Package: pepitope (via r-universe)

June 18, 2026

Title Extract, QC and Screen Peptide Epitopes

Version 0.99.2

Description Supports T cell receptor neoantigen co-culture screen analysis from peptide library design to sequencing quality control and differential abundance testing. The package annotates somatic variants and RNA fusions, extracts mutant and reference peptide context, prepares barcoded minigene construct tables, counts sample and construct barcodes from sequencing data, and identifies immunogenic epitopes in dropout screens.

URL https://github.com/mschubert/pepitope

BugReports https://github.com/mschubert/pepitope/issues

Depends R (>= 4.5.0)

Imports Biobase, BiocGenerics, Biostrings, BSgenome, DESeq2, dplyr, ensembldb, GenomeInfoDb, GenomicFeatures, GenomicRanges, ggplot2, ggpp, ggrepel, grDevices, gtools, IRanges, MatrixGenerics, methods, patchwork, purrr, RColorBrewer, readr, reshape2, Rcpp, rlang, S4Vectors, stats, stringdist, stringr, SummarizedExperiment, tibble, tidyr, utils, VariantAnnotation, XVector

LinkingTo Rcpp

License GPL-3

Encoding UTF-8

Suggests AnnotationHub, BiocCheck, BSgenome.Hsapiens.NCBI.GRCh38, BSgenome.Hsapiens.UCSC.hg38, DT, htmltools, knitr, pkgdown, plotly, readxl, rmarkdown, testthat, writexl

biocViews DNASeq, VariantAnnotation, ImmunoOncology

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/pak/sysreqs make libbz2-dev libicu-dev liblzma-dev libpng-dev libxml2-dev libssl-dev libx11-dev xz-utils zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-06-18 17:26:59 UTC

RemoteUrl https://github.com/BiocStaging/pepitope

RemoteRef HEAD

RemoteSha 0146b1ec6ca0fa8828e66acd41610bfadd6e4b1e

Contents

`annotate_coding` *Annotate VCF variants with coding changes*

Description

Annotate VCF variants with coding changes

Usage

```
annotate_coding(vr, txdb, asm)
```

Arguments

`vr`	A VRanges object with SNVs and small indels
`txdb`	TxDb or EnsDb object
`asm`	Genomic sequence BSGenome object

Value

A GRanges object with annotated variants

Examples

```
if (interactive()) {
    vcf = system.file("my_variants.vcf", package="pepitope")
    vr = readVcfAsVRanges(vcf)
    txdb = AnnotationHub::AnnotationHub()[["AH100643"]]
    asm = BSgenome.Hsapiens.NCBI.GRCh38::BSgenome.Hsapiens.NCBI.GRCh38
    annotate_coding(vr, txdb, asm)
}
```

---

annotate_fusions *Aggregate fusion VCFs into a table*

Description

Aggregate fusion VCFs into a table

Usage

```
annotate_fusions(vr, txdb, asm)
```

Arguments

vr	A VRanges object with RNA fusions from readVcfAsRanges
txdb	A transcription database, eg. AnnotationHub()[["AH100643"]]
asm	A Genome sequence package object, eg. ::BSgenome.Hsapiens.NCBI.GRCh38

Value

A DataFrame objects with fusions

Examples

```
if (interactive()) {
    vcf = system.file("my_fusions.vcf", package="pepitope")
    vr = readVcfAsVRanges(vcf)
    txdb = AnnotationHub::AnnotationHub()[["AH100643"]]
    asm = BSgenome.Hsapiens.NCBI.GRCh38::BSgenome.Hsapiens.NCBI.GRCh38
    annotate_fusions(vr, txdb, asm)
}
```

count_fastq *Count barcodes directly from source FASTQ files*

Description

Count barcodes directly from source FASTQ files

Usage

```
count_fastq(
  fq,
  samples,
  all_constructs,
  valid_barcodes,
  read_structure,
  verbose = TRUE
)
```

Arguments

fq	Path to one FASTQ file
samples	A sample sheet as data.frame in tsv format. Requires the columns 'sample_id', 'patient', 'rep', 'origin', 'barcode'
all_constructs	A named list of all construct libraries
valid_barcodes	A character vector of all possible construct barcodes
read_structure	A character string describing the FASTQ read structure. If missing, this will be inferred from the first reads in 'fq'.
verbose	Whether to print progress messages (default: TRUE)

Value

A SummarizedExperiment object with counts and metadata

Examples

```
samples = data.frame(sample_id="sample1", patient="pat1", rep="1",
    origin="library", barcode="GGG")
constructs = list(pat1=data.frame(gene_name="GENE1", mut_id="GENE1_A1V",
    pep_id="GENE1_A1V", pep_type="alt", tiled="ATGGCCGCC", barcode_1="AAAA"))
fq = tempfile(fileext=".fq")
writeLines(c("@r1", "GGGAAAA", "+", "IIIIIII", "@r2", "GGGAAAA", "+", "IIIIIII"), fq)
count_fastq(fq, samples, constructs, read_structure="3B4M", verbose=FALSE)
```

`filter_fusions` *Filter a fusion VRanges object by number of reads and tools*

Description

Filter a fusion VRanges object by number of reads and tools

Usage

```
filter_fusions(vr, min_reads = NULL, min_split_reads = NULL, min_tools = NULL)
```

Arguments

`vr`	A VRanges object with RNA fusions from readVcfAsRanges
`min_reads`	The minimum number of linked read support for a fusion
`min_split_reads`	The minimum number of split read support for a fusion
`min_tools`	The minimum number of tools that identify a fusion

Value

A filtered VRanges object

Examples

```
vr = VariantAnnotation::VRanges("chr1", IRanges::IRanges(1, 1), ref="A", alt="T")
vr$DV = 3L
vr$RV = 2L
vr$TOOL_HITS = IRanges::IntegerList(2L)
filter_fusions(vr, min_reads=4, min_split_reads=2, min_tools=1)
```

`filter_variants` *Make results report to save as xlsx sheets (full, filtered, peptides)*

Description

Make results report to save as xlsx sheets (full, filtered, peptides)

Usage

```
filter_variants(
  vr,
  ...,
  min_cov = 2,
  min_af = 0.05,
  pass = TRUE,
  sample = NULL,
  chrs = NULL
)
```

Arguments

vr	A VRanges object from ‘readVcfAsVRanges‘
...	Force filters by name (ignored)
min_cov	Minimum number of reads to span the ALT allele
min_af	Minimum allele frequency of the ALT allele
pass	Whether to only include softFilterMatrix PASS
sample	Only include if in ‘sampleNames(vr)‘ (required if more than one present)
chrs	Either "default" or a character vector of chromosome names

Value

A filtered VRanges object

Examples

```
vcf = system.file("my_variants.vcf", package="pepitope")
vr = readVcfAsVRanges(vcf)
filter_variants(vr, min_cov=2, min_af=0.05, pass=TRUE)
```

---

make_peptides *Make a variants report as named list of tables*

Description

Make a variants report as named list of tables

Usage

```
make_peptides(subs, fus = DataFrame())
```

Arguments

subs	Variants within mutation context from ‘subset_context()‘
fus	Variants within fusion context from ‘subset_context_fusions()‘

Value

A data.frame with cDNA and peptide sequences

Examples

```
if (interactive()) {
    peptides = make_peptides(subs)
}
```

`make_report` *Make a variants report as named list of tables*

Description

Make a variants report as named list of tables

Usage

```
make_report(vars, subs, fus = DataFrame(), tiled)
```

Arguments

`vars`	Variant results from ‘annotate_coding()‘
`subs`	Variants within mutation context from ‘subset_context()‘
`fus`	Variants within fusion context from ‘subset_context_fusions()‘
`tiled`	A data.frame of the tiled peptide sequences

Value

A named list of data.frames for report output

Examples

```
if (interactive()) {
    make_report(vars, subs, fus, tiled)
}
```

pep_tile *Tile cDNA into peptide sequences*

Description

Tile cDNA into peptide sequences

Usage

```
pep_tile(peptides, tile_size = 93, tile_ov = 45)
```

Arguments

peptides	A ‘data.frame‘ with context-subset peptide/minigene data
tile_size	Oligo tiling size
tile_ov	Oligo tiling overlap

Value

A data.frame with tiled nucleotide and peptide sequences

Examples

```
peptides = data.frame(var_id="v1", mut_id="M1_A1V", gene_name="M1",
    gene_id="gene1", tx_id="tx1", pep_id="M1_A1V", cDNA="ATGGCCGCCGCC")
pep_tile(peptides, tile_size=9, tile_ov=3)
```

pepitope *Pepitope: peptide epitopes from reference genome and variant (VCF) file*

Description

Given a reference genome and VCF file, this package will provide the upstream/downstream peptide context of a variant. It will generate a summary report for protein-coding variants including the reference and mutated allele, read coverage, amino acid sequence, and other information. It can also be used to remove restriction sites from cDNA, alongside other helper functions.

Value

Package documentation for pepitope

Author(s)

Maintainer: Michael Schubert <mschu.dev@gmail.com> (ORCID)

Authors:

- Michael Schubert <mschu.dev@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/mschubert/pepitope
- Report bugs at https://github.com/mschubert/pepitope/issues

---

plot_barcode_overlap *Plot barcode overlap between different samples*

Description

Plot barcode overlap between different samples

Usage

```
plot_barcode_overlap(all_constructs, valid_barcodes)
```

Arguments

all_constructs	A named list of all constructs
valid_barcodes	A character vector of possible barcodes (optional)

Value

A ggplot2 object

Examples

```
if (interactive()) {
    constructs = list(pat1=data.frame(gene_name="GENE1", mut_id="GENE1_A1V",
        pep_id="GENE1_A1V", pep_type="alt", tiled="ATGGCCGCC", barcode_1="AAAA"))
    plot_barcode_overlap(constructs, valid_barcodes=c("AAAA", "CCCC"))
}
```

plot_read_count *Plot the overall read counts*

Description

Plot the overall read counts

Usage

```
plot_read_count(dset, n_cutoff = 10)

plot_reads(dset)
```

Arguments

dset	The 'SummarizedExperiment' object from 'count_fastq'
n_cutoff	Minimum reads for counting a barcode as represented

Value

A patchwork object containing the read count summary plots

Examples

```
if (interactive()) {
    samples = data.frame(sample_id="sample1", patient="pat1", rep="1",
        origin="library", barcode="GGG")
    constructs = list(pat1=data.frame(gene_name="GENE1", mut_id="GENE1_A1V",
        pep_id="GENE1_A1V", pep_type="alt", tiled="ATGGCCGCC", barcode_1="AAAA"))
    fq = tempfile(fileext=".fq")
    writeLines(c("@r1", "GGGAAAA", "+", "IIIIIII", "@r2", "GGGAAAA", "+", "IIIIIII"), fq)
    dset = count_fastq(fq, samples, constructs, read_structure="3B4M", verbose=FALSE)
    plot_read_count(dset)
}
```

plot_read_distr *Plot the read distribution across barcodes*

Description

Plot the read distribution across barcodes

Usage

```
plot_read_distr(dset)

plot_distr(dset)
```

Arguments

dset	The ‘SummarizedExperiment‘ object from ‘count_fastq‘

Value

A ‘ggplot2‘ object with cumulative read distribution plots

Examples

```
if (interactive()) {
    samples = data.frame(sample_id="sample1", patient="pat1", rep="1",
        origin="library", barcode="GGG")
    constructs = list(pat1=data.frame(gene_name="GENE1", mut_id="GENE1_A1V",
        pep_id="GENE1_A1V", pep_type="alt", tiled="ATGGCCGCC", barcode_1="AAAA"))
    fq = tempfile(fileext=".fq")
   writeLines(c("@r1", "GGGAAAA", "+", "IIIIIII", "@r2", "GGGAAAA", "+", "IIIIIII"), fq)
    dset = count_fastq(fq, samples, constructs, read_structure="3B4M", verbose=FALSE)
    plot_read_distr(dset)
}
```

---

plot_read_structure *Plot annotated read structure examples*

Description

Plot annotated read structure examples

Usage

```
plot_read_structure(fq, samples, all_constructs, nrec = 100000L)
```

Arguments

fq	Path to one FASTQ file
samples	A sample sheet as ‘data.frame‘ in tsv format. Requires the columns ’sample_id’, ’patient’, ’rep’, ’origin’, ’barcode’
all_constructs	A named list of all construct libraries
nrec	Number of FASTQ records to inspect

Value

A ‘ggplot2‘ object

Examples

```
if (interactive()) {
    samples = data.frame(sample_id="sample1", patient="pat1", rep="1",
        origin="library", barcode="GGG")
    constructs = list(pat1=data.frame(gene_name="GENE1", mut_id="GENE1_A1V",
        pep_id="GENE1_A1V", pep_type="alt", tiled="ATGGCCGCC", barcode_1="AAAA"))
    fq = tempfile(fileext=".fq")
    writeLines(c("@r1", "GGGAAAA", "+", "IIIIIII", "@r2", "GGGAAAA", "+", "IIIIIII"), fq)
    plot_read_structure(fq, samples, constructs, nrec=10)
}
```

`plot_screen` *Plot screen results*

Description

Plot screen results

Usage

```
plot_screen(
  res,
  sample = NULL,
  links = TRUE,
  labs = TRUE,
  min_reads = 10,
  cap_fc = 8,
  p_sig = 0.05
)
```

Arguments

`res`	A results 'data.frame' from 'screen_calc()'
`sample`	Which library to plot (default: all)
`links`	Whether to draw arrows between ref and significant alt peptides
`labs`	Whether to label genes in less dense areas
`min_reads`	Minimum number of reads to show as points (default: 10)
`cap_fc`	Maximum amount of fold-change to limit values to
`p_sig`	Maximum adjusted p-value to consider significant (default: 0.05)

Value

A 'ggplot2' object of the differential expression results

Examples

```
if (interactive()) {
    res = data.frame(baseMean=c(20, 30), log2FoldChange=c(-1, 1), padj=c(0.2, 0.05),
        gene_name=c("GENE1", "GENE1"), pep_type=c("ref", "alt"),
        bc_type=c("pat1", "pat1"), mut_id=c("GENE1_A1", "GENE1_A1"))
    plot_screen(res, links=FALSE, labs=FALSE)
}
```

`remove_cutsite` *Remove a Restriction Enzyme cut site but keep AA in a tiled peptide data.frame*

Description

Remove a Restriction Enzyme cut site but keep AA in a tiled peptide data.frame

Usage

```
remove_cutsite(pep, ...)
```

Arguments

`pep`	A data.frame of tiled peptides
`...`	Named argumennts of cut sites, e.g. 'BbsI="GAAGAC"'

Value

A data.frame with replace nucleotides and number of replacements

Examples

```
pep = data.frame(pep_id="p1", tiled="ATGGCCGCC")
remove_cutsite(pep, BbsI="GAAGAC")
```

screen_calc *Calculate differential abundance of construct barcodes*

Description

Calculate differential abundance of construct barcodes

Usage

```
screen_calc(dset, comparisons)
```

Arguments

dset	A 'SummarizedExperiment' object from 'count_fastq()'
comparisons	A character vector of sample and reference condition, or list thereof

Value

A data.frame of DESeq2 results, or a named list of data.frames

Examples

```
if (interactive()) {
    screen_calc(dset, c("screen", "library"))
}
```

subset_context *Subset nucleotide/protein sequences to codon +/- 45 bp context*

Description

Subset nucleotide/protein sequences to codon +/- 45 bp context

Usage

```
subset_context(codv, ctx_codons)
```

Arguments

codv	Annotated variants from 'annotate_coding()'
ctx_codons	How many flanking codons each to include in the context

Value

GRanges object with sequence information of only context

Examples

```
if (interactive()) {
    subset_context(codv, ctx_codons=15)
}
```

`subset_context_fusion` *Subset the peptide context for gene fusions*

Description

Subset the peptide context for gene fusions

Usage

```
subset_context_fusion(res, ctx_codons)
```

Arguments

`res`	A DataFrame object from ‘fusions‘
`ctx_codons`	How many flanking codons each to include in the context

Value

A DataFrame object of gene fusions

Examples

```
if (interactive()) {
    subset_context_fusion(fusions, ctx_codons=15)
}
```

Index