

Package: drugannotR (via r-universe)

September 29, 2026

Type Package

Title Coordinate-Level Pharmacological Annotation of Genomic Regions

Version 0.99.0

Description Maps genomic regions to gene features and then to traceable drug-gene relationships. The package preserves genomic coordinates, relationship categories, evidence scope, and source provenance. It deliberately reports coordinate-level pharmacological relevance rather than predicting treatment response, efficacy, toxicity, or dose.

License BSD_3_clause + file LICENSE

Encoding UTF-8

Depends R (>= 4.5.0)

Imports AnnotationDbi, BiocFileCache, GenomeInfoDb, GenomicFeatures, GenomicRanges, IRanges, S4Vectors, methods, rtracklayer, stats, utils

Suggests BiocStyle, covr, knitr, org.Hs.eg.db, rmarkdown, testthat (>= 3.0.0), txdbmaker, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg38.knownGene

VignetteBuilder knitr

biocViews Annotation, DataRepresentation, DataImport, GeneTarget, GenomeAnnotation, Pharmacogenetics, Pharmacogenomics, VariantAnnotation

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Config/testthat/edition 3

Config/pak/sysreqs

make libbz2-dev libicu-dev liblzma-dev libpng-dev libxml2-dev libssl-dev xz-utils zlib1g-dev

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

Date/Publication 2026-07-31 09:49:32 UTC

RemoteUrl <https://github.com/BiocStaging/drugannotR>

RemoteRef HEAD

RemoteSha cdff92230f7f31837ccc17f57ca7faf3fa38038e

Contents

annotateDrugRegions	2
buildDrugGeneSets	3
mapDrugGenes	5
summariseDrugAnnotations	7

Index	9
--------------	----------

annotateDrugRegions	<i>Annotate genomic regions with pharmacological relationships</i>
---------------------	--

Description

Finds overlaps between input regions and mapped drug-gene features while preserving coordinates, feature type, evidence scope, and source provenance.

Usage

```
annotateDrugRegions.bed, drug_ranges)
```

Arguments

bed	A GenomicRanges::GRanges , an existing BED file path, or a data frame with seqnames, start, and end. Data-frame coordinates are interpreted as 1-based closed coordinates; BED files are imported with <code>rtracklayer::import()</code> and retain BED coordinate semantics.
drug_ranges	A GenomicRanges::GRanges returned by <code>mapDrugGenes()</code> .

Details

Overlaps ignore strand because the input represents observed genomic regions rather than stranded transcripts. The returned interpretation never asserts a functional effect or predicts drug response.

Value

A [GenomicRanges::GRanges](#) containing input region coordinates. Rows are specific to a region, source evidence record, relationship, feature type, and coordinate evidence scope. Multiple transcript pieces belonging to the same logical mapping are collapsed. Metadata includes `overlap_bp`, `overlap_fraction`, `overlap_segments`, and a cautious interpretation.

Examples

```
drug_sets <- buildDrugGeneSets(
  sources = "custom",
  source_data = list(custom = data.frame(
    drug_name = "example drug",
    gene_symbol = "GENE1",
    relationship_category = "target",
```

```

        evidence_scope = "gene_association",
        source = "example"
    ))
)
genes <- GenomicRanges::GRanges(
  "chr1",
  IRanges::IRanges(1001, 2000),
  strand = "+",
  gene_symbol = "GENE1"
)
drug_ranges <- mapDrugGenes(
  drug_sets,
  gene_model = genes,
  features = "gene_body"
)
regions <- GenomicRanges::GRanges(
  "chr1",
  IRanges::IRanges(1200, 1300),
  region_id = "region_1"
)
annotateDrugRegions(regions, drug_ranges)

```

buildDrugGeneSets	<i>Build a controlled drug-gene relationship table</i>
-------------------	--

Description

Imports supported source tables into a common long-form [S4Vectors::DataFrame](#). DGIdb and DrugCentral are target-interaction sources and are therefore mapped only to relationship_category = "target". They are never used to infer efficacy, toxicity, or dosage associations.

Usage

```

buildDrugGeneSets(
  sources = c("PharmGKB", "DGIdb", "DrugCentral"),
  source_data = NULL,
  cache = NULL,
  refresh = FALSE,
  organism = "Homo sapiens"
)

```

Arguments

sources	Character vector containing any of "ClinPGx" (or its "PharmGKB" alias), "DGIdb", "DrugCentral", or "custom".
source_data	Named list of local file paths, data frames, or S4Vectors::DataFrame objects. Names identify sources. A ClinPGx entry can be summaryAnnotations.zip, summary_annotations.tsv, or a compatible older clinical annotation table. A custom entry must already use the canonical columns described below.

cache	NULL, a cache directory, or a BiocFileCache::BiocFileCache object used for public downloads.
refresh	Logical scalar. If TRUE, public cached resources are downloaded again.
organism	Character vector used to filter DrugCentral records. The default retains human records. Use NULL to retain all organisms.

Details

ClinPGx/PharmGKB summary annotations, DGIdb interactions, and DrugCentral targets are downloaded into a [BiocFileCache::BiocFileCache](#) when their entries are omitted from source_data.

Supported relationship categories are target, pharmacodynamics, metabolism, transport, carrier, efficacy, toxicity, dosage, Other, resistance, and exposure.

evidence_scope describes the scope of the source statement. Coordinate scope is recalculated by [mapDrugGenes\(\)](#) for each mapped genomic feature. In particular, an rsID is not coordinate-exact until matching coordinates are supplied to mapDrugGenes().

Value

A [S4Vectors::DataFrame](#) with one row per source evidence statement. Canonical columns are drug_id, drug_name, gene_id, gene_symbol, relationship_category, interaction_type, direction, phenotype, phenotype_category, evidence_level, evidence_scope, source, source_record_id, genome_build, and variant_id. Source-specific provenance columns are retained.

Examples

```
dgidb <- data.frame(
  gene_claim_name = "HGNC:1595",
  gene_concept_id = "hgnc:1595",
  gene_name = "CYP2C19",
  drug_claim_name = "CHEBI:37941",
  drug_concept_id = "chembl:1771",
  drug_name = "CLOPIDOGREL",
  interaction_source_db_name = "ExampleSource",
  interaction_types = "substrate",
  evidence_score = 1
)
sets <- buildDrugGeneSets(
  sources = "DGIdb",
  source_data = list(DGIdb = dgidb)
)
sets
```

mapDrugGenes

Map drug-associated genes to genomic features

Description

Joins a canonical drug-gene table to an explicit gene model and returns genomic ranges for each requested feature and evidence statement. Promoters are derived from transcript starts for TxDb inputs. An untyped GRanges model is treated as gene bodies and can also yield strand-aware promoters.

Usage

```
mapDrugGenes(
  drug_sets,
  genome = "GRCh38",
  gene_model = NULL,
  features = c("gene_body", "promoter"),
  promoter_upstream = 2000L,
  promoter_downstream = 500L,
  org_db = NULL,
  org_db_keytype = "ENTREZID",
  org_db_symbol_column = "SYMBOL",
  variant_ranges = NULL
)
```

Arguments

drug_sets	A canonical S4Vectors::DataFrame , normally returned by buildDrugGeneSets() .
genome	Genome build label. "GRCh38" and "GRCh37" can resolve installed human TxDb packages when gene_model is NULL.
gene_model	A TxDb or GenomicRanges::GRanges object. For a GRanges, provide gene_id or gene_symbol metadata. Typed models may also provide feature_type.
features	Any of coding_exon, utr, intron, gene_body, and promoter.
promoter_upstream, promoter_downstream	Non-negative promoter widths relative to transcript or gene-body transcription starts.
org_db	Optional AnnotationDbi::AnnotationDb or mapping table used to add gene symbols to identifier-only models.
org_db_keytype, org_db_symbol_column	Columns used with org_db.
variant_ranges	Optional GenomicRanges::GRanges with a variant_id metadata column. A matching record yields an exact_variant range.

Details

Feature-derived coordinate scope is explicit:

- promoter ranges use `regulatory_overlap`;
- gene bodies, coding exons, UTRs, and introns use `gene_association`;
- only supplied variant coordinates use `exact_variant`.

A source statement mentioning an rsID is not treated as an exact genomic overlap unless coordinates for that rsID are supplied.

Value

A `GenomicRanges::GRanges` with one or more genomic pieces per source record and feature type. `mapping_id` identifies pieces belonging to the same logical annotation. Unmapped drug-gene records are retained in `metadata(result)$unmapped_records`.

Examples

```
drug_sets <- buildDrugGeneSets(
  sources = "custom",
  source_data = list(custom = data.frame(
    drug_id = "drug:1",
    drug_name = "example drug",
    gene_id = "gene:1",
    gene_symbol = "GENE1",
    relationship_category = "metabolism",
    evidence_scope = "gene_association",
    source = "example"
  ))
)
genes <- GenomicRanges::GRanges(
  "chr1",
  IRanges::IRanges(1001, 2000),
  strand = "+",
  gene_id = "gene:1",
  gene_symbol = "GENE1"
)
mapDrugGenes(
  drug_sets,
  genome = "GRCh38",
  gene_model = genes,
  features = c("gene_body", "promoter")
)
```

`summariseDrugAnnotations`*Summarise coordinate-level drug annotations*

Description

Aggregates annotation rows without calculating enrichment p-values, drug rankings, or effect-direction predictions.

Usage

```
summariseDrugAnnotations(annotations, by = c("region", "drug", "relationship"))
```

Arguments

<code>annotations</code>	A GenomicRanges::GRanges returned by annotateDrugRegions() .
<code>by</code>	Grouping dimensions. Supported values are region, gene, drug, relationship, feature, evidence_scope, and source.

Details

Overlap base pairs are unioned within each input region before summation, preventing duplicated source records or transcript fragments from inflating coverage. `strongest_evidence_level` ranks only recognised PharmGKB-style levels (1A, 1B, 2A, 2B, 3, 4). It is NA when multiple unranked evidence labels cannot be compared safely.

Value

A [S4Vectors::DataFrame](#) with the grouping columns plus applicable counts, unioned overlap base pairs, represented categories, evidence scopes, evidence levels, exact-variant presence, and source provenance. Summary columns for dimensions already present in `by` are omitted.

Examples

```
drug_sets <- buildDrugGeneSets(  
  sources = "custom",  
  source_data = list(custom = data.frame(  
    drug_name = "example drug",  
    gene_symbol = "GENE1",  
    relationship_category = "target",  
    evidence_scope = "gene_association",  
    source = "example"  
  ))  
)  
genes <- GenomicRanges::GRanges(  
  "chr1",  
  IRanges::IRanges(1001, 2000),  
  strand = "+",
```

```
      gene_symbol = "GENE1"
    )
mapped <- mapDrugGenes(
  drug_sets,
  gene_model = genes,
  features = "gene_body"
)
regions <- GenomicRanges::GRanges(
  "chr1",
  IRanges::IRanges(1200, 1300),
  region_id = "region_1"
)
annotations <- annotateDrugRegions(regions, mapped)
summariseDrugAnnotations(annotations)
```

Index

`annotateDrugRegions`, [2](#)
`annotateDrugRegions()`, [7](#)
`AnnotationDbi::AnnotationDb`, [5](#)

`BiocFileCache::BiocFileCache`, [4](#)
`buildDrugGeneSets`, [3](#)
`buildDrugGeneSets()`, [5](#)

`GenomicRanges::GRanges`, [2](#), [5–7](#)

`mapDrugGenes`, [5](#)
`mapDrugGenes()`, [2](#), [4](#)

`S4Vectors::DataFrame`, [3–5](#), [7](#)
`summariseDrugAnnotations`, [7](#)