprocessing.genes

preprocessing.species

label Species

type `basic:string`

description Species latin name. This field is required if gene subset is set.

required False

hidden !preprocessing.genes

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

preprocessing.log2

label Log-transform expressions

type `basic:boolean`

description Transform expressions with log2(x + 1) before clustering.

default `True`

preprocessing.z_score

label Z-score normalization

type `basic:boolean`

description Use Z-score normalization of gene expressions before clustering.

default `True`

processing.distance_metric

label Distance metric

type `basic:string`

default `pearson`

choices

- Euclidean: `euclidean`
- Pearson: `pearson`
- Spearman: `spearman`

processing.linkage_method

label Linkage method

type `basic:string`

default `average`

choices

- single: `single`
- average: `average`
- complete: `complete`

postprocessing.order

label Order samples optimally

type `basic:boolean`

default `True`

Output results cluster

label Hierarchical clustering

type `basic:json`

required False

Indel Realignment and Base Recalibration

`data:alignment:bam:vcvc-realign-recalibrate` (*data:alignment:bam* **alignment**, *data:genome:fasta* **genome**, *list:data:variants:vcf* **known_vars**, *list:data:variants:vcf* **known_indels**) [Source: v1.0.2]

Preprocess BAM file and prepare for Variant Calling.

Input arguments alignment

label Alignment file (BAM)

type `data:alignment:bam`

genome

label Genome

type `data:genome:fasta`

known_vars

label Known sites (dbSNP)

type `list:data:variants:vcf`

known_indels

label Known indels

type `list:data:variants:vcf`

Output results bam

label Alignment file

type `basic:file`

bai

label Index BAI

type `basic:file`

stats

label Stats

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

LoFreq (call)

`data:variants:vcf:lofreqlofreq` (*data:alignment:bam* **alignment**, *data:genome:fasta* **genome**, *data:masterfile:amplicon* **intervals**, *basic:integer* **min_bq**, *basic:integer* **min_alt_bq**) [Source: v0.4.1]

Lofreq (call) Variant Calling.

Input arguments alignment

label Alignment file (BAM)

type `data:alignment:bam`

genome

label Genome

type `data:genome:fasta`

intervals

label Intervals

type `data:masterfile:amplicon`

description Use this option to perform the analysis over only part of the genome.

min_bq

label Min baseQ

type `basic:integer`

description Skip any base with baseQ smaller than the default value.

default `6`

min_alt_bq

label Min alternate baseQ

type `basic:integer`

description Skip alternate bases with baseQ smaller than the default value.

default `6`

Output results vcf

label Variants

type `basic:file`

tbi

label Tabix index

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

MACS 1.4

`data:chipseq:callpeak:macs14macs14` (*data:alignment:bam* **treatment**, *data:alignment:bam* **control**, *basic:string* **pvalue**) [Source: v3.2.1]

Model-based Analysis of ChIP-Seq (MACS 1.4) empirically models the length of the sequenced ChIP fragments, which tends to be shorter than sonication or library construction size estimates, and uses it to improve the spatial resolution of predicted binding sites. MACS also uses a dynamic Poisson distribution to effectively capture local biases in the genome sequence, allowing for more sensitive and robust prediction. See the [original paper](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2592715/) for more information.

Input arguments treatment

label BAM File

type `data:alignment:bam`

control

label BAM Background File

type `data:alignment:bam`

required False

pvalue

label P-value

type `basic:string`

default `1e-9`

choices

- 1e-9: `1e-9`
- 1e-6: `1e-6`

Output results peaks_bed

label Peaks (BED)

type `basic:file`

summits_bed

label Summits (BED)

type `basic:file`

peaks_xls

label Peaks (XLS)

type `basic:file`

wiggle

label Wiggle

type `basic:file`

control_bigwig

label Control (bigWig)

type `basic:file`

required False

treat_bigwig

label Treat (bigWig)

type `basic:file`

peaks_bigbed_igv_ucsc

label Peaks (bigBed)

type `basic:file`

required False

summits_bigbed_igv_ucsc

label Summits (bigBed)

type `basic:file`

required False

peaks_tbi_jbrowse

label JBrowse track peaks file

type `basic:file`

summits_tbi_jbrowse

label JBrowse track summits file

type `basic:file`

model

label Model

type `basic:file`

required False

neg_peaks

label Negative peaks (XLS)

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

MACS 2.0

`data:chipseq:callpeak:macs2macs2-callpeak` (*data:alignment:bam* **case**, *data:alignment:bam* **control**, *data:bed* **promoter**, *basic:boolean* **tagalign**, *basic:integer* **q_threshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**, *basic:string* **duplicates**, *basic:string* **duplicates_prepeak**, *basic:decimal* **qvalue**, *basic:decimal* **pvalue**, *basic:decimal* **pvalue_prepeak**, *basic:integer* **cap_num**, *basic:integer* **mfold_lower**, *basic:integer* **mfold_upper**, *basic:integer* **slocal**, *basic:integer* **llocal**, *basic:integer* **extsize**, *basic:integer* **shift**, *basic:integer* **band_width**, *basic:boolean* **nolambda**, *basic:boolean* **fix_bimodal**, *basic:boolean* **nomodel**, *basic:boolean* **nomodel_prepeak**, *basic:boolean* **down_sample**, *basic:boolean* **bedgraph**, *basic:boolean* **spmr**, *basic:boolean* **call_summits**, *basic:boolean* **broad**, *basic:decimal* **broad_cutoff**) [Source: v4.0.4]

Model-based Analysis of ChIP-Seq (MACS 2.0), is used to identify transcript factor binding sites. MACS 2.0 captures the influence of genome complexity to evaluate the significance of enriched ChIP regions, and MACS improves the spatial resolution of binding sites through combining the information of both sequencing tag position and orientation. It has also an option to link nearby peaks together in order to call broad peaks. See [here](https://github.com/taoliu/MACS/) for more information.

In addition to peak-calling, this process computes ChIP-Seq and ATAC-Seq QC metrics. Process returns a QC metrics report, fragment length estimation, and a deduplicated tagAlign file. QC report contains ENCODE 3 proposed QC metrics – [NRF](https://www.encodeproject.org/data-standards/terms/), [PBC bottlenecking coefficients, NSC, and RSC](https://genome.ucsc.edu/ENCODE/qualityMetrics.html#chipSeq).

Input arguments case

label Case (treatment)

type `data:alignment:bam`

control

label Control (background)

type `data:alignment:bam`

required False

promoter

label Promoter regions BED file

type `data:bed`

description BED file containing promoter regions (TSS+-1000bp for example). Needed to get the number of peaks and reads mapped to promoter regions.

required False

tagalign

label Use tagAlign files

type `basic:boolean`

description Use filtered tagAlign files as case (treatment) and control (background) samples. If extsize parameter is not set, run MACS using input's estimated fragment length.

default `False`

prepeakqc_settings.q_threshold

label Quality filtering threshold

type `basic:integer`

default `30`

prepeakqc_settings.n_sub

label Number of reads to subsample

type `basic:integer`

default `15000000`

prepeakqc_settings.tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on "+" strand by 4bp and reads on "-" strand by 5bp.

default `False`

prepeakqc_settings.shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

required False

settings.duplicates

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden tagalign

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.duplicates_prepeak

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden !tagalign

default `all`

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.qvalue

label Q-value cutoff

type `basic:decimal`

description The q-value (minimum FDR) cutoff to call significant regions. Q-values are calculated from p-values using Benjamini-Hochberg procedure.

required False

disabled settings.pvalue && settings.pvalue_prepeak

settings.pvalue

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

required False

disabled settings.qvalue

hidden tagalign

settings.pvalue_prepeak

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

disabled settings.qvalue

hidden !tagalign || settings.qvalue

default `1e-05`

settings.cap_num

label Cap number of peaks by taking top N peaks

type `basic:integer`

description To keep all peaks set value to 0.

disabled settings.broad

default `500000`

settings.mfold_lower

label MFOLD range (lower limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.mfold_upper

label MFOLD range (upper limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.slocal

label Small local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.llocal

label Large local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.extsize

label extsize

type `basic:integer`

description While '–nomodel' is set, MACS uses this parameter to extend reads in 5'->3' direction to fix-sized fragments. For example, if the size of binding region for your transcription factor is 200 bp, and you want to bypass the model building by MACS, this parameter can be set as 200. This option is only valid when –nomodel is set or when MACS fails to build model and –fix-bimodal is on.

required False

settings.shift

label Shift

type `basic:integer`

description Note, this is NOT the legacy –shiftsize option which is replaced by –extsize! You can set an arbitrary shift in bp here. Please Use discretion while setting it other than default value (0). When –nomodel is set, MACS will use this value to move cutting ends (5') then apply –extsize from 5' to 3' direction to extend them to fragments. When this value is negative, ends will be moved toward 3'->5' direction, otherwise 5'->3' direction. Recommended to keep it as default 0 for ChIP-Seq datasets, or -1 * half of EXTSIZE together with –extsize option for detecting enriched cutting loci such as certain DNAseI-Seq datasets. Note, you can't set values other than 0 if format is BAMPE for paired-end data. Default is 0.

required False

settings.band_width

label Band width

type `basic:integer`

description The band width which is used to scan the genome ONLY for model building. You can set this parameter as the sonication fragment size expected from wet experiment. The previous side effect on the peak detection process has been removed. So this parameter only affects the model building.

required False

settings.nolambda

label Use backgroud lambda as local lambda

type `basic:boolean`

description With this flag on, MACS will use the background lambda as local lambda. This means MACS will not consider the local bias at peak candidate regions.

default `False`

settings.fix_bimodal

label Turn on the auto paired-peak model process

type `basic:boolean`

description Whether turn on the auto paired-peak model process. If it's set, when MACS failed to build paired model, it will use the nomodel settings, the '–extsize' parameter to extend each tags. If set, MACS will be terminated if paired-peak model is failed.

default `False`

settings.nomodel

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden tagalign

default `False`

settings.nomodel_prepeak

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden !tagalign

default `True`

settings.down_sample

label Down-sample

type `basic:boolean`

description When set, random sampling method will scale down the bigger sample. By default, MACS uses linear scaling. This option will make the results unstable and irreproducible since each time, random reads would be selected, especially the numbers (pileup, pvalue, qvalue) would change. Consider to use 'randsample' script before MACS2 runs instead.

default `False`

settings.bedgraph

label Save fragment pileup and control lambda

type `basic:boolean`

description If this flag is on, MACS will store the fragment pileup, control lambda, -log10pvalue and -log10qvalue scores in bedGraph files. The bedGraph files will be stored in current directory named NAME+'_treat_pileup.bdg' for treatment data, NAME+'_control_lambda.bdg' for local lambda values from control, NAME+'_treat_pvalue.bdg' for Poisson pvalue scores (in -log10(pvalue) form), and NAME+'_treat_qvalue.bdg' for q-value scores from Benjamini-Hochberg-Yekutieli procedure.

default `True`

settings.spmr

label Save signal per million reads for fragment pileup profiles

type `basic:boolean`

disabled settings.bedgraph === false

default `True`

settings.call_summits

label Call summits

type `basic:boolean`

description MACS will now reanalyze the shape of signal profile (p or q-score depending on cutoff setting) to deconvolve subpeaks within each peak called from general procedure. It's highly recommended to detect adjacent binding events. While used, the output subpeaks of a big peak region will have the same peak boundaries, and different scores and peak summit positions.

default `False`

settings.broad

label Composite broad regions

type `basic:boolean`

description When this flag is on, MACS will try to composite broad regions in BED12 (a gene-model-like format) by putting nearby highly enriched regions into a broad region with loose cutoff. The broad region is controlled by another cutoff through –broad-cutoff. The maximum length of broad region length is 4 times of d from MACS.

disabled settings.call_summits === true

default `False`

settings.broad_cutoff

label Broad cutoff

type `basic:decimal`

description Cutoff for broad region. This option is not available unless –broad is set. If -p is set, this is a p-value cutoff, otherwise, it's a q-value cutoff. DEFAULT = 0.1

required False

disabled settings.call_summits === true || settings.broad !== true

Output results called_peaks

label Called peaks

type `basic:file`

narrow_peaks

label Narrow peaks

type `basic:file`

required False

chip_qc

label QC report

type `basic:file`

required False

case_prepeak_qc

label Pre-peak QC report (case)

type `basic:file`

case_tagalign

label Filtered tagAlign (case)

type `basic:file`

control_prepeak_qc

label Pre-peak QC report (control)

type `basic:file`

required False

control_tagalign

label Filtered tagAlign (control)

type `basic:file`

required False

narrow_peaks_bigbed_igv_ucsc

label Narrow peaks (BigBed)

type `basic:file`

required False

summits

label Peak summits

type `basic:file`

required False

summits_tbi_jbrowse

label Peak summits tbi index for JBrowse

type `basic:file`

required False

summits_bigbed_igv_ucsc

label Summits (bigBed)

type `basic:file`

required False

broad_peaks

label Broad peaks

type `basic:file`

required False

gappedPeak

label Broad peaks (bed12/gappedPeak)

type `basic:file`

required False

treat_pileup

label Treatment pileup (bedGraph)

type `basic:file`

required False

treat_pileup_bigwig

label Treatment pileup (bigWig)

type `basic:file`

required False

control_lambda

label Control lambda (bedGraph)

type `basic:file`

required False

control_lambda_bigwig

label Control lambda (bigwig)

type `basic:file`

required False

model

label Model

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

MACS2 - ROSE2

data:workflow:chipseq:macs2rose2workflow-macs-rose (*data:alignment:bam* **case**, *data:alignment:bam* **control**, *data:bed* **promoter**, *basic:boolean* **tagalign**, *basic:integer* **q_threshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**, *basic:string* **duplicates**, *basic:string* **duplicates_prepeak**, *basic:decimal* **qvalue**, *basic:decimal* **pvalue**, *basic:decimal* **pvalue_prepeak**, *basic:integer* **cap_num**, *basic:integer* **mfold_lower**, *basic:integer* **mfold_upper**, *basic:integer* **slocal**, *basic:integer* **llocal**, *basic:integer* **extsize**, *basic:integer* **shift**, *basic:integer* **band_width**, *basic:boolean* **nolambda**, *basic:boolean* **fix_bimodal**, *basic:boolean* **nomodel**, *basic:boolean* **nomodel_prepeak**, *basic:boolean* **down_sample**, *basic:boolean* **bedgraph**, *basic:boolean* **spmr**, *basic:boolean* **call_summits**, *basic:boolean* **broad**, *basic:decimal* **broad_cutoff**, *basic:integer* **tss**, *basic:integer* **stitch**, *data:bed* **mask**) [Source: v1.0.1]

Input arguments case

label Case (treatment)

type `data:alignment:bam`

control

label Control (background)

type `data:alignment:bam`

required False

promoter

label Promoter regions BED file

type `data:bed`

description BED file containing promoter regions (TSS+-1000bp for example). Needed to get the number of peaks and reads mapped to promoter regions.

required False

tagalign

label Use tagAlign files

type `basic:boolean`

description Use filtered tagAlign files as case (treatment) and control (background) samples. If extsize parameter is not set, run MACS using input's estimated fragment length.

default `False`

prepeakqc_settings.q_threshold

label Quality filtering threshold

type `basic:integer`

default `30`

prepeakqc_settings.n_sub

label Number of reads to subsample

type `basic:integer`

default `15000000`

prepeakqc_settings.tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on "+" strand by 4bp and reads on "-" strand by 5bp.

default `False`

prepeakqc_settings.shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

required False

settings.duplicates

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden tagalign

choices

- 1: `1`

- auto: `auto`
- all: `all`

settings.duplicates_prepeak

label Number of duplicates

type `basic:string`

description It controls the MACS behavior towards duplicate tags at the exact same location – the same coordination and the same strand. The 'auto' option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff and the 'all' option keeps all the tags. If an integer is given, at most this number of tags will be kept at the same location. The default is to keep one tag at the same location.

required False

hidden !tagalign

default `all`

choices

- 1: `1`
- auto: `auto`
- all: `all`

settings.qvalue

label Q-value cutoff

type `basic:decimal`

description The q-value (minimum FDR) cutoff to call significant regions. Q-values are calculated from p-values using Benjamini-Hochberg procedure.

required False

disabled settings.pvalue && settings.pvalue_prepeak

settings.pvalue

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

required False

disabled settings.qvalue

hidden tagalign

settings.pvalue_prepeak

label P-value cutoff

type `basic:decimal`

description The p-value cutoff. If specified, MACS2 will use p-value instead of q-value cutoff.

disabled settings.qvalue

hidden !tagalign || settings.qvalue

default `1e-05`

settings.cap_num

label Cap number of peaks by taking top N peaks

type `basic:integer`

description To keep all peaks set value to 0.

disabled settings.broad

default `500000`

settings.mfold_lower

label MFOLD range (lower limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.mfold_upper

label MFOLD range (upper limit)

type `basic:integer`

description This parameter is used to select the regions within MFOLD range of high-confidence enrichment ratio against background to build model. The regions must be lower than upper limit, and higher than the lower limit of fold enrichment. DEFAULT:10,30 means using all regions not too low (>10) and not too high (<30) to build paired-peaks model. If MACS can not find more than 100 regions to build model, it will use the –extsize parameter to continue the peak detection ONLY if –fix-bimodal is set.

required False

settings.slocal

label Small local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.llocal

label Large local region

type `basic:integer`

description Slocal and llocal parameters control which two levels of regions will be checked around the peak regions to calculate the maximum lambda as local lambda. By default, MACS considers 1000bp for small local region (–slocal), and 10000bps for large local region (–llocal) which captures the bias from a long range effect like an open chromatin domain. You can tweak these according to your project. Remember that if the region is set too small, a sharp spike in the input data may kill the significant peak.

required False

settings.extsize

label extsize

type `basic:integer`

description While '–nomodel' is set, MACS uses this parameter to extend reads in 5'->3' direction to fix-sized fragments. For example, if the size of binding region for your transcription factor is 200 bp, and you want to bypass the model building by MACS, this parameter can be set as 200. This option is only valid when –nomodel is set or when MACS fails to build model and –fix-bimodal is on.

required False

settings.shift

label Shift

type `basic:integer`

description Note, this is NOT the legacy –shiftsize option which is replaced by –extsize! You can set an arbitrary shift in bp here. Please Use discretion while setting it other than default value (0). When –nomodel is set, MACS will use this value to move cutting ends (5') then apply –extsize from 5' to 3' direction to extend them to fragments. When this value is negative, ends will be moved toward 3'->5' direction, otherwise 5'->3' direction. Recommended to keep it as default 0 for ChIP-Seq datasets, or -1 * half of EXTSIZE together with –extsize option for detecting enriched cutting loci such as certain DNAseI-Seq datasets. Note, you can't set values other than 0 if format is BAMPE for paired-end data. Default is 0.

required False

settings.band_width

label Band width

type `basic:integer`

description The band width which is used to scan the genome ONLY for model building. You can set this parameter as the sonication fragment size expected from wet experiment. The previous side effect on the peak detection process has been removed. So this parameter only affects the model building.

required False

settings.nolambda

label Use backgroud lambda as local lambda

type `basic:boolean`

description With this flag on, MACS will use the background lambda as local lambda. This means MACS will not consider the local bias at peak candidate regions.

default `False`

settings.fix_bimodal

label Turn on the auto paired-peak model process

type `basic:boolean`

description Whether turn on the auto paired-peak model process. If it's set, when MACS failed to build paired model, it will use the nomodel settings, the '–extsize' parameter to extend each tags. If set, MACS will be terminated if paired-peak model is failed.

default `False`

settings.nomodel

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden tagalign

default `False`

settings.nomodel_prepeak

label Bypass building the shifting model

type `basic:boolean`

description While on, MACS will bypass building the shifting model.

hidden !tagalign

default `True`

settings.down_sample

label Down-sample

type `basic:boolean`

description When set, random sampling method will scale down the bigger sample. By default, MACS uses linear scaling. This option will make the results unstable and irreproducible since each time, random reads would be selected, especially the numbers (pileup, pvalue, qvalue) would change. Consider to use 'randsample' script before MACS2 runs instead.

default `False`

settings.bedgraph

label Save fragment pileup and control lambda

type `basic:boolean`

description If this flag is on, MACS will store the fragment pileup, control lambda, -log10pvalue and -log10qvalue scores in bedGraph files. The bedGraph files will be stored in current directory named NAME+'_treat_pileup.bdg' for treatment data, NAME+'_control_lambda.bdg' for local lambda values from control, NAME+'_treat_pvalue.bdg' for Poisson pvalue scores (in -log10(pvalue) form), and NAME+'_treat_qvalue.bdg' for q-value scores from Benjamini-Hochberg-Yekutieli procedure.

default `True`

settings.spmr

label Save signal per million reads for fragment pileup profiles

type `basic:boolean`

disabled settings.bedgraph === false

default `True`

settings.call_summits

label Call summits

type `basic:boolean`

description MACS will now reanalyze the shape of signal profile (p or q-score depending on cutoff setting) to deconvolve subpeaks within each peak called from general procedure. It's highly recommended to detect adjacent binding events. While used, the output subpeaks of a big peak region will have the same peak boundaries, and different scores and peak summit positions.

default `False`

settings.broad

label Composite broad regions

type `basic:boolean`

description When this flag is on, MACS will try to composite broad regions in BED12 (a gene-model-like format) by putting nearby highly enriched regions into a broad region with loose cutoff. The broad region is controlled by another cutoff through –broad-cutoff. The maximum length of broad region length is 4 times of d from MACS.

disabled settings.call_summits === true

default `False`

settings.broad_cutoff

label Broad cutoff

type `basic:decimal`

description Cutoff for broad region. This option is not available unless –broad is set. If -p is set, this is a p-value cutoff, otherwise, it's a q-value cutoff. DEFAULT = 0.1

required False

disabled settings.call_summits === true || settings.broad !== true

rose_settings.tss

label TSS exclusion

type `basic:integer`

description Enter a distance from TSS to exclude. 0 = no TSS exclusion

default `0`

rose_settings.stitch

label Stitch

type `basic:integer`

description Enter a max linking distance for stitching. If not given, optimal stitching parameter will be determined automatically.

required False

rose_settings.mask

label Masking BED file

type `data:bed`

description Mask a set of regions from analysis. Provide a BED of masking regions.

required False

Output results

Mappability

`data:mappability:bcmmappability-bcm` (*data:genome:fasta* **genome**, *data:annotation:gff3* **gff**, *basic:integer* **length**) [Source: v2.0.1]

Compute genome mappability. Developed by Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia and Shaulsky's Lab, Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX, USA.

Input arguments genome

label Reference genome

type `data:genome:fasta`

gff

label General feature format

type `data:annotation:gff3`

length

label Read length

type `basic:integer`

default `50`

Output results mappability

label Mappability

type `basic:file`

Mappability info

`data:mappability:bcmupload-mappability` (*basic:file* **src**) [Source: v1.1.1]

Upload mappability information.

Input arguments src

label Mappability file

type `basic:file`

description Mappability file: 2 column tab separated

validate_regex `\.(tab)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

Output results mappability

label Uploaded mappability

type `basic:file`

Merge Expressions (ETC)

data:expressionset:etcmergeetc (*list:data:etc* **exps**, *list:basic:string* **genes**) [Source: v1.1.1]

Merge Expression Time Course (ETC) data.

Input arguments exps

label Expression Time Course (ETC)

type `list:data:etc`

genes

label Filter genes

type `list:basic:string`

required False

Output results expset

label Expression set

type `basic:file`

expset_type

label Expression set type

type `basic:string`

Metabolic pathway file

data:metabolicpathwayupload-metabolic-pathway (*basic:file* **src**, *basic:string* **source**, *basic:string* **species**) [Source: v1.0.1]

Upload pathway json.

Input arguments src

label Pathway file

type `basic:file`

description JSON file. Supported extensions: '.json', '.json.gz'

validate_regex `(\.json)(\.gz)?$`

source

label Gene ID database

type `basic:string`

choices

- BIGG: `BIGG`

species

label Species

type `basic:string`

choices

- Homo Sapiens: `Homo Sapiens`

- Mus musculus: `Mus musculus`

Output results pathway

label Pathway json

type `basic:json`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

MultiQC

`data:multiqcmultiqc` (*list:data* **data**, *basic:boolean* **dirs**, *basic:boolean* **fullnames**, *basic:boolean* **config**, *basic:string* **cl_config**) [Source: v1.1.2]

Aggregate results from bioinformatics analyses across many samples into a single report. [MultiQC](http://www.multiqc.info) searches a given directory for analysis logs and compiles a HTML report. It's a general use tool, perfect for summarising the output from numerous bioinformatics tools.

Input arguments data

label Input data

type `list:data`

description Select multiple data objects for which the MultiQC report is to be generated.

advanced.dirs

label –dirs

type `basic:boolean`

description Prepend directory to sample names.

default `True`

advanced.fullnames

label –fullnames

type `basic:boolean`

description Do not clean the sample names (leave as full file name).

default `False`

advanced.config

label Use configuration file

type `basic:boolean`

description Use Genialis configuration file for MultiQC report.

default `True`

advanced.cl_config

label –cl-config

type `basic:string`

description Enter text with command-line configuration options to override the defaults (e.g. custom_logo_url: https://www.genialis.com).

required False

Output results report

label MultiQC report

type `basic:file:html`

report_data

label Report data

type `basic:dir`

OBO file

`data:ontology:oboupload-obo` (*basic:file* **src**) [Source: v1.1.1]

Upload gene ontology in OBO format.

Input arguments src

label Gene ontology (OBO)

type `basic:file`

description Gene ontology in OBO format.

required True

validate_regex `\.obo(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

Output results obo

label Ontology file

type `basic:file`

obo_obj

label OBO object

type `basic:file`

PCA

`data:pcapca` (*list:data:expression* **exps**, *list:basic:string* **genes**, *basic:string* **source**, *basic:string* **species**) [Source: v2.1.1]

Principal component analysis (PCA)

Input arguments exps

label Expressions

type `list:data:expression`

genes

label Gene subset

type `list:basic:string`

required False

source

label Gene ID database of selected genes

type `basic:string`

description This field is required if gene subset is set.

required False

species

label Species

type `basic:string`

description Species latin name. This field is required if gene subset is set.

required False

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

Output results pca

label PCA

type `basic:json`

Picard CollectTargetedPcrMetrics

`data:picard:coveragepicard-pcrmetrics` (*data:alignment:bam* **alignment**, *data:masterfile:amplicon* **master_file**, *data:genome:fasta* **genome**) [Source: v0.2.1]

Calculate PCR-related metrics from targeted sequencing data using the Picard CollectTargetedPcrMetrics tool

Input arguments alignment

label Alignment file (BAM)

type `data:alignment:bam`

master_file

label Master file

type `data:masterfile:amplicon`

genome

label Genome

type `data:genome:fasta`

Output results target_pcr_metrics

label Target PCR metrics

type `basic:file`

target_coverage

label Target coverage

type `basic:file`

Pre-peakcall QC

`data:prepeakqcqc-prepeak` (*data:alignment:bam* **alignment**, *basic:integer* **q_treshold**, *basic:integer* **n_sub**, *basic:boolean* **tn5**, *basic:integer* **shift**) [Source: v0.2.2]

ChIP-Seq and ATAC-Seq QC metrics. Process returns a QC metrics report, fragment length estimation, and a deduplicated tagAlign file. Both fragment length estimation and the tagAlign file can be used as inputs in MACS 2.0. QC report contains ENCODE 3 proposed QC metrics – [NRF, PBC bottlenecking coefficients](https://www.encodeproject.org/data-standards/terms/), [NSC, and RSC](https://genome.ucsc.edu/ENCODE/qualityMetrics.html#chipSeq).

Input arguments alignment

label Aligned reads

type `data:alignment:bam`

q_treshold

label Quality filtering treshold

type `basic:integer`

default `30`

n_sub

label Number of reads to subsample

type `basic:integer`

default `15000000`

tn5

label TN5 shifting

type `basic:boolean`

description Tn5 transposon shifting. Shift reads on “+” strand by 4bp and reads on “-” strand by 5bp.

default `False`

shift

label User-defined cross-correlation peak strandshift

type `basic:integer`

description If defined, SPP tool will not try to estimate fragment length but will use the given value as fragment length.

required False

Output results chip_qc

label QC report

type `basic:file`

tagalign

label Filtered tagAlign

type `basic:file`

fraglen

label Fragnment length

type `basic:integer`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Prepare GEO - ChIP-Seq

`data:other:geo:chipseqprepare-geo-chipseq` (*list:data:reads:fastq* **reads**, *list:data:chipseq:callpeak* **macs**, *basic:string* **name**) [Source: v2.0.2]

Prepare ChIP-seq data for GEO upload.

Input arguments reads

label Reads

type `list:data:reads:fastq`

description List of reads objects. Fastq files will be used.

macs

label MACS

type `list:data:chipseq:callpeak`

description List of MACS2 or MACS14 objects. BedGraph (MACS2) or Wiggle (MACS14) files will be used.

name

label Collection name

type `basic:string`

Output results tarball

label GEO folder

type `basic:file`

table

label Annotation table

type `basic:file`

Prepare GEO - RNA-Seq

`data:other:geo:rnaseqprepare-geo-rnaseq` (*list:data:reads:fastq* **reads**, *list:data:expression* **expressions**, *basic:string* **name**) [Source: v0.1.1]

Prepare RNA-Seq data for GEO upload.

Input arguments reads

label Reads

type `list:data:reads:fastq`

description List of reads objects. Fastq files will be used.

expressions

label Expressions

type `list:data:expression`

description Cuffnorm data object. Expression table will be used.

name

label Collection name

type `basic:string`

Output results tarball

label GEO folder

type `basic:file`

table

label Annotation table

type `basic:file`

Quantify shRNA species using bowtie2

`data:expression:shrna2quantshrna-quant` (*data:alignment:bam* **alignment**, *basic:integer* **readlengths**, *basic:integer* **alignscores**) [Source: v1.0.0]

Based on ‘bowtie2‘ output (.bam file) calculate number of mapped species. Input is limited to results from ‘bowtie2‘ since ‘YT:Z:‘ tag used to fetch aligned species is specific to this process. Result is a count matrix (successfully mapped reads) where species are in rows columns contain read specifics (count, species name, sequence, ‘AS:i:‘ tag value).

Input arguments alignment

label Alignment

type `data:alignment:bam`

required True

readlengths

label Species lengths threshold

type `basic:integer`

description Species with read lengths below specified threshold will be removed from final output. Default is no removal.

alignscores

label Align scores filter threshold

type `basic:integer`

description Species with align score below specified threshold will be removed from final output. Default is no removal.

Output results exp

label Normalized expression

type `basic:file`

rc

label Read counts

type `basic:file`

required False

exp_json

label Expression (json)

type `basic:json`

exp_type

label Expression type

type `basic:string`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

mapped_species

label Mapped species

type `basic:file`

RNA-Seq (Cuffquant)

`data:workflow:rnaseq:cuffquantworkflow-rnaseq-cuffquant` (*data:reads:fastq* **reads**, *data:genome:fasta* **genome**, *data:annotation* **annotation**) [Source: v1.0.0]

Input arguments reads

label Input reads

type `data:reads:fastq`

genome

label genome

type `data:genome:fasta`

annotation

label Annotation file

type `data:annotation`

Output results

ROSE2

`data:chipseq:rose2rose2` (*data:chipseq:callpeak* **input**, *data:bed* **input_upload**, *data:alignment:bam* **rankby**, *data:alignment:bam* **control**, *basic:integer* **tss**, *basic:integer* **stitch**, *data:bed* **mask**) [Source: v4.2.1]

For identification of super enhancers R2 uses the Rank Ordering of Super-Enhancers algorithm (ROSE2). This takes the peaks called by RSEG for acetylation and calculates the distances in-between to judge whether they can be considered super-enhancers. The ranked values can be plotted and by locating the inflection point in the resulting graph, super-enhancers can be assigned. It can also be used with the MACS calculated data. See [here](http://younglab.wi.mit.edu/super_enhancer_code.html) for more information.

Input arguments input

label BED/narrowPeak file (MACS results)

type `data:chipseq:callpeak`

required False

input_upload

label BED file (Upload)

type `data:bed`

required False

rankby

label BAM File

type `data:alignment:bam`

description bamfile to rank enhancer by

control

label Control BAM File

type `data:alignment:bam`

description bamfile to rank enhancer by

required False

tss

label TSS exclusion

type `basic:integer`

description Enter a distance from TSS to exclude. 0 = no TSS exclusion

default `0`

stitch

label Stitch

type `basic:integer`

description Enter a max linking distance for stitching. If not given, optimal stitching parameter will be determined automatically.

required False

mask

label Masking BED file

type `data:bed`

description Mask a set of regions from analysis. Provide a BED of masking regions.

required False

Output results all_enhancers

label All enhancers table

type `basic:file`

enhancers_with_super

label Super enhancers table

type `basic:file`

plot_points

label Plot points

type `basic:file`

plot_panel

label Plot panel

type `basic:file`

enhancer_gene

label Enhancer to gene

type `basic:file`

enhancer_top_gene

label Enhancer to top gene

type `basic:file`

gene_enhancer

label Gene to Enhancer

type `basic:file`

stitch_parameter

label Stitch parameter

type `basic:file`

required False

all_output

label All output

type `basic:file`

scatter_plot

label Super-Enhancer plot

type `basic:json`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

RSEM

`data:expression:rsemrsem` (*data:alignment:bam* **alignments**, *basic:string* **read_type**, *data:index:expression* **expression_index**, *basic:string* **strandedness**) [Source: v1.1.1]

RSEM is a software package for estimating gene and isoform expression levels from RNA-Seq data. The RSEM package supports threads for parallel computation of the EM algorithm, single-end and paired-end read data, quality scores, variable-length reads and RSPD estimation. See [here](https://deweylab.github.io/RSEM/README.html) and the [original paper](https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-323) for more information.

Input arguments alignments

label Aligned reads

type `data:alignment:bam`

read_type

label Type of reads

type `basic:string`

default `se`

choices

- Single-end: `se`
- Paired-end: `pe`

expression_index

label Gene expression indices

type `data:index:expression`

strandedness

label Strandedness

type `basic:string`

default `none`

choices

- None: `none`
- Forward: `forward`
- Reverse: `reverse`

Output results rc

label Read counts

type `basic:file`

fpkm

label FPKM

type `basic:file`

exp

label TPM (Transcripts Per Million)

type `basic:file`

exp_json

label TPM (json)

type `basic:json`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

genes

label Results grouped by gene

type `basic:file`

transcripts

label Results grouped by transcript

type `basic:file`

log

label RSEM log

type `basic:file`

exp_type

label Type of expression

type `basic:string`

source

label Transcript ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

Reads (QSEQ multiplexed, paired)

`data:multiplexed:qseq:pairedupload-multiplexed-paired` (*basic:file* **reads**, *basic:file* **reads2**, *basic:file* **barcodes**, *basic:file* **annotation**) [Source: v1.1.1]

Upload multiplexed NGS reds in QSEQ format.

Input arguments reads

label Multiplexed upstream reads

type `basic:file`

description NGS reads in QSeq format. Supported extensions: .qseq.txt.bz2 (preferred), .qseq.* or .qseq.txt.*.

required True

validate_regex `((\.qseq|\.qseq\.txt)(\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

reads2

label Multiplexed downstream reads

type `basic:file`

description NGS reads in QSeq format. Supported extensions: .qseq.txt.bz2 (preferred), .qseq.* or .qseq.txt.*.

required True

validate_regex `((\.qseq|\.qseq\.txt)(\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

barcodes

label NGS barcodes

type `basic:file`

description Barcodes in QSeq format. Supported extensions: .qseq.txt.bz2 (preferred), .qseq.* or .qseq.txt.*.

required True

validate_regex `((\.qseq|\.qseq\.txt)(\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

annotation

label Barcode mapping

type `basic:file`

description A tsv file mapping barcodes to experiment name, e.g. “TCGCAGG\tHr00”.

required True

validate_regex `(\.csv|\.tsv)$`

Output results qseq_reads

label Multiplexed upstream reads

type `basic:file`

qseq_reads2

label Multiplexed downstream reads

type `basic:file`

qseq_barcodes

label NGS barcodes

type `basic:file`

annotation

label Barcode mapping

type `basic:file`

matched

label Matched

type `basic:string`

notmatched

label Not matched

type `basic:string`

badquality

label Bad quality

type `basic:string`

skipped

label Skipped

type `basic:string`

Reads (QSEQ multiplexed, single)

`data:multiplexed:qseq:singleupload-multiplexed-single` (*basic:file* **reads**, *basic:file* **barcodes**, *basic:file* **annotation**) [Source: v1.1.1]

Upload multiplexed NGS reds in QSEQ format.

Input arguments reads

label Multiplexed NGS reads

type `basic:file`

description NGS reads in QSeq format. Supported extensions: .qseq.txt.bz2 (preferred), .qseq.* or .qseq.txt.*.

required True

validate_regex `(\.(qseq)(|\.txt)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

barcodes

label NGS barcodes

type `basic:file`

description Barcodes in QSeq format. Supported extensions: .qseq.txt.bz2 (preferred), .qseq.* or .qseq.txt.*.

required True

validate_regex `(\.(qseq)(|\.txt)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z))|(\.bz2)$`

annotation

label Barcode mapping

type `basic:file`

description A tsv file mapping barcodes to experiment name, e.g. “TCGCAGG\tHr00”.

required True

validate_regex `(\.csv|\.tsv)$`

Output results qseq_reads

label Multiplexed NGS reads

type `basic:file`

qseq_barcodes

label NGS barcodes

type `basic:file`

annotation

label Barcode mapping

type `basic:file`

matched

label Matched

type `basic:string`

notmatched

label Not matched

type `basic:string`

badquality

label Bad quality

type `basic:string`

skipped

label Skipped

type `basic:string`

SAM header

`data:sam:headerupload-header-sam` (*basic:file* **src**) [Source: v1.1.1]

Upload a mapping file header in SAM format.

Input arguments src

label Header (SAM)

type `basic:file`

description A mapping file header in SAM format.

validate_regex `\.(sam)$`

Output results sam

label Uploaded file

type `basic:file`

SRA data

data:sraimport-sra (*basic:string* **sra_accession**, *basic:boolean* **show_advanced**, *basic:integer* **min_spot_id**, *basic:integer* **max_spot_id**, *basic:integer* **min_read_len**, *basic:boolean* **clip**, *basic:boolean* **aligned**, *basic:boolean* **unaligned**) [Source: v0.1.1]

Import single or paired-end reads from Sequence Read Archive (SRA) via an SRA accession number. SRA stores raw sequencing data and alignment information from high-throughput sequencing platforms.

Input arguments sra_accession

label SRA accession

type `basic:string`

show_advanced

label Show advanced options

type `basic:boolean`

default `False`

advanced.min_spot_id

label Minimum spot ID

type `basic:integer`

required False

advanced.max_spot_id

label Maximum spot ID

type `basic:integer`

required False

advanced.min_read_len

label Minimum read length

type `basic:integer`

required False

advanced.clip

label Clip adapter sequences

type `basic:boolean`

default `False`

advanced.aligned

label Dump only aligned sequences

type `basic:boolean`

default `False`

advanced.unaligned

label Dump only unaligned sequences

type `basic:boolean`

default `False`

Output results

SRA data (paired-end)

`data:reads:fastq:pairedimport-sra-paired` (*basic:string* **sra_accession**, *basic:boolean* **show_advanced**, *basic:integer* **min_spot_id**, *basic:integer* **max_spot_id**, *basic:integer* **min_read_len**, *basic:boolean* **clip**, *basic:boolean* **aligned**, *basic:boolean* **unaligned**) [Source: v0.1.1]

Import paired-end reads from Sequence Read Archive (SRA) via an SRA accession number. SRA stores raw sequencing data and alignment information from high-throughput sequencing platforms.

Input arguments sra_accession

label SRA accession

type `basic:string`

show_advanced

label Show advanced options

type `basic:boolean`

default `False`

advanced.min_spot_id

label Minimum spot ID

type `basic:integer`

required False

advanced.max_spot_id

label Maximum spot ID

type `basic:integer`

required False

advanced.min_read_len

label Minimum read length

type `basic:integer`

required False

advanced.clip

label Clip adapter sequences

type `basic:boolean`

default `False`

advanced.aligned

label Dump only aligned sequences

type `basic:boolean`

default `False`

advanced.unaligned

label Dump only unaligned sequences

type `basic:boolean`

default `False`

Output results fastq

label Reads file (mate 1)

type `list:basic:file`

fastq2

label Reads file (mate 2)

type `list:basic:file`

fastqc_url

label Quality control with FastQC (Upstream)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (Downstream)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (Upstream)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (Downstream)

type `list:basic:file`

SRA data (single-end)

`data:reads:fastq:singleimport-sra-single` (*basic:string* **sra_accession**, *basic:boolean* **show_advanced**, *basic:integer* **min_spot_id**, *basic:integer* **max_spot_id**, *basic:integer* **min_read_len**, *basic:boolean* **clip**, *basic:boolean* **aligned**, *basic:boolean* **unaligned**) [Source: v0.1.1]

Import single-end reads from Sequence Read Archive (SRA) via an SRA accession number. SRA stores raw sequencing data and alignment information from high-throughput sequencing platforms.

Input arguments sra_accession

label SRA accession

type `basic:string`

show_advanced

label Show advanced options

type `basic:boolean`

default `False`

advanced.min_spot_id

label Minimum spot ID

type `basic:integer`

required False

advanced.max_spot_id

label Maximum spot ID

type `basic:integer`

required False

advanced.min_read_len

label Minimum read length

type `basic:integer`

required False

advanced.clip

label Clip adapter sequences

type `basic:boolean`

default `False`

advanced.aligned

label Dump only aligned sequences

type `basic:boolean`

default `False`

advanced.unaligned

label Dump only unaligned sequences

type `basic:boolean`

default `False`

Output results fastq

label Reads file

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

STAR

`data:alignment:bam:staralignment-star` (*data:reads:fastq* **reads**, *data:genomeindex:star* **genome**, *data:annotation* **annotation**, *basic:string* **exon_name**, *basic:integer* **sjdbOverhang**, *basic:boolean* **unstranded**, *basic:boolean* **noncannonical**, *basic:boolean* **chimeric**, *basic:integer* **chimSegmentMin**, *basic:boolean* **quantmode**, *basic:boolean* **singleend**, *basic:boolean* **gene_counts**, *basic:string* **outFilterType**, *basic:integer* **outFilterMultimapNmax**, *basic:integer* **outFilterMismatchNmax**, *basic:decimal* **outFilterMismatchNoverLmax**, *basic:integer* **outFilterScoreMin**, *basic:decimal* **outFilterMismatchNoverReadLmax**, *basic:integer* **alignSJoverhangMin**, *basic:integer* **alignSJDBoverhangMin**, *basic:integer* **alignIntronMin**, *basic:integer* **alignIntronMax**, *basic:integer* **alignMatesGapMax**, *basic:string* **alignEndsType**, *basic:boolean* **two_pass_mode**, *basic:string* **outSAMunmapped**, *basic:string* **outSAMattributes**, *basic:string* **outSAMattrRGline**, *basic:string* **tool_bigwig**, *basic:integer* **bin_size_bigwig**) [Source: v1.8.11]

Spliced Transcripts Alignment to a Reference (STAR) software is based on an alignment algorithm that uses sequential maximum mappable seed search in uncompressed suffix arrays followed by seed clustering and stitching procedure. In addition to unbiased de novo detection of canonical junctions, STAR can discover non-canonical splices and chimeric (fusion) transcripts, and is also capable of mapping full-length RNA sequences. More information can be found in the [STAR manual](http://labshare.cshl.edu/shares/gingeraslab/www-data/dobin/STAR/STAR.posix/doc/STARmanual.pdf) and in the [original paper](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3530905/).

Input arguments reads

label Reads

type `data:reads:fastq`

genome

label Indexed reference genome

type `data:genomeindex:star`

description Genome index prepared by STAR aligner indexing tool.

annotation

label Annotation file (GTF/GFF3)

type `data:annotation`

description Insert known annotations into genome indices at the mapping stage.

required False

annotation_options.exon_name

label –sjdbGTFfeatureExon

type `basic:string`

description Feature type in GTF file to be used as exons for building transcripts

default `exon`

annotation_options.sjdbOverhang

label Junction length (sjdbOverhang)

type `basic:integer`

description This parameter specifies the length of the genomic sequence around the annotated junction to be used in constructing the splice junction database. Ideally, this length should be equal to the ReadLength-1, where ReadLength is the length of the reads. For instance, for Illumina 2x100b paired-end reads, the ideal value is 100-1=99. In case of reads of varying length, the ideal value is max(ReadLength)-1. In most cases, the default value of 100 will work as well as the ideal value.

default `100`

unstranded

label The data is unstranded

type `basic:boolean`

description For unstranded RNA-seq data, Cufflinks/Cuffdiff require spliced alignments with XS strand attribute, which STAR will generate with –outSAMstrandField intronMotif option. As required, the XS strand attribute will be generated for all alignments that contain splice junctions. The spliced alignments that have undefined strand (i.e. containing only non-canonical unannotated junctions) will be suppressed. If you have stranded RNA-seq data, you do not need to use any specific STAR options. Instead, you need to run Cufflinks with the library option –library-type options. For example, cufflinks –library-type fr-firststrand should be used for the standard dUTP protocol, including Illumina's stranded Tru-Seq. This option has to be used only for Cufflinks runs and not for STAR runs.

default `False`

noncannonical

label Remove non-cannonical junctions (Cufflinks compatibility)

type `basic:boolean`

description It is recommended to remove the non-canonical junctions for Cufflinks runs using –outFilterIntronMotifs RemoveNoncanonical.

default `False`

detect_chimeric.chimeric

label Detect chimeric and circular alignments

type `basic:boolean`

description To switch on detection of chimeric (fusion) alignments (in addition to normal mapping), –chimSegmentMin should be set to a positive value. Each chimeric alignment consists of two "segments". Each segment is non-chimeric on its own, but the segments are chimeric to each other (i.e. the segments belong to different chromosomes, or different strands, or are far from each other). Both segments may contain splice junctions, and one of the segments may contain portions of both mates.

–chimSegmentMin parameter controls the minimum mapped length of the two segments that is allowed. For example, if you have 2x75 reads and used –chimSegmentMin 20, a chimeric alignment with 130b on one chromosome and 20b on the other will be output, while 135 + 15 won't be.

default `False`

detect_chimeric.chimSegmentMin

label –chimSegmentMin

type `basic:integer`

disabled detect_chimeric.chimeric != true

default `20`

t_coordinates.quantmode

label Output in transcript coordinates

type `basic:boolean`

description With –quantMode TranscriptomeSAM option STAR will output alignments translated into transcript coordinates in the Aligned.toTranscriptome.out.bam file (in addition to alignments in genomic coordinates in Aligned.*.sam/bam files). These transcriptomic alignments can be used with various transcript quantification software that require reads to be mapped to transcriptome, such as RSEM or eXpress.

default `False`

t_coordinates.singleend

label Allow soft-clipping and indels

type `basic:boolean`

description By default, the output satisfies RSEM requirements: soft-clipping or indels are not allowed. Use –quantTranscriptomeBan Singleend to allow insertions, deletions ans soft-clips in the transcriptomic alignments, which can be used by some expression quantification software (e.g. eXpress).

disabled t_coordinates.quantmode != true

default `False`

t_coordinates.gene_counts

label Count reads

type `basic:boolean`

description With –quantMode GeneCounts option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options: column 1: gene ID; column 2: counts for unstranded RNA-seq; column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes); column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse).

disabled t_coordinates.quantmode != true

default `False`

filtering.outFilterType

label Type of filtering

type `basic:string`

description Normal: standard filtering using only current alignment; BySJout: keep only those reads that contain junctions that passed filtering into SJ.out.tab

default `Normal`

choices

- Normal: `Normal`
- BySJout: `BySJout`

filtering.outFilterMultimapNmax

label –outFilterMultimapNmax

type `basic:integer`

description Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 10).

required False

filtering.outFilterMismatchNmax

label –outFilterMismatchNmax

type `basic:integer`

description Alignment will be output only if it has fewer mismatches than this value (default: 10).

required False

filtering.outFilterMismatchNoverLmax

label –outFilterMismatchNoverLmax

type `basic:decimal`

description Max number of mismatches per pair relative to read length: for 2x100b, max number of mismatches is 0.06*200=8 for the paired read.

required False

filtering.outFilterScoreMin

label –outFilterScoreMin

type `basic:integer`

description Alignment will be output only if its score is higher than or equal to this value (default: 0).

required False

filtering.outFilterMismatchNoverReadLmax

label –outFilterMismatchNoverReadLmax

type `basic:decimal`

description Alignment will be output only if its ratio of mismatches to *read* length is less than or equal to this value (default: 1.0).

required False

alignment.alignSJoverhangMin

label –alignSJoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for spliced alignments (default: 5).

required False

alignment.alignSJDBoverhangMin

label –alignSJDBoverhangMin

type `basic:integer`

description Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments (default: 3).

required False

alignment.alignIntronMin

label –alignIntronMin

type `basic:integer`

description Minimum intron size: genomic gap is considered intron if its length >= alignIntronMin, otherwise it is considered Deletion (default: 21).

required False

alignment.alignIntronMax

label –alignIntronMax

type `basic:integer`

description Maximum intron size, if 0, max intron size will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignMatesGapMax

label –alignMatesGapMax

type `basic:integer`

description Maximum gap between two mates, if 0, max intron gap will be determined by (2pow(winBinNbits)*winAnchorDistNbins) (default: 0).

required False

alignment.alignEndsType

label –alignEndsType

type `basic:string`

description Type of read ends alignment (default: Local).

required False

default `Local`

choices

- Local: `Local`
- EndToEnd: `EndToEnd`
- Extend5pOfRead1: `Extend5pOfRead1`
- Extend5pOfReads12: `Extend5pOfReads12`

two_pass_mapping.two_pass_mode

label –twopassMode

type `basic:boolean`

description Perform first-pass mapping, extract junctions, insert them into genome index, and re-map all reads in the second mapping pass.

default `False`

output_sam_bam.outSAMunmapped

label –outSAMunmapped

type `basic:string`

description Output of unmapped reads in the SAM format.

required False

default `None`

choices

- None: `None`
- Within: `Within`

output_sam_bam.outSAMattributes

label –outSAMattributes

type `basic:string`

description a string of desired SAM attributes, in the order desired for the output SAM.

required False

default `Standard`

choices

- None: `None`
- Standard: `Standard`
- All: `All`

output_sam_bam.outSAMattrRGline

label –outSAMattrRGline

type `basic:string`

description SAM/BAM read group line. The first word contains the read group identifier and must start with "ID:", e.g. –outSAMattrRGline ID:xxx CN:yy "DS:z z z"

required False

output_sam_bam.tool_bigwig

label Tool to calculate BigWig

type `basic:string`

description Tool to calculate BigWig.

default `deeptools`

choices

- deepTools: `deeptools`

- UCSC BedGraphToBigWig: `bedgraphtobigwig`

output_sam_bam.bin_size_bigwig

label Bin Size for the output of BigWig

type `basic:integer`

description Size of the bins, in bases, for the output of the bigwig. Only possible if 'Tool to calculate BigWig' is deepTools. If BigWig is calculated by UCSC BedGraphToBigWig then bin size is 1.

default `50`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped_f

label Unmapped reads (mate 1)

type `basic:file`

required False

unmapped_r

label Unmapped reads (mate 2)

type `basic:file`

required False

sj

label Splice junctions

type `basic:file`

chimeric

label Chimeric alignments

type `basic:file`

required False

alignment_transcriptome

label Alignment (trancriptome coordinates)

type `basic:file`

required False

gene_counts

label Gene counts

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

STAR genome index

`data:genomeindex:staralignment-star-index` (*data:genome:fasta* **genome**, *data:seq:nucleotide* **genome2**, *data:annotation* **annotation**, *basic:string* **exon_name**, *basic:integer* **sjdbOverhang**, *basic:integer* **genomeSAindexNbases**, *basic:integer* **genomeChrBinNbits**, *basic:integer* **genomeSAsparseD**) [Source: v1.5.4]

Generate genome indices files from the supplied reference genome sequence and GTF files.

Input arguments genome

label Reference genome (indexed)

type `data:genome:fasta`

required False

genome2

label Reference genome (nucleotide sequence)

type `data:seq:nucleotide`

required False

annotation

label Annotation file (GTF/GFF3)

type `data:annotation`

required False

annotation_options.exon_name

label –sjdbGTFfeatureExon

type `basic:string`

description Feature type in GTF file to be used as exons for building transcripts.

default `exon`

annotation_options.sjdbOverhang

label Junction length (sjdbOverhang)

type `basic:integer`

description This parameter specifies the length of the genomic sequence around the annotated junction to be used in constructing the splice junction database. Ideally, this length should be equal to the ReadLength-1, where ReadLength is the length of the reads. For instance, for Illumina 2x100b paired-end reads, the ideal value is 100-1=99. In case of reads of varying length, the ideal value is max(ReadLength)-1. In most cases, the default value of 100 will work as well as the ideal value.

default `100`

advanced.genomeSAindexNbases

label Small genome adjustment

type `basic:integer`

description For small genomes, the parameter –genomeSAindexNbases needs to be scaled down, with a typical value of min(14, log2(GenomeLength)/2 - 1). For example, for 1 megaBase genome, this is equal to 9, for 100 kiloBase genome, this is equal to 7.

required False

advanced.genomeChrBinNbits

label Large number of references adjustment

type `basic:integer`

description If you are using a genome with a large (>5,000) number of references (chrosomes/scaffolds), you may need to reduce the –genomeChrBinNbits to reduce RAM consumption. The following scaling is recommended: –genomeChrBinNbits = min(18, log2(GenomeLength / NumberOfReferences)). For example, for 3 gigaBase genome with 100,000 chromosomes/scaffolds, this is equal to 15.

required False

advanced.genomeSAsparseD

label Sufflux array sparsity

type `basic:integer`

description Suffux array sparsity, i.e. distance between indices: use bigger numbers to decrease needed RAM at the cost of mapping speed reduction (integer > 0, default = 1).

required False

Output results index

label Indexed genome

type `basic:dir`

source

label Gene ID source

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Salmon Index

`data:index:salmonsalmon-index` (*data:seq:nucleotide* **nucl**, *data:file* **decoys**, *basic:boolean* **gencode**, *basic:boolean* **keep_duplicates**, *basic:boolean* **perfect_hash**, *basic:string* **source**, *basic:string* **species**, *basic:string* **build**, *basic:integer* **kmerlen**) [Source: v1.0.0]

Generate index files for Salmon transcript quantification tool.

Input arguments nucl

label Nucleotide sequence

type `data:seq:nucleotide`

description A CDS sequence file in .FASTA format.

decoys

label Decoys

type `data:file`

description Treat these sequences as decoys that may have sequence homologous to some known transcript.

required False

gencode

label Gencode

type `basic:boolean`

description This flag will expect the input transcript FASTA to be in GENCODE format, and will split the transcript name at the first '|' character. These reduced names will be used in the output and when looking for these transcripts in a gene to transcript GTF.

default `False`

keep_duplicates

label Keep duplicates

type `basic:boolean`

description This flag will disable the default indexing behavior of discarding sequence-identical duplicate transcripts. If this flag is passed, then duplicate transcripts that appear in the input will be retained and quantified separately.

default `False`

perfect_hash

label Perfect hash

type `basic:boolean`

description Build the index using a perfect hash rather than a dense hash. This will require less memory (especially during quantification), but will take longer to construct.

default `False`

source

label Source of attribute ID

type `basic:string`

choices

- DICTYBASE: `DICTYBASE`
- ENSEMBL: `ENSEMBL`
- NCBI: `NCBI`
- UCSC: `UCSC`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`
- Dictyostelium discoideum: `Dictyostelium discoideum`

build

label Genome build

type `basic:string`

kmerlen

label Size of k-mers

type `basic:integer`

description The size of k-mers that should be used for the quasi index. We find that a k of 31 seems to work well for reads of 75bp or longer, but you might consider a smaller k if you plan to deal with shorter reads.

default `31`

Output results index

label Salmon index

type `basic:dir`

source

label Source of attribute ID

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Secondary hybrid BAM file

`data:alignment:bam:secondaryupload-bam-secondary` (*data:alignment:bam* **bam**, *basic:file* **src**, *basic:string* **species**, *basic:string* **build**) [Source: v0.5.0]

Upload a secondary mapping file in BAM format.

Input arguments bam

label Hybrid bam

type `data:alignment:bam`

description Secondary bam will be appended to the same sample where hybrid bam is.

required False

src

label Mapping (BAM)

type `basic:file`

description A mapping file in BAM format. The file will be indexed on upload, so additional BAI files are not required.

validate_regex `\.(bam)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Drosophila melanogaster: `Drosophila melanogaster`
- Mus musculus: `Mus musculus`

build

label Build

type `basic:string`

Output results bam

label Uploaded file

type `basic:file`

bai

label Index BAI

type `basic:file`

stats

label Alignment statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Spike-ins quality control

`data:spikeinsspikein-qc` (*list:data:expression* **samples**, *basic:string* **mix**) [Source: v0.0.3]

Plot spike-ins measured abundances for samples quality control. The process will output graphs showing the correlation between known concentration of ERCC spike-ins and sample's measured abundance.

Input arguments samples

label Expressions with spike-ins

type `list:data:expression`

mix

label Spike-ins mix

type `basic:string`

description Select spike-ins mix.

choices

- ERCC Mix 1: `ercc_mix1`
- ERCC Mix 2: `ercc_mix2`
- SIRV-Set 3: `sirv_set3`

Output results plots

label Plot figures

type `list:basic:file`

report

label HTML report with results

type `basic:file:html`

hidden True

report_zip

label ZIP file contining HTML report with results

type `basic:file`

Subread

`data:alignment:bam:subreadalignment-subread` (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:integer* **indel**, *basic:integer* **consensus**, *basic:integer* **mis_matched_bp**, *basic:integer* **cpu_number**, *basic:boolean* **multi_mapping**, *basic:string* **reads_orientation**, *basic:integer* **consensus_subreads**) [Source: v2.1.1]

Subread is an accurate and efficient general-purpose read aligner which can align both genomic DNA-seq and RNA-seq reads. It can also be used to discover genomic mutations including short indels and structural variants. See [here](http://subread.sourceforge.net/) and a paper by [Liao and colleagues](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3664803/) (2013) for more information.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

options.indel

label Number of INDEL bases

type `basic:integer`

description Specify the number of INDEL bases allowed in the mapping.

required False

default `5`

options.consensus

label Consensus threshold

type `basic:integer`

description Specify the consensus threshold, which is the minimal number of consensus subreads required for reporting a hit.

required False

default `3`

options.mis_matched_bp

label Max number of mis-matched bases

type `basic:integer`

description Specify the maximum number of mis-matched bases allowed in the alignment.

required False

default `3`

options.cpu_number

label Number of threads/CPUs

type `basic:integer`

description Specify the number of threads/CPUs used for mapping

required False

default `1`

options.multi_mapping

label Report multi-mapping reads in addition to uniquely mapped reads.

type `basic:boolean`

description Reads that were found to have more than one best mapping location are going to be reported.

required False

PE_options.reads_orientation

label reads orientation

type `basic:string`

description Specify the orientation of the two reads from the same pair.

required False

default `fr`

choices

- ff: `ff`
- fr: `fr`
- rf: `rf`

PE_options.consensus_subreads

label Minimum number of consensus subreads

type `basic:integer`

description Specify the minimum number of consensus subreads both reads from the sam pair must have.

required False

default `1`

Output results bam

label Alignment file

type `basic:file`

description Position sorted alignment

bai

label Index BAI

type `basic:file`

unmapped

label Unmapped reads

type `basic:file`

required False

stats

label Statistics

type `basic:file`

bigwig

label BigWig file

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Subsample FASTQ (paired-end)

`data:reads:fastq:paired:seqtkseqtk-sample-paired` (*data:reads:fastq:paired* **reads**, *basic:integer* **n_reads**, *basic:integer* **seed**, *basic:decimal* **fraction**, *basic:boolean* **two_pass**) [Source: v1.0.3]

[Seqtk](https://github.com/lh3/seqtk) is a fast and lightweight tool for processing sequences in the FASTA or FASTQ format. The Seqtk "sample" command enables subsampling of the large FASTQ file(s).

Input arguments reads

label Reads

type `data:reads:fastq:paired`

n_reads

label Number of reads

type `basic:integer`

default `1000000`

advanced.seed

label Seed

type `basic:integer`

default `11`

advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads [0 - 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the “Number of reads” input parameter.

required False

advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

Output results fastq

label Remaining mate 1 reads

type `list:basic:file`

fastq2

label Remaining mate 2 reads

type `list:basic:file`

fastqc_url

label Mate 1 quality control with FastQC

type `list:basic:file:html`

fastqc_url2

label Mate 2 quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download mate 1 FastQC archive

type `list:basic:file`

fastqc_archive2

label Download mate 2 FastQC archive

type `list:basic:file`

Subsample FASTQ (single-end)

data:reads:fastq:single:seqtkseqtk-sample-single (*data:reads:fastq:single* **reads**, *basic:integer* **n_reads**, *basic:integer* **seed**, *basic:decimal* **fraction**, *basic:boolean* **two_pass**) [Source: v1.0.3]

[Seqtk](https://github.com/lh3/seqtk) is a fast and lightweight tool for processing sequences in the FASTA or FASTQ format. The Seqtk "sample" command enables subsampling of the large FASTQ file(s).

Input arguments reads

label Reads

type `data:reads:fastq:single`

n_reads

label Number of reads

type `basic:integer`

default `1000000`

advanced.seed

label Seed

type `basic:integer`

default `11`

advanced.fraction

label Fraction

type `basic:decimal`

description Use the fraction of reads [0 - 1.0] from the orignal input file instead of the absolute number of reads. If set, this will override the "Number of reads" input parameter.

required False

advanced.two_pass

label 2-pass mode

type `basic:boolean`

description Enable two-pass mode when down-sampling. Two-pass mode is twice as slow but with much reduced memory.

default `False`

Output results fastq

label Remaining reads

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

Test basic fields

`data:test:fieldstest-basic-fields` (*basic:boolean* **boolean**, *basic:date* **date**, *basic:datetime* **datetime**, *basic:decimal* **decimal**, *basic:integer* **integer**, *basic:string* **string**, *basic:text* **text**, *basic:url:download* **url_download**, *basic:url:view* **url_view**, *basic:string* **string2**, *basic:string* **string3**, *basic:string* **string4**, *basic:string* **string5**, *basic:string* **string6**, *basic:string* **string7**, *basic:string* **tricky2**) [Source: v1.1.1]

Test with all basic input fields whose values are printed by the processor and returned unmodified as output fields.

Input arguments boolean

label Boolean

type `basic:boolean`

default `True`

date

label Date

type `basic:date`

default `2013-12-31`

datetime

label Date and time

type `basic:datetime`

default `2013-12-31 23:59:59`

decimal

label Decimal

type `basic:decimal`

default `-123.456`

integer

label Integer

type `basic:integer`

default `-123`

string

label String

type `basic:string`

default `Foo b-a-r.gz 1.23`

text

label Text

type `basic:text`

default `Foo bar in 3 lines.`

url_download

label URL download

type `basic:url:download`

default `{'url': 'http://www.w3.org/TR/1998/REC-html40-19980424/html40.pdf'}`

url_view

label URL view

type `basic:url:view`

default `{'name': 'Something', 'url': 'http://www.something.com/'}`

group.string2

label String 2 required

type `basic:string`

description String 2 description.

required True

disabled false

hidden false

placeholder `Enter string`

group.string3

label String 3 disabled

type `basic:string`

description String 3 description.

disabled true

default `disabled`

group.string4

label String 4 hidden

type `basic:string`

description String 4 description.

hidden True

default `hidden`

group.string5

label String 5 choices

type `basic:string`

description String 5 description.

hidden False

default `choice_2`

choices

- Choice 1: `choice_1`
- Choice 2: `choice_2`
- Choice 3: `choice_3`

group.string6

label String 6 regex only “Aa”

type `basic:string`

default `AAaAaaa`

validate_regex `^[aA]*$`

group.string7

label String 7 optional choices

type `basic:string`

description String 7 description.

required False

hidden False

default `choice_2`

choices

- Choice 1: `choice_1`
- Choice 2: `choice_2`
- Choice 3: `choice_3`

tricky.tricky1.tricky2

label Tricky 2

type `basic:string`

default `true`

Output results output

label Result

type `basic:url:view`

out_boolean

label Boolean

type `basic:boolean`

out_date

label Date

type `basic:date`

out_datetime

label Date and time

type `basic:datetime`

out_decimal

label Decimal

type `basic:decimal`

out_integer

label Integer

type `basic:integer`

out_string

label String

type `basic:string`

out_text

label Text

type `basic:text`

out_url_download

label URL download

type `basic:url:download`

out_url_view

label URL view

type `basic:url:view`

out_group.string2

label String 2 required

type `basic:string`

description String 2 description.

out_group.string3

label String 3 disabled

type `basic:string`

description String 3 description.

out_group.string4

label String 4 hidden

type `basic:string`

description String 4 description.

out_group.string5

label String 5 choices

type `basic:string`

description String 5 description.

out_group.string6

label String 6 regex only “Aa”

type `basic:string`

out_group.string7

label String 7 optional choices

type `basic:string`

out_tricky.tricky1.tricky2

label Tricky 2

type `basic:string`

Test disabled inputs

`data:test:disabledtest-disabled` (*basic:boolean* **broad**, *basic:integer* **broad_width**, *basic:string* **width_label**, *basic:integer* **if_and_condition**) [Source: v1.1.1]

Test disabled input fields.

Input arguments broad

label Broad peaks

type `basic:boolean`

default `False`

broad_width

label Width of peaks

type `basic:integer`

disabled broad === false

default `5`

width_label

label Width label

type `basic:string`

disabled broad === false

default `FD`

if_and_condition

label If width is 5 and label FDR

type `basic:integer`

disabled broad_width == 5 && width_label == ‘FDR’

default `5`

Output results output

label Result

type `basic:string`

Test hidden inputs

data:test:hiddentest-hidden (*basic:boolean* **broad**, *basic:integer* **broad_width**, *basic:integer* **parameter1**, *basic:integer* **parameter2**, *basic:integer* **broad_width2**) [Source: v1.1.1]

Test hidden input fields

Input arguments broad

label Broad peaks

type `basic:boolean`

default `False`

broad_width

label Width of peaks

type `basic:integer`

hidden broad === false

default `5`

parameters_broad_f.parameter1

label parameter1

type `basic:integer`

default `10`

parameters_broad_f.parameter2

label parameter2

type `basic:integer`

default `10`

parameters_broad_t.broad_width2

label Width of peaks2

type `basic:integer`

default `5`

Output results output

label Result

type `basic:string`

Test select controler

data:test:resulttest-list (*data:test:result* **single**, *list:data:test:result* **multiple**) [Source: v1.1.1]

Test with all basic input fields whose values are printed by the processor and returned unmodified as output fields.

Input arguments single

label Single

type `data:test:result`

multiple

label Multiple

type `list:data:test:result`

Output results output

label Result

type `basic:string`

Test sleep progress

`data:test:resulttest-sleep-progress` (*basic:integer* **t**) [Source: v1.1.1]

Test for the progress bar by sleeping 5 times for the specified amount of time.

Input arguments t

label Sleep time

type `basic:integer`

default `5`

Output results output

label Result

type `basic:string`

Trim, align and quantify using a library as a reference.

`data:workflow:trimalquantworkflow-trim-align-quant` (*data:reads:fastq:single* **reads**, *list:basic:string* **up_primers_seq**, *list:basic:string* **down_primers_seq**, *basic:decimal* **error_rate_5end**, *basic:decimal* **error_rate_3end**, *data:genome:fasta* **genome**, *basic:string* **mode**, *basic:integer* **N**, *basic:integer* **L**, *basic:integer* **gbar**, *basic:string* **mp**, *basic:string* **rdg**, *basic:string* **rfg**, *basic:string* **score_min**, *basic:integer* **readlengths**, *basic:integer* **alignscores**) [Source: v0.0.1]

Input arguments reads

label Untrimmed reads.

type `data:reads:fastq:single`

description First stage of shRNA pipeline. Trims 5’ adapters, then 3’ adapters using the same error rate setting, aligns reads to a reference library and quantifies species.

trimming_options.up_primers_seq

label 5’ adapter sequence

type `list:basic:string`

description A string of 5’ adapter sequence.

required True

trimming_options.down_primers_seq

label 3’ adapter sequence

type `list:basic:string`

description A string of 3’ adapter sequence.

required True

trimming_options.error_rate_5end

label Error rate for 5’

type `basic:decimal`

description Maximum allowed error rate (no. of errors divided by the length of the matching region) for 5’ trimming.

required False

default `0.1`

trimming_options.error_rate_3end

label Error rate for 3’

type `basic:decimal`

description Maximum allowed error rate (no. of errors divided by the length of the matching region) for 3’ trimming.

required False

default `0.1`

alignment_options.genome

label Reference library

type `data:genome:fasta`

description Choose the reference library against which to align reads.

alignment_options.mode

label Alignment mode

type `basic:string`

description End to end: Bowtie 2 requires that the entire read align from one end to the other, without any trimming (or “soft clipping”) of characters from either end. local: Bowtie 2 does not require that the entire read align from one end to the other. Rather, some characters may be omitted (“soft clipped”) from the ends in order to achieve the greatest possible alignment score.

default `--end-to-end`

choices

- end to end mode: `--end-to-end`
- local: `--local`

alignment_options.N

label Number of mismatches allowed in seed alignment (N)

type `basic:integer`

description Sets the number of mismatches to allowed in a seed alignment during multiseed alignment. Can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity. Default: 0.

required False

alignment_options.L

label Length of seed substrings (L)

type `basic:integer`

description Sets the length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more sensitive. Default: the –sensitive preset is used by default for end-to-end alignment and –sensitive-local for local alignment. See documentation for details.

required False

alignment_options.gbar

label Disallow gaps within positions (gbar)

type `basic:integer`

description Disallow gaps within <int> positions of the beginning or end of the read. Default: 4.

required False

alignment_options.mp

label Maximal and minimal mismatch penalty (mp)

type `basic:string`

description Sets the maximum (MX) and minimum (MN) mismatch penalties, both integers. A number less than or equal to MX and greater than or equal to MN is subtracted from the alignment score for each position where a read character aligns to a reference character, the characters do not match, and neither is an N. If –ignore-quals is specified, the number subtracted quals MX. Otherwise, the number subtracted is MN + floor((MX-MN)(MIN(Q, 40.0)/40.0)) where Q is the Phred quality value. Default for MX, MN: 6,2.

required False

alignment_options.rdg

label Set read gap open and extend penalties (rdg)

type `basic:string`

description Sets the read gap open (<int1>) and extend (<int2>) penalties. A read gap of length N gets a penalty of <int1> + N * <int2>. Default: 5,3.

required False

alignment_options.rfg

label Set reference gap open and close penalties (rfg)

type `basic:string`

description Sets the reference gap open (<int1>) and extend (<int2>) penalties. A reference gap of length N gets a penalty of <int1> + N * <int2>. Default: 5,3.

required False

alignment_options.score_min

label Minimum alignment score needed for "valid" alignment (score-min)

type `basic:string`

description Sets a function governing the minimum alignment score needed for an alignment to be considered "valid" (i.e. good enough to report). This is a function of read length. For instance, specifying L,0,-0.6 sets the minimum-score function to f(x) = 0 + -0.6 * x, where x is the read length. The default in –end-to-end mode is L,-0.6,-0.6 and the default in –local mode is G,20,8.

required False

quant_options.readlengths

label Species lengths threshold

type `basic:integer`

description Species with read lengths below specified threshold will be removed from final output. Default is no removal.

quant_options.alignscores

label Align scores filter threshold

type `basic:integer`

description Species with align score below specified threshold will be removed from final output. Default is no removal.

Output results

Trimmomatic (paired-end)

`data:reads:fastq:paired:trimmomatictrimmomatic-paired` (*data:reads:fastq:paired* **reads**, *data:seq:nucleotide* **adapters**, *basic:integer* **seed_mismatches**, *basic:integer* **simple_clip_threshold**, *basic:integer* **palindrome_clip_threshold**, *basic:integer* **min_adapter_length**, *basic:boolean* **keep_both_reads**, *basic:integer* **window_size**, *basic:integer* **required_quality**, *basic:integer* **target_length**, *basic:decimal* **strictness**, *basic:integer* **leading**, *basic:integer* **trailing**, *basic:integer* **crop**, *basic:integer* **headcrop**, *basic:integer* **minlen**, *basic:integer* **average_quality**) [Source: v2.1.2]

Trimmomatic performs a variety of useful trimming tasks including removing adapters for Illumina paired-end and single-end data. FastQC is performed for quality control checks on trimmed raw sequence data, which are the output of Trimmomatic. See [Trimmomatic official website](http://www.usadellab.org/cms/?page=trimmomatic), the [introductory paper](https://www.ncbi.nlm.nih.gov/pubmed/24695404), and the [FastQC official website](https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) for more information.

Input arguments reads

label Reads

type `data:reads:fastq:paired`

illuminaclip.adapters

label Adapter sequences

type `data:seq:nucleotide`

description Adapter sequence in FASTA format that will be removed from the read. This field as well as 'Seed mismatches', 'Simple clip threshold' and 'Palindrome clip threshold' parameters are needed to perform Illuminacliping. 'Minimum adapter length' and 'Keep both reads' are optional parameters.

required False

illuminaclip.seed_mismatches

label Seed mismatches

type `basic:integer`

description Specifies the maximum mismatch count which will still allow a full match to be performed. This field as well as 'Adapter sequence', 'Simple clip threshold' and 'Palindrome clip threshold' parameters are needed to perform Illuminacliping.

required False

disabled !illuminaclip.adapters

illuminaclip.simple_clip_threshold

label Simple clip threshold

type `basic:integer`

description Specifies how accurate the match between any adapter etc. sequence must be against a read. This field as well as 'Adapter sequence', 'Seed mismatches' and 'Palindrome clip threshold' parameters are needed to perform Illuminacliping.

required False

disabled !illuminaclip.adapters

illuminaclip.palindrome_clip_threshold

label Palindrome clip threshold

type `basic:integer`

description Specifies how accurate the match between the two 'adapter ligated' reads must be for PE palindrome read alignment. This field as well as 'Adapter sequence', 'Simple clip threshold' and 'Seed mismatches' parameters are needed to perform Illuminacliping.

required False

disabled !illuminaclip.adapters

illuminaclip.min_adapter_length

label Minimum adapter length

type `basic:integer`

description In addition to the alignment score, palindrome mode can verify that a minimum length of adapter has been detected. If unspecified, this defaults to 8 bases, for historical reasons. However, since palindrome mode has a very low false positive rate, this can be safely reduced, even down to 1, to allow shorter adapter fragments to be removed. This field is optional for preforming Illuminaclip. 'Adapter sequences', 'Seed mismatches', 'Simple clip threshold' and 'Palindrome clip threshold' are also needed in order to use this parameter.

disabled !illuminaclip.seed_mismatches && !illuminaclip.simple_clip_threshold && !illuminaclip.palindrome_clip_threshold

default `8`

illuminaclip.keep_both_reads

label Keep both reads

type `basic:boolean`

description After read-though has been detected by palindrome mode, and the adapter sequence removed, the reverse read contains the same sequence information as the forward read, albeit in reverse complement. For this reason, the default behaviour is to entirely drop the reverse read.By specifying this parameter, the reverse read will also be retained, which may be useful e.g. if the downstream tools cannot handle a combination of paired and unpaired reads. This field is optional for preforming Illuminaclip. 'Adapter sequence', 'Seed mismatches', 'Simple clip threshold', 'Palindrome clip threshold' and also 'Minimum adapter length' are needed in order to use this parameter.

required False

disabled !illuminaclip.seed_mismatches && !illuminaclip.simple_clip_threshold && !illuminaclip.palindrome_clip_threshold && !illuminaclip.min_adapter_length

slidingwindow.window_size

label Window size

type `basic:integer`

description Specifies the number of bases to average across. This field as well as 'Required quality' are needed to perform a 'Sliding window' trimming (cutting once the average quality within the window falls below a threshold).

required False

slidingwindow.required_quality

label Required quality

type `basic:integer`

description Specifies the average quality required. This field as well as 'Window size' are needed to perform a 'Sliding window' trimming (cutting once the average quality within the window falls below a threshold).

required False

maxinfo.target_length

label Target length

type `basic:integer`

description This specifies the read length which is likely to allow the location of the read within the target sequence to be determined. This field as well as 'Strictness' are needed to perform 'Maxinfo' feature (an adaptive quality trimmer which balances read length and error rate to maximise the value of each read).

required False

maxinfo.strictness

label Strictness

type `basic:decimal`

description This value, which should be set between 0 and 1, specifies the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (<0.2) favours longer reads, while a high value (>0.8) favours read correctness. This field as well as 'Target length' are needed to perform 'Maxinfo' feature (an adaptive quality trimmer which balances read length and error rate to maximise the value of each read).

required False

trim_bases.leading

label Leading quality

type `basic:integer`

description Remove low quality bases from the beginning. Specifies the minimum quality required to keep a base.

required False

trim_bases.trailing

label Trailing

type `basic:integer`

description Remove low quality bases from the end. Specifies the minimum quality required to keep a base.

required False

trim_bases.crop

label Crop

type `basic:integer`

description Cut the read to a specified length by removing bases from the end.

required False

trim_bases.headcrop

label Headcrop

type `basic:integer`

description Cut the specified number of bases from the start of the read.

required False

reads_filtering.minlen

label Minimum length

type `basic:integer`

description Drop the read if it is below a specified length.

required False

reads_filtering.average_quality

label Average quality

type `basic:integer`

description Drop the read if the average quality is below the specified level.

required False

Output results fastq

label Reads file (mate 1)

type `list:basic:file`

fastq_unpaired

label Reads file

type `basic:file`

required False

fastq2

label Reads file (mate 2)

type `list:basic:file`

fastq2_unpaired

label Reads file

type `basic:file`

required False

fastqc_url

label Quality control with FastQC (Upstream)

type `list:basic:file:html`

fastqc_url2

label Quality control with FastQC (Downstream)

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive (Upstream)

type `list:basic:file`

fastqc_archive2

label Download FastQC archive (Downstream)

type `list:basic:file`

Trimmomatic (single-end)

data:reads:fastq:single:trimmomatictrimmomatic-single (*data:reads:fastq:single* **reads**, *data:seq:nucleotide* **adapters**, *basic:integer* **seed_mismatches**, *basic:integer* **simple_clip_threshold**, *basic:integer* **window_size**, *basic:integer* **required_quality**, *basic:integer* **target_length**, *basic:decimal* **strictness**, *basic:integer* **leading**, *basic:integer* **trailing**, *basic:integer* **crop**, *basic:integer* **headcrop**, *basic:integer* **minlen**, *basic:integer* **average_quality**) [Source: v2.1.2]

Trimmomatic performs a variety of useful trimming tasks including removing adapters for Illumina paired-end and single-end data. FastQC is performed for quality control checks on trimmed raw sequence data, which are the output of Trimmomatic. See [Trimmomatic official website](http://www.usadellab.org/cms/?page=trimmomatic), the [introductory paper](https://www.ncbi.nlm.nih.gov/pubmed/24695404), and the [FastQC official website](https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) for more information.

Input arguments reads

label Reads

type `data:reads:fastq:single`

illuminaclip.adapters

label Adapter sequences

type `data:seq:nucleotide`

description Adapter sequence in FASTA format that will be removed from the read. This field as well as 'Seed mismatches' and 'Simple clip threshold' parameters are needed to perform Illuminacliping.

required False

illuminaclip.seed_mismatches

label Seed mismatches

type `basic:integer`

description Specifies the maximum mismatch count which will still allow a full match to be performed. This field as well as 'Adapter sequences' and 'Simple clip threshold' parameter are needed to perform Illuminacliping.

required False

disabled !illuminaclip.adapters

illuminaclip.simple_clip_threshold

label Simple clip threshold

type `basic:integer`

description Specifies how accurate the match between any adapter etc. sequence must be against a read. This field as well as 'Adapter sequences' and 'Seed mismatches' parameter are needed to perform Illuminacliping.

required False

disabled !illuminaclip.adapters

slidingwindow.window_size

label Window size

type `basic:integer`

description Specifies the number of bases to average across. This field as well as 'Required quality' are needed to perform a 'Sliding window' trimming (cutting once the average quality within the window falls below a threshold).

required False

slidingwindow.required_quality

label Required quality

type `basic:integer`

description Specifies the average quality required in window size. This field as well as 'Window size' are needed to perform a 'Sliding window' trimming (cutting once the average quality within the window falls below a threshold).

required False

maxinfo.target_length

label Target length

type `basic:integer`

description This specifies the read length which is likely to allow the location of the read within the target sequence to be determined. This field as well as 'Strictness' are needed to perform 'Maxinfo' feature (an adaptive quality trimmer which balances read length and error rate to maximise the value of each read).

required False

maxinfo.strictness

label Strictness

type `basic:decimal`

description This value, which should be set between 0 and 1, specifies the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (<0.2) favours longer reads, while a high value (>0.8) favours read correctness. This field as well as 'Target length' are needed to perform 'Maxinfo' feature (an adaptive quality trimmer which balances read length and error rate to maximise the value of each read).

required False

trim_bases.leading

label Leading quality

type `basic:integer`

description Remove low quality bases from the beginning, if below a threshold quality.

required False

trim_bases.trailing

label Trailing quality

type `basic:integer`

description Remove low quality bases from the end, if below a threshold quality.

required False

trim_bases.crop

label Crop

type `basic:integer`

description Cut the read to a specified length by removing bases from the end.

required False

trim_bases.headcrop

label Headcrop

type `basic:integer`

description Cut the specified number of bases from the start of the read.

required False

reads_filtering.minlen

label Minimum length

type `basic:integer`

description Drop the read if it is below a specified length.

required False

reads_filtering.average_quality

label Average quality

type `basic:integer`

description Drop the read if the average quality is below the specified level.

required False

Output results fastq

label Reads file

type `list:basic:file`

fastqc_url

label Quality control with FastQC

type `list:basic:file:html`

fastqc_archive

label Download FastQC archive

type `list:basic:file`

Trimmomatic - HISAT2 - HTSeq-count (paired-end)

data:workflow:rnaseq:htseqworkflow-rnaseq-paired (*data:reads:fastq:paired* **reads**, *data:genome:fasta* **genome**, *data:annotation:gtf* **annotation**, *data:seq:nucleotide* **adapters**, *basic:integer* **seed_mismatches**, *basic:integer* **palindrome_clip_threshold**, *basic:integer* **simple_clip_threshold**, *basic:integer* **minlen**, *basic:integer* **trailing**, *basic:string* **stranded**, *basic:string* **id_attribute**) [Source: v1.0.1]

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __Trimmomatic__ which performs a variety of useful trimming tasks including removing adapters for Illumina paired-end and single-end high-throughput sequencing reads. Next, preprocessed reads are aligned by __HISAT2__ aligner. HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label Input reads

type `data:reads:fastq:paired`

genome

label Genome

type `data:genome:fasta`

annotation

label Annotation (GTF)

type `data:annotation:gtf`

adapters

label Adapter sequences (FASTA)

type `data:seq:nucleotide`

required False

illuminaclip.seed_mismatches

label Seed mismatches

type `basic:integer`

description Specifies the maximum mismatch count which will still allow a full match to be performed.

default `2`

illuminaclip.palindrome_clip_threshold

label Palindrome clip threshold

type `basic:integer`

description Specifies how accurate the match between the two 'adapter ligated' reads must be for PE palindrome read alignment.

default `30`

illuminaclip.simple_clip_threshold

label Simple clip threshold

type `basic:integer`

description Specifies how accurate the match between any adapter etc. sequence must be against a read.

default `10`

minlen

label Min length

type `basic:integer`

description Drop the read if it is below a specified length.

default `10`

trailing

label Trailing quality

type `basic:integer`

description Remove low quality bases from the end. Specifies the minimum quality required to keep a base.

default `28`

stranded

label Is data from a strand specific assay?

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GFF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

Output results

Trimmomatic - HISAT2 - HTSeq-count (single-end)

`data:workflow:rnaseq:htseqworkflow-rnaseq-single` (*data:reads:fastq:single* **reads**, *data:genome:fasta* **genome**, *data:annotation:gtf* **annotation**, *data:seq:nucleotide* **adapters**, *basic:integer* **seed_mismatches**, *basic:integer* **simple_clip_threshold**, *basic:integer* **minlen**, *basic:integer* **trailing**, *basic:string* **stranded**, *basic:string* **id_attribute**) [Source: v1.0.1]

This RNA-seq pipeline is comprised of three steps, preprocessing, alignment, and quantification.

First, reads are preprocessed by __Trimmomatic__ which performs a variety of useful trimming tasks including removing adapters for Illumina paired-end and single-end high-throughput sequencing reads. Next, preprocessed reads are aligned by __HISAT2__ aligner. HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads For more information see [this comparison of RNA-seq aligners](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/). Finally, aligned reads are summarized to genes by __HTSeq-count__. Compared to featureCounts, HTSeq-count is not as computationally efficient. All three tools in this workflow support parallelization to accelerate the analysis.

Input arguments reads

label Input reads

type `data:reads:fastq:single`

genome

label Genome

type `data:genome:fasta`

annotation

label Annotation (GTF)

type `data:annotation:gtf`

adapters

label Adapter sequences (FASTA)

type `data:seq:nucleotide`

required False

illuminaclip.seed_mismatches

label Seed mismatches

type `basic:integer`

description Specifies the maximum mismatch count which will still allow a full match to be performed.

default `2`

illuminaclip.simple_clip_threshold

label Simple clip threshold

type `basic:integer`

description Specifies how accurate the match between any adapter etc. sequence must be against a read.

default `10`

minlen

label Minimum length

type `basic:integer`

description Drop the read if it is below a specified length.

default `10`

trailing

label Trailing quality

type `basic:integer`

description Remove low quality bases from the end. Specifies the minimum quality required to keep a base.

default `28`

stranded

label Is data from a strand specific assay?

type `basic:string`

description In strand non-specific assay a read is considered overlapping with a feature regardless of whether it is mapped to the same or the opposite strand as the feature. In strand-specific forward assay and single reads, the read has to be mapped to the same strand as the feature. For paired-end reads, the first read has to be on the same strand and the second read on the opposite strand. In strand-specific reverse assay these rules are reversed.

default `no`

choices

- Strand non-specific: `no`
- Strand-specific forward: `yes`
- Strand-specific reverse: `reverse`

id_attribute

label ID attribute

type `basic:string`

description GFF attribute to be used as feature ID. Several GFF lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identity the counts in the output table.

default `gene_id`

Output results

Upload Picard CollectTargetedPcrMetrics

`data:picard:coverage:uploadupload-picard-pcrmetrics` (*basic:file* **target_pcr_metrics**, *basic:file* **target_coverage**) [Source: v1.1.1]

Upload Picard CollectTargetedPcrMetrics result files.

Input arguments target_pcr_metrics

label Target PCR metrics

type `basic:file`

target_coverage

label Target coverage

type `basic:file`

Output results target_pcr_metrics

label Target PCR metrics

type `basic:file`

target_coverage

label Target coverage

type `basic:file`

VCF file

`data:variants:vcfupload-variants-vcf` (*basic:file* **src**, *basic:string* **species**, *basic:string* **build**) [Source: v2.1.1]

Upload variants in VCF format.

Input arguments src

label Variants (VCF)

type `basic:file`

description Variants in VCF format.

required True

validate_regex `\.(vcf)(|\.gz|\.bz2|\.tgz|\.tar\.gz|\.tar\.bz2|\.zip|\.rar|\.7z)$`

species

label Species

type `basic:string`

description Species latin name.

choices

- Homo sapiens: `Homo sapiens`
- Mus musculus: `Mus musculus`
- Rattus norvegicus: `Rattus norvegicus`

- Dictyostelium discoideum: `Dictyostelium discoideum`
- Odocoileus virginianus texanus: `Odocoileus virginianus texanus`
- Solanum tuberosum: `Solanum tuberosum`

build

label Genome build

type `basic:string`

Output results vcf

label Uploaded file

type `basic:file`

tbi

label Tabix index

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Variant calling (CheMut)

`data:variants:vcf:chemutvc-chemut` (*data:genome:fasta* **genome**, *list:data:alignment:bam* **parental_strains**, *list:data:alignment:bam* **mutant_strains**, *basic:boolean* **br_and_ind_ra**, *basic:boolean* **dbsnp**, *data:variants:vcf* **known_sites**, *list:data:variants:vcf* **known_indels**, *basic:string* **PL**, *basic:string* **LB**, *basic:string* **PU**, *basic:string* **CN**, *basic:date* **DT**, *basic:integer* **stand_emit_conf**, *basic:integer* **stand_call_conf**, *basic:integer* **ploidy**, *basic:string* **glm**, *list:basic:string* **intervals**, *basic:boolean* **rf**) [Source: v1.2.2]

“CheMut varint calling using multiple BAM input files. Note: Usage of Genome Analysis Toolkit requires a licence.”

Input arguments genome

label Reference genome

type `data:genome:fasta`

parental_strains

label Parental strains

type `list:data:alignment:bam`

mutant_strains

label Mutant strains

type `list:data:alignment:bam`

br_and_ind_ra

label Do variant base recalibration and indel realignment

type `basic:boolean`

default `False`

dbsnp

label Use dbSNP file

type `basic:boolean`

description rsIDs from this file are used to populate the ID column of the output. Also, the DB INFO flag will be set when appropriate. dbSNP is not used in any way for the calculations themselves.

default `False`

known_sites

label Known sites (dbSNP)

type `data:variants:vcf`

required False

hidden br_and_ind_ra === false && dbsnp === false

known_indels

label Known indels

type `list:data:variants:vcf`

required False

hidden br_and_ind_ra === false

reads_info.PL

label Platform/technology

type `basic:string`

description Platform/technology used to produce the reads.

default `Illumina`

choices

- Capillary: `Capillary`
- Ls454: `Ls454`
- Illumina: `Illumina`
- SOLiD: `SOLiD`
- Helicos: `Helicos`
- IonTorrent: `IonTorrent`
- Pacbio: `Pacbio`

reads_info.LB

label Library

type `basic:string`

default `x`

reads_info.PU

label Platform unit

type `basic:string`

description Platform unit (e.g. flowcell-barcode.lane for Illumina or slide for SOLiD). Unique identifier.

default `x`

reads_info.CN

label Sequencing center

type `basic:string`

description Name of sequencing center producing the read.

default `x`

reads_info.DT

label Date

type `basic:date`

description Date the run was produced.

default `2017-01-01`

Varc_param.stand_emit_conf

label Emission confidence threshold

type `basic:integer`

description The minimum confidence threshold (phred-scaled) at which the program should emit sites that appear to be possibly variant.

default `10`

Varc_param.stand_call_conf

label Calling confidence threshold

type `basic:integer`

description The minimum confidence threshold (phred-scaled) at which the program should emit variant sites as called. If a site's associated genotype has a confidence score lower than the calling threshold, the program will emit the site as filtered and will annotate it as LowQual. This threshold separates high confidence calls from low confidence calls.

default `30`

Varc_param.ploidy

label Sample ploidy

type `basic:integer`

description Ploidy (number of chromosomes) per sample. For pooled data, set to (Number of samples in each pool * Sample Ploidy).

default `2`

Varc_param.glm

label Genotype likelihoods model

type `basic:string`

description Genotype likelihoods calculation model to employ – SNP is the default option, while INDEL is also available for calling indels and BOTH is available for calling both together.

default `SNP`

choices

- SNP: `SNP`
- INDEL: `INDEL`
- BOTH: `BOTH`

Varc_param.intervals

label Intervals

type `list:basic:string`

description Use this option to perform the analysis over only part of the genome. This argument can be specified multiple times. You can use samtools-style intervals (e.g. -L chr1 or -L chr1:100-200).

required False

Varc_param.rf

label ReasignOneMappingQuality Filter

type `basic:boolean`

description This read transformer will change a certain read mapping quality to a different value without affecting reads that have other mapping qualities. This is intended primarily for users of RNA-Seq data handling programs such as TopHat, which use MAPQ = 255 to designate uniquely aligned reads. According to convention, 255 normally designates "unknown" quality, and most GATK tools automatically ignore such reads. By reassigning a different mapping quality to those specific reads, users of TopHat and other tools can circumvent this problem without affecting the rest of their dataset.

default `False`

Output results vcf

label Called variants file

type `basic:file`

tbi

label Tabix index

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

Variant filtering (CheMut)

data:variants:vcf:filteringfiltering-chemut (*data:variants:vcf* **variants**, *basic:string* **analysis_type**, *basic:string* **parental_strain**, *basic:string* **mutant_strain**, *basic:integer* **read_depth**) [Source: v1.3.1]

Filtering and annotation of Variant Calling data - Chemical mutagenesis in Dictyostelium discoideum

Input arguments variants

label Variants file (VCF)

type `data:variants:vcf`

analysis_type

label Analysis type

type `basic:string`

description Choice of the analysis type. Use "SNV" or "INDEL" options for the analysis of haploid VCF files prepared by using GATK UnifiedGenotyper -glm option "SNP" or "INDEL", respectively. Choose options SNV_CHR2 or INDEL_CHR2 to run the GATK analysis only on the diploid portion of CHR2 (-ploidy 2 -L chr2:2263132-3015703).

default `snv`

choices

- SNV: `snv`
- INDEL: `indel`
- SNV_CHR2: `snv_chr2`
- INDEL_CHR2: `indel_chr2`

parental_strain

label Parental Strain Prefix

type `basic:string`

default `parental`

mutant_strain

label Mutant Strain Prefix

type `basic:string`

default `mut`

read_depth

label Read Depth Cutoff

type `basic:integer`

default `5`

Output results summary

label Summary

type `basic:file`

description Summarize the input parameters and results.

vcf

label Variants

type `basic:file`

description A genome VCF file of variants that passed the filters.

tbi

label Tabix index

type `basic:file`

variants_filtered

label Variants filtered

type `basic:file`

description A data frame of variants that passed the filters.

required False

variants_filtered_alt

label Variants filtered (multiple alt. alleles)

type `basic:file`

description A data frame of variants that contain more than two alternative alleles. These vairants are likely to be false positives.

required False

gene_list_all

label Gene list (all)

type `basic:file`

description Genes that are mutated at least once.

required False

gene_list_top

label Gene list (top)

type `basic:file`

description Genes that are mutated at least twice.

required False

mut_chr

label Mutations (by chr)

type `basic:file`

description List mutations in individual chromosomes.

required False

mut_strain

label Mutations (by strain)

type `basic:file`

description List mutations in individual strains.

required False

strain_by_gene

label Strain (by gene)

type `basic:file`

description List mutants that carry mutations in individual genes.

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

WALT

`data:alignment:mr:waltwalt` (*data:genome:fasta* **genome**, *data:reads:fastq* **reads**, *basic:boolean* **rm_dup**, *basic:integer* **mismatch**, *basic:integer* **number**) [Source: v1.0.2]

WALT (Wildcard ALignment Tool) is a read mapping program for bisulfite sequencing in DNA methylation studies.

Input arguments genome

label Reference genome

type `data:genome:fasta`

reads

label Reads

type `data:reads:fastq`

rm_dup

label Remove duplicates

type `basic:boolean`

default `True`

mismatch

label Maximum allowed mismatches

type `basic:integer`

required False

number

label Number of reads to map in one loop

type `basic:integer`

description Sets the number of reads to mapping in each loop. Larger number results in program taking more memory. This is especially evident in paired-end mapping.

required False

Output results mr

label Alignment file

type `basic:file`

description Position sorted alignment

stats

label Statistics

type `basic:file`

unmapped_f

label Unmapped reads (mate 1)

type `basic:file`

required False

unmapped_r

label Unmapped reads (mate 2)

type `basic:file`

required False

species

label Species

type `basic:string`

build

label Build

type `basic:string`

WGBS

`data:workflow:wgbsworkflow-wgbs` (*data:reads:fastq* **reads**, *data:genome:fasta* **genome**, *basic:boolean* **rm_dup**, *basic:integer* **mismatch**, *basic:integer* **number**, *basic:boolean* **cpgs**, *basic:boolean* **symmetric_cpgs**) [Source: v1.0.2]

This WGBS pipeline is comprised of three steps - alignment, computation of methylation levels, and identification of hypo-methylated regions (HMRs).

First, reads are aligned by __WALT__ aligner. [WALT (Wildcard ALignment Tool)](https://github.com/smithlabcode/walt) is fast and accurate read mapping for bisulfite sequencing. Then, methylation level at each genomic cytosine is calculated using __methcounts__. Finally, hypo-methylated regions are identified using __hmr__. Both methcounts and hmr are part of [MethPipe](http://smithlabresearch.org/software/methpipe/) package.

Input arguments reads

label Select sample(s)

type `data:reads:fastq`

genome

label Genome

type `data:genome:fasta`

alignment.rm_dup

label Remove duplicates

type `basic:boolean`

default `True`

alignment.mismatch

label Maximum allowed mismatches

type `basic:integer`

default `6`

alignment.number

label Number of reads to map in one loop

type `basic:integer`

description Sets the number of reads to mapping in each loop. Larger number results in program taking more memory. This is especially evident in paired-end mapping.

required False

methcounts.cpgs

label Only CpG context sites

type `basic:boolean`

description Output file will contain methylation data for CpG context sites only. Choosing this option will result in CpG content report only.

disabled methcounts.symmetric_cpgs

default `False`

methcounts.symmetric_cpgs

label Merge CpG pairs

type `basic:boolean`

description Merging CpG pairs results in symmetric methylation levels. Methylation is usually symmetric (cytosines at CpG sites were methylated on both DNA strands). Choosing this option will only keep the CpG sites data.

disabled methcounts.cpgs

default `True`

Output results

Whole exome sequencing (WES) analysis

data:workflow:wesworkflow-wes (*data:reads:fastq:paired* **reads**, *data:genome:fasta* **genome**, *data:bedpe* **bamclipper_bedpe**, *list:data:variants:vcf* **known_sites**, *data:bed* **intervals**, *data:variants:vcf* **hc_dbsnp**, *basic:string* **validation_stringency**, *data:seq:nucleotide* **adapters**, *basic:integer* **seed_mismatches**, *basic:integer* **simple_clip_threshold**, *basic:integer* **min_adapter_length**, *basic:integer* **palindrome_clip_threshold**, *basic:integer* **leading**, *basic:integer* **trailing**, *basic:integer* **minlen**, *basic:integer* **seed_l**, *basic:integer* **band_w**, *basic:boolean* **m**, *basic:decimal* **re_seeding**, *basic:integer* **match**, *basic:integer* **mismatch**, *basic:integer* **gap_o**, *basic:integer* **gap_e**, *basic:integer* **clipping**, *basic:integer* **unpaired_p**, *basic:integer* **report_tr**, *basic:boolean* **md_skip**, *basic:boolean* **md_remove_duplicates**, *basic:string* **md_assume_sort_order**, *basic:string* **read_group**, *basic:integer* **stand_call_conf**, *basic:integer* **mbq**) [Source: v2.0.0]

Whole exome sequencing pipeline analyzes Illumina panel data. It consists of trimming, aligning, soft clipping, (optional) marking of duplicates, recalibration of base quality scores and finally, calling of variants.

The tools used are Trimmomatic which performs trimming. Aligning is performed using BWA (mem). Soft clipping of Illumina primer sequences is done using bamclipper tool. Marking of duplicates (MarkDuplicates), recalibration of base quality scores (ApplyBQSR) and calling of variants (HaplotypeCaller) is done using GATK4 bundle of bioinformatics tools.

To successfully run this pipeline, you will need a genome (FASTA), paired-end (FASTQ) files, BEDPE file for bamclipper, known sites of variation (dbSNP) (VCF), dbSNP database of variations (can be the same as known sites of variation), intervals on which target capture was done (BED) and illumina adapter sequences (FASTA). Make sure that specified resources match the genome used in the alignment step.

Result is a file of called variants (VCF).

Input arguments reads

label Raw untrimmed reads

type `data:reads:fastq:paired`

description Raw paired-end reads.

required True

genome

label Reference genome

type `data:genome:fasta`

description Against which genome to align. Further processes depend on this genome (e.g. BQSR step).

required True

bamclipper_bedpe

label BEDPE file used for clipping using Bamclipper

type `data:bedpe`

description BEDPE file used for clipping using Bamclipper tool.

required True

known_sites

label Known sites of variation used in BQSR

type `list:data:variants:vcf`

description Known sites of variation as a VCF file.

required True

intervals

label Intervals

type `data:bed`

description Use intervals to narrow the analysis to defined regions. This usually help cutting down on process time.

required True

hc_dbsnp

label dbSNP for GATK4's HaplotypeCaller

type `data:variants:vcf`

description dbSNP database of variants for variant calling.

required True

validation_stringency

label Validation stringency for all SAM files read by this program. Setting stringency to SILENT can improve performance when processing a BAM file in which variable-length data (read, qualities, tags) do not otherwise need to be decoded. Default is STRICT. This setting is used in BaseRecalibrator and ApplyBQSR processes.

type `basic:string`

default `STRICT`

choices

- STRICT: `STRICT`
- SILENT: `SILENT`
- LENIENT: `LENIENT`

advanced.trimming.adapters

label Adapter sequences

type `data:seq:nucleotide`

description Adapter sequence in FASTA format that will be removed from the read. This field as well as 'Seed mismatches', 'Simple clip threshold' and 'Palindrome clip threshold' parameters are needed to perform Illuminacliping. 'Minimum adapter length' and 'Keep both reads' are optional parameters.

required False

advanced.trimming.seed_mismatches

label Seed mismatches

type `basic:integer`

description Specifies the maximum mismatch count which will still allow a full match to be performed. This field as well as ‘Adapter sequence’, ‘Simple clip threshold’ and ‘Palindrome clip threshold’ parameters are needed to perform Illuminacliping.

required False

disabled !advanced.trimming.adapters

advanced.trimming.simple_clip_threshold

label Simple clip threshold

type `basic:integer`

description Specifies how accurate the match between any adapter etc. sequence must be against a read. This field as well as ‘Adapter sequences’ and ‘Seed mismatches’ parameter are needed to perform Illuminacliping.

required False

disabled !advanced.trimming.adapters

advanced.trimming.min_adapter_length

label Minimum adapter length

type `basic:integer`

description In addition to the alignment score, palindrome mode can verify that a minimum length of adapter has been detected. If unspecified, this defaults to 8 bases, for historical reasons. However, since palindrome mode has a very low false positive rate, this can be safely reduced, even down to 1, to allow shorter adapter fragments to be removed. This field is optional for preforming Illuminaclip. ‘Adapter sequences’, ‘Seed mismatches’, ‘Simple clip threshold’ and ‘Palindrome clip threshold’ are also needed in order to use this parameter.

disabled !advanced.trimming.seed_mismatches && !advanced.trimming.simple_clip_threshold && !advanced.trimming.palindrome_clip_threshold

default `8`

advanced.trimming.palindrome_clip_threshold

label Palindrome clip threshold

type `basic:integer`

description Specifies how accurate the match between the two ‘adapter ligated’ reads must be for PE palindrome read alignment. This field as well as ‘Adapter sequence’, ‘Simple clip threshold’ and ‘Seed mismatches’ parameters are needed to perform Illuminaclipping.

required False

disabled !advanced.trimming.adapters

advanced.trimming.leading

label Leading quality

type `basic:integer`

description Remove low quality bases from the beginning, if below a threshold quality.

required False

advanced.trimming.trailing

label Trailing quality

type `basic:integer`

description Remove low quality bases from the end, if below a threshold quality.

required False

advanced.trimming.minlen

label Minimum length

type `basic:integer`

description Drop the read if it is below a specified length.

required False

advanced.align.seed_l

label Minimum seed length

type `basic:integer`

description Minimum seed length. Matches shorter than minimum seed length will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20.

default `19`

advanced.align.band_w

label Band width

type `basic:integer`

description Gaps longer than this will not be found.

default `100`

advanced.align.m

label Mark shorter split hits as secondary

type `basic:boolean`

description Mark shorter split hits as secondary (for Picard compatibility)

default `False`

advanced.align.re_seeding

label Re-seeding factor

type `basic:decimal`

description Trigger re-seeding for a MEM longer than minSeedLen*FACTOR. This is a key heuristic parameter for tuning the performance. Larger value yields fewer seeds, which leads to faster alignment speed but lower accuracy.

default `1.5`

advanced.align.scoring.match

label Score of a match

type `basic:integer`

default `1`

advanced.align.scoring.mismatch

label Mismatch penalty

type `basic:integer`

default `4`

advanced.align.scoring.gap_o

label Gap open penalty

type `basic:integer`

default `6`

advanced.align.scoring.gap_e

label Gap extension penalty

type `basic:integer`

default `1`

advanced.align.scoring.clipping

label Clipping penalty

type `basic:integer`

description Clipping is applied if final alignment score is smaller than (best score reaching the end of query) - (Clipping penalty)

default `5`

advanced.align.scoring.unpaired_p

label Penalty for an unpaired read pair

type `basic:integer`

description Affinity to force pair. Score: scoreRead1+scoreRead2-Penalty

default `9`

advanced.align.report_tr

label Report threshold score

type `basic:integer`

description Don't output alignment with score lower than defined number. This option only affects output.

default `30`

advanced.markduplicates.md_skip

label Skip GATK's MarkDuplicates step

type `basic:boolean`

default `False`

advanced.markduplicates.md_remove_duplicates

label Remove found duplicates

type `basic:boolean`

default `False`

advanced.markduplicates.md_assume_sort_order

label Assume sort oder

type `basic:string`

default

choices

- as in BAM header (default):
- unsorted: `unsorted`
- queryname: `queryname`
- coordinate: `coordinate`
- duplicate: `duplicate`
- unknown: `unknown`

advanced.bqsr.read_group

label Read group (@RG)

type `basic:string`

description If BAM file has not been prepared using a @RG tag, you can add it here. This argument enables the user to replace all read groups in the INPUT file with a single new read group and assign all reads to this read group in the OUTPUT BAM file. Addition or replacement is performed using Picard's AddOrReplaceReadGroups tool. Input should take the form of -name=value delimited by a \t, e.g. "-ID=1\t-PL=Illumina\t-SM=sample_1". See AddOrReplaceReadGroups documentation for more information on tag names. Note that PL, LB, PU and SM are required fields. See caveats of rewriting read groups in the documentation linked above.

required False

advanced.hc.stand_call_conf

label Min call confidence threshold

type `basic:integer`

description The minimum phred-scaled confidence threshold at which variants should be called.

default `20`

advanced.hc.mbq

label Min Base Quality

type `basic:integer`

description Minimum base quality required to consider a base for calling.

default `20`

Output results

coverageBed

`data:coveragecoveragebed` (*data:alignment:bam* **alignment**, *data:masterfile:amplicon* **master_file**) [Source: v4.1.1]

Bedtools coverage (coveragebed)

Input arguments alignment

label Alignment (BAM)

type `data:alignment:bam`

master_file

label Master file

type `data:masterfile:amplicon`

Output results cov_metrics

label Coverage metrics

type `basic:file`

mean_cov

label Mean amplicon coverage

type `basic:file`

amplicon_cov

label Amplicon coverage file (nomergebed)

type `basic:file`

covplot_html

label HTML coverage plot

type `basic:file:html`

edgeR

`data:differentialexpression:edgerdifferentialexpression-edger` (*list:data:expression* **case**, *list:data:expression* **control**, *basic:integer* **filter**) [Source: v1.1.1]

Empirical Analysis of Digital Gene Expression Data in R (edgeR). Differential expression analysis of RNA-seq expression profiles with biological replication. Implements a range of statistical methodology based on the negative binomial distributions, including empirical Bayes estimation, exact tests, generalized linear models and quasi-likelihood tests. As well as RNA-seq, it be applied to differential signal analysis of other types of genomic data that produce counts, including ChIP-seq, Bisulfite-seq, SAGE and CAGE. See [here](https://www.bioconductor.org/packages/devel/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf) for more information.

Input arguments case

label Case

type `list:data:expression`

description Case samples (replicates)

control

label Control

type `list:data:expression`

description Control samples (replicates)

filter

label Raw counts filtering threshold

type `basic:integer`

description Filter genes in the expression matrix input. Remove genes where the number of counts in all samples is below the threshold.

default `10`

Output results raw

label Differential expression

type `basic:file`

de_json

label Results table (JSON)

type `basic:json`

de_file

label Results table (file)

type `basic:file`

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

featureCounts

data:expression:featurecountsfeature_counts (*data:alignment:bam* **aligned_reads**, *basic:string* **assay_type**, *data:index:salmon* **cdna_index**, *basic:integer* **n_reads**, *data:annotation* **annotation**, *basic:string* **feature_class**, *basic:string* **feature_type**, *basic:string* **id_attribute**, *basic:string* **normalization_type**, *data:mappability:bcm* **mappability**, *basic:boolean* **show_advanced**, *basic:boolean* **count_features**, *basic:boolean* **allow_multi_overlap**, *basic:integer* **min_overlap**, *basic:decimal* **frac_overlap**, *basic:decimal* **frac_overlap_feature**, *basic:boolean* **largest_overlap**, *basic:integer* **read_extension_5**, *basic:integer* **read_extension_3**, *basic:integer* **read_to_pos**, *basic:boolean* **count_multi_mapping_reads**, *basic:boolean* **fraction**, *basic:integer* **min_mqs**, *basic:boolean* **split_only**, *basic:boolean* **non_split_only**, *basic:boolean* **primary**, *basic:boolean* **ignore_dup**, *basic:boolean* **junc_counts**, *data:genome* **genome**, *basic:boolean* **is_paired_end**, *basic:boolean* **require_both_ends_mapped**, *basic:boolean* **check_frag_length**, *basic:integer* **min_frag_length**, *basic:integer* **max_frag_length**, *basic:boolean* **do_not_count_chimeric_fragments**, *basic:boolean* **do_not_sort**, *basic:boolean* **by_read_group**, *basic:boolean* **count_long_reads**, *basic:boolean* **report_reads**, *basic:integer* **max_mop**, *basic:boolean* **verbose**) [Source: v2.4.1]

featureCounts is a highly efficient general-purpose read summarization program that counts mapped reads for genomic features such as genes, exons, promoter, gene bodies, genomic bins and chromosomal locations. It can be used to count both RNA-seq and genomic DNA-seq reads. See the [official website](http://bioinf.wehi.edu.au/featureCounts/) and the [introductory paper](https://academic.oup.com/bioinformatics/article/30/7/923/232889) for more information.

Input arguments alignment.aligned_reads

label Aligned reads

type `data:alignment:bam`

alignment.assay_type

label Assay type

type `basic:string`

description Indicate if strand-specific read counting should be performed. For paired-end reads, strand of the first read is taken as the strand of the whole fragment. FLAG field is used to tell if a read is first or second read in a pair. Automated strand detection is enabled using the [Salmon](https://salmon.readthedocs.io/en/latest/library_type.html) tool's build-in functionality. To use this option, cDNA (transcriptome) index file crated using the Salmon indexing tool must be provided.

default `non_specific`

choices

- Strand non-specific: `non_specific`
- Strand-specific forward: `forward`
- Strand-specific reverse: `reverse`
- Detect automatically: `auto`

alignment.cdna_index

label cDNA index file

type `data:index:salmon`

description Transcriptome index file created using the Salmon indexing tool. cDNA (transcriptome) sequences used for index file creation must be derived from the same species as the input sequencing reads to obtain the reliable analysis results.

required False

hidden alignment.assay_type != 'auto'

alignment.n_reads

label Number of reads in subsampled alignment file

type `basic:integer`

description Alignment (.bam) file subsample size. Increase the number of reads to make automatic detection more reliable. Decrease the number of reads to make automatic detection run faster.

hidden alignment.assay_type != 'auto'

default `5000000`

annotation.annotation

label Annotation

type `data:annotation`

description GTF and GFF3 annotation formats are supported.

annotation.feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GTF/GFF3 file) to be used. All other features will be ignored.

default `exon`

annotation.feature_type

label Feature type

type `basic:string`

description The type of feature the quantification program summarizes over (e.g. gene or transcript-level analysis). The value of this parameter needs to be chosen in line with 'ID attribute' below.

default `gene`

choices

- gene: `gene`
- transcript: `transcript`

annotation.id_attribute

label ID attribute

type `basic:string`

description GTF/GFF3 attribute to be used as feature ID. Several GTF/GFF3 lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identify the counts in the output table. In GTF files this is usually 'gene_id', in GFF3 files this is often 'ID', and 'transcript_id' is frequently a valid choice for both annotation formats.

default `gene_id`

choices

- gene_id: `gene_id`
- transcript_id: `transcript_id`
- ID: `ID`
- geneid: `geneid`

normalization_type

label Normalization type

type `basic:string`

description The default expression normalization type.

default `TPM`

choices

- TPM: `TPM`
- CPM: `CPM`
- FPKM: `FPKM`
- RPKUM: `RPKUM`

mappability

label Mappability

type `data:mappability:bcm`

description Genome mappability information

required False

hidden normalization_type != 'RPKUM'

show_advanced

label Show advanced options

type `basic:boolean`

description Inspect and modify parameters

default `False`

advanced.summarization_level.count_features

label Perform read counting at feature level

type `basic:boolean`

description Count reads for exons rather than genes.

default `False`

advanced.overlap.allow_multi_overlap

label Assign reads to all their overlapping features or meta-features

type `basic:boolean`

default `False`

advanced.overlap.min_overlap

label Minimum number of overlapping bases in a read that is required for read assignment

type `basic:integer`

description Number of overlapping bases is counted from both reads if paired-end. If a negative value is provided, then a gap of up to specified size will be allowed between read and the feature that the read is assigned to.

default `1`

advanced.overlap.frac_overlap

label Minimum fraction of overlapping bases in a read that is required for read assignment

type `basic:decimal`

description Value should be within range [0, 1]. Number of overlapping bases is counted from both reads if paired end. Both this option and 'Minimum number of overlapping bases in a read that is required for read assignment' need to be satisfied for read assignment.

default `0.0`

advanced.overlap.frac_overlap_feature

label Minimum fraction of overlapping bases included in a feature that is required for overlapping with a read or a read pair

type `basic:decimal`

description Value should be within range [0, 1].

default `0.0`

advanced.overlap.largest_overlap

label Assign reads to a feature or meta-feature that has the largest number of overlapping bases

type `basic:boolean`

default `False`

advanced.overlap.read_extension_5

label Number of bases to extend reads upstream by from their 5' end

type `basic:integer`

default `0`

advanced.overlap.read_extension_3

label Number of bases to extend reads upstream by from their 3' end

type `basic:integer`

default `0`

advanced.overlap.read_to_pos

label Reduce reads to their 5'-most or 3'-most base

type `basic:integer`

description Read counting is performed based on the single base the read is reduced to.

required False

advanced.multi_mapping_reads.count_multi_mapping_reads

label Count multi-mapping reads

type `basic:boolean`

description For a multi-mapping read, all its reported alignments will be counted. The 'NH' tag in BAM input is used to detect multi-mapping reads.

default `False`

advanced.fractional_counting.fraction

label Assign fractional counts to features

type `basic:boolean`

description This option must be used together with 'Count multi-mapping reads' or 'Assign reads to all their overlapping features or meta-features' or both. When 'Count multi-mapping reads' is checked, each reported alignment from a multi-mapping read (identified via 'NH' tag) will carry a count of 1 / x, instead of 1 (one), where x is the total number of alignments reported for the same read. When 'Assign reads to all their overlapping features or meta-features' is checked, each overlapping feature will receive a count of 1 / y, where y is the total number of features overlapping with the read. When both 'Count multi-mapping reads' and 'Assign reads to all their overlapping features or meta-features' are specified, each alignment will carry a count of 1 / (x * y).

required False

disabled !advanced.multi_mapping_reads.count_multi_mapping_reads && !advanced.overlap.allow_multi_overlap

default `False`

advanced.read_filtering.min_mqs

label Minimum mapping quality score

type `basic:integer`

description The minimum mapping quality score a read must satisfy in order to be counted. For paired-end reads, at least one end should satisfy this criterion.

default `0`

advanced.read_filtering.split_only

label Count only split alignments

type `basic:boolean`

default `False`

advanced.read_filtering.non_split_only

label Count only non-split alignments

type `basic:boolean`

default `False`

advanced.read_filtering.primary

label Count only primary alignments

type `basic:boolean`

description Primary alignments are identified using bit 0x100 in BAM FLAG field.

default `False`

advanced.read_filtering.ignore_dup

label Ignore duplicate reads in read counting

type `basic:boolean`

description Duplicate reads are identified using bit Ox400 in BAM FLAG field. The whole read pair is ignored if one of the reads is a duplicate read for paired-end data.

default `False`

advanced.exon_exon_junctions.junc_counts

label Count number of reads supporting each exon-exon junction

type `basic:boolean`

description Junctions are identified from those exon-spanning reads in input (containing 'N' in CIGAR string).

default `False`

advanced.exon_exon_junctions.genome

label Genome

type `data:genome`

description Reference sequences used in read mapping that produced the provided BAM files. This optional argument can be used to improve read counting for junctions.

required False

disabled !advanced.exon_exon_junctions.junc_counts

advanced.paired_end.is_paired_end

label Count fragments (or templates) instead of reads

type `basic:boolean`

default `True`

advanced.paired_end.require_both_ends_mapped

label Count only read pairs that have both ends aligned

type `basic:boolean`

default `False`

advanced.paired_end.check_frag_length

label Check fragment length when assigning fragments to meta-features or features

type `basic:boolean`

description Use minimum and maximum fragment/template length to set thresholds.

default `False`

advanced.paired_end.min_frag_length

label Minimum fragment/template length

type `basic:integer`

required False

disabled !advanced.paired_end.check_frag_length

default `50`

advanced.paired_end.max_frag_length

label Maximum fragment/template length

type `basic:integer`

required False

disabled !advanced.paired_end.check_frag_length

default `600`

advanced.paired_end.do_not_count_chimeric_fragments

label Do not count chimeric fragments

type `basic:boolean`

description Do not count read pairs that have their two ends mapped to different chromosomes or mapped to same chromosome but on different strands.

default `False`

advanced.paired_end.do_not_sort

label Do not sort reads in BAM input

type `basic:boolean`

default `False`

advanced.read_groups.by_read_group

label Assign reads by read group

type `basic:boolean`

description RG tag is required to be present in the input BAM files.

default `False`

advanced.long_reads.count_long_reads

label Count long reads such as Nanopore and PacBio reads

type `basic:boolean`

default `False`

advanced.miscellaneous.report_reads

label Output detailed assignment results for each read or read pair

type `basic:boolean`

default `False`

advanced.miscellaneous.max_mop

label Maximum number of ‘M’ operations allowed in a CIGAR string

type `basic:integer`

description Both ‘X’ and ‘=’ are treated as ‘M’ and adjacent ‘M’ operations are merged in the CIGAR string.

default `10`

advanced.miscellaneous.verbose

label Output verbose information

type `basic:boolean`

description Output verbose information for debugging, such as unmatched chromosome / contig names.

default `False`

Output results rc

label Read counts

type `basic:file`

fpkm

label FPKM

type `basic:file`

tpm

label TPM

type `basic:file`

cpm

label CPM

type `basic:file`

exp

label Default expression output

type `basic:file`

exp_json

label Default expression output (json)

type `basic:json`

exp_type

label Expression normalization type (on default output)

type `basic:string`

exp_set

label Expressions

type `basic:file`

exp_set_json

label Expressions (json)

type `basic:json`

feature_counts_output

label featureCounts output

type `basic:file`

counts_summary

label Counts summary

type `basic:file`

read_assignments

label Read assignments

type `basic:file`

description Read assignment results for each read (or fragment if paired end).

required False

strandedness_report

label Strandedness report file

type `basic:file`

required False

source

label Gene ID database

type `basic:string`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

feature_type

label Feature type

type `basic:string`

methcounts

data:wgbs:methcountsmethcounts (*data:genome:fasta* **genome**, *data:alignment:mr* **alignment**, *basic:boolean* **cpgs**, *basic:boolean* **symmetric_cpgs**) [Source: v1.0.1]

The methcounts program takes the mapped reads and produces the methylation level at each genomic cytosine, with the option to produce only levels for CpG-context cytosines.

Input arguments genome

label Reference genome

type `data:genome:fasta`

alignment

label Mapped reads

type `data:alignment:mr`

description WGBS alignment file in Mapped Read (.mr) format.

cpgs

label Only CpG context sites

type `basic:boolean`

description Output file will contain methylation data for CpG context sites only. Choosing this option will result in CpG content report only.

disabled symmetric_cpgs

default `False`

symmetric_cpgs

label Merge CpG pairs

type `basic:boolean`

description Merging CpG pairs results in symmetric methylation levels. Methylation is usually symmetric (cytosines at CpG sites were methylated on both DNA strands). Choosing this option will only keep the CpG sites data.

disabled cpgs

default `True`

Output results meth

label Methylation levels

type `basic:file`

stats

label Statistics

type `basic:file`

bigwig

label Methylation levels BigWig file

type `basic:file`

species

label Species

type `basic:string`

build

label Build

type `basic:string`

miRNA pipeline

data:workflow:mirnaworkflow-mirna (*data:reads:fastq* **reads**, *data:genome:fasta* **genome**, *data:annotation* **annotation**, *basic:string* **id_attribute**, *basic:string* **feature_class**) [Source: v0.0.5]

Input arguments reads

label Input miRNA reads.

type `data:reads:fastq`

description Note that these reads should already be void of adapters.

genome

label Genome

type `data:genome:fasta`

annotation

label Annotation (GTF/GFF3)

type `data:annotation`

id_attribute

label ID attribute

type `basic:string`

description GTF/GFF3 attribute to be used as feature ID. Several GTF/GFF3 lines with the same feature ID will be considered as parts of the same feature. The feature ID is used to identify the counts in the output table. In GTF files this is usually 'gene_id', in GFF3 files this is often 'ID', and 'transcript_id' is frequently a valid choice for both annotation formats.

default `gene_id`

choices

- gene_id: `gene_id`
- transcript_id: `transcript_id`
- ID: `ID`
- geneid: `geneid`

feature_class

label Feature class

type `basic:string`

description Feature class (3rd column in GFF file) to be used, all features of other types are ignored.

default `miRNA`

Output results

snpEff

data:snpeff:uploadupload-snpeff (*basic:file* **annotation**, *basic:file* **summary**, *basic:file* **snpeff_genes**) [Source: v1.1.1]

Upload snpEff result files.

Input arguments annotation

label Annotation file

type `basic:file`

summary

label Summary

type `basic:file`

snpeff_genes

label SnpEff genes

type `basic:file`

Output results annotation

label Annotation file

type `basic:file`

summary

label Summary

type `basic:file:html`

snpeff_genes

label SnpEff genes

type `basic:file`

snpEff

data:snpeffsnpeff (*data:variants:vcf* **variants**, *basic:string* **var_source**, *basic:string* **database**, *list:data:variants:vcf* **known_vars_annot**) [Source: v0.2.1]

Variant annotation using snpEff package.

Input arguments variants

label Variants (VCF)

type `data:variants:vcf`

var_source

label Input VCF source

type `basic:string`

choices

- GATK HC: `gatk_hc`

- loFreq: `lofreq`

database

label snpEff database

type `basic:string`

default `GRCh37.75`

choices

- GRCh37.75: `GRCh37.75`

known_vars_annot

label Known variants

type `list:data:variants:vcf`

Output results annotation

label Annotation file

type `basic:file`

summary

label Summary

type `basic:file:html`

snpeff_genes

label SnpEff genes

type `basic:file`

1.3 Descriptor schemas

When working with the biological data, it is recommended (and often required) to properly annotate samples. The annotation information attached to the samples includes information about *organism*, *source*, *cell type*, *library preparation protocols* and others.

The annotation fields associated with the samples or related sample files are defined in the descriptor schemas. This tutorial describes the descriptor schemas that are attached to the sample objects, raw sequencing reads and differential expressions files.

Other available descriptor schemas can be explored at the Resolwe-bio GitHub page. Customized descriptor schemas can be created using the Resolwe SDK.

1.3.1 Sample

When a new data object that represents a biological sample (i.e. fastq files, bam files) is uploaded to the database, the unannotated sample (presample) is automatically created. When annotation is attached to the presample object, this object is automatically converted to the annotated sample. To annotate the sample, we need to define a descriptor schema that will be used for the annotation. Together with the descriptor schema, we need to provide the annotations (descriptors) that populate the annotation fields defined in the descriptor shema. The details of this process are described in the Resolwe SDK documentation.

To annotate the sample in a GEO compliant way, we prepared the sample annotation schema. An example of the customized descriptor schema is also available.

1.3.2 Reads

To annotate raw sequencing reads we have prepared two descriptor schemas: reads and reads_detailed.

1.3.3 Differential expression

To define the default thresholds for `p-value`, `log fold change (FC)` and to describe which samples are used as cases and which as controls in the calculation of differential expression we have prepared diffexp descriptor schema.

1.4 Reference

1.4.1 Utilities

Test helper functions.

class `resolwe_bio.utils.test.`**`BioProcessTestCase`**(*methodName='runTest'*)

Base class for writing bioinformatics process tests.

It is a subclass of Resolwe's `ProcessTestCase` with some specific functions used for testing bioinformatics processes.

`prepare_adapters`(*fn='adapters.fasta'*)

Prepare adapters FASTA.

`prepare_amplicon_master_file`(*mfile='56G_masterfile_test.txt', pname='56G panel, v2'*)

Prepare amplicon master file.

`prepare_annotation`(*fn='sp_test.gtf', source='DICTYBASE', species='Dictyostelium discoideum', build='dd-05-2009'*)

Prepare annotation GTF.

`prepare_annotation_gff`(*fn='annotation dicty.gff.gz', source='DICTYBASE', species='Dictyostelium discoideum', build='dd-05-2009'*)

Prepare annotation GFF3.

`prepare_bam`(*fn='sp_test.bam', species='Dictyostelium discoideum', build='dd-05-2009'*)

Prepare alignment BAM.

`prepare_expression`(*f_rc='exp_1_rc.tab.gz', f_exp='exp_1_tpm.tab.gz', f_type='TPM', name='Expression', source='DICTYBASE', descriptor=None, feature_type='gene', species='Dictyostelium discoideum', build='dd-05-2009'*)

Prepare expression.

`prepare_genome`()

Prepare genome FASTA.

`prepare_paired_reads`(*mate1=['fw reads.fastq.gz'], mate2=['rw reads.fastq.gz']*)

Prepare NGS reads FASTQ.

`prepare_reads`(*fn=['reads.fastq.gz']*)

Prepare NGS reads FASTQ.

`setUp`()

Initialize test files path.

class resolwe_bio.utils.test.**KBBioProcessTestCase**(*methodName='runTest'*)

Class for bioinformatics process tests that use knowledge base.

It is based on *BioProcessTestCase* and Django's LiveServerTestCase. The latter launches a live Django server in a separate thread so that the tests may use it to query the knowledge base.

setUp()

Set-up test gene information knowledge base.

resolwe_bio.utils.test.**skipDockerFailure**(*reason*)

Skip decorated tests due to failures when run in Docker.

Unless TESTS_SKIP_DOCKER_FAILURES Django setting is set to False. reason should describe why the test is being skipped.

resolwe_bio.utils.test.**skipUnlessLargeFiles**(**files*)

Skip decorated tests unless large files are available.

Parameters ***files** (*list*) – variable lenght files list, where each element represents a large file path relative to the TEST_LARGE_FILES_DIR directory

1.5 Change Log

All notable changes to this project are documented in this file. This project adheres to Semantic Versioning.

1.5.1 Unreleased

Added

- Add alleyoop-rates process
- Add alleyoop-utr-rates process
- Add alleyoop-summary process
- Add alleyoop-snpeval process
- Add alleyoop-collapse process
- Add slam-count process
- Add workflow-slamdunk-paired workflow

Changed

- **BACKWARD INCOMPATIBLE:** Refactor slamdunk-all-paired process to support genome browser visualization and add additional output fields
- Append sample and genome reference information to the summary output file in the filtering-chemut process
- Bigwig output field in bamclipper, bqsr and markduplicates processes is no longer required
- Freeze docutils package version to 0.15.2 because Sphinx has problems parsing development version numbers
- Support Slamdunk/Alleyoop processes in MultiQC
- Enable sorting of files in alignment-star process using samtools

1.5.2 24.0.0 - 2019-11-15

Added

- Add `resolwebio/slamdunk` Docker image
- Add Tabix (1.7-2) to `resolwebio/bamliquidator:1.2.0` Docker image
- Add `seqtk-rev-complement-single` and `seqtk-rev-complement-paired` process
- Add `slamdunk-all-paired` process

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 20.x
- Make BaseSpace file download more robust
- Bump `rose2` to 1.1.0, `bamliquidator` to 1.3.8, and use `resolwebio/base:ubuntu-18.04` Docker image as a base image in `resolwebio/bamliquidator:1.1.0` Docker image
- Use `resolwebio/bamliquidator:1.2.0` in `rose2` process
- Bump CPU, memory and Docker image (`resolwebio/rnaseq:4.9.0`) requirements in `alignment-bwa-mem`, `alignment-bwa-sw` and `alignment-bwa-aln` processes
- Use multi-threading option in Samtools commands in `alignment-bwa-mem`, `alignment-bwa-sw` and `alignment-bwa-aln` processes
- Support merging of multi-lane sequencing data into a single (pair) of FASTQ files in the `upload-fastq-single`, `upload-fastq-paired`, `files-to-fastq-single` and `files-to-fastq-paired` processes.

1.5.3 23.1.1 - 2019-10-11

Changed

- Renamed `workflow-trim-align-quant` workflow to make the name more informative

1.5.4 23.1.0 - 2019-09-30

Added

- Add `Macaca mulatta` species choice to the `sample` descriptor schema
- Add `workflow-cutadapt-star-fc-quant-wo-depletion-single` process

Changed

- Test files improved for `workflow-wes`, `bamclipper`, `markduplicates` and `bqsr`
- Fix typo in `differentialexpression-shrna` process docstring

Fixed

- Fix transcript-to-gene_id mapping for Salmon expressions in differentialexpression-deseq2 process. Transcript versions are now ignored when matching IDs using the transcript-to-gene_id mapping table.
- Fix workflow-cutadapt-star-fc-quant-single process description

1.5.5 23.0.0 - 2019-09-17

Changed

- Update order of QC reports in MultiQC configuration file. The updated configuration file is part of the resolwebio/common:1.3.1 Docker image.
- Bump Jbrowse to version 1.16.6 in resolwebio/rnaseq:4.9.0 Docker image
- Use JBrowse generate-names.pl script to index GTF/GFF3 features upon annotation file upload
- Support Salmon reports in MultiQC and expose dirs_depth parameter
- Expose transcript-level expression file in the salmon-quant process

Added

- Add workflow-bbduk-salmon-qc-single and workflow-bbduk-salmon-qc-paired workflows

Fixed

- Give process upload-bedpe access to network

1.5.6 22.0.0 - 2019-08-20

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 19.x
- **BACKWARD INCOMPATIBLE:** Unify cutadapt-single and cutadapt-paired process inputs and refactor to use Cutadapt v2.4
- Expose BetaPrior parameter in differentialexpression-deseq2 process
- Install R from CRAN-maintained repositories in Docker images build from the resolwebio/base:ubuntu-18.04 base image
- Prepare resolwebio/common:1.3.0 Docker image:
  - Install R v3.6.1
  - Bump Resdk to v10.1.0
  - Install gawk package
  - Fix Docker image build issues
- Use resolwebio/common:1.3.0 as a base image for resolwebio/rnaseq:4.8.0
- Update StringTie to v2.0.0 in resolwebio/rnaseq:4.8.0

- Support StringTie analysis results in DESeq2 tool

Added

- Add `cutadapt-3prime-single` process
- Add `workflow-cutadapt-star-fc-quant-single` process
- Add argument `skip` to `bamclipper` which enables skipping of the said process
- Add `cutadapt-corall-single` and `cutadapt-corall-paired` processes for pre-processing of reads obtained using Corall Total RNA-seq library prep kit
- Add `umi-tools-dedup` process
- Add `stringtie` process
- Add `workflow-corall-single` and `workflow-corall-paired` workflows optimized for Corall Total RNA-seq library prep kit data

Fixed

- Fix warning message in hierarchical clustering of genes. Incorrect gene names were reported in the warning message about removed genes. Computation of hierarchical clustering was correct.

1.5.7 21.0.1 - 2019-07-26

Changed

- Bump Cutadapt to v2.4 and use `resolwebio/common:1.2.0` as a base image in `resolwebio/rnaseq:4.6.0`

Added

- Add pigz package to `resolwebio/common:1.2.0` Docker image
- Add StringTie and UMI-tools to `resolwebio/rnaseq:4.7.0` Docker image

Fixed

- Fix `spikeins-qc` process to correctly handle the case where all expressions are without spikeins
- Fix an error in `macs2-callpeak` process that prevented correct reporting of build/species mismatch between inputs
- Support UCSC annotations in `feature_counts` process by assigning empty string gene_ids to the "unknown" gene

1.5.8 21.0.0 - 2019-07-16

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 18.x

- Bump the number of allocated CPU cores to 20 in `alignment-bwa-mem` process
- Bump memory requirements in `seqtk-sample-single` and `seqtk-sample-paired` processes
- Bump Salmon to v0.14.0 in `resolwebio/rnaseq:4.5.0` Docker image
- Expose additional inputs in `salmon-index` process
- Use `resolwebio/rnaseq:4.5.0` Docker image in processes that call Salmon tool (`library-strandedness`, `feature_counts` and `qorts-qc`)
- Implement dropdown menu for `upload-bedpe` process
- Add validation stringency parameter to `bqsr` process and propagate it to the `workflow-wes` as well
- Add LENIENT value to validation stringency parameter of the `markduplicates` process
- Improve performance of RPKUM normalization in `featureCounts` process

Added

- Add `salmon-quant` process

Fixed

- Fix genome upload process to correctly handle filenames with dots
- Fix merging of expressions in `archive-samples` process. Previously some genes were missing in the merged expression files. The genes that were present had expression values correctly assigned. The process was optimized for performance and now supports parallelization.

1.5.9 20.0.0 2019-06-19

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 17.x
- **BACKWARD INCOMPATIBLE:** Use Elasticsearch version 6.x
- **BACKWARD INCOMPATIBLE:** Bump Django requirement to version 2.2
- **BACKWARD INCOMPATIBLE:** Remove obsolete RNA-seq workflows `workflow-bbduk-star-featurecounts-single`, `workflow-bbduk-star-featurecounts-paired`, `workflow-cutadapt-star-featurecounts-single` and `workflow-cutadapt-star-featurecounts-pai`
- **BACKWARD INCOMPATIBLE:** Remove obsolete descriptor schemas: `rna-seq-bbduk-star-featurecounts`, `quantseq`, `rna-seq-cutadapt-star-featurecounts` and `kapa-rna-seq-bbduk-star-featurecounts`
- **BACKWARD INCOMPATIBLE:** In `upload-fasta-nucl` process, store compressed and uncompressed FASTA files in `fastagz` and `fasta` ouput fields, respectively
- Allow setting the Java memory usage flags for the QoRTs tool in `resolwebio/common:1.1.3` Docker image
- Use `resolwebio/common:1.1.3` Docker image as a base image for `resolwebio/rnaseq:4.4.2`
- Bump GATK4 version to 4.1.2.0 in `resolwebio/dnaseq:4.2.0`
- Use MultiQC configuration file and prepend directory name to sample names by default in `multiqc` process

- Bump `resolwebio/common` to 1.1.3 in `resolwebio/dnaseq:4.2.0`
- Process `vc-gatk4-hc` now also accepts BED files through parameter `intervals_bed`

Added

- Support Python 3.7
- Add Tabix (1.7-2) to `resolwebio/wgbs` docker image
- Add JBrowse index output to `hmr` process
- Add `bamclipper` tool and `parallel` package to `resolwebio/dnaseq:4.2.0` image
- Support `hg19_mm10` hybrid genome in `bam-split` process
- Support mappability-based normalization (RPKUM) in featureCounts
- Add BEDPE upload process
- Add `bamclipper` process
- Add `markduplicates` process
- Add `bqsr` (BaseQualityScoreRecalibrator) process
- Add whole exome sequencing (WES) pipeline

Fixed

- Fix building problems of `resolwebio/dnaseq` docker
- Fix handling of no-adapters input in workflows `workflow-bbduk-star-featurecounts-qc-single` and `workflow-bbduk-star-featurecounts-qc-paired`

1.5.10 19.0.1 2019-05-13

Fixed

- Use `resolwebio/rnaseq:4.4.2` Docker image that enforces the memory limit and bump memory requirements for `qorts-qc` process
- Bump memory requirements for `multiqc` process

1.5.11 19.0.0 2019-05-07

Changed

- Use Genialis fork of MultiQC 1.8.0b in `resolwebio/common:1.1.2`
- Support Samtools idxstats and QoRTs QC reports in `multiqc` process
- Support `samtools-idxstats` QC step in workflows:
  - `workflow-bbduk-star-featurecounts-qc-single`
  - `workflow-bbduk-star-featurecounts-qc-paired`
  - `workflow-bbduk-star-fc-quant-single`

  - workflow-bbduk-star-fc-quant-paired
- Simplify cellranger-count outputs folder structure
- Bump STAR aligner to version 2.7.0f in resolwebio/rnaseq:4.4.1 Docker image
- Use resolwebio/rnaseq:4.4.1 in alignment-star and alignment-star-index processes
- Save filtered count-matrix output file produced by DESeq2 differential expression process

Added

- Add samtools-idxstats process
- Improve cellranger-count and cellranger-mkref logging
- Add FastQC report to upload-sc-10x process

Fixed

- Fix archive-samples to work with data:chipseq:callpeak:macs2 data objects when downloading only peaks without QC reports
- Fix parsing gene set files with empty lines to avoid saving gene sets with empty string elements

1.5.12 18.0.0 2019-04-16

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 16.x
- **BACKWARD INCOMPATIBLE:** Rename and improve descriptions of processes specific to CATS RNA-seq kits. Remove related cutadapt-star-htseq descriptor schema.
- **BACKWARD INCOMPATIBLE:** Remove workflow-accel-gatk4 pipeline. Remove amplicon-panel, amplicon-panel-advanced and amplicon-master-file descriptor schemas.
- **BACKWARD INCOMPATIBLE:** Remove obsolete processes and descriptor schemas: rna-seq-quantseq, bcm-workflow-rnaseq, bcm-workflow-chipseq, bcm-workflow-wgbs, dicty-align-reads, dicty-etc, affy and workflow-chip-seq
- Expose additional parameters of bowtie2 process
- Support strandedness auto detection in qorts-qc process

Added

- Add shRNAde (v1.0) R package to the resolwebio/rnaseq:4.4.0 Docker image
- Add resolwebio/scseq Docker image
- Add shRNA differential expression process. This is a two-step process which trims, aligns and quantifies short hairpin RNA species. These are then used in a differential expression.
- Add sc-seq processes:
  - cellranger-mkref

  - cellranger-count
  - upload-sc-10x
  - upload-bam-scseq-indexed

Fixed

- Bump memory requirements in seqtk-sample-single and seqtk-sample-paired processes
- Fix cellranger-count html report
- Mark spliced-alignments with XS flags in workflow-rnaseq-cuffquant
- Fix whitespace handling in cuffnorm process

1.5.13 17.0.0 2019-03-19

Added

- Add qorts-qc (Quality of RNA-seq Tool-Set QC) process
- Add workflow-bbduk-star-fc-quant-single and workflow-bbduk-star-fc-quant-paired processes
- Add independent gene filtering and gene filtering based on Cook's distance in DESeq2 differential expression process

Changed

- **BACKWARD INCOMPATIBLE**: Move gene filtering by expression count input to filter.min_count_sum in DESeq2 differential expression process
- **BACKWARD INCOMPATIBLE:** Require Resolwe 15.x
- Update resolwebio/common:1.1.0 Docker image:
  - add QoRTs (1.3.0) package
  - bump MultiQC to 1.7.0
  - bump Subread package to 1.6.3
- Expose maxns input parameter in bbduk-single and bbduk-paired processes. Make this parameter available in workflows workflow-bbduk-star-featurecounts-qc-single, workflow-bbduk-star-featurecounts-qc-paired, workflow-bbduk-star-featurecounts-single and workflow-bbduk-star-featurecounts-paired.
- Save CPM-normalized expressions in feature_counts process. Control the default expression normalization type (exp_type) using the normalization_type input.
- Bump MultiQC to version 1.7.0 in multiqc process
- Use resolwebio/rnaseq:4.3.0 with Subread/featureCounts version 1.6.3 in feature_counts process

1.5.14 16.3.0 2019-02-19

Changed

- Bump STAR aligner version to 2.7.0c in `resolwebio/rnaseq:4.2.2`
- Processes `alignment-star` and `alignment-star-index` now use Docker image `resolwebio/rnaseq:4.2.2` which contains STAR version `2.7.0c`
- Persistence of `basespace-file-import` process changed from `RAW` to `TEMP`

Added

- Make `prepare-geo-chipseq` work with both `data:chipseq:callpeak:macs2` and `data:chipseq:callpeak:macs14` as inputs

Fixed

- Report correct total mapped reads and mapped reads percentage in prepeak QC report for `data:alignment:bam:bowtie2` inputs in `macs2-callpeak` process

1.5.15 16.2.0 2019-01-28

Changed

- Enable multithreading mode in `alignment-bwa-aln` and `alignment-bwa-sw`
- Lineary lower the timeout for BigWig calculation when running on multiple cores

Fixed

- Remove `pip --process-dependency-links` argument in testenv settings
- Fix walt getting killed when `sort` runs out of memory. The `sort` command buffer size was limited to the process memory limit.

1.5.16 16.1.0 2019-01-17

Changed

Added

- Add the `FASTQ` file validator script to the `upload-fastq-single`, `upload-fastq-paired`, `files-to-fastq-single` and `files-to-fastq-paired` processes
- Add `spikein-qc` process
- Add to `resolwebio/rnaseq:4.1.0` Docker image:
  - `dnaio` Python library
- Add to `resolwebio/rnaseq:4.2.0` Docker image:
  - ERCC table

  - common Genialis fonts and css file
  - spike-in QC report template
- Set MPLBACKEND environment variable to Agg in resolwebio/common:1.0.1 Docker image

Fixed

- Fix the format of the output FASTQ file in the demultiplex.py script
- Fix NSC and RSC QC metric calculation for ATAC-seq and paired-end ChIP-seq samples in macs2-callpeak and qc-prepeak processes

1.5.17 16.0.0 2018-12-19

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 14.x
- **BACKWARD INCOMPATIBLE:** Remove obsolete processes findsimilar
- **BACKWARD INCOMPATIBLE:** Include ENCODE-proposed QC analysis metrics methodology in the macs2-callpeak process. Simplified MACS2 analysis inputs now allow the use of sample relations (treatment/background) concept to trigger multiple MACS2 jobs automatically using the macs2-batch or macs2-rose2-batch processes.
- **BACKWARD INCOMPATIBLE:** Update workflow-atac-seq inputs to match the updated macs2-callpeak process
- Use resolwebio/rnaseq:4.0.0 Docker image in alignment-star-index, bbduk-single, bbduk-paired, cuffdiff, cufflinks, cuffmerge, cuffnorm, cuffquant, cutadapt-custom-single, cutadapt-custom-paired, cutadapt-single, cutadapt-paired, differentialexpression-deseq2, differentialexpression-edger, expression-aggregator, feature_counts, goenrichment, htseq-count, htseq-count-raw, index-fasta-nucl, library-strandedness, pca, regtools-junctions-annotate, rsem, salmon-index, trimmomatic-single, trimmomatic-paired, upload-expression, upload-expression-cuffnorm, upload-expression-star, upload-fasta-nucl, upload-fastq-single, upload-fastq-paired, files-to-fastq-single, files-to-fastq-paired, upload-gaf, upload-genome, upload-gff3, upload-gtf and upload-obo
- Order statistical groups in expression aggregator output by sample descriptor field value
- Use resolwebio/biox:1.0.0 Docker image in etc-bcm, expression-dicty and mappability-bcm processes
- Use resolwebio/common:1.0.0 Docker image in amplicon-table, mergeexpressions, upload-diffexp, upload-etc, upload-multiplexed-single and upload-multiplexed-paired processes
- Use resolwebio/base:ubuntu-18.04 Docker image in create-geneset, create-geneset-venn, mergeetc, prepare-geo-chipseq, prepare-geo-rnaseq, upload-cxb, upload-geneset, upload-header-sam, upload-mappability, upload-snpeff and upload-picard-pcrmetrics processes
- Update GATK4 to version 4.0.11.0 in resolwebio/dnaseq:4.1.0 Docker image. Install and use JDK v8 by default to ensure compatibility with GATK4 package.

- Use resolwebio/dnaseq:4.1.0 Docker image in align-bwa-trim, coveragebed, filtering-chemut, lofreq, picard-pcrmetrics, upload-master-file, upload-variants-vcf and vc-gatk4-hc processes
- Expose reads quality filtering (q) parameter, reorganize inputs and rename the stats output file in alignment-bwa-aln process
- Use resolwebio/chipseq:4.0.0 Docker image in chipseq-genescore, chipseq-peakscore, macs14, upload-bed and qc-prepeak processes
- Use resolwebio/bamliquidator:1.0.0 Docker image in bamliquidator and bamplot processes

Added

- Add biosample source field to sample descriptor schema
- Add background_pairs Jinja expressions filter that accepts a list of data objects and orders them in a list of pairs (case, background) based on the background relation between corresponding samples
- Add chipseq-bwa descriptor schema. This schema specifies the default inputs for BWA ALN aligner process as defined in ENCODE ChIP-Seq experiments.
- Add support for MACS2 result files to MultiQC process
- Add macs2-batch, macs2-rose2-batch and workflow-macs-rose processes
- Add feature symbols to expressions in archive-samples process

Fixed

- Make ChIP-seq fields in sample descriptor schema visible when ChIPmentation assay type is selected
- Fix handling of whitespace in input BAM file name in script detect_strandedness.sh
- Set available memory for STAR aligner to 36GB. Limit the available memory for STAR aligner --limitBAMsortRAM parameter to 90% of the Docker requirements setting
- Set bbduk-single and bbduk-paired memory requirements to 8GB
- Fix wrong file path in archive-samples process

1.5.18 15.0.0 2018-11-20

Changed

- **BACKWARD INCOMPATIBLE:** Remove obsolete processes: bsmap, mcall, coverage-garvan, igv, jbrowse-bed, jbrowse-gff3, jbrowse-gtf, jbrowse-bam-coverage, jbrowse-bam-coverage-normalized, jbrowse-refseq, fastq-mcf-single, fastq-mcf-paired, hsqutils-trim, prinseq-lite-single, prinseq-lite-paired, sortmerna-single, sortmerna-paired, bam-coverage, hsqutils-dedup, vc-samtools, workflow-heat-seq and alignment-tophat2
- **BACKWARD INCOMPATIBLE:** Remove jbrowse-bam-coverage process step from the workflow-accel workflow. The bigwig coverage track is computed in align-bwa-trim process instead.
- **BACKWARD INCOMPATIBLE:** Remove resolwebio/utils Docker image. This image is replaced by the resolwebio/common image.

- **BACKWARD INCOMPATIBLE:** Use `resolwebio/common` Docker image as a base image for the `resolwebio/biox`, `resolwebio/chipseq`, `resolwebio/dnaseq` and `resolwebio/rnaseq` images
- **BACKWARD INCOMPATIBLE:** Remove `resolwebio/legacy` Docker image.
- Use sample name as the name of the data object in:
  - `alignment-bwa-aln`
  - `alignment-bowtie2`
  - `qc-prepeak`
  - `macs2-callpeak`
- Attach `macs2-callpeak`, `macs14` and `rose2` process data to the case/treatment sample
- Use `resolwebio/dnaseq:4.0.0` docker image in `align-bwa-trim` process
- Use `resolwebio/rnaseq:4.0.0` docker image in aligners: `alignment-bowtie`, `alignment-bowtie2`, `alignment-bwa-mem`, `alignment-bwa-sw`, `alignment-bwa-aln`, `alignment-hisat2`, `alignment-star` and `alignment-subread`.
- Set memory limits in `upload-genome`, `trimmomatic-single` and `trimmomatic-paired` processes
- Improve error messages in differential expression process `DESeq2`

Added

- Add `makedb (WALT 1.01)` - callable as `makedb-walt`, tool to create genome index for WALT aligner, to `resolwebio/rnaseq` docker image
- Add `resolwebio/wgbs` docker image including the following tools:
  - `MethPipe (3.4.3)`
  - `WALT (1.01)`
  - `wigToBigWig (kent-v365)`
- Add `resolwebio/common` Docker image. This image includes common bioinformatics utilities and can serve as a base image for other, specialized `resolwebio` Docker images: `resolwebio/biox`, `resolwebio/chipseq`, `resolwebio/dnaseq` and `resolwebio/rnaseq`.
- Add `shift` (user-defined cross-correlation peak strandshift) input to `qc-prepeak` process
- Add ATAC-seq workflow
- Compute index for `WALT` aligner on genome upload and support uploading the index together with the genome
- Add `Whole genome bisulfite sequencing` workflow and related WGBS processes:
  - `WALT`
  - `methcounts`
  - `HMR`
- Add bedClip to *resolwebio/chipseq:3.1.0* docker image
- Add `resolwebio/biox` Docker image. This image is based on the `resolwebio/common` image and includes Biox Python library for Dictyostelium RNA-Seq analysis support.
- Add `resolwebio/snpeff` Docker image. The image includes SnpEff (4.3K) tool.
- Add spike-in names, rRNA and globin RNA cromosome names in `resolwebio/common` image

- Add UCSC bedGraphtoBigWig tool for calculating BigWig in `bamtobigwig.sh` script. In `align-bwa-trim` processor set this option (that BigWig is calculated by UCSC tool instead of deepTools), because it is much faster for amplicon files. In other processors update the input parameters for `bamtobigwig.sh`: `alignment-bowtie`, `alignment-bowtie2`, `alignment-bwa-mem`, `alignment-bwa-sw`, `alignment-bwa-aln`, `alignment-hisat2`, `alignment-star` `alignment-subread`, `upload-bam`, `upload-bam-indexed` and `upload-bam-secondary`.
- In `bamtobigwig.sh` don't create BigWig when bam file was aligned on globin RNA or rRNA (this are QC steps and BigWig is not needed)

Fixed

- **BACKWARD INCOMPATIBLE:** Use user-specificed distance metric in hierarchical clustering
- Handle integer expression values in hierarchical clustering
- Fix Amplicon table gene hyperlinks for cases where multiple genes are associated with detected variant
- Handle empty gene name in expression files in PCA
- Fix PBC QC reporting in `qc-prepeak` process for a case where there are no duplicates in the input bam
- Fix `macs2-callpeak` process so that user defined fragment lenth has priority over the `qc-prepeak` estimated fragment length when shifting reads for post-peakcall QC
- Fix `macs2-callpeak` to prevent the extension of intervals beyond chromosome boundaries in MACS2 bedgraph outputs
- Fix warning message in hierarchical clustering of genes to display gene names

1.5.19 14.0.2 2018-10-23

Fixed

- Fix `htseq-count-raw` process to correctly map features with associated feature symbols.

1.5.20 14.0.1 2018-10-23

Fixed

- Handle missing gene expression in hierarchical clustering of genes. If one or more genes requested in gene filter are missing in selected expression files a warning is issued and hierarchical clustering of genes is computed with the rest of the genes instead of failing.
- Fix PCA computation for single sample case

1.5.21 14.0.0 2018-10-09

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 13.x
- **BACKWARD INCOMPATIBLE:** Remove `gsize` input from `macs2-callpeak` process and automate genome size selection

- **BACKWARD INCOMPATIBLE:** Set a new default `sample` and `reads` descriptor schema. Change slug from `sample2` to `sample`, modify group names, add `cell_type` field to the new `sample` descriptor schema, and remove the original `sample`, `sample-detailed`, and `reads-detailed` descriptor schemas.
- **BACKWARD INCOMPATIBLE:** Unify types of `macs14` and `macs2-callpeak` processes and make `rose2` work with both
- **BACKWARD INCOMPATIBLE:** Remove `replicates` input in `cuffnorm` process. Use sample relation information instead.
- Use `resolwebio/chipseq:3.0.0` docker image in the following processes:
  - `macs14`
  - `macs2-callpeak`
  - `rose2`
- Downgrade primerclip to old version (v171018) in `resolwebio/dnaseq:3.3.0` docker image and move it to google drive.
- Move `bam-split` process to `resolwebio/rnaseq:3.7.1` docker image
- Count unique and multimmaping reads in `regtools-junctions-annotate` process

Added

- Add `qc-prepeak` process that reports ENCODE3 accepted ChIP-seq and ATAC-seq QC metrics
- Add QC report to `macs2-callpeak` process
- Add combining ChIP-seq QC reports in `archive-samples` process
- Add detection of globin-derived reads as an additional QC step in the `workflow-bbduk-star-featurecounts-qc-single` and `workflow-bbduk-star-featurecounts-qc-pai` processes.
- Add mappings from ENSEMBL or NCBI to UCSC chromosome names and deepTools (v3.1.0) to `resolwebio/dnaseq:3.3.0` docker image
- Add BigWig output field to following processors:
  - `align-bwa-trim`
  - `upload-bam`
  - `upload-bam-indexed`
  - `upload-bam-secondary`
- Add `replicate_groups` Jinja expressions filter that accepts a list of data objects and returns a list of labels determining replicate groups.
- Add 'Novel splice junctions in BED format' output to `regtools-junctions-annotate` process, so that user can visualize only novel splice juntions in genome browsers.

Fixed

- Fix handling of numerical feature_ids (NCBI source) in `create_expression_set.py` script
- Make `chipseq-peakscore` work with gzipped narrowPeak input from `macs2-callpeak`

- Use uncompressed FASTQ files as input to STAR aligner to prevent issues on (network) filesystems without FIFO support

1.5.22 13.0.0 2018-09-18

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 12.x
- **BACKWARD INCOMPATIBLE:** Remove obsolete processes: `assembler-abyss`, `cutadapt-amplicon`, `feature_location`, `microarray-affy-qc`, `reads-merge`, `reference_compatibility`, `transmart-expressions`, `upload-hmmer-db`, `upload-mappability-bigwig`, `upload-microarray-affy`.
- **BACKWARD INCOMPATIBLE:** Remove obsolete descriptor schema: `transmart`.
- **BACKWARD INCOMPATIBLE:** Remove tools which are not used by any process: `clustering_leaf_ordering.py`, `go_genesets.py`, `VCF_ad_extract.py`, `volcanoplot.py`, `xgff.py`, `xgtf2gff.py`.
- **BACKWARD INCOMPATIBLE:** Management command for inserting features and mappings requires PostgreSQL version 9.5 or newer
- Update the meta data like name, description, category, etc. of most of the processes
- Speed-up management command for inserting mappings
- Change location of cufflinks to Google Drive for resolwebio/rnaseq Docker build
- Calculate alignment statistics for the uploaded alignment (.bam) file in the `upload-bam`, `upload-bam-indexed` and `upload-bam-secondary` processes.
- Annotation (GTF/GFF3) file input is now optional for the creation of the STAR genome index files. Annotation file can be used at the alignment stage to supplement the genome indices with the set of known features.
- Trigger process warning instead of process error in the case when `bamtobigwig.sh` scripts detects an empty .bam file.
- Set the default reads length filtering parameter to 30 bp in the `rna-seq-bbduk-star-featurecounts` and `kapa-rna-seq-bbduk-star-featurecounts` experiment descriptor schema. Expand the kit selection choice options in the latter descriptor schema.

Added

- Add `MultiQC (1.6.0)` and `Seqtk (1.2-r94)` to the `resolwebio/utils:1.5.0` Docker image
- Add `sample2` descriptor schema which is the successor of the original `sample` and `reads` descriptor schemas
- Add bedToBigBed and Tabix to resolwebio/rnaseq:3.7.0 docker image
- Add `HS Panel` choice option to the `amplicon-master-file` descriptor schema
- Add MultiQC process
- Add process for the Seqtk tool `sample` sub-command. This process allows sub-sampling of `.fastq` files using either a fixed number of reads or the ratio of the input file.
- Add MultiQC analysis step to the `workflow-bbduk-star-featurecounts-single` and `workflow-bbduk-star-featurecounts-single` processes.

- Add `workflow-bbduk-star-featurecounts-qc-single` and `workflow-bbduk-star-featurecounts-qc-` processes which support MultiQC analysis, input reads down-sampling (using Seqtk) and rRNA sequence detection using STAR aligner.
- Add to `resolwebio/chipseq` Docker image:
  - `bedtools (2.25.0-1)`
  - `gawk (1:4.1.3+dfsg-0.1)`
  - `picard-tools (1.113-2)`
  - `run_spp.R (1.2) (as spp)`
  - `SPP (1.14)`
- Add `regtools-junctions-annotate` process that annotates novel splice junctions.
- Add `background` relation type to fixtures

Fixed

- Track `source` information in the `upload-fasta-nucl` process.
- When STAR aligner produces an empty alignment file, re-sort the alignment file to allow successful indexing of the output `.bam` file.
- Create a symbolic link to the alignment file in the `feature_counts` process, so that relative path is used in the quantification results. This prevent the FeatureCounts output to be listed as a separate sample in the MultiQC reports.
- Fix handling of expression objects in `archive-samples` process

1.5.23 12.0.0 - 2018-08-13

Changed

- **BACKWARD INCOMPATIBLE:** Require Resolwe 11.x
- **BACKWARD INCOMPATIBLE:** Use read count instead of sampling rate in strandedness detection
- **BACKWARD INCOMPATIBLE:** Remove `genome` input from `rose2` process and automate its selection
- **BACKWARD INCOMPATIBLE:** Refactor `cutadapt-paired` process
- **BACKWARD INCOMPATIBLE:** Improve leaf ordering performance in gene and sample hierarchical clustering. We now use exact leaf ordering which has been recently added to `scipy` instead of an approximate in-house solution based on nearest neighbor algorithm. Add informative warning and error messages to simplify troubleshooting with degenerate datasets.
- Remove `igvtools` from `resolwebio/utils` Docker image
- Improve helper text and labels in processes used for sequencing data upload
- Allow using custom adapter sequences in the `workflow-bbduk-star-featurecounts-single` and `workflow-bbduk-star-featurecounts-paired` processes
- Change chromosome names from ENSEMBL / NCBI to UCSC (example: "1" to "chr1") in BigWig files. The purpose of this is to enable viewing BigWig files in UCSC genome browsers for files aligned with ENSEBML or NCBI genome. This change is done by adding script bigwig_chroms_to_ucsc.py to bamtobigwig.sh script.
- Reduce RAM requirement in SRA import processes

Added

- Add two-pass mode to `alignment-star` process
- Add `regtools (0.5.0)` to `resolwebio/rnaseq` Docker image
- Add KAPA experiment descriptor schema
- Add `resdk` Python 3 package to `resolwebio/utils` Docker image
- Add to `cutadapt-single` process an option to discard reads having more 'N' bases than specified.
- Add workflows for single-end `workflow-cutadapt-star-featurecounts-single` and paired-end reads `workflow-cutadapt-star-featurecounts-paired`. Both workflows consist of preprocessing with Cutadapt, alignment with STAR two pass mode and quantification with featureCounts.
- Add descriptor schema `rna-seq-cutadapt-star-featurecounts`

Fixed

- **BACKWARD INCOMPATIBLE:** Fix the `stitch` parameter handling in `rose2`
- fix `upload-gtf` to create JBrowse track only if GTF file is ok
- Pin `sra-toolkit` version to 2.9.0 in `resolwebio/utils` Docker image.
- Fix and improve `rose2` error messages
- Fail gracefully if bam file is empty when producing bigwig files
- Fail gracefully if there are no matches when mapping chromosome names

1.5.24 11.0.0 - 2018-07-17

Changed

- **BACKWARD INCOMPATIBLE:** Remove management command module
- **BACKWARD INCOMPATIBLE:** Remove filtering of genes with low expression in PCA analysis
- **BACKWARD INCOMPATIBLE:** Remove obsolete RNA-seq DSS process
- Expand error messages in `rose2` process
- Check for errors during download of FASTQ files and use `resolwebio/utils:1.3.0` Docker image in import SRA process
- Increase Feature's full name's max length to 350 to support a long full name of "Complement C3 Complement C3 beta chain C3-beta-c Complement C3 alpha chain C3a anaphylatoxin Acylation stimulating protein Complement C3b alpha' chain Complement C3c alpha' chain fragment 1 Complement C3dg fragment Complement C3g fragment Complement C3d fragment Complement C3f fragment Complement C3c alpha' chain fragment 2" in Ensembl

Added

- Add *exp_set* and *exp_set_json* output fields to expression processes:
  - `feature_counts`
  - `htseq-count`

  - `htseq-count-raw`
  - `rsem`
  - `upload-expression`
  - `upload-expression-cuffnorm`
  - `upload-expression-star`
- Add ‘Masking BED file’ input to `rose2` process which allows masking reagions from the analysis
- Add `filtering.outFilterMismatchNoverReadLmax` input to `alignment-star` process
- Add mappings from ENSEMBL or NCBI to UCSC chromosome names to `resolwebio/rnaseq:3.5.0` docker image

Fixed

- Fix peaks BigBed output in `macs14` process
- Remove duplicated forward of `alignIntronMax` input field in BBDuk - STAR - featureCounts workflow
- Make `cuffnorm` process attach correct expression data objects to samples
- Fix `upload-gtf` in a way that GTF can be shown in JBrowse. Because JBrowse works only with GFF files, input GTF is converted to GFF from which JBrowse track is created.

1.5.25 10.0.1 - 2018-07-06

Fixed

- Fix `bamtobigwig.sh` to timeout the `bamCoverage` calculation after defined time

1.5.26 10.0.0 - 2018-06-19

Added

- Add to `resolwebio/chipseq` Docker image:
  - `Bedops (v2.4.32)`
  - `Tabix (v1.8)`
  - `python3-pandas`
  - `bedGraphToBigWig (kent-v365)`
  - `bedToBigBed (kent-v365)`
- Add to `resolwebio/rnaseq:3.2.0` Docker image:
  - `genometools (1.5.9)`
  - `igvtools (v2.3.98)`
  - `jbrowse (v1.12.0)`
  - `Bowtie (v1.2.2)`
  - `Bowtie2 (v2.3.4.1)`

  - BWA (0.7.17-r1188)
  - TopHat (v2.1.1)
  - Picard Tools (v2.18.5)
  - bedGraphToBigWig (kent-v365)
- Add Debian package file to resolwebio/rnaseq:3.3.0 Docker image
- Support filtering by type on feature API endpoint
- Add BigWig output field to following processes:
  - alignment-bowtie
  - alignment-bowtie2
  - alignment-tophat2
  - alignment-bwa-mem
  - alignment-bwa-sw
  - alignment-bwa-aln
  - alignment-hisat2
  - alignment-star
- Add Jbrowse track output field to upload-genome processor.
- Use reslowebio/rnaseq Docker image and add Jbrowse track and IGV sorting and indexing to following processes:
  - upload-gff3
  - upload-gtf
  - gff-to-gtf
- Add Tabix index for Jbrowse to upload-bed processor and use reslowebio/rnaseq Docker image
- Add BigWig, BigBed and JBrowse track outputs to macs14 process
- Add Species and Build outputs to rose2 process
- Add Species, Build, BigWig, BigBed and JBrowse track outputs to macs2 process
- Add scipy (v1.1.0) Python 3 package to resolwebio/utils Docker image

Changed

- **BACKWARD INCOMPATIBLE:** Drop support for Python 3.4 and 3.5
- **BACKWARD INCOMPATIBLE:** Require Resolwe 10.x
- **BACKWARD INCOMPATIBLE:** Upgrade to Django Channels 2
- **BACKWARD INCOMPATIBLE:** Count fragments (or templates) instead of reads by default in featureCounts process and BBDuk - STAR - featureCounts pipeline. The change applies only to paired-end data.
- **BACKWARD INCOMPATIBLE:** Use resolwebio/rnaseq:3.2.0 Docker image in the following processes that output reads:
  - upload-fastq-single

  - upload-fastq-paired
  - files-to-fastq-single
  - files-to-fastq-paired
  - reads-merge
  - bbduk-single
  - bbduk-paired
  - cutadapt-single
  - cutadapt-paired
  - cutadapt-custom-single
  - cutadapt-custom-paired
  - trimmomatic-single
  - trimmomatic-paired.

  This change unifies the version of FastQC tool (0.11.7) used for quality control of reads in the aforementioned processes. The new Docker image comes with an updated version of Cutadapt (1.16) which affects the following processes:

  - cutadapt-single
  - cutadapt-paired
  - cutadapt-custom-single
  - cutadapt-custom-paired.

  The new Docker image includes also an updated version of Trimmomatic (0.36) used in the following processes:

  - upload-fastq-single
  - upload-fastq-paired
  - files-to-fastq-single
  - files-to-fastq-paired
  - trimmomatic-single
  - trimmomatic-paired.

- **BACKWARD INCOMPATIBLE:** Change Docker image in alignment-subread from resolwebio/legacy:1.0.0 with Subread (v1.5.1) to resolwebio/rnaseq:3.2.0 with Subread (v1.6.0). --multiMapping option was added instead of --unique_reads. By default aligner report uniquely mapped reads only.
- Update wigToBigWig to kent-v365 version in resolwebio/chipseq Docker image
- Change paths in HTML amplicon report template in resolwebio/dnaseq Docker image
- Move assay type input in BBDuk - STAR - featureCounts pipeline descriptor schema to advanced options
- Use resolwebio/rnaseq:3.2.0 Docker image with updated versions of tools instead of resolwebio/legacy:1.0.0 Docker image in following processes:
  - alignment-bowtie with Bowtie (v1.2.2) instead of Bowtie (v1.1.2)
  - alignment-bowtie2 with Bowtie2 (v2.3.4.1) instead of Bowtie2 (v2.2.6)
  - alignment-tophat2 with TopHat (v2.1.1) instead of TopHat (v2.1.0)

  - alignment-bwa-mem, alignment-bwa-sw` and ``alignment-bwa-aln with BWA (v0.7.17-r1188) instead of BWA (v0.7.12-r1039)
  - `alignment-hisat2` with HISAT2 (v2.1.0) instead of HISAT2 (v2.0.3-beta)
  - `upload-genome`
- Use `resolwebio/base:ubuntu-18.04` Docker image as a base image in `resolwebio/utils` Docker image
- Update Python 3 packages in `resolwebio/utils` Docker image:
  - `numpy` (v1.14.4)
  - `pandas` (v0.23.0)
- Replace `bedgraphtobigwig` with `deepTools` in `resolwebio/rnaseq` Docker image, due to faster performance
- Use `resolwebio/rnaseq:3.3.0` Docker image in `alignment-star-index` with STAR (v2.5.4b)

Fixed

- Make management commands use a private random generator instance
- Fix output `covplot_html` of `coveragebed` process
- Fix process `archive-samples` and `amplicon-archive-multi-report` to correctly handle nested file paths
- Change `rose2` and `chipseq-peakscore` to work with `.bed` or `.bed.gz` input files
- Fix the `expression-aggregator` process so that it tracks the `species` of the input expression data
- Fix `bamtobigwig.sh` to use `deepTools` instead of `bedtools` with `bedgraphToBigWig` due to better time performance

1.5.27 9.0.0 - 2018-05-15

Changed

- **BACKWARD INCOMPATIBLE:** Simplify the `amplicon-report` process inputs by using Latex report template from the `resolwebio/latex` Docker image assets
- **BACKWARD INCOMPATIBLE:** Simplify the `coveragebed` process inputs by using Bokeh assets from the `resolwebio/dnaseq` Docker image
- **BACKWARD INCOMPATIBLE:** Require Resolwe 9.x
- Update `wigToBigWig` tool in `resolwebio/chipseq` Docker image
- Use `resolwebio/rnaseq:3.1.0` Docker image in the following processes:
  - `cufflinks`
  - `cuffnorm`
  - `cuffquant`
- Remove `differentialexpression-limma` process
- Use `resolwebio/rnaseq:3.1.0` docker image and expand error messages in:
  - `cuffdiff`

  - differentialexpression-deseq2
  - differentialexpression-edger
- Update workflow-bbduk-star-htseq
- Update quantseq descriptor schema
- Assert species and build in htseq-count-normalized process
- Set amplicon report template in resolwebio/latex Docker image to landscape mode

Added

- Support Python 3.6
- Add template_amplicon_report.tex to resolwebio/latex Docker image assets
- Add SnpEff tool and bokeh assets to resolwebio/dnaseq Docker image
- Add automated library strand detection to feature_counts quantification process
- Add FastQC option nogroup to bbduk-single and bbduk-paired processes
- Add CPM normalization to htseq-count-raw process
- Add workflow-bbduk-star-htseq-paired
- Add legend to amplicon report template in resolwebio/latex Docker image

Fixed

- Fix manual installation of packages in Docker images to handle dots and spaces in file names correctly
- Fix COSMIC url template in amplicon-table process
- Fix Create IGV session in Archive samples process
- Fix source tracking in cufflinks and cuffquant processes
- Fix amplicon master file validation script. Check and report error if duplicated amplicon names are included. Validation will now pass also for primer sequences in lowercase.
- Fix allele frequency (AF) calculation in snpeff process
- Fix bug in script for calculating FPKM. Because genes of raw counts from featureCounts were not lexicographically sorted, division of normalized counts was done with values from other, incorrect, genes. Results from featureCounts, but not HTSeq-count process, were affected.

1.5.28 8.1.0 - 2018-04-13

Changed

- Use the latest versions of the following Python packages in resolwebio/rnaseq docker image: Cutadapt 1.16, Apache Arrow 0.9.0, pysam 0.14.1, requests 2.18.4, appdirs 1.4.3, wrapt 1.10.11, PyYAML 3.12
- Bump tools version in resolwebio/rnaseq docker image:
  - Salmon to 0.9.1
  - FastQC to 0.11.7

- Generalize the no-extraction-needed use-case in resolwebio/base Docker image download_and_verify script

Added

- Add the following Python packages to resolwebio/rnaseq docker image: six 1.11.0, chardet 3.0.4, urllib3 1.22, idna 2.6, and certifi 2018.1.18
- Add edgeR R library to resolwebio/rnaseq docker image
- Add Bedtools to resolwebio/rnaseq docker image

Fixed

- Handle filenames with spaces in the following processes:
  - alignment-star-index
  - alignment-tophat2
  - cuffmerge
  - index-fasta-nucl
  - upload-fasta-nucl
- Fix COSMIC url template in (multisample) amplicon reports

1.5.29 8.0.0 - 2018-04-11

Changed

- **BACKWARD INCOMPATIBLE:** Refactor trimmomatic-single, trimmomatic-paired, bbduk-single, and bbduk-paired processes
- **BACKWARD INCOMPATIBLE:** Merge align-bwa-trim and align-bwa-trim2 process functionality. Retain only the refactored process under slug align-bwa-trim
- **BACKWARD INCOMPATIBLE:** In processes handling VCF files, the output VCF files are stored in bgzip-compressed form. Tabix index is not referenced to an original VCF file anymore, but stored in a separate tbi output field
- **BACKWARD INCOMPATIBLE:** Remove an obsolete workflow-accel-2 workflow
- **BACKWARD INCOMPATIBLE:** Use Elasticsearch version 5.x
- **BACKWARD INCOMPATIBLE:** Parallelize execution of the following processes:
  - alignment-bowtie2
  - alignment-bwa-mem
  - alignment-hisat2
  - alignment-star
  - alignment-tophat2
  - cuffdiff
  - cufflinks

  - cuffquant
- Require Resolwe 8.x
- Bump STAR aligner version in resolwebio/rnaseq docker image to 2.5.4b
- Bump Primerclip version in resolwebio/dnaseq docker image
- Use resolwebio/dnaseq Docker image in picard-pcrmetrics process
- Run vc-realign-recalibrate process using multiple cpu cores to optimize the processing time
- Use resolwebio/rnaseq Docker image in alignment-star process

Added

- Add CNVKit, LoFreq and GATK to resolwebio/dnaseq docker image
- Add BaseSpace files download tool
- Add process to import a file from BaseSpace
- Add process to convert files to single-end reads
- Add process to convert files to paired-end reads
- Add vc-gatk4-hc process which implements GATK4 HaplotypeCaller variant calling tool
- Add workflow-accel-gatk4 pipeline that uses GATK4 HaplotypeCaller as an alternative to GATK3 used in workflow-accel pipeline
- Add amplicon-master-file descriptor schema
- Add workflow-bbduk-star-featurecounts pipeline
- Add rna-seq-bbduk-star-featurecounts RNA-seq descriptor schema

Fixed

- Fix iterative trimming in bowtie and bowtie2 processes
- Fix archive-samples to use sample names for headers when merging expressions
- Improve goea.py tool to handle duplicated mapping results
- Handle filenames with spaces in the following processes:
  - alignment-hisat2
  - alignment-bowtie
  - prepare-geo-chipseq
  - prepare-geo-rnaseq
  - cufflinks
  - cuffquant

1.5.30 7.0.1 - 2018-03-27

Fixed

- Use name-ordered BAM file for counting reads in `HTSeq-count` process by default to avoid buffer overflow with large BAM files

1.5.31 7.0.0 - 2018-03-13

Changed

- **BACKWARD INCOMPATIBLE:** Remove Ubuntu 17.04 base Docker image since it has has reached its end of life and change all images to use the new ubuntu 17.10 base image
- **BACKWARD INCOMPATIBLE:** Require `species` and `build` inputs in the following processes:
  - `upload-genome`
  - `upload-gtf`
  - `upload-gff3`
  - `upload-bam`
  - `upload-bam-indexed`
- **BACKWARD INCOMPATIBLE:** Track `species` and `build` information in the following processes:
  - `cuffmerge`
  - alignment processes
  - variant calling processes
  - JBrowse processes
- **BACKWARD INCOMPATIBLE:** Track `species`, `build` and `feature_type` in the following processes:
  - `upload-expression-star`
  - quantification processes
  - differential expression processes
- **BACKWARD INCOMPATIBLE:** Track `species` in gene set (Venn) and `goenrichment` processes
- **BACKWARD INCOMPATIBLE:** Rename `genes_source` input to `source` in hierarchical clustering and PCA processes
- **BACKWARD INCOMPATIBLE:** Remove the following obsolete processes:
  - Dictyostelium-specific ncRNA quantification
  - `go-geneset`
  - bayseq differential expression
  - `cuffmerge-gtf-to-gff3`
  - `transdecoder`
  - `web-gtf-dictybase`
  - `upload-rmsk`
  - `snpdat`

- **BACKWARD INCOMPATIBLE:** Unify output fields of processes of type `data:annotation`
- **BACKWARD INCOMPATIBLE:** Rename the organism field names to species in `rna-seq` and `cutadapt-star-htseq` descriptor schemas
- **BACKWARD INCOMPATIBLE:** Rename the `genome_and_annotation` field name to `species` in `bcm-*` descriptor schemas and use the full species name for the `species` field values
- **BACKWARD INCOMPATIBLE:** Refactor `featureCounts` process
- **BACKWARD INCOMPATIBLE:** Change `import-sra` process to work with `resolwebio/utils` Docker image and refactor its inputs
- Require Resolwe 7.x
- Add environment export for Jenkins so that the manager will use a globally-unique channel name
- Set `scheduling_class` of gene and sample hierarchical clustering processes to `interactive`
- Change base Docker images of `resolwebio/rnaseq` and `resolwebio/dnaseq` to `resolwebio/base:ubuntu-18.04`
- Use the latest versions of the following Python packages in `resolwebio/rnaseq` Docker image: Cutadapt 1.15, Apache Arrow 0.8.0, pysam 0.13, and xopen 0.3.2
- Use the latest versions of the following Python packages in `resolwebio/dnaseq` Docker image: Bokeh 0.12.13, pandas 0.22.0, Matplotlib 2.1.2, six 1.11.0, PyYAML 3.12, Jinja2 2.10, NumPy 1.14.0, Tornado 4.5.3, and pytz 2017.3
- Use the latest version of `wigToBigWig` tool in `resolwebio/chipseq` Docker image
- Use `resolwebio/rnaseq:3.0.0` Docker image in `goenrichment`, `upload-gaf` and `upload-obo` processes
- Use `resolwebio/dnaseq:3.0.0` Docker image in `filtering_chemut` process
- Change `cuffnorm` process type to `data:cuffnorm`
- Set type of `coverage-garvan` process to `data:exomecoverage`
- Remove `gsize` input from `macs14` process and automate genome size selection
- Adjust `bam-split` process so it can be included in workflows
- Make ID attribute labels in `featureCounts` more informative
- Change 'source' to 'gene ID database' in labes and descriptions
- Change `archive-samples` process to create different IGV session files for `build` and `species`
- Expose advanced parameters in Chemical Mutagenesis workflow
- Clarify some descriptions in the `filtering_chemut` process and `chemut` workflow
- Change expected genome build formatting for hybrid genomes in `bam-split` process
- Set the `cooksCutoff` parameter to `FALSE` in `deseq.R` tool
- Rename 'Expressions (BCM)' to 'Dicty expressions'

Added

- Mechanism to override the manager's control channel prefix from the environment
- Add Ubuntu 17.10 and Ubuntu 18.04 base Docker images
- Add `resolwebio/utils` Docker image

- Add BBMap, Trimmomatic, Subread, Salmon, and dexseq_prepare_annotation2 tools and DEXSeq and loadSubread R libraries to resolwebio/rnaseq Docker image
- Add abstract processes that ensure that all processes that inherit from them have the input and output fields that are defined in them:
  - abstract-alignment
  - abstract-annotation
  - abstract-expression
  - abstract-differentialexpression
  - abstract-bed
- Add miRNA workflow
- Add prepare-geo-chipseq and prepare-geo-rnaseq processes that produce a tarball with necessary data and folder structure for GEO upload
- Add library-strandedness process which uses the Salmon tool built-in functionality to detect the library strandedness information
- Add species and genome build output fields to macs14 process
- Expose additional parameters in alignment-star, cutadapt-single and cutadapt-paired processes
- Add merge expressions to archive-samples process
- Add description of batch mode to Expression aggregator process
- Add error and warning messages to the cuffnorm process
- Add optional species input to hierarchical clustering and PCA processes
- Add Rattus norvegicus species choice to the rna-seq descriptor schema to allow running RNA-seq workflow for this species from the Recipes

Fixed

- Fix custom argument passing script for Trimmomatic in resolwebio/rnaseq Docker image
- Fix installation errors for dexseq-prepare-annotation2 in resolwebio/rnaseq Docker image
- Fix consensus_subreads input option in Subread process
- Limit variant-calling process in the chemical mutagenesis workflow and the Picard tools run inside to 16 GB of memory to prevent them from crashing because they try to use too much memory
- The chemical mutagenesis workflow was erroneously categorized as data:workflow:rnaseq:cuffquant type. This is switched to data:workflow:chemut type.
- Fix handling of NA values in Differential expression results table. NA values were incorrectly replaced with value 0 instead of 1
- Fix cuffnorm process to work with samples containing dashes in their name and dispense prefixing sample names starting with numbers with 'X' in the cuffnorm normalization outputs
- Fix cuffnorm process' outputs to correctly track species and build information
- Fix typos and sync parameter description common to featureCounts and miRNA workflow

1.5.32 6.2.2 - 2018-02-21

Fixed

- Fix cuffnorm process to correctly use sample names as labels in output files and expand cuffnorm tests

1.5.33 6.2.1 - 2018-01-28

Changed

- Update description text of cutadapt-star-htseq descriptor schema to better describe the difference between gene/transcript-type analyses
- Speed-up management command for inserting mappings

1.5.34 6.2.0 - 2018-01-17

Added

- Add R, tabix, and CheMut R library to resolwebio/dnaseq Docker image
- Add SRA Toolkit to resolwebio/rnaseq Docker image

Changed

- Require Resolwe 6.x
- Extend pathway map with species and source field
- Move template and logo for multi-sample report into resolwebio/latex Docker image
- Refactor amplicon-report process to contain all relevant inputs for amplicon-archive-multi-report
- Refactor amplicon-archive-multi-report
- Use resolwebio/dnaseq:1.2.0 Docker image in filtering_chemut process

Fixed

- Enable DEBUG setting in tests using Django's LiveServerTestCase
- Wait for ElasticSeach to index the data in KBBioProcessTestCase
- Remove unused parameters in TopHat (2.0.13) process and Chip-seq workflow

1.5.35 6.1.0 - 2017-12-12

Added

- Add amplicon-archive-multi-report process
- Add upload-metabolic-pathway process
- Add memory-optimized primerclip as a separate align-bwa-trim2 process

- Add workflow-accel-2 workflow

Changed

- Improve PCA process performance
- Use resolwebio/chipseq:1.1.0 Docker image in macs14 process
- Change formatting of EFF[*].AA column in snpeff process
- Save unmapped reads in aligment-hisat2 process
- Turn off test profiling

Fixed

- Fix pre-sorting in upload-master-file process
- Revert align-bwa-trim process to use non-memory-optimized primerclip
- Fix file processing in cutadapt-custom-paired process

1.5.36 6.0.0 - 2017-11-28

Added

- Add AF filter to amplicon report
- Add number of samples to the output of expression aggregator
- Add ChIP-Rx, ChIPmentation and eClIP experiment types to reads descriptor schema
- Add pandas Python package to resolwebio/latex Docker image
- Add primerclip, samtools, picard-tools and bwa to resolwebio/dnaseq Docker image
- Add cufflinks, RNASeqT R library, pyarrow and sklearn Python packages to resolwebio/rnaseq Docker image
- Add wigToBigWig tool to resolwebio/chipseq Docker image

Changed

- **BACKWARD INCOMPATIBLE:** Drop Python 2 support, require Python 3.4 or 3.5
- **BACKWARD INCOMPATIBLE:** Make species part of the feature primary key
- **BACKWARD INCOMPATIBLE:** Substitute Python 2 with Python 3 in resolwebio/rnaseq Docker image. The processes to be updated to this version of the Docker image should also have their Python scripts updated to Python 3.
- Require Resolwe 5.x
- Set maximum RAM requirement in bbduk process
- Move *Assay type* input parameter in RNA-Seq descriptor schema from advanced options to regular options
- Use resolwebio/rnaseq Docker image in Cutadapt processes
- Use additional adapter trimming option in cutadapt-custom-single/paired processes

- Show antibody information in `reads` descriptor for `ChIP-Seq`, `ChIPmentation`, `ChIP-Rx`, `eClIP`, `MNase-Seq`, `MeDIP-Seq`, `RIP-Seq` and `ChIA-PET` experiment types
- Use `resolwebio/dnaseq` Docker image in `align-bwa-trim` process
- Refactor `resolwebio/chipseq` Docker image
- Use Resolwe's Test Runner for running tests and add ability to only run a partial test suite based on what proceses have Changed
- Configure Jenkins to only run a partial test suite when testing a pull request
- Make tests use the live Resolwe API host instead of external server

Fixed

- Fix merging multiple expressions in DESeq process
- Fix `resolwebio/rnaseq` Docker image's README
- Handle multiple ALT values in amplicon report
- Fix BAM file input in `rsem` process

1.5.37 5.0.1 - 2017-11-14

Fixed

- Update Features and Mappings ElasticSearch indices building to be compatible with Resolwe 4.0

1.5.38 5.0.0 - 2017-10-25

Added

- Add automatic headers extractor to `bam-split` process
- Add HTML amplicon plot in `coveragebed` process
- Add raw RSEM tool output to *rsem* process output
- Add support for transcript-level differential expression in `deseq2` process

Changed

- **BACKWARD INCOMPATIBLE:** Bump Django requirement to version 1.11.x
- **BACKWARD INCOMPATIBLE:** Make `BioProcessTestCase` non-transactional
- Require Resolwe 4.x
- Add the advanced options checkbox to the `rna-seq` descriptor schema
- Remove static amplicon plot from `coveragebed` and `amplicon-report` processes
- Update Dockerfile for `resolwebio/latex` with newer syntax and add some additional Python packages

1.5.39 4.2.0 - 2017-10-05

Added

- Add `resolwebio/base` Docker image based on Ubuntu 17.04
- Add `resolwebio/dnaseq` Docker image
- Add `DESeq2` tool to `resolwebio/rnaseq` docker image
- Add input filename regex validator for `upload-master-file` process

Changed

- Remove obsolete mongokey escape functionality
- Report novel splice-site junctions in HISAT2
- Use the latest stable versions of the following bioinformatics tools in `resolwebio/rnaseq` docker image: Cutadapt 1.14, FastQC 0.11.5, HTSeq 0.9.1, and SAMtools 1.5

1.5.40 4.1.0 - 2017-09-22

Added

- Add Mus musculus to all BCM workflows' schemas
- Add `bam-split` process with supporting processes `upload-bam-primary`, `upload-bam-secondary` and `upload-header-sam`

Changed

- Enable Chemut workflow and process tests

Fixed

- Fix chemut `intervals` input option

1.5.41 4.0.0 - 2017-09-14

Added

- New base and legacy Docker images for processes, which support non-root execution as implemented by Resolwe

Changed

- **BACKWARD INCOMPATIBLE:** Modify all processes to explicitly use the new Docker images
- **BACKWARD INCOMPATIBLE:** Remove `clustering-hierarchical-genes-etc` process
- Require Resolwe 3.x

1.5.42 3.2.0 2017-09-13

Added

- Add index-fasta-nucl and rsem process
- Add custom Cutadapt - STAR - RSEM workflow

1.5.43 3.1.0 2017-09-13

Added

- Add statistics of logarithmized expressions to expression-aggregator
- Add input field description to cutadapt-star-htseq descriptor schema
- Add HISAT2 and RSEM tool to resolwebio/rnaseq docker image

Changed

- Remove eXpress tool from resolwebio/rnaseq docker image
- Use system packages of RNA-seq tools in resolwebio/rnaseq docker image
- Set hisat2 process' memory resource requirement to 32GB
- Use resolwebio/rnaseq docker image in hisat2 process

1.5.44 3.0.0 2017-09-07

Added

- Add custom Cutadapt - STAR - HT-seq workflow
- Add expression aggregator process
- Add resolwebio/rnaseq docker image
- Add resolwebio/latex docker image
- Add access to sample field of data objects in processes via sample filter

Changed

- **BACKWARD INCOMPATIBLE:** Remove threads input in STAR aligner process and replace it with the cores resources requirement
- **BACKWARD INCOMPATIBLE:** Allow upload of custom amplicon master files (make changes to amplicon-panel descriptor schema, upload-master-file and amplicon-report processes and workflow-accel workflow)
- **BACKWARD INCOMPATIBLE:** Remove threads input in cuffnorm process and replace it with the cores resources requirement
- Add sample descriptor to prepare_expression test function
- Prettify amplicon report

Fixed

- Fix `upload-expression-star` process to work with arbitrary file names
- Fix STAR aligner to work with arbitrary file names
- Fix `cuffnorm` group analysis to work correctly
- Do not crop Amplicon report title as this may result in malformed LaTeX command
- Escape LaTeX's special characters in `make_report.py` tool
- Fix validation error in `Test sleep progress` process

1.5.45 2.0.0 2017-08-25

Added

- Support bioinformatics process test case based on Resolwe's `TransactionProcessTestCase`
- Custom version of Resolwe's `with_resolwe_host` test decorator which skips the decorated tests on non-Linux systems
- Add optimal leaf ordering and simulated annealing to gene and sample hierarchical clustering
- Add `resolwebio/chipseq` docker image and use it in ChIP-Seq processes
- Add Odocoileus virginianus texanus (deer) organism to sample descriptor
- Add test for `import-sra` process
- Add RNA-seq DSS test
- Add Cutadapt and custom Cutadapt processes

Changed

- Require Resolwe 2.0.x
- Update processes to support new input sanitization introduced in Resolwe 2.0.0
- Improve variant table name in amplicon report
- Prepend `api/` to all URL patterns in the Django test project
- Set `hisat2` process' memory resource requirement to 16GB and cores resource requirement to 1
- Filter LoFreq output VCF files to remove overlapping indels
- Add *Non-canonical splice sites penalty*, *Disallow soft clipping* and *Report alignments tailored specifically for Cufflinks* parameters to `hisat2` process
- Remove `threads` input from `cuffquant` and `rna-seq` workfows
- Set core resource requirement in `cuffquant` process to 1

Fixed

- Correctly handle paired-end parameters in `featureCount`
- Fix `NaN` in explained variance in PCA. When PC1 alone explained more than 99% of variance, explained variance for PC2 was not returned

- Fix input sanitization error in `dss-rna-seq` process
- Fix gene source check in hierarchical clustering and PCA
- Enable network access for all import processes
- Fix RNA-seq DSS adapters bug
- Fix sample hierarchical clustering output for a single sample case

1.5.46 1.4.1 2017-07-20

Changed

- Optionally report all amplicons in Amplicon table

Fixed

- Remove remaining references to calling `pip` with `--process-dependency-links` argument

1.5.47 1.4.0 2017-07-04

Added

- Amplicon workflow
- Amplicon descriptor schemas
- Amplicon report generator
- Add Rattus norvegicus organism choice to sample schema
- Transforming form Phred 64 to Phred 33 when uploading fastq reads
- Add primertrim process
- RNA-Seq experiment descriptor schema
- iCount sample and reads descriptor schemas
- iCount demultiplexing and sample annotation
- ICount QC
- Add MM8, RN4 and RN6 options to rose2 process
- Add RN4 and RN6 options to bamplot process
- Archive-samples process
- Add bamliquidator
- CheMut workflow
- Dicty primary analysis descriptor schema
- IGV session to Archive-samples process
- Use Resolwe’s field projection mixins for knowledge base endpoints
- `amplicon-table` process
- Add C. griseus organism choice to Sample descriptor schema

- Add S. tuberosum organism choice to Sample descriptor schema
- Add log2 to gene and sample hierarchical clustering
- Add new inputs to import SRA, add read type selection process
- Set memory resource requirement in jbrowse annotation gff3 and gtf processes to 16GB
- Set memory resource requirement in star alignment and index processes to 32GB
- Add C. elegans organism choice to Sample descriptor schema
- Add D. melanogaster organism choice to Sample descriptor schema
- Set core resource requirement in Bowtie process to 1
- Set memory resource requirement in amplicon BWA trim process to 32GB
- Add new master file choices to amplicon panel descriptor schema
- Add S. tuberosum organism choice to RNA-seq workflow
- Add Cutadapt process
- Add leaf ordering to gene and sample hierarchical clustering

Fixed

- Use new import paths in `resolwe.flow`
- Upload reads (paired/single) containing whitespace in the file name
- Fix reads filtering processes for cases where input read file names contain whitespace
- Add additional filtering option to STAR aligner
- Fix bbduk-star-htseq_count workflow
- Fix cuffnorm process: Use sample names as labels (boxplot, tables), remove group labels input, auto assign group labels, add outputs for Rscript output files which were only available compressed
- Derive output filenames in hisat2 from the first reads filename
- Correctly fetch KB features in `goea.py`
- Append JBrowse tracks to sample
- Replace the BAM MD tag in *align-bwa-trim* process to correct for an issue with the primerclip tool
- Fix typo in trimmomatic and bbduk processes
- Use re-import in *etc* and *hmmer_database* processes

Changed

- Support Resolwe test framework
- Run tests in parallel with Tox
- Use Resolwe's new `FLOW_DOCKER_COMMAND` setting in test project
- Always run Tox's `docs`, `linters` and `packaging` environments with Python 3
- Add `extra` Tox testing environment with a check that there are no large test files in `resolwe_bio/tests/files`

- Replace Travis CI with Genialis' Jenkins for running the tests
- Store compressed and uncompressed .fasta files in `data:genome:fasta` objects
- Change sample_geo descriptor schema to have strain option available for all organisms
- More readable rna-seq-quantseq schema, field stranded
- Remove obsolete Gene Info processes
- Change log2(fc) default from 2 to 1 in diffexp descriptor schema
- Change Efective genome size values to actual values in macs14 process
- Change variable names in bowtie processes
- Remove iClip processes, tools, files and tests

1.5.48 1.3.0 2017-01-28

Changed

- Add option to save expression JSON to file before saving it to Storage
- Update `upload-expression` process
- No longer treat `resolwe_bio/tools` as a Python package
- Move processes' test files to the `resolwe_bio/tests/files` directory to generalize and simplify handling of tests' files
- Update differential expression (DE) processors
- Update `generate_diffexpr_cuffdiff` django-admin command
- Save gene_id source to `output.source` for DE, expression and related objects
- Refactor `upload-diffexp` processor
- Update sample descriptor schema
- Remove obsolete descriptor schemas
- Add stitch parameter to rose2 processor
- Add filtering to DESeq2
- Set Docker Compose's project name to `resolwebio` to avoid name clashes
- GO enrichment analysis: map features using gene Knowledge base
- Add option to upload .gff v2 files with upload-gtf processor
- Replace Haystack with Resolwe Elastic Search API
- Require Resolwe 1.4.1+
- Update processes to be compatible with Resolwe 1.4.0

Added

- Process definition documentation style and text improvements
- Add `resolwe_bio.kb` app, Resolwe Bioinformatics Knowledge Base
- Add tests to ensure generators produce the same results

- Upload Gene sets (`data:geneset`)
- Add `generate_geneset` django-admin command
- Add `generate_diffexpr_deseq` django-admin command
- Add 'Generate GO gene sets' processor
- Add generic file upload processors
- Add upload processor for common image file types (.jpg/.tiff/.png/.gif)
- Add upload processor for tabular file formats (.tab/.tsv/.csv/.txt/.xls/.xlsx)
- Add Trimmomatic process
- Add featureCounts process
- Add Subread process
- Add process for hierarchical clusteing of samples
- Add gff3 to gtf file converter
- Add microarray data descriptor schema
- Add process for differential expression edgeR
- `BioCollectionFilter` and `BidDataFilter` to support filtering collections and data by samples on API
- Added processes for automatically downloading single and paired end SRA files from NCBI and converting them to FASTQ
- Added process for automatically downloading SRA files from NCBI and converting them to FASTQ
- Add HEAT-Seq pipeline tools
- Add HEAT-Seq workflow
- Add `create-geneset`, `create-geneset-venn` processors
- Add `source` filter to feature search endpoint
- Add bamplot process
- Add gene hiererhical clustering
- Add cuffquant workflow
- Support Django 1.10 and versionfield 0.5.0
- django-admin commands `insert_features` and `insert_mappings` for importing features and mappings to the Knowledge Base
- Add bsmap and mcall to analyse WGBS data
- Vaccinesurvey sample descriptor schema
- Add RNA-Seq single and paired-end workflow

Fixed

- Set `presample` to `False` for Samples created on Sample endpoint
- Fix FastQC report paths in processors
- Fix `htseq_count` and `featureCounts` for large files
- Fix `upload gtf annotation`

- Fix gene_id field type for differential expression storage objects
- Order data objects in `SampleViewSet`
- Fix sample hiererhical clustering
- Fix name in gff to gtf process
- Fix clustering to read expressed genes as strings
- Fix protocol labels in `rna-seq-quantseq` descriptor schema

1.5.49 1.2.1 2016-07-27

Changed

- Update `resolwe` requirement

1.5.50 1.2.0 2016-07-27

Changed

- Decorate all tests that currently fail on Docker with `skipDockerFailure`
- Require Resolwe's `master` git branch
- Put packaging tests in a separate Tox testing environment
- Rename DB user in test project
- Change PostgreSQL port in test project
- Add ROSE2 results parser
- Compute index for HISAT2 aligner on genome upload
- Updated Cuffquant/Cuffnorm tools
- Change ROSE2 enhancer rank plot labels
- Refactor processor syntax
- Move processes tests into `processes` subdirectory
- Split `sample` API endpoint to `sample` for annotated `Samples` and `presample` for unannotated `Samples`
- Rename test project's data and upload directories to `.test_data` and `.test_upload`
- Save fastq files to `lists:basic:file` field. Refactor related processors.
- Reference genome-index path when running aligners.
- Add pre-computed genome-index files when uploading reference fasta file.
- Include all necessary files for running the tests in source distribution
- Exclude tests from built/installed version of the package
- Move testing utilities from `resolwe_bio.tests.processes.utils` to `resolwe_bio.utils.test`
- Update Cuffdiff processor inputs and results table parsing
- Refactor processes to use the updated `resolwe.flow.executors.run` command

- Refactor STAR aligner - export expressions as separate objects

Fixed

- Make Tox configuration more robust to different developer environments
- Set `required: false` in processor input/output fields where necessary
- Add `Sample`'s `Data objects` to `Collection` when `Sample` is added
- Fixed/renamed Cufflinks processor field names

Added

- `skipDockerFailure` test decorator
- Expand documentation on running tests
- Use Travis CI to run the tests
- Add `Sample` model and corresponding viewset and filter
- Add docker-compose command for PostgreSQL
- API endpoint for adding `Samples` to `Collections`
- HISAT2 aligner
- Use Git Large File Storage (LFS) for large test files
- Test for `generate_samples` django-admin command
- django-admin command: `generate_diffexpr`

1.5.51 1.1.0 2016-04-18

Changed

- Remove obsolete utilities superseded by resolwe-runtime-utils
- Require Resolwe 1.1.0

Fixed

- Update sample descriptor schema
- Include all source files and supplementary package data in sdist

Added

- `flow_collection: sample` to processes
- MACS14 processor
- Initial Tox configuration for running the tests
- Tox tests for ensuring high-quality Python packaging
- ROSE2 processor

- django-admin command: `generate_samples`

1.5.52 1.0.0 2016-03-31

Changed

- Renamed assertFileExist to assertFileExists
- Restructured processes folder hierarchy
- Removed re-require and hard-coded tools' paths

Fixed

- Different line endings are correctly handled when opening gzipped files
- Fail gracefully if the field does not exist in assertFileExists
- Enabled processor tests (GO, Expression, Variant Calling)
- Enabled processor test (Upload reads with old Illumina QC encoding)
- Made Resolwe Bioinformatics work with Resolwe and Docker

Added

- Import expressions from tranSMART
- Limma differential expression (tranSMART)
- VC filtering tool (Chemical mutagenesis)
- Additional analysis options to Abyss assembler
- API endpoint for Sample
- Initial documentation

1.6 Contributing

1.6.1 Installing prerequisites

Make sure you have Python 3.6 installed on your system. If you don't have it yet, follow these instructions.

Resolwe Bioinformatics requires PostgreSQL (9.4+). Many Linux distributions already include the required version of PostgreSQL (e.g. Fedora 22+, Debian 8+, Ubuntu 15.04+) and you can simply install it via distribution's package manager. Otherwise, follow these instructions.

The pip tool will install all Resolwe Bioinformatics' dependencies from PyPI. Installing some (indirect) dependencies from PyPI will require having a C compiler (e.g. GCC) as well as Python development files installed on the system.

Note: The preferred way to install the C compiler and Python development files is to use your distribution's packages, if they exist. For example, on a Fedora/RHEL-based system, that would mean installing `gcc` and `python3-devel` packages.

Optional prerequisites

If you want to run or develop tests with large input or output files, then install the Git Large File Storage extension.

1.6.2 Preparing environment

Fork the main Resolwe Bioinformatics' git repository.

If you don't have Git installed on your system, follow these instructions.

Clone your fork (replace `<username>` with your GitHub account name) and change directory:

```
git clone https://github.com/<username>/resolwe-bio.git
cd resolwe-bio
```

Prepare Resolwe Bioinformatics for development:

```
pip install --pre -e .[docs,package,test]
```

Note: We recommend using pyvenv to create an isolated Python environment for Resolwe Bioinformatics.

1.6.3 Preparing database

Add a postgres user:

```
createuser -s -r postgres
```

1.6.4 Running tests

Manually

Change directory to the `tests` Django project:

```
cd tests
```

Run docker:

```
docker-compose up
```

Note: On Mac or Windows, Docker might complain about non-mounted volumes. You can edit volumes in *Docker => Preferences => File Sharing* The following volumes need to be shared:

- /private
- /tmp
- /var/folders

`/private` is shared by default. When you attempt to add `/var/folders` it might try to add `/private/var/folders` which will cause Docker complaining about overlapping volumes. Here's a workaround: Change `/private` to `/var/folders` and then add `/private` again.

To run the tests, use:

```
./manage.py test resolwe_bio --parallel 2
```

Note: If you don't specify the number of parallel test processes (i.e. you just use `--parallel`), Django will run one test process per each core available on the machine.

Warning: If you run Docker in a virtual machine (i.e. if you use MacOS or Windows) rather that directly on your machine, the virtual machine can become totally unresponsive if you set the number of parallel test processes too high. We recommend using at most `--parallel 2` in such cases.

To run a specific test, use:

```
./manage.py test resolwe_bio.tests.<module-name>.<class-name>.<method-name>
```

For example, to run the `test_macs14` test of the `ChipSeqProcessorTestCase` class in the `test_chipseq` module, use:

```
./manage.py test resolwe_bio.tests.processes.test_chipseq.ChipSeqProcessorTestCase.
↪test_macs14
```

Using Tox

To run the tests with Tox, use:

```
tox
```

To re-create the virtual environment before running the tests, use:

```
tox -r
```

To only run the tests with a specific Python version, use:

```
tox -e py<python-version>
```

For example, to only run the tests with Python 3.5, use

```
tox -e py35
```

Note: To see the list of available Python versions, see `tox.ini`.

Note: To control the number of test processes Django will run in parallel, set the `DJANGO_TEST_PROCESSES` environment variable.

Since running tests for all processes may take a long time, there is an option to run partial tests based on what files have been changed between HEAD and a specific commit (e.g. master). The Tox environments that run partial tests have the `-partial` suffix, e.g.:

```
tox -e py35-partial
```

To configure the commit against which the changes are compared you should set the `RESOLWE_TEST_ONLY_CHANGES_TO` environmental variable (it is set to master by default).

Running tests skipped on Docker

To run the tests that are skipped on Docker due to failures and errors, set the `RESOLWEBIO_TESTS_SKIP_DOCKER_FAILURES` environment variable to `no`.

For example, to run the skipped tests during a single test run, use:

```
RESOLWEBIO_TESTS_SKIP_DOCKER_FAILURES=no ./manage.py test resolwe_bio
```

To run the skipped tests for the whole terminal session, execute:

```
export RESOLWEBIO_TESTS_SKIP_DOCKER_FAILURES=no
```

and then run the tests as usual.

Running tests with large files

To run the tests with large input or output files, ensure you have the Git Large File Storage extension installed and run the tests as usual.

Adding tests with large files

If a test file is larger than 1 MiB, then put it in the `resolwe_bio/tests/files/large/` directory. Git Large File Storage (LFS) extension will automatically pick it up and treat it appropriately.

To ensure contributors without Git LFS or users using the source distribution can smoothly run the tests, decorate the tests using large files with the following:

```
@skipUnlessLargeFiles(<large-file1>, <large-file2>, ...)
```

where `<large-file1>`, `<large-file2>`, ... represent the names of large files used inside a particular test.

The decorator will ensure the test is skipped unless these files are present and represent real large files (not just Git LFS pointers).

1.6.5 Building documentation

```
python setup.py build_sphinx
```

Note: To build the documentation, you must use Python 3 (Python 2 is not supported).

1.6.6 Preparing release

Follow Resolwe's documentation on preparing a release. Resolwe code is automatically released to PyPI when tagged, but this is not supported in Resolwe Bioinformatics yet. After you have completed the first part, follow the steps below to release the code on PyPI.

Clean `build` directory:

```
python setup.py clean -a
```

Remove previous distributions in `dist` directory:

```
rm dist/*
```

Remove previous `egg-info` directory:

```
rm -r *.egg-info
```

Create source distribution:

```
python setup.py sdist
```

Build wheel:

```
python setup.py bdist_wheel
```

Upload distribution to PyPI:

```
twine upload dist/*
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index