SEQprocess

June 25, 2018

Type Package

Title R-based pipelines for NGS data processing

Version 0.99.0

Date 2018-1-3

Author Hyun Goo Woo

Maintainer Hyun Goo Woo <HyunGooWoo@gmail.com>

Description R-based pipelines for NGS data processing

License GPL2

biocViews Alignment, Annotation, Preprocessing, QualityControl, GenomeAssembly

RoxygenNote 6.0.1

Suggests knitr, BiocStyle

Depends R (>= 3.3.2)

Imports parallel, rmarkdown, GenomicRanges, limma, R.utils, Biobase, SummarizedExperiment

Encoding UTF-8

SEQprocess

June 25, 2018

R topics documented:

bowtie2 . . . 2
build.star.idx . . . 3
bwa . . . 4
cufflinks . . . 5
cutadapt . . . 6
eset2SE . . . 6
fastQC . . . 7
gatk.baserecalibrator . . . 8
gatk.combinevariants . . . 9
gatk.depthOFcoverage . . . 10
gatk.haplotypecaller . . . 10
gatk.mutect2.normal . . . 11
gatk.printreads . . . 12
gatk.realign . . . 13
gatk.targetcreator . . . 14
gatk.variantfilter . . . 15
generate.GC . . . 16
get.annovar.report . . . 17
get.collectmetrics.report . . . 17
get.fns . . . 18
get.proc.report . . . 18
get.process.names . . . 19
get.qc.report . . . 19
get.Robject.report . . . 20
get.single_end.metrics.report . . . 20
get.tophat.report . . . 21
get.trim.gal.report . . . 21
get.trim_cut.report . . . 22
gff2gr . . . 22
gtf2gr . . . 23
htseq.add.info . . . 23
htseq_count . . . 24
make.cset . . . 25
make.eset . . . 25
make.vset . . . 26
multiple.reads.pileup . . . 26
MuSE.call . . . 27

MuSE.sump . . . 28
Muse.tabix2.vcf . . . 29
mut.type.somatic . . . 29
mutect2 . . . 30
picard.addrg . . . 31
picard.collectmetrics . . . 32
picard.reorder . . . 33
picard.rmdu . . . 34
pileup2seqz . . . 35
ploidyNcellularity . . . 35
print_message . . . 36
processSomatic . . . 36
qc_aggregate . . . 37
read.pileup.gz . . . 38
refGene2gr . . . 38
report . . . 39
SEQprocess . . . 39
seqz2rda . . . 41
seqz2seg . . . 41
somaticsniper . . . 42
STAR . . . 43
table.annovar . . . 45
tophat2 . . . 46
trim.gal . . . 47
varscan . . . 48
vcf2annovar . . . 49
vcf2gz . . . 50
vep . . . 50
vset.preprocess . . . 51

Index **52**

`bowtie2` *bowtie2*

Description

A wrapper function to run bowtie2.

Usage

```
bowtie2(fq1, output.dir, sample.name, bowtie.idx, mc.cores=1, run.cmd=TRUE)
```

Arguments

`fq1`	Path to read1 fastq files
`output.dir`	Output directory
`sample.name`	A character vector for the sample names
`bowtie.idx`	Path to bowtie index files
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
`run.cmd`	Whether to execute the command line (default=TRUE)

Details

Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. Bowtie2 is used only for single-end sequencing data.

Value

Aligned SAM files

References

Fast gapped-read alignment with Bowtie 2

See Also

http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

---

build.star.idx *STAR*

Description

A wrapper function to run STAR (runMode genomeGenerate)

Usage

```
build.star.idx(star.idx.dir=file.path(reference.dir, "STAR.idx"), sample.name, a
```

Arguments

star.idx.dir	Directory of STAR index files
sample.name	A character vector for the sample names
ref.fa	Reference fasta file path
ref.gtf	Reference gtf file path (e.g., gencode.gtf)
sjdbOverhang	A parameter value for the –sjdbOverhang in STAR. Length of the donor/acceptor sequence on each side of the junctions, ideally=(mate_length-1) (default=100)
fasta.idx	Indexing reference fasta file (when first indexing => TRUE)
SJ.idx	Indexing splicing junction (when second indexing => TRUE)
run.cmd	Whether to execute the command line (default=TRUE)
star_thread_number	A parameter value for –runThreadN in STAR. A numeric value of the number of threads (default: 8)

Details

Indexing reference fasta file and splicing junction site from fastq files.

Value

STAR reference and splicing junction indexing

References

STAR: ultrafast universal RNA-seq aligner

See Also

{https://github.com/alexdobin/STAR

`bwa` *bwa*

Description

A wrapper function to run BWA.

Usage

```
bwa(bwa.method, fq1, fq2, output.dir, sample.name, ref.fa, bwa.idx, bwa_thread_n
```

Arguments

`bwa.method`	bwa algorithms of mem and aln can be used(mem: for paired-end data, aln: for single-end data)
`fq1`	Path to read1 fastq files
`fq2`	Path to read2 fastq files (bwa-mem only)
`output.dir`	Output directory
`sample.name`	A character vector for the sample names
`ref.fa`	Path to reference fasta file
`bwa.idx`	Path to bwa index files
`bwa.thread.number`	A parameter value for -t in BWA. A numeric value of the number of threads (default: 4)
`run.cmd`	Whether to execute the command line (default=TRUE)
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. "bwa" can be run with option either of BWA-mem or BWA-aln.

Value

Aligned BAM files

References

Fast and accurate short read alignment with Burrows–Wheeler transform

See Also

http://bio-bwa.sourceforge.net/bwa.html

cufflinks *cufflinks*

Description

A wrapper function to run Cufflinks for mRNA quantitation

Usage

```
cufflinks(fns.bam, sample.name, output.dir, cufflinks_thread_number=4, cufflinks
```

Arguments

fns.bam	Path to bam files
output.dir	Output directory
sample.name	A character vector for the sample names
cufflinks.thread.number	A parameter value for -p in Cufflinks. A numeric value of the number of threads (default: 4)
cufflinks.gtf	If you set -G, Output will not include novel genes and isoforms that are assembled. (default: -G)
ref.gtf	Path to reference gtf file
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Cufflinks algorithms for transcript assembly and expression quantification are much more accurate with paired-end reads.

Value

mRNA quantification text files

References

http://cole-trapnell-lab.github.io/cufflinks/papers/

See Also

http://cole-trapnell-lab.github.io/cufflinks/

`cutadapt` *cutadapt*

Description

A wrapper function to run Cutadapt

Usage

```
cutadpat(fq1, output.dir, adpat.seq="TGGAATTCTCGGGTGCCAAGG", m=1, mc.cores=1, ru
```

Arguments

`fq1`	Path to fastq files
`output.dir`	Output directory
`adapt.seq`	A parameter value for -b in cutadapt. Adapter sequences user wants to remove
`m`	A parameter value for -m in cutadapt. Discards processed reads that are shorter than m option. Reads that are too short before adapter removal are also discarded.
`run.cmd`	Whether to execute the command line (default=TRUE)
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Cutadapt finds and removes adapter sequences, primers, poly-A tails and other types of unwanted sequence.

References

Cutadapt Removes Adapter Sequences from High-Throughput Sequencing Reads

See Also

https://cutadapt.readthedocs.io/en/stable/

`eset2SE` *eset2SE*

Description

Convert ExpressionSet to SummarizedExperiment

Usage

```
eset2SE(eset = NULL, vset = NULL, cset = NULL, Robject.dir)
```

Arguments

eset	RNA abundance ExpressionSet
vset	Variant ExpressionSet
cset	CNV ExpressionSet
Robject.dir	Output directory

Details

SEQprocess also provides SummarizedExperiment data format for comportable data analysis and management

---

fastQC *fastQC*

Description

A wrapper function to run fastQC

Usage

```
fastqc(fq1, fq2, output.dir, run.cmd=TRUE)
```

Arguments

fq1	Path to read1 fastq files
fq2	Path to read2 fastq files
output.dir	Output directory
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

FastQC aims to provide a QC report that detects problems originating from either the sequencer or the starting library material.

Value

Quality check report for sequence data. (e.g., .html)

References

FastQC: a quality control tool for high throughput sequence data. Andrews S. (2010).

See Also

http://www.bioinformatics.babraham.ac.uk/projects/fastqc

gatk.baserecalibrator *gatk.baserecalibrator*

Description

A wrapper function to run GATK (BaseRecalibrator)

Usage

```
gatk.baserecalibrator(fns.bam, output.dir, sample.name, ref.fa, ref.dbSNP, ref.g
```

Arguments

fns.bam	Path to input BAM files
output.dir	Output directory
sample.name	A character vector for the sample names
ref.fa	Reference fasta file
ref.dbSNP	Known SNP sites reference(VCF)
ref.gold_indels	Known sites to indel variant call format(VCF)
unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

First pass of the base quality score recalibration. Generates a recalibration table based on various covariates. The default covariates are read group, reported quality score, machine cycle, and nucleotide context. This walker generates tables based on specified covariates via by-locus traversal operating only at sites that are in the known sites VCF.

Value

GATK report file contained recalibration table by read group, quality scores and all the optional covariates. (e.g., .grp)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

gatk.combinevariants
gatk.combinevariants

Description

A wrapper function to run GATK (CombineVariants)

Usage

```
gatk.combinevariants(ref.fa, normal.vcf, minN=2, filteredrecordsmergetype="KEEP_
```

Arguments

ref.fa	Reference fasta file
normal.vcf	Normal sample vcfs list
minN	Parameter value for -minN in GATK CombineVariants. Minimum number of samples to call the variant (default=2)
filteredrecordsmergetype	A parameter value for –filteredrecordsmergetype in GATK CombineVariants. Determines how to handle records seen at the same site in the VCF
output.dir	Output directory
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The MuTect2 pipeline employs a "Panel of Normal" to identify additional germline mutations. This method enables a higher level of confidence to be assigned to somatic variants that are called by the MuTect2 pipeline.

Value

pon(panel of normal) vcf file

References

Sensitive detection of somatic point mutations in heterogeneous cancer samples

See Also

https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org_broadinstitute_gatk_tools_walkers_cancer_m2_MuTect2.php

gatk.depthOFcoverage *gatk.depthOFcoverage*

Description

Calculate the read depth of the position with single nucleotide variations

Usage

```
gatk.depthOFcoverage(vcf.dir, annot.dir, Robject.dir, ref.fa, unsafe,
  minBaseQuality = 1, minMappingQuality = 1, run.cmd = TRUE,
  mc.cores = 1)
```

Arguments

vcf.dir	Output of variant call step (directory of vcf files)
annot.dir	Output of annotation step (directory of .annovar files)
ref.fa	Reference fasta file path
unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
minBaseQuality	Minimum base quality (default=1)
minMappingQuality	Minimum mapping quality (default=1)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to dedicate. Must be at least one(default=1), and parallelization requires at least two cores.

Details

When creating a vSet, use read depth to determine whether a mutation exists. GATK DepthOfCoverage uses the interval bed file to calculate the depth of the position.

gatk.haplotypecaller *gatk.haplotypecaller*

Description

A wrapper function to run GATK (HaplotypeCaller)

Usage

```
gatk.haplotypecaller(fns.bam, output.dir, sample.name, ref.fa, genotyping_mode="
```

Arguments

fns.bam	Path to BAM files
output.dir	Output directory
sample.name	A character vector for the sample names
ref.fa	Referance fasta file
genotyping_mode	A parameter value for –genotyping_mode in GATK. A character vector to determine the alternate alleles to use for genotyping (default: DISCOVERY)
output_mode	A parameter value for –output_mode in GATK. A character vector to produces variant calls (default: EMIT_VARIANTS_ONLY)
stand_call_conf_number	A parameter value for -stand_call_conf in GATK. A numeric value of The minimum phred-scaled confidence threshold at which variants should be called (default: 30)
unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

HaplotypeCaller is capable of calling SNPs and indels simultaneously via local de-novo assembly of haplotypes in an active region.

Value

Variant calling format files (.vcf)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

---

gatk.mutect2.normal

gatk.mutect2.normal

Description

A wrapper function to run GATK (MuTect2)

Usage

```
gatk.mutect2(normal.bam, sample.name, ref.dbSNP, cosmic.vcf, output.dir, run.cmd
```

Arguments

normal.bam	BAM files of normal samples
sample.name	A character vector for the sample names
ref.dbSNP	Known SNP sites reference vcf
cosmic.vcf	Known variant sites of cosmic database vcf file
output.dir	Output directory
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

MuTect2 is a somatic SNP and indel caller that combines the DREAM challenge-winning somatic genotyping engine of the original MuTect (Cibulskis et al., 2013) with the assembly-based machinery of HaplotypeCaller. This function takes normal samples as input to make the panel of normal (pon).

Value

Only normal sample vcf files.

References

Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples

See Also

https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org_broadinstitute_gatk_tools_walkers_cancer_m2_MuTect2.php

---

gatk.printreads *gatk.printreads*

Description

A wrapper function to run GATK (PrintReads)

Usage

```
gatk.printreads(fn.realign.bam, output.dir, sample.name, ref.fa, fns.grp, unsafe
```

Arguments

fns.bam	Path to input BAM files
fns.grp	GATK report file created by BaseRecalibrator
output.dir	Output directory
sample.name	A character vector for the sample names
ref.fa	Reference fasta file

unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Writes a new file using reads from SAM format file (SAM/BAM/CRAM) that pass criteria. Improves the accuracy of variant calling based on Base Quality Score Recalibration.

Value

GATK PrintReads returns a Base quality score recalibrated BAM files (eg. recal.bam)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

---

gatk.realign *gatk.realign*

Description

A wrapper function to run GATK (IndelRealigner)

Usage

```
gatk.realign(fn.rmdu.bam, fn.intervals, output.dir, fn.realign.bam, ref.fa, ref.
```

Arguments

fns.bam	Path to input BAM files
fns.interval	Interval list file created by gatk.targetcreator
output.dir	Output directory
sample.name	A character vector for the sample names
ref.fa	Reference fasta file
ref.gold_indels	Known sites to indel variant call format(VCF)
unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Perform local realignment of reads around indels. IndelRealigner takes a coordinate-sorted and indexed BAM and a target intervals file generated by RealignerTargetCreator. IndelRealigner then performs local realignment on reads coincident with the target intervals using consensus from indels present in the original alignment.

Value

GATK IndelRealigner returns a Realigned bam file (e.g., realign.bam)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

---

gatk.targetcreator *gatk.targetcreator*

Description

A wrapper function to run GATK (RealignerTargetCreator)

Usage

```
gatk.targetcreator(fns.bam, output.dir, sample.name, ref.fa, ref.indels, unsafe=
```

Arguments

fns.bam	Path to BAM files
output.dir	Output directory
sample.name	A character vector for the sample names
ref.fa	Reference fasta file
ref.gold_indels	Known sites to indel variant call format(VCF)
unsafe	A parameter value for -U ALLOW_N_CIGAR_READS in GATK. This parameter must be TRUE in RNA-seq data.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Define insertion and deletion intervals to target for local realignment

Value

Interval file included indel positions (e.g., .intervals)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

---

gatk.variantfilter *gatk.variantfilter*

Description

A wrapper function to run GATK (VariantFiltration)

Usage

```
gatk.variantfilter(fns.vcf, output.dir, sample.name, ref.fa, FS=30.0, QD=2.0, QU
```

Arguments

fns.vcf	Path to VCF files
output.dir	Output directory
ref.fa	Reference fasta file
FS	A parameter value for FS in GATK. FisherStrand. (default=30.0)
QD	A parameter value for QD in GATK. Quality by Depth. (default=2.0)
QUAL	A parameter value for QUAL in GATK. Low quality. (default=50)
DP	A parameter value for DP in GATK. Low depth. (default=5)
gatk.window	A parameter value for -window in GATK. The window size (in bases) in which to evaluate clustered SNPs.
cluster	A parameter value for -cluster in GATK. The number of SNPs which make up a cluster. Must be at least 2.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
sample.name	A character vector for the sample names

Details

Filter variant calls based on INFO and/or FORMAT annotations. This tool is designed for hard-filtering variant calls based on certain criteria.

Value

Filtered VCF file (eg. .f.vcf)

References

The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.

See Also

https://software.broadinstitute.org/gatk/

`generate.GC` *generate.GC*

Description

A wrapper function to run sequenza-utils.py in sequenza (GC-windows)

Usage

```
generate.GC(window=1,000,000, output.dir, ref.fa, run.cmd=TRUE)
```

Arguments

`window`	A parameter value for -w in sequenza. Indicate a window size (in bases), to be used for the binning. The heterozygous positions and the positions carrying variant calls are not affected by binning.
`output.dir`	Output directory
`ref.fa`	Reference fasta file path
`run.cmd`	Whether to execute the command line (default=TRUE)

Details

Calculation GC contents from reference fasta file

References

Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data.

See Also

https://cran.r-project.org/web/packages/sequenza/vignettes/sequenza.pdf

get.annovar.report *get.annovar.report*

Description

Creates a data frame using the annovar output files

Usage

```
get.annovar.report(annot.dir)
```

Arguments

annot.dir	Path to directory with annovar output files

Details

Provide data frame of annotation information

Value

list of result summary

get.collectmetrics.report *get.collectmetrics.report*

Description

Creates a data frame using picard.collectmultiplemetrics()

Usage

```
get.collectmetrics.report(bam.dir, collectmetrics.idx=".alignment_summary_metric
```

Arguments

bam.dir	Path to directory with bam files
collectmetric.idx	Index of files (default=".alignment_summary_metrics$")

Details

Creates a data frame using the txt file, the output of picard.collectmultiplemetrics().

Value

data frame of result summary

`get.fns` *get.fns*

Description

Gets path of input files

Usage

```
get.fns(input.dir, idx)
```

Arguments

`input.dir`	Path to directory including input files
`idx`	Suffix of input files

Value

Path to the input files

`get.proc.report` *get.proc.report*

Description

Writes report accoding to process

Usage

```
get.proc.report(proc=c("qc", "trim","cutadapt", "bwa-mem", "bwa-aln", "tophat2",
```

Arguments

`proc`	Process name
`output.dir`	Output directory

Details

Creates report information according to processing result

get.process.names *get.process.names*

Description

Process names to be used in report

Usage

```
get.process.names(qc, trim.method, align.method, bwa.method , rm.dup, realign, v
```

Arguments

qc	As the quality check progresses, qc is added to the report processes.
trim.method	As the trimming progresses, trimming method is added to the report processes.
align.method	As the alignment progresses, alignment method is added to the report processes.
bwa.method	When alignment is performed with bwa, bwa method is added to the report processes
rm.dup	As the removal of duplicates progresses, removal method is added to the report processes.
realign	As the re-alignment progresses, re-alignment is added to the report processes.
variant.call.method	As the variant calling progresses, variant calling method is added to the report processes.
annotation.methodd	As the variant annotation progresses, annotation method is added to the report processes.
rseq.quant.method	As the RNA quantification progresses, RNA quantification method is added to the report processes.

get.qc.report *get.qc.report*

Description

Creates a data frame using a fastq file

Usage

```
get.qc.report(qc.dir)
```

Arguments

qc.dir	Path to directory with fastQC output files

Details

Adds information (ex.Q30) using the data frame combined with the fastqc file and output the result as a data frame

Value

data frame of result summary

---

get.Robject.report *get.Robject.report*

Description

Reads rda file

Usage

```
get.Robject.report(Robject.dir)
```

Arguments

Robject.dir	Path to Robject directory

Details

Reads information according to data set.

Value

list of data set summary

---

get.single_end.metrics.report *get.single.end.metrics.report*

Description

Creates a data frame using picard.collectmultiplemetrics()

Usage

```
get.single_end.metrics.report(sam.dir, collectmetrics.idx=".alignment_summary_me
```

Arguments

sam.dir	Path to directory with sam files
collectmetric.idx	Index of file (default=".alignment_summary_metrics$")

Details

Creates a data frame using the txt file, the output of picard.collectmultiplemetrics(). Used for Single-end data(ex. miRSEQ).

Value

data frame of the result summary

`get.tophat.report` *get.tophat.report*

Description

Creates a data frame using the tophat align summary

Usage

```
get.tophat.report(tophat.dir)
```

Arguments

`tophat.dir`	Path to directory with bam files

Details

Creates a data frame using the txt file, the output of Tophat.

Value

data frame of result summary

`get.trim.gal.report` *get.trim.gal.report*

Description

Creates a data frame using the trimming output

Usage

```
get.trim.gal.report(trim.dir)
```

Arguments

`trim.dir`	Path to directory with trimmed fastq files

Details

Creates a data frame using the txt file, the output of Trim galore.

Value

data frame of result summary

`get.trim_cut.report` *get.trim_cut.report*

Description

Creates a data frame using the trimming output

Usage

```
get.trim_cut.report(trim.dir)
```

Arguments

`trim.dir`	Path to directory with trimmed fastq files

Details

Creates a data frame using the txt file, the output of Cutadapt.

Value

data frame of result summary

`gff2gr` *gff2gr*

Description

Converts reference gff file to GRanges form

Usage

```
gff2gr(mir.gff, output.dir)
```

Arguments

`mir.gff`	Directoy stored at reference gff file
`output.dir`	Output directory

`gtf2gr` *gtf2gr*

Description

Converts reference gtf file to GRanges form to execute FPKM estimation

Usage

```
gtf2gr(ref.gtf, output.dir)
```

Arguments

`ref.gtf`	Directoy stored at reference gtf file (e.g. gencode.v22.gtf)
`output.dir`	Output directory

Details

To normalize the number of reads of each feature calculated in the previous step to the value of FPKM, convert the reference gtf file to GRanges format.

`htseq.add.info` *htseq.add.info*

Description

Add information to the htseq output file

Usage

```
htseq.add.info(RNAtype="mRNA", count.fns, output.dir, output.dir, mc.cores=1)
```

Arguments

`RNAtype`	RNAtype (default="mRNA")
`output.dir`	Directory stored at FPKM conunt files
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
`fns.count`	count file paths

Details

Adds information to the output file of htseq. (Gene name, chromosome, start position, end position, gene size, FPKM value)

htseq_count *htseq_count*

Description

A wrapper function to run htseq-count for mRNA or miRNA quantitation

Usage

```
htseq_count(RNAtype="mRNA", fns.bam, sample.name, output.dir, Mode="intersection
```

Arguments

fns.bam	Path to input BAM or SAM files
sample.name	A character vector for the sample names
output.dir	Output directory
stranded	A parameter value for -s in htseq-count. Whether the data is from a strand-specific assay (default:no)
idattr	A parameter value for -i in htseq-count. GFF attribute to be used as feature ID (default:"gene_id")
htseq.r	A parameter value for -r in htseq-count. Sorting order method (default:"pos")
htseq.a	A parameter value for -a in htseq-count. Skip all reads with alignment quality lower than the given minimum value (default: 10)
ref.gtf	Directoy stored at reference gtf file
mir.gff	Directoy stored at micro-RNA reference gff file
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
MODE	A parameter value for -m in htseq-count. Mode to handle reads overlapping more than one feature (default:intersection-nonempty)

Details

Counting reads in features. Given a file with aligned sequencing reads and a list of genomic features, a common task is to count how many reads map to each feature.

Value

Text file included read count information

References

HTSeq—a Python framework to work with high-throughput sequencing data

See Also

{https://htseq.readthedocs.io/en/release_0.9.1/

make.cset *make.cset*

Description

The copy number variants data are transformed to a file with extension .cSet

Usage

```
make.cset(cnv.dir, Robject.dir)
```

Arguments

cnv.dir	sequenza output directory
Robject.dir	Ouptut directory

make.eset *make.eset*

Description

For the expression data are transformed to a file with extension .eSet

Usage

```
make.eset(RNAquant.dir, Robject.dir, mc.cores=1)
```

Arguments

RNAquant.dir	Cufflinks or htseq-count output directory
Robject.dir	Output directory
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Quantify mRNA and miRNA and store them in ExpressionSet format for convenient analysis

make.vset *make.vset*

Description

The mutations data are transformed to a file with extension .vSet

Usage

```
make.vset(annot.dir, Robject.dir, mc.cores = 1)
```

Arguments

annot.dir	ANNOVAR output directory
Robject.dir	Ouptut directory
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
mut.cnt.cutoff	Depth filter

multiple.reads.pileup *multiple.reads.pileup*

Description

A wrapper function to run samtools (mpileup)

Usage

```
multiple.reads.pileup(ref.fa, normal.bam, tumor.bam, sample.name, output.dir, ma
```

Arguments

ref.fa	Reference fasta file path
normal.bam	Path to normal sample recalibration bam files
tumor.bam	Path to tumor sample recalibration bam files as tumor-normal pair
sample.name	A character vector for the sample names
output.dir	Output directory
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
mapQ	A parameter value for mapQ in varscan2. Skip alignments with mapQ smaller than mapQ value (default:1)

Details

Generate VCF, BCF or pileup for one or multiple BAM files. Alignment records are grouped by sample (SM) identifiers in @RG header lines. If sample identifiers are absent, each input file is regarded as one sample.

See Also

{http://www.htslib.org/doc/samtools.html

`MuSE.call` *MuSE.call*

Description

A wrapper function to run MuSE (call)

Usage

```
MuSE.call(tumor.bam, normal.bam, output.dir, sample.name, ref.fa, run.cmd=TRUE,
```

Arguments

`tumor.bam`	path to tumor sample recalibration bam files as tumor-normal pair
`normal.bam`	path to normal sample recalibration bam files
`output.dir`	Output directory
`sample.name`	A character vector for the sample names
`ref.fa`	Reference fasta file
`run.cmd`	Whether to execute the command line (default=TRUE)
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The first step of MuSE, MuSE.call takes as input indexed reference fasta file and BAM files. The BAM files require aligning all the sequence reads against the reference genome using the Burrows-Wheeler alignment tool BWA-mem algorithm. In addition, the BAM files need to be processed by following the GATK-MarkDuplicates, realigning the paired tumor-normal BAMs jointly and recalibrating base quality scores.

Value

MuSE.call output txt file.

References

MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data

See Also

http://bioinformatics.mdanderson.org/main/MuSE

MuSE.sump *MuSE.sump*

Description

A wrapper function to run MuSE (sump)

Usage

```
MuSE.sump(MuSE.txt, output.dir, ref.dbSNP, ref.gold_indels, data.type="E", run.c
```

Arguments

MuSE.txt	Path to MuSE.call output text file
output.dir	Output directory
MuSE.data.type	E is used for WXS data and G can be used for WGS data
ref.dbSNP	Known SNP sites reference
ref.gold_indels	Known Indel sites reference
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The second step, MuSE sump, takes as input the output file from MuSE.call and dbSNP variant call format file. MuSE provide two options for building the sample-specific error model. One is applicable to WES data (option '-E'), and the other to WGS data (option -G).

Value

vcf files included variant call information

References

MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data

See Also

http://bioinformatics.mdanderson.org/main/MuSE

`Muse.tabix2.vcf` *VCFtabix*

Description

Tabix indexes a TAB-delimited genome position file in.tab.bgz.

Usage

```
tabix.vcf(vcf.gz.file, run.cmd = TRUE)
```

Arguments

`vcf.gz.file`	Path to dbSNP, indel reference vcf.gz file
`run.cmd`	Whether to execute the command line (default=TRUE)

`mut.type.somatic` *mut.type.somatic*

Description

Write mutation type

Usage

```
mut.type.somatic(df, ref="Ref",alt="Alt")
```

Arguments

`df`	data frame from get.annovar.report()
`ref`	column name of the reference (default="Ref")
`alt`	column name of the alteration (default="Alt")

Details

In the data frame, enter the mutation type using reference and alteration

Value

data frame with mutation type

mutect2 *mutect2*

Description

A wrapper function to run GATK (MuTect2) Processed through the variant calling as tumor-normal pairs.

Usage

```
run.mutect2(output.dir, ref.fa, tumor.bam, normal.bam, pon.vcf, cosmic.vcf, ref.
```

Arguments

output.dir	Output directory
ref.fa	Reference fasta file
tumor.bam	Tumor sample bam files
normal.bam	Bam files form normal samples obtained from a function gatk.mutect.normal
pon.vcf	Panel of normal samples obtained from a function gatk.combinedvariant
cosmic.vcf	Known variant sites of cosmic database vcf file
ref.dbSNP	Known SNP sites reference vcf
contamination_fraction_to_filter	A parameter value for –contamination_fraction_to_filter in GATK MuTect2. Fraction of contamination to aggressively remove (default=0.02)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

MuTect2 is designed to produce somatic variant calls only, and includes some logic to skip variant sites that are very clearly germline based on the evidence present in the Normal sample compared to the Tumor sample.

Value

VCF files

References

Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples

See Also

https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org_broadinstitute_gatk_tools_walkers_cancer_m2_MuTect2.php

picard.addrg *picard.addrg*

Description

A wrapper function to run Picard (AddOrReplaceReadGroups)

Usage

```
picard.addrg(fns.bam, output.dir, sample.name, RGLB="lC", RGPL="Illumina", RGPU=
```

Arguments

fns.bam	Path to BAM files
output.dir	Output directory
RGLB	A parameter value for RGLB in picard. A character value of Read Group library (default="LC")
RGPL	A parameter value for RGPL in picard. A character value of Read Group platform (default="Illumina")
RGPU	A parameter value for RGPU in picard. A character value of Read Group platform unit (default="runbarcode")
RGSM	A parameter value for RGSM in picard. character value of Read Group sample name (default=sample.name)
SORT_ORDER	A parameter value for SO in picard. Sort order, a character value of sorting method (default="coordinate")
VALIDATION_STRINGENCY	A parameter value for VALIDATION_STRINGENCY in picard. A character value of validation stringency (default="LENIENT")
CREATE_INDEX	A parameter value for CREATE_INDEX in picard. A character value whether to create .bam index files (default="true")
tmp.dir	Temporary directory path (default=./tmp)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
sample.name	A character vector for the sample names

Details

This tool 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 files.

Value

BAM files added read groups

See Also

http://broadinstitute.github.io/picard/

picard.collectmetrics *picard.collectmetrics*

Description

Provide read alignment information.

Usage

```
picard.collectmetrics(fns.bam, out.fns, ref.fa, run.cmd=T, mc.cores=1)
```

Arguments

fns.bam	Path to BAM files
ref.fa	Reference fasta file path
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
output.dir	Output directory

Details

Provides a summary of the alignment process using the bam files

Value

txt files

Author(s)

Ji-Hye Lee

References

Currently there is no journal reference for picard.

See Also

http://broadinstitute.github.io/picard/

picard.reorder *picard.reorder*

Description

A wrapper function to run Picard (ReordrSam)

Usage

```
picard.reorder(fns.bam, output.dir, sample.name, ref.fa, ALLOW_INCOMPLETE_DICT_C
```

Arguments

fns.bam	Path to BAM files
output.dir	Output directory
ref.fa	Reference fasta file (eg. GRCh38.fa)
ALLOW_INCOMPLETE_DICT_CONCORDANCE	A parameter value for ALLOW_INCOMPLETE_DICT_CONCORDANCE in picard. Logical, allow discordant contig (default=FALSE)
ALLOW_CONTIG_LENGTH_DISCORDANCE	A parameter value for ALLOW_CONTIG_LENGTH_DISCORDANCE in picard. Logical, allow contig of different length (default=FALSE)
CREATE_INDEX	A parameter value for CREATE_INDEX in picard. Logical, whether to create .bam index files (default=TRUE)
tmp.dir	Temporary directory (default= ./tmp)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
sample.name	A character vector for the sample names

Details

ReorderSam reorders reads in a BAM file to match the contig ordering in a provided reference file, as determined by exact name matching of contigs. Reads mapped to contigs but absent in the new reference are dropped. Runs substantially faster if the input is an indexed BAM file.

Value

Reordered BAM files (e.g., .rg.od.bam)

See Also

http://broadinstitute.github.io/picard/

picard.rmdu *picard.rmdu*

Description

A wrapper function to run Picard (MarkDuplicates)

Usage

```
picard.rmdu(fns.bam, output.dir, sample.name, out.metrics, CREATE_INDEX=TRUE, RE
```

Arguments

fns.bam	Path to BAM files
output.dir	Output directory
sample.name	A character vector for the sample names
CREATE_INDEX	A parameter value for CREATE_INDEX in picard. Logical, whether to create bam index files (default=TRUE)
REMOVE_DUPLICATES	A parameter value for REMOVE_DUPLICATES in picard. Logical, whether to remove duplicates (default=TRUE)
VALIDATION_STRINGENCY	A parameter value for VALIDATION_STRINGENCY in picard. A character value of validation stringency (default="LENIENT")
tmp.dir	Temporary directory (default= ./tmp)
BARCODE_TAG	A parameter value for BARCODE_TAG in picard. If barcode sequencing data, set this option TRUE. Duplicated BARCODE is removed.
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.#' @details The MarkDuplicates tool works by comparing sequences in the 5 prime positions of both reads and read-pairs in a SAM/BAM file. An BARCODE_TAG option is available to facilitate duplicate marking using molecular barcodes. After duplicate reads are collected, the tool differentiates the primary and duplicate reads using an algorithm that ranks reads by the sums of their base-quality scores.

Value

Removing duplicate reads in bam files (e.g., .rmdu.bam)

See Also

http://broadinstitute.github.io/picard/

`pileup2seqz` *generate.seqz*

Description

A wrapper function to run sequenza-utils.py in sequenza (pileup2seqz, seqz-binning)

Usage

```
pileup2seqz(gc.fn, normal.pileup.gz, window=1000000, tumor.pileup.gz, output.dir
```

Arguments

`normal.pileup.gz`	samtools pileup file of normal sample
`window`	A parameter value for -w in sequenza. Indicate a window size (in bases), to be used for the binning.
`tumor.pileup.gz`	samtools pileup file of tumor sample
`fn.gc`	output file of generate.GC function

Details

A seqz file contains genotype information, alleles and mutation frequency, and other features. This file is used as input for the R-based part of Sequenza.

References

Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data.

See Also

https://cran.r-project.org/web/packages/sequenza/vignettes/sequenza.pdf

---

`ploidyNcellularity` *ploidyNcellularity*

Description

Calculate ploidy and cellularity

Usage

```
ploidyNcellularity(cnv.dir)
```

Arguments

`cnv.dir`	Output directory

Details

Calculate ploidy and cellularity for each paired-sample and quantify the copy number

References

Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data.

See Also

https://cran.r-project.org/web/packages/sequenza/vignettes/sequenza.pdf

---

print_message *print_message*

Description

Show command line

Usage

```
print_message(cmd)
```

Arguments

cmd	What users want to show as a message

Value

message()

---

processSomatic *processSomatic*

Description

A wrapper function to run VarScan2 (processSomatic)

Usage

```
processSomatic(fns.vcf, output.dir, min_tumor_freq=0.1, max_normal_freq=0.05, p_
```

Arguments

fns.vcf	varscan output file path
output.dir	Output directory
min_tumor_freq	A parameter value for –min-tumor-freq in varscan2. Minimum variant allele frequency in tumor (default:0.10)
max_normal_freq	A parameter value for –max-normal-freq in varscan2. Maximum variant allele frequency in normal (default:0.05)
p_value	A parameter value for –p-value in varscan2. P-value for high-confidence calling (default:0.07)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

processSomatic will separate a somatic output file by somatic_status (Germline, Somatic, LOH). Somatic mutations will further be classified as high-confidence (.hc) or low-confidence (.lc).

References

VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing.

See Also

{http://varscan.sourceforge.net/

---

qc_aggregate *qc_aggregate*

Description

Aggregates the information from the fastq file

Usage

```
qc_aggregate(qc.dir, pattern="fastqc.zip$")
```

Arguments

qc.dir	Path to directory with fastqc.zip files
pattern	Index of file (default="fastqc.zip$")

Details

Combine the fastqc results into one data.

`read.pileup.gz` *read.pileup.gz*

Description

A wrapper function to run samtools (mpileup)

Usage

```
read.pileup.gz(ref.fa, fns.bam, sample.name, output.dir, mapQ=1, run.cmd=TRUE, m
```

Arguments

`ref.fa`	Reference fasta file path
`sample.name`	A character vector for the sample names
`mapQ`	Mapping quality (default=1)
`run.cmd`	Whether to execute the command line (default=TRUE)
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
`normal.bam`	BAM files of normal sample
`tumor.bam`	BAM files of tumor sample

Details

Generates VCF, BCF or pileup for one or multiple BAM files. Alignment records are grouped by sample (SM) identifiers in @RG header lines. If the sample identifiers are absent, each input file is regarded as one sample.

See Also

{http://www.htslib.org/doc/samtools.html

`refGene2gr` *refGene2gr*

Description

Convert gene reference file to GRanges form

Usage

```
refGene2gr(refGene.path, cnv.dir)
```

Arguments

`refGene.path`	Path to refGene.txt
`cnv.dir`	Copy number variation directory

report *report*

Description

Reports the result of using SEQprocess()

Usage

```
report(envList)
```

Arguments

envList	R environment list

Details

Provides a report that summarizes the processing steps and visualized tables and plots for the processed results. The report file is automatically generated recording the workflows of the data processing steps, the options used in the processing, and the outcome results.

Value

pdf file include data processing result information

SEQprocess *SEQprocess*

Description

Run the NGS data processing pipeline

Usage

```
SEQprocess(fastq.dir = NULL, output.dir = file.path(getwd(), "result",
  "SEQprocess_result"), argList = list(program.name = "SEQprocess"),
  project.name = "SEQprocess", type = c("WGS", "WES", "BarSEQ", "RSEQ",
  "miRSEQ"), pipeline = c("none", "GDC", "GATK", "BarSEQ", "Tuxedo",
  "miRSEQ"), mc.cores = 1, run.cmd = TRUE, report.mode = FALSE,
  config.fn = system.file("data/config.R", package = "SEQprocess"),
  qc = TRUE, trim.method = c("trim.galore", "cutadapt", "none"),
  align.method = c("bwa", "bowtie2", "tophat2", "star", "none"),
  build.transcriptome.idx = FALSE, tophat.thread.number = 4,
  bwa.method = c("mem", "aln"), bwa.thread.number = 4,
  star.thread.number = 8, rm.dup = c("MarkDuplicates", "BARCODE", "none"),
  realign = TRUE, variant.call.method = c("none", "gatk", "varscan2",
  "mutect2", "muse", "somaticsniper"), gatk.thread.number = 4,
  annotation.method = c("annovar", "vep", "none"), ref = "hg38",
  rseq.abundance.method = c("none", "cufflinks", "htseq"),
  cufflinks.gtf = c("G", "g"), cufflinks.thread.number = 4,
```

```
RNAtype = c("mRNA", "miRNA"), CNV = FALSE, make.eSet = FALSE,
eset2SummarizedExperiment = FALSE, mut.cnt.cutoff = 8,
qc.dir = file.path(output.dir, "00_qc"), trim.dir = file.path(output.dir,
"01_trim"), align.dir = file.path(output.dir, "02_align"),
rmdup.dir = file.path(output.dir, "03_rmdup"),
realign.dir = file.path(output.dir, "04_realign"),
vcf.dir = file.path(output.dir, "05_vcf"),
annot.dir = file.path(output.dir, "06_annot"),
RNAquant.dir = file.path(output.dir, "07_RNAquant"),
cnv.dir = file.path(output.dir, "08_cnv"),
Robject.dir = file.path(output.dir, "09_Robject"))
```

Arguments

`fastq.dir`	If the user starts the process with fastq files, set the directory for the fastq files.
`output.dir`	Output directory
`argList`	The argument list used by the user in the shell
`project.name`	User's project name
`type`	Sequence data type
`pipeline`	One of the six pipelines provided by SEQprocess
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
`run.cmd`	Whether to execute the command line (default=TRUE)
`report.mode`	Whether the process is finished and report is generated
`config.fn`	Congifure file path
`qc`	Whether quality check
`trim.method`	Set trimming method
`align.method`	Set alignment method
`build.transcriptome.idx`	(tophat) A transcriptome index and the associated data files (the original GFF file) can be thus reused for multiple TopHat runs with this option, so these files are only created for the first run with a given set of transcripts. (default=FALSE)
`tophat.thread.number`	(tophat) A numeric value of the number of threads
`bwa.method`	(bwa) Set bwa method
`bwa.thread.number`	(bwa) A numeric value of the number of threads
`star.thread.number`	(STAR) A numeric value of the number of threads
`rm.dup`	Set the remove duplicates method
`realign`	Whether realignment
`variant.call.method`	Set variant call method
`annotation.method`	Set variant annotation method
`ref`	(annovar) Set annovar reference version

rseq.abundance.method	Set RNA quantification method
cufflinks.gtf	(cufflinks) If you set "-G", Output will not include novel genes and isoforms that are assembled.
cufflinks.thread.number	(cufflinks) A numeric value of the number of threads
RNAtype	(htseq) Choose mRNA or miRNA.
CNV	Whether estimate copy number variation
make.eSet	Make ExpressionSet R data(RNA expression, Copy number variation, Mutation)
eset2SE	Convert ExpressionSet R data to SummarizedExperiment R data

---

`seqz2rda` *seqz2rda*

Description

Saves seqz file in R data format.

Usage

```
seqz2rda(cnv.dir)
```

Arguments

cnv.dir	output directory

`seqz2seg` *seqz2seg*

Description

Segmentation to estimate DNA copy number variation.

Usage

```
seqz2seg(cnv.dir, window=1,000,000)
```

Arguments

cnv.dir	Output directory
window	A parameter value for -w in sequenza. Indicate a window size (in bases), to be used for the binning.

Details

Normalization of depth ratio and DNA segmentation

References

Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data.

See Also

https://cran.r-project.org/web/packages/sequenza/vignettes/sequenza.pdf

---

`somaticsniper` *somaticsniper*

Description

A wrapper function to run SomaticSniper

Usage

```
somaticsniper(ref.fa, tumor.bam, normal.bam, output.dir, sample.name, mapQual=1,
```

Arguments

`tumor.bam`	Tumor sample bam files
`normal.bam`	Normal sample bam files
`output.dir`	Output directory
`sample.name`	A character vector for the sample names
`ref.fa`	Reference fasta file
`mapQual`	A parameter value for -q in SomaticSniper Filtering reads with mapping quality less than INT (default:1)
`LOH`	A parameter value for -L in SomaticSniper. Do not report LOH variants as determined by genotypes (logical)
`Genotype`	A parameter value for -G in SomaticSniper. Do not report Gain of Referene variants as determined by genotypes (logical)
`somaticQual`	A parameter value for -Q in SomaticSniper. Filtering somatic SNV output with somatic quality less than INT (default:15)
`somaticMutation`	A parameter value for -s in SomaticSniper. Prior probability of a somatic mutation (default:0.01)
`Theta`	A parameter value for -T in SomaticSniper. Theta in maq consensus calling model (default:0.85)
`Hap.number`	A parameter value for -N in SomaticSniper. Number of haplotypes in the sample (default:2)
`Hap.diff`	A parameter value for -r in SomaticSniper. Prior of a difference between two haplotypes (default:0.001)
`out.format`	A parameter value for -F in SomaticSniper. Select output format (vcf or classic) (default:vcf)
`run.cmd`	Whether to execute the command line (default=TRUE)
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The purpose of this program is to identify single nucleotide positions that are different between tumor and normal (or in theory, any two bam files). It takes a tumor bam and a normal bam and compares the two to determine the differences.

Value

VCF files

References

SomaticSniper: identification of somatic point mutations in whole genome sequencing data.

See Also

http://gmt.genome.wustl.edu/packages/somatic-sniper/

STAR *STAR*

Description

A wrapper function to run STAR.

Usage

```
STAR(STAR.idx, fq1, fq2, sample.name, output.dir, run.cmd=TRUE, mc.cores=1)
```

Arguments

star.idx.dir	Directory of STAR index
output.dir	Output directory
sample.name	A character vector for the sample names
fq1	path to read1 fastq files
fq2	path to read2 fastq files
outFilterMultimapScoreRange	A parameter value for –outFilterMultimapScoreRange in STAR. The score range below the maximum score for multimapping alignments (default: 1)
outFilterMultimapNmax	A parameter value for –outFilterMultimapNmax in STAR. Read alignments will be output only if the read maps fewer than this value, otherwise no alignments will be output (default: 20)
outFilterMismatchNmax	A parameter value for –outFilterMismatchNmax in STAR. Alignment will be output only if it has fewer mismatches than this value (default: 10)
alignIntronMax	A parameter value for –alignIntronMax in STAR. Maximum intron length (default: 500,000)

`alignMatesGapMax`
: A parameter value for –alignMatesGapMax in STAR. Maximum genomic distance between mates (default: 1,000,000)

`sjdbScore`
: A parameter value for –sjdbScore in STAR. Extra alignment score for alignmets that cross database junctions (default: 2)

`alignSJDBoverhangMin`
: A parameter value for –alignSJDBoverhangMin in STAR. Minimum overhang for annotated junctions (default: 1)

`outFilterMatchNminOverLread`
: A parameter value for –outFilterMatchNminOverLread in STAR. Float: outFilterMatchNmin normalized to read length (sum of mates' lengths for paired-end reads) (default: 0.33)

`outFilterScoreMinOverLread`
: A parameter value for –outFilterScoreMinOverLread in STAR. Float: outFilterScoreMin normalized to read length (sum of mates' lengths for paired-end reads) (default: 0.33)

`sjdbOverhang`
: A parameter value for –sjdbOverhang in STAR. >=0: Length of the donor/acceptor sequence on each side of the junctions, if =0, splice junction database is not used (default: 100)

`SJ.detect`
: First align, detection of splicing junction (default=TRUE)

`SJ.align`
: Second align, mapping reads to fastq files (default=FALSE)

`run.cmd`
: Whether to execute the command line (default=TRUE)

`mc.cores`
: The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

`star_thread_number`
: A parameter value for –runThreadN in STAR. A numeric value of the number of threads (default: 8)

Details

Spliced Transcripts Alignment to a Reference (STAR), which was designed to specifically address many of the challenges of RNA-seq data mapping, and uses a novel strategy for spliced alignments.

Value

Aligned BAM files

References

STAR: ultrafast universal RNA-seq aligner

See Also

{https://github.com/alexdobin/STAR

table.annovar *table.annovar*

Description

A wrapper function to run table_annovar.pl in ANNOVAR

Usage

```
table.annovar(fn.annovar, output.dir, sample.name, annovar.db, ref="hg38", proto
```

Arguments

fns.annovar	Path to annovar files
output.dir	Output directory
sample.name	A character vector for the sample names
annovar.db.dir	Path to directory with ANNOVAR database
ref	A parameter value for -buildver in ANNOVAR. Specify the genome build version (default: hg38)
protocol	A parameter value for -protocol in ANNOVAR Database names in ANNOVAR (default: "refGene,cytoBand,genomicSuperDups,esp6500siv2_all,1000g2015aug_all,exac03,avsnp1
protocol.type	A parameter value for -operation in ANNOVAR. Strings separated by commas that specify the types of operation for each protocol (g: genome, r: region, f: filter, default="g,r,r,f,f,f,f,f,f")
nastring	A parameter value for -nastring in ANNOVAR. Strings to display when a score is not available (default: ".")
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

This function takes an input variant file (such as a VCF file) and generate a tab-delimited output file with many columns, each representing one set of annotations. Additionally, if the input is a VCF file, the program also generates a new output VCF file with the INFO field filled with annotation information.

Value

CSV files from annovar format

References

ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data

See Also

http://annovar.openbioinformatics.org/en/latest/user-guide/

tophat2 *tophat2*

Description

A wrapper function to run tophat2.

Usage

```
tophat2(fq1, fq2, output.dir, sample.name, ref.gtf, bowtie.idx, tophat_thread_nu
```

Arguments

fq1	Path to read1 fastq files
fq2	Path to read2 fastq files
output.dir	Output directory
sample.name	A character vector for the sample names
ref.gtf	Path to reference gtf file
bowtie.idx	Path to directory with bowtie indexes and a prefix for the bowtie indexes
build.transcriptome.idx	A parameter value for –transcriptome-index in tophat2. A transcriptome index and the associated data files (the original GFF file) can be thus reused for multiple TopHat runs with this option, so these files are only created for the first run with a given set of transcripts. (default=FALSE)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
tophat_thread_number	A parameter value for -p in tophat2. A numeric value of the number of threads (default: 4)

Details

TopHat is a program that aligns RNA-Seq reads to a genome to identify exon-exon splice junctions. It is built on the ultrafast short read mapping program Bowtie.

Value

Aligned BAM files

References

TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions

See Also

https://ccb.jhu.edu/software/tophat/manual.shtml

trim.gal *trim.gal*

Description

A wrapper function to run Trim Galore

Usage

```
trim.gal(fq1, fq2, trim.quality=30, trim.clip_R1=13, trim.clip_R2=13, output.dir
```

Arguments

fq1	Path to read1 fastq files
fq2	Path to read2 fastq files
output.dir	Output directory
trim.quality	A parameter value for –quality in trimgalore. A numeric value of phred score cutoff to trim (default=30)
trim.clip_R1	A parameter value for –clip_R1 in trimgalore. A numeric value of bp to remove adaptor from the 5-prime end of read1 files (default=13)
trim.clip_R2	A parameter value for –clip_R2 in trimgalore. A numeric value of bp to remove adaptor from the 5-prime end of read2 files (default=13)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Trims low quality bases, and cleans up adapter sequences for paired-end files.

Value

Trimmed fastq files (e.g., .val_1.fastq, .val_1.fastq)

References

Krueger F. Trim Galore!

See Also

https://www.bioinformatics.babraham.ac.uk/projects/trim_galore

varscan *varscan*

Description

A wrapper function to run VarScan2

Usage

```
varscan(fn.pileup, output.dir, sample.name, min_coverage_normal=8, min_coverage_
```

Arguments

fn.pileup	samtools mpileup output file path
output.dir	Output directory
sample.name	A character vector for the sample names
min_coverage_normal	A parameter value for –min-coverage-normal in VarScan2. Minimum coverage in normal to call somatic (default:8)
min_coverage_tumor	A parameter value for –min-coverage-tumor in VarScan2. Minimum coverage in tumor to call somatic (default:6)
min_var_freq	A parameter value for –min-var-freq in VarScan2. Minimum variant frequency to call a heterozygote (default:0.10)
min_freq_for_hom	A parameter value for –min-freq-for-hom in VarScan2. Minimum frequency to call homozygote (default:0.75)
somatic_p_value	A parameter value for –somatic-p-value in VarScan2. P-value threshold to call a somatic site (default:0.05)
strand_filter	A parameter value for –strand-fiter in VarScan2. If set to 1, removes variants with > 90 percent strand bias(default:0)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

VarScan is a platform-independent mutation caller for targeted, exome, and whole-genome sequencing data.

References

VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing.

See Also

{http://varscan.sourceforge.net/

vcf2annovar *vcf2annovar*

Description

A wrapper function to run vcf2annovar.pl in ANNOVAR

Usage

```
vcf2annovar(fns.vcf, output.dir, sample.name, format="vcf4", coverage=0, run.cmd
```

Arguments

fns.vcf	Path to VCF files
output.dir	Output directory
sample.name	A character vector for the sample names
format	A parameter value for -format in ANNOVAR Input files format (.vcf)
coverage	A parameter value for -coverage in ANNOVAR Read coverage threshold in pileup file (default:0)
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The convert2annovar.pl script convert other "genotype calling" format into ANNOVAR format. Additionally, the program can generate ANNOVAR input files from a list of dbSNP identifiers, or from transcript identifiers, or from a genomic region.

Value

Converted annovar files from variant calling format (e.g., .annovar)

References

ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data

See Also

http://annovar.openbioinformatics.org/en/latest/user-guide/

vcf2gz *vcf2gz*

Description

Compress VCF file to gz file

Usage

```
vcf2gz(vcf)
```

Arguments

vcf	Path to dbSNP, indel reference vcf file

vep *Vairant Effect Predictor (VEP)*

Description

A wrapper function to run VEP

Usage

```
vep(fns.vcf, output.dir, sample.name, perl5.10.path="/usr/bin", vep.db.dir, run.
```

Arguments

fns.vcf	Path to VCF files
output.dir	Output directory
vep.db.dir	Specify the cache directory to use.
perl5.10.path	Absolute path to perl 5.10 version. VEP is a Perl based tool. We recommend version 5.10. (default="/usr/bin")
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

The VEP uses the coordinates and alleles in the VCF file to infer biological context for each variant including the location of each mutation, its biological consequence (frameshift/ silent mutation), and the affected genes. Variants in the VCF files are also matched to known variants from external mutation databases.

Value

text file and html file included variant information

References

The Ensembl Variant Effect Predictor

See Also

https://asia.ensembl.org/info/docs/tools/vep/script/vep_options.html

---

`vset.preprocess` *vset.preprocess*

Description

Use the depth of the variants position to determine the presence or absence of the mutation.

Usage

```
vset.preprocess(Robject.dir, mut.cnt.cutoff)
```

Arguments

`annot.dir`	Output of annotation step (directory of .annovar files)
`Robject.dir`	Ouptut directory
`mut.cnt.cutoff`	Criterion of depth (default=8)
`mc.cores`	The number of cores to dedicate. Must be at least one(default=1), and parallelization requires at least two cores.

Details

vSet before preprocessing is simply denoted as 1 if there is a mutation and 0 otherwise. After the depth of the mutated position is calculated, the variants position with a depth that does not meet a certain criterion is not included in the analysis.

Index

bowtie2, 2
build.star.idx, 3
bwa, 4

cufflinks, 5
cutadapt, 6

eset2SE, 6

fastQC, 7

gatk.baserecalibrator, 8
gatk.combinevariants, 9
gatk.depthOFcoverage, 10
gatk.haplotypecaller, 10
gatk.mutect2.normal, 11
gatk.printreads, 12
gatk.realign, 13
gatk.targetcreator, 14
gatk.variantfilter, 15
generate.GC, 16
get.annovar.report, 17
get.collectmetrics.report, 17
get.fns, 18
get.proc.report, 18
get.process.names, 19
get.qc.report, 19
get.Robject.report, 20
get.single_end.metrics.report, 20
get.tophat.report, 21
get.trim.gal.report, 21
get.trim_cut.report, 22
gff2gr, 22
gtf2gr, 23

htseq.add.info, 23
htseq_count, 24

make.cset, 25
make.eset, 25
make.vset, 26
multiple.reads.pileup, 26
MuSE.call, 27
MuSE.sump, 28
Muse.tabix2.vcf, 29
mut.type.somatic, 29
mutect2, 30

picard.addrg, 31
picard.collectmetrics, 32
picard.reorder, 33
picard.rmdu, 34
pileup2seqz, 35
ploidyNcellularity, 35
print_message, 36
processSomatic, 36

qc_aggregate, 37

read.pileup.gz, 38
refGene2gr, 38
report, 39

SEQprocess, 39
seqz2rda, 41
seqz2seg, 41
somaticsniper, 42
STAR, 43

tabix.vcf *(Muse.tabix2.vcf)*, 29
table.annovar, 45
tophat2, 46
trim.gal, 47

varscan, 48
vcf2annovar, 49
vcf2gz, 50
vep, 50
vset.preprocess, 51