

# Package: GeneTrackR (via r-universe)

October 1, 2026

**Type** Package

**Title** Genomic Track Visualization and Haplotype Analysis

**Version** 0.99.2

**Date** 2026-09-01

**Description** Provides tools to read, validate, query, transform, write, and visualize gene annotations, genomic signal tracks, and variant tracks using a shared genomic coordinate framework. Supports GenePred, GTF, GFF3, and BED annotations; bedGraph, wig, and bigWig signals; VCF variants; integrated locus visualization; haplotype construction; phenotype association; linkage disequilibrium; haplotype refinement; and variant-effect analysis. The package interoperates with GenomicRanges objects and uses a pure-R signal I/O layer so core workflows do not require a compiled extension.

**License** GPL-3

**URL** <https://renscq.github.io/GeneTrackR/>,  
<https://github.com/Renscq/GeneTrackR>

**BugReports** <https://github.com/Renscq/GeneTrackR/issues>

**biocViews** Software, GenomeAnnotation, VariantAnnotation, Coverage, Sequencing, Visualization

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 4.5.0)

**Imports** data.table, ggplot2, patchwork, rlang, utils, stats, grid, grDevices, tools, GenomicRanges, IRanges, S4Vectors, RColorBrewer

**Suggests** testthat (>= 3.0.0), roxygen2, knitr, rmarkdown, Rsamtools, openxlsx

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/Needs/website** pkgdown  
**NeedsCompilation** no  
**Config/roxygen2/version** 8.1.0  
**Repository** <https://biocstaging.r-universe.dev>  
**Date/Publication** 2026-09-01 09:05:02 UTC  
**RemoteUrl** <https://github.com/BiocStaging/GeneTrackR>  
**RemoteRef** HEAD  
**RemoteSha** 5d4a1b2b9dc145527eafe4a201bc5d838feeffb5

## Contents

|                                              |    |
|----------------------------------------------|----|
| as_exon_table . . . . .                      | 4  |
| as_feature . . . . .                         | 4  |
| as_feature_table . . . . .                   | 5  |
| as_gene_table . . . . .                      | 6  |
| as_genepred . . . . .                        | 6  |
| as_granges . . . . .                         | 7  |
| as_transcript_table . . . . .                | 8  |
| benchmark_tabix_query . . . . .              | 8  |
| bin_bwg . . . . .                            | 9  |
| BwgTrack . . . . .                           | 10 |
| compute_ld_block . . . . .                   | 11 |
| diagnose_tabix . . . . .                     | 12 |
| Feature . . . . .                            | 13 |
| FeatureTrack . . . . .                       | 14 |
| GenePred . . . . .                           | 15 |
| GeneTrackR-advanced-parameters . . . . .     | 16 |
| GeneTrackR-demo-data . . . . .               | 18 |
| get_gene_length_distribution_table . . . . . | 19 |
| hap_gene_variant . . . . .                   | 20 |
| hap_region_variant . . . . .                 | 22 |
| hap_variant . . . . .                        | 23 |
| head.BwgTrack . . . . .                      | 25 |
| LDTrack . . . . .                            | 25 |
| merge_bwg . . . . .                          | 26 |
| merge_feature . . . . .                      | 27 |
| norm_bwg . . . . .                           | 28 |
| plot_feature_track . . . . .                 | 29 |
| plot_gene . . . . .                          | 31 |
| plot_gene_length_distribution . . . . .      | 32 |
| plot_hap_pheno . . . . .                     | 34 |
| plot_hap_refined . . . . .                   | 36 |
| plot_hap_variant . . . . .                   | 37 |
| plot_ld_block . . . . .                      | 40 |
| plot_pheno . . . . .                         | 42 |

|                                     |     |
|-------------------------------------|-----|
| plot_refined_hap_pheno . . . . .    | 43  |
| plot_refined_hap_variant . . . . .  | 46  |
| plot_region . . . . .               | 49  |
| plot_signal_gene . . . . .          | 51  |
| plot_signal_region . . . . .        | 54  |
| plot_signal_transcript . . . . .    | 57  |
| plot_tracks . . . . .               | 61  |
| plot_transcript . . . . .           | 65  |
| plot_variant . . . . .              | 67  |
| plot_variant_effect . . . . .       | 68  |
| plot_variant_pheno . . . . .        | 70  |
| plot_variant_track . . . . .        | 72  |
| print.GeneTrackRPhenoPlot . . . . . | 74  |
| read_bed . . . . .                  | 74  |
| read_bwg . . . . .                  | 75  |
| read_genepred . . . . .             | 77  |
| read_gff . . . . .                  | 78  |
| read_gtf . . . . .                  | 79  |
| read_pheno . . . . .                | 80  |
| read_vcf . . . . .                  | 81  |
| read_vcf_track . . . . .            | 82  |
| refine_haplotype . . . . .          | 83  |
| retrieve_bwg . . . . .              | 84  |
| retrieve_feature . . . . .          | 86  |
| retrieve_vcf . . . . .              | 88  |
| seqinfo_bwg . . . . .               | 89  |
| slice_bwg . . . . .                 | 90  |
| summary_bwg . . . . .               | 91  |
| summary_feature . . . . .           | 92  |
| summary_pheno . . . . .             | 93  |
| summary_vcf . . . . .               | 94  |
| validate_bwg . . . . .              | 95  |
| validate_feature . . . . .          | 95  |
| validate_vcf . . . . .              | 96  |
| VariantTrack . . . . .              | 97  |
| write_bwg . . . . .                 | 97  |
| write_feature . . . . .             | 98  |
| write_feature_track . . . . .       | 100 |
| write_genepred . . . . .            | 101 |
| write_variant_track . . . . .       | 102 |
| write_vcf . . . . .                 | 103 |

---

|               |                                                  |
|---------------|--------------------------------------------------|
| as_exon_table | <i>Extract exon table from a GenePred object</i> |
|---------------|--------------------------------------------------|

---

**Description**

Extract exon table from a GenePred object

**Usage**

```
as_exon_table(object)
```

**Arguments**

object            A GenePred object.

**Value**

A data.table with exon-level annotation.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
head(as_exon_table(gp))
```

---

|            |                                                                |
|------------|----------------------------------------------------------------|
| as_feature | <i>Coerce an annotation object to a unified Feature object</i> |
|------------|----------------------------------------------------------------|

---

**Description**

Coerce an annotation object to a unified Feature object

**Usage**

```
as_feature(object)
```

```
## S3 method for class 'Feature'
as_feature(object)
```

```
## S3 method for class 'FeatureTrack'
as_feature(object)
```

```
## S3 method for class 'GenePred'
as_feature(object)

## S3 method for class 'data.frame'
as_feature(object)
```

### Arguments

object                    A GenePred, FeatureTrack, or data.frame-like object.

### Value

A Feature object.

### Examples

```
x <- data.frame(chrom = "chr1", start = 1L, end = 100L)
as_feature(x)
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
feature <- as_feature(gp)
feature
```

---

|                  |                                               |
|------------------|-----------------------------------------------|
| as_feature_table | <i>Extract the standardized feature table</i> |
|------------------|-----------------------------------------------|

---

### Description

Extract the standardized feature table

### Usage

```
as_feature_table(object)
```

### Arguments

object                    A Feature, FeatureTrack, or GenePred object.

### Value

A data.table with standardized feature columns.

### Examples

```
x <- Feature(data.frame(chrom = "chr1", start = 1L, end = 100L))
as_feature_table(x)
```

---

|               |                                                  |
|---------------|--------------------------------------------------|
| as_gene_table | <i>Extract gene table from a GenePred object</i> |
|---------------|--------------------------------------------------|

---

**Description**

Extract gene table from a GenePred object

**Usage**

```
as_gene_table(object)
```

**Arguments**

object            A GenePred object.

**Value**

A data.table with gene-level annotation.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
head(as_gene_table(gp))
```

---

|             |                                                                         |
|-------------|-------------------------------------------------------------------------|
| as_genepred | <i>Convert a Feature object with gene-model information to GenePred</i> |
|-------------|-------------------------------------------------------------------------|

---

**Description**

Convert a Feature object with gene-model information to GenePred

**Usage**

```
as_genepred(object)
```

**Arguments**

object            A Feature object containing transcript and exon features.

**Value**

A GenePred object.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
as_genepred(gp)
```

as\_granges

*Convert a GeneTrackR genomic track object to GRanges***Description**

Converts GeneTrackR annotation, signal, and variant objects to `GenomicRanges::GRanges` so they can be used directly with Bioconductor genomic interval infrastructure. Annotation objects support gene, transcript, exon, and feature levels. In-memory `BwgTrack` objects expose `sample_id` and signal score metadata, while `VariantTrack` objects retain their VCF-derived metadata columns. Lazy signal or variant objects should be retrieved by genomic region before conversion.

**Usage**

```
as_granges(object, level = c("gene", "transcript", "exon", "feature"))
```

**Arguments**

|                     |                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | A <code>GenePred</code> , <code>Feature</code> , <code>FeatureTrack</code> , <code>BwgTrack</code> , or <code>VariantTrack</code> object.                                                             |
| <code>level</code>  | Annotation feature level. One of <code>gene</code> , <code>transcript</code> , <code>exon</code> , or <code>feature</code> . Ignored for <code>BwgTrack</code> and <code>VariantTrack</code> objects. |

**Value**

A `GenomicRanges::GRanges` object.

**Examples**

```
gp_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
gp <- read_genepred(gp_file, format = "genePredExt", verbose = FALSE)
as_granges(gp, level = "gene")

signal <- BwgTrack(
  samples = data.frame(sample_id = "sampleA"),
  data = data.frame(
    sample_id = "sampleA", chrom = "chr1", start = 10L, end = 12L,
    value = 1.5, strand = "*"
  )
)
as_granges(signal)
```

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
as_granges(vcf)
```

---

`as_transcript_table`     *Extract transcript table from a GenePred object*

---

### Description

Extract transcript table from a GenePred object

### Usage

```
as_transcript_table(object)
```

### Arguments

`object`                    A GenePred object.

### Value

A data.table with transcript-level annotation.

### Examples

```
gp_file <- system.file(
  "extdata", "gtr_demo_genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
head(as_transcript_table(gp))
```

---

`benchmark_tabix_query`     *Test whether a bedGraph tabix query returns records without full-file fallback*

---

### Description

Test whether a bedGraph tabix query returns records without full-file fallback

### Usage

```
benchmark_tabix_query(object, chrom, start, end, samples = NULL)
```

**Arguments**

|         |                      |
|---------|----------------------|
| object  | A BwgTrack object.   |
| chrom   | Chromosome name.     |
| start   | Region start.        |
| end     | Region end.          |
| samples | Optional sample IDs. |

**Value**

A data.table with per-sample query timing and number of records.

**Examples**

```
signal_file <- system.file(
  "extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"
)
signal <- read_bwg(
  signal_file, format = "bedgraph", sample_names = "RNA_plus",
  strand = "+", mode = "lazy", verbose = FALSE
)
benchmark_tabix_query(signal, "chr1", 12339001L, 12340000L)
```

---

|         |                                                   |
|---------|---------------------------------------------------|
| bin_bwg | <i>Bin signal tracks into fixed-width windows</i> |
|---------|---------------------------------------------------|

---

**Description**

Bin signal tracks into fixed-width windows

**Usage**

```
bin_bwg(data, bin_size = 50L)
```

**Arguments**

|          |                                                                                                                |
|----------|----------------------------------------------------------------------------------------------------------------|
| data     | Signal table returned by retrieve_bwg. The table must contain sample_id, chrom, start, end, and value columns. |
| bin_size | Bin size in bases.                                                                                             |

**Value**

A binned signal table.

## Examples

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
dt <- retrieve_bwg(
  rnaseq, chrom = "chr1", start = 12339001, end = 12352000, strand = "+"
)
bin_bwg(dt, bin_size = 50)
```

---

BwgTrack

*Create a BwgTrack object*


---

## Description

Creates a standardized BwgTrack object. GeneTrackR schema version 2 keeps sample metadata, optional in-memory signal records, optional chromosome metadata, object metadata, and validation results in stable named slots. Public signal coordinates use 1-based closed intervals.

## Usage

```
BwgTrack(
  samples,
  data = NULL,
  meta = list(),
  validation = make_empty_validation(),
  seqinfo = NULL
)
```

## Arguments

|                         |                                                                                                                                                                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>samples</code>    | Sample metadata table containing <code>sample_id</code> .                                                                                                                                                                                           |
| <code>data</code>       | Optional in-memory signal table.                                                                                                                                                                                                                    |
| <code>meta</code>       | Metadata list. <code>coordinate</code> , <code>schema_version</code> , <code>backend</code> , and <code>mode</code> are normalized by the constructor.                                                                                              |
| <code>validation</code> | Validation result list.                                                                                                                                                                                                                             |
| <code>seqinfo</code>    | Optional chromosome metadata containing <code>chrom</code> and <code>length</code> . If <code>sample_id</code> is absent, the chromosome metadata are shared across all samples. Unknown chromosome lengths can be represented by <code>NA</code> . |

## Value

A BwgTrack object with `samples`, `data`, `seqinfo`, `meta`, and `validation` slots.

**Examples**

```
samples <- data.frame(sample_id = "sample1")
BwgTrack(samples)
```

---

|                  |                                                                  |
|------------------|------------------------------------------------------------------|
| compute_ld_block | <i>Compute a linkage disequilibrium block from VCF genotypes</i> |
|------------------|------------------------------------------------------------------|

---

**Description**

Calculates pairwise LD for variants in a genomic region. Genotypes are parsed from the sample columns preserved by `read_vcf(..., keep_genotype = TRUE)`. The current implementation uses genotype dosage, where 0 is homozygous reference, 1 is heterozygous/non-reference carrier, and 2 is homozygous alternate for diploid biallelic records. Multi-allelic non-reference alleles are collapsed into total non-reference dosage.

**Usage**

```
compute_ld_block(
  vcf,
  chrom = NULL,
  start = NULL,
  end = NULL,
  variant_type = c("both", "snp", "ind"),
  method = c("r2", "Dprime"),
  samples = NULL,
  min_pair_samples = 3L,
  ploidy = 2L,
  keep_genotype_matrix = FALSE,
  verbose = TRUE
)
```

**Arguments**

|                               |                                                                                                                                          |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <code>vcf</code>              | A VariantTrack object returned by <code>read_vcf()</code> or a VCF path.                                                                 |
| <code>chrom</code>            | Chromosome. If NULL, a single-chromosome in-memory VCF uses its full chromosome range.                                                   |
| <code>start</code>            | Region start. If NULL, the minimum retained variant position is used.                                                                    |
| <code>end</code>              | Region end. If NULL, the maximum retained variant position is used.                                                                      |
| <code>variant_type</code>     | One of both, snp, or ind. ind keeps INS/DEL/MNV records.                                                                                 |
| <code>method</code>           | LD value used as the primary ld column. Currently r2 and Dprime are supported. Dprime is a dosage-based approximation for unphased VCFs. |
| <code>samples</code>          | Optional sample names to keep.                                                                                                           |
| <code>min_pair_samples</code> | Minimum number of paired non-missing samples required for a pairwise estimate.                                                           |

|                      |                                                                                       |
|----------------------|---------------------------------------------------------------------------------------|
| ploidy               | Ploidy used for dosage-based allele-frequency and Dprime approximation. Default is 2. |
| keep_genotype_matrix | Logical. Whether to store the variant-by-sample dosage matrix in the returned object. |
| verbose              | Logical. Whether to print progress messages.                                          |

**Value**

An LDTrack object.

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
ld <- compute_ld_block(
  vcf,
  chrom = "chr1",
  start = 12342620,
  end = 12355500,
  variant_type = "snp",
  method = "r2",
  verbose = FALSE
)
ld$data
```

---

diagnose\_tabix

*Diagnose tabix availability for bedGraph tracks*


---

**Description**

Diagnose tabix availability for bedGraph tracks

**Usage**

```
diagnose_tabix(object)
```

**Arguments**

object            A BwgTrack object created by read\_bwg().

**Value**

A data.table reporting index presence, backend availability, and whether indexed querying is enabled.

## Examples

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph",
  sample_names = "RNA_seq_plus",
  strand = "+",
  mode = "lazy",
  verbose = FALSE
)
diagnose_tabix(rnaseq)
```

---

Feature

---

Create a standardized Feature object

---

## Description

`Feature()` is the unified annotation class used by `GeneTrackR`. BED, GFF, GTF, and `GenePred` annotations are all normalized to a `Feature`-compatible object with a flat interval table in `object$data`. `GenePred`-compatible inputs additionally store derived genes, transcripts, and exons tables, allowing the same object to be used by gene model plotting functions.

Required coordinate convention is 1-based closed intervals.

## Usage

```
Feature(
  data,
  meta = list(),
  genes = NULL,
  transcripts = NULL,
  exons = NULL,
  validation = make_empty_validation()
)
```

## Arguments

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code>        | A <code>data.frame</code> or <code>data.table</code> containing standardized interval columns. Required columns are <code>chrom</code> , <code>start</code> , and <code>end</code> . Recommended columns include <code>feature_id</code> , <code>name</code> , <code>type</code> , <code>score</code> , <code>strand</code> , <code>source</code> , <code>gene_id</code> , <code>transcript_id</code> , <code>parent_id</code> , <code>level</code> , and <code>gene_type</code> . |
| <code>meta</code>        | Metadata list such as source file, original format, and coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>genes</code>       | Optional gene-level table.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>transcripts</code> | Optional transcript-level table.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>exons</code>       | Optional exon-level table.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>validation</code>  | Optional validation list.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

**Value**

A Feature object, also inheriting from FeatureTrack.

**Examples**

```
x <- data.frame(chrom = "chr1", start = 1L, end = 100L)
Feature(x)
```

---

FeatureTrack

---

*Create a FeatureTrack object*


---

**Description**

FeatureTrack() is kept as a user-facing alias of Feature() for interval tracks. It returns an object inheriting from both Feature and FeatureTrack.

**Usage**

```
FeatureTrack(
  data,
  meta = list(),
  genes = NULL,
  transcripts = NULL,
  exons = NULL,
  validation = make_empty_validation()
)
```

**Arguments**

|             |                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data        | A data.frame or data.table containing standardized interval columns. Required columns are chrom, start, and end. Recommended columns include feature_id, name, type, score, strand, source, gene_id, transcript_id, parent_id, level, and gene_type. |
| meta        | Metadata list such as source file, original format, and coordinate system.                                                                                                                                                                           |
| genes       | Optional gene-level table.                                                                                                                                                                                                                           |
| transcripts | Optional transcript-level table.                                                                                                                                                                                                                     |
| exons       | Optional exon-level table.                                                                                                                                                                                                                           |
| validation  | Optional validation list.                                                                                                                                                                                                                            |

**Value**

A Feature object inheriting from FeatureTrack.

**Examples**

```
x <- data.frame(chrom = "chr1", start = 1L, end = 100L)
FeatureTrack(x)
```

---

GenePred*Create a GenePred object*

---

**Description**

Create a GenePred object

**Usage**

```
GenePred(  
  transcripts,  
  exons,  
  genes = NULL,  
  meta = list(),  
  validation = make_empty_validation()  
)
```

**Arguments**

|             |                                    |
|-------------|------------------------------------|
| transcripts | Transcript-level annotation table. |
| exons       | Exon-level annotation table.       |
| genes       | Gene-level annotation table.       |
| meta        | Metadata list.                     |
| validation  | Validation result list.            |

**Value**

A GenePred object.

**Examples**

```
gp_file <- system.file(  
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"  
)  
gp <- read_genepred(  
  gp_file,  
  format = "genePredExt",  
  verbose = FALSE,  
  progress = FALSE  
)  
GenePred(gp$transcripts, gp$exons, gp$genes)
```

---

GeneTrackR-advanced-parameters

*Advanced parameter guide for GeneTrackR*


---

## Description

This help topic summarizes frequently used advanced parameters that are shared by several GeneTrackR functions. It is intended as a quick reference for custom colors, highlighted intervals, signal transformations, sample grouping, and strand handling.

## Value

No value is returned; this is a documentation-only help topic.

## Gene model colors

Gene model functions such as `plot_gene()`, `plot_transcript()`, and `plot_region()` use `gene_palette` and `gene_colors` for exon/CDS/UTR fills. `gene_palette` accepts any palette from `RColorBrewer::brewer.pal.info`. If the number of discrete groups exceeds the palette limit, colors are interpolated automatically.

`gene_colors` can be either unnamed or named. For gene models, the most stable named form is:

```
gene_colors = c(
  CDS = "#33a02c",
  UTR = "#b2df8a",
  exon = "#fb9a99"
)
```

Unnamed colors are matched to the fixed gene-model levels UTR, CDS, and exon, so colors remain stable even when a plot contains only one feature type.

## Highlight intervals

`highlight` must be a data frame with at least `start` and `end` columns:

```
highlight = data.frame(start = 105000, end = 106000)
```

In genomic plots, `start` and `end` are genomic coordinates. In transcript coordinate plots, they are spliced transcript coordinates.

## Signal colors

Signal plotting functions use `signal_palette`, `signal_palette_direction`, `signal_colors`, `sample_groups`, and `signal_color_by`.

```
signal_palette = "YlGnBu"
signal_palette_direction = -1
signal_colors = c(sampleA = "#2166AC", sampleB = "#B2182B")
```

`signal_palette_direction` reverses the standard discrete palette order. Discrete sample/group/frame colors otherwise follow the RColorBrewer class order exactly, while heatmaps use a continuous gradient. Explicit `signal_colors` are treated as exact user mappings and are not reversed.

Group-level coloring is controlled by a named vector or a two-column data frame:

```
sample_groups = c(K01 = "K0", K02 = "K0", WT1 = "WT", WT2 = "WT")
signal_color_by = "group"
```

### Signal y-axis

Signal plots support transformed values and per-sample or shared y-axis ranges:

```
signal_transform = "sqrt" # none, log2, log10, or sqrt
signal_y_scale = "fixed" # free or fixed
signal_y_ticks = "range" # range or pretty
signal_y_limits = c(0, 20) # limits after signal_transform
```

`signal_y_ticks = "range"` displays integer y-axis limits only. Supplying `signal_y_limits` changes `signal_y_scale` to "fixed".

### Plot appearance

Standard track panels use `plot_theme`, `show_panel_border`, and black text. Signal geometry can be adjusted with `signal_alpha` and `signal_bar_width`:

```
plot_theme = "classic" # bw, classic, light, or minimal
show_panel_border = FALSE
signal_alpha = 0.80
signal_bar_width = 0.85
signal_track_height = 4
gene_track_height = 1
```

`signal_track_height` and `gene_track_height` control the relative panel heights in `plot_signal_transcript()`, `plot_signal_gene()`, and `plot_signal_region()`. `plot_tracks()` uses its named heights vector for the same purpose.

`signal_bar_width` is relative to each BedGraph interval. It changes the visual bar width around the interval center without changing genomic positions.

### Strand handling

Standard bigWig and wig tracks are unstranded. For such tracks, `strand_policy = "ignore_unstranded"` lets `strand = "+"` or `strand = "-"` return the same unstranded signal. Use `strand_policy = "strict"` when you only want records with explicit strand information, such as plus/minus bedGraph tracks.

### See Also

[plot\\_gene\(\)](#), [plot\\_transcript\(\)](#), [plot\\_region\(\)](#), [plot\\_signal\\_gene\(\)](#), [plot\\_signal\\_transcript\(\)](#), [plot\\_signal\\_region\(\)](#), [plot\\_tracks\(\)](#), [retrieve\\_bwg\(\)](#), [read\\_bwg\(\)](#)

---

GeneTrackR-demo-data    *GeneTrackR deterministic demo files*

---

## Description

GeneTrackR ships a small deterministic genomic dataset under `inst/extdata`. The files support executable examples, tests, and vignettes without downloading external data.

## Value

No value is returned; this is a documentation-only help topic.

## Files

The installed `extdata` directory contains:

- `gtr_demo.genePredExt`, `gtr_demo.gtf`, and `gtr_demo.gff3` for the same deterministic gene annotation;
- `gtr_demo_features.bed` for genomic interval features;
- `gtr_demo_variants.vcf` for designed variant and genotype patterns;
- `gtr_demo_pheno.tsv` for deterministic phenotypes;
- `gtr_demo.chrom.sizes` for chromosome lengths; and
- plus/minus RNA-seq and Ribo-seq `gtr_demo_*bedgraph` signal tracks.

## Provenance

The dataset is synthetic and deterministic. Canonical source tables are stored under `inst/scripts/demo_model` in the source package. `inst/scripts/generate_demo_data.R` generates the installed `gtr_demo_*` files and `inst/scripts/validate_demo_data.R` validates their designed dimensions and relationships. No external biological dataset is bundled.

## Coordinate conventions

Canonical source tables use 1-based closed coordinates. Format-specific output follows the conventions of GenePredExt, BED, bedGraph, GTF, GFF3, and VCF, and GeneTrackR readers convert those inputs to their documented internal coordinate representation.

## Examples

```
system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
```

---

get\_gene\_length\_distribution\_table

*Extract length records from a gene annotation object*


---

## Description

Build a tidy feature-length table from a Feature/GenePred-compatible annotation object. The function supports gene spans, transcript lengths, exon lengths, CDS lengths, total UTR lengths, 5' UTR lengths, and 3' UTR lengths. CDS and UTR lengths can be summarized per transcript or returned as individual genomic segments.

## Usage

```
get_gene_length_distribution_table(
  object,
  feature = c("all", "gene", "transcript", "exon", "cds", "utr", "five_utr", "three_utr"),
  unit = c("auto", "transcript", "segment"),
  transcript_length = c("spliced", "genomic"),
  chrom = NULL,
  start = NULL,
  end = NULL,
  mode = c("overlap", "within", "trim"),
  keep_zero = FALSE
)

get_genepred_length_table(...)
```

## Arguments

|                   |                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object            | A Feature or GenePred-compatible annotation object.                                                                                                                                                                                                                                                                        |
| feature           | Feature type to extract. Use all to return gene, transcript, exon, CDS, UTR, 5' UTR, and 3' UTR records.                                                                                                                                                                                                                   |
| unit              | Output unit. auto uses gene-level records for genes, transcript-level records for transcripts, transcript-level totals for CDS/UTR, and segment-level records for exons. transcript summarizes exonic features per transcript where applicable. segment returns individual exon/CDS/UTR genomic segments where applicable. |
| transcript_length | Transcript length definition. spliced uses the sum of exon lengths. genomic uses transcript span length.                                                                                                                                                                                                                   |
| chrom             | Optional chromosome filter.                                                                                                                                                                                                                                                                                                |
| start             | Optional region start in 1-based closed coordinates.                                                                                                                                                                                                                                                                       |
| end               | Optional region end in 1-based closed coordinates.                                                                                                                                                                                                                                                                         |
| mode              | Region selection mode passed to <a href="#">retrieve_feature()</a> .                                                                                                                                                                                                                                                       |

|           |                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| keep_zero | Logical. Whether to keep zero-length transcript-level CDS/UTR records. Default is FALSE.                                                        |
| ...       | Deprecated compatibility arguments forwarded by <code>get_genepred_length_table()</code> to <code>get_gene_length_distribution_table()</code> . |

### Details

Use `unit = "segment"` to inspect each exon/CDS/UTR segment separately, and `unit = "transcript"` to summarize feature lengths per transcript. `transcript_length = "spliced"` uses exon lengths, whereas `"genomic"` uses transcript span length including introns.

### Value

A `data.table` with feature-length records.

### Examples

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
get_gene_length_distribution_table(gp, feature = "cds", unit = "transcript")
```

---

|                  |                                                                      |
|------------------|----------------------------------------------------------------------|
| hap_gene_variant | <i>Build haplotypes from variants in a gene or transcript region</i> |
|------------------|----------------------------------------------------------------------|

---

### Description

Extracts variants for a specified gene or transcript, optionally including upstream and downstream flanking regions, and converts sample genotype profiles into haplotypes.

### Usage

```
hap_gene_variant(
  vcf,
  annotation,
  gene_id = NULL,
  transcript_id = NULL,
  upstream = 0L,
  downstream = 0L,
  strand_aware = TRUE,
  samples = NULL,
  variant_type = NULL,
  genotype_mode = c("code", "string"),
  missing_genotype = NA_character_,
  min_variant_number = NULL
)
```

**Arguments**

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>vcf</code>                | A VariantTrack object or VCF file path.                                                                                                                                                                                                                                                                                                                                                                     |
| <code>annotation</code>         | A gene annotation object used to locate <code>gene_id</code> or <code>transcript_id</code> .                                                                                                                                                                                                                                                                                                                |
| <code>gene_id</code>            | Optional gene ID. Use exactly one of <code>gene_id</code> or <code>transcript_id</code> .                                                                                                                                                                                                                                                                                                                   |
| <code>transcript_id</code>      | Optional transcript ID. Use exactly one of <code>gene_id</code> or <code>transcript_id</code> .                                                                                                                                                                                                                                                                                                             |
| <code>upstream</code>           | Upstream flanking length in bp. When <code>strand_aware</code> = TRUE, upstream is interpreted relative to the gene/transcript strand.                                                                                                                                                                                                                                                                      |
| <code>downstream</code>         | Downstream flanking length in bp. When <code>strand_aware</code> = TRUE, downstream is interpreted relative to the gene/transcript strand.                                                                                                                                                                                                                                                                  |
| <code>strand_aware</code>       | Logical. Whether upstream/downstream should follow strand direction. Default TRUE.                                                                                                                                                                                                                                                                                                                          |
| <code>samples</code>            | Optional sample names to keep.                                                                                                                                                                                                                                                                                                                                                                              |
| <code>variant_type</code>       | Optional variant types to keep.                                                                                                                                                                                                                                                                                                                                                                             |
| <code>genotype_mode</code>      | Genotype representation. <code>code</code> converts genotypes to compact 0/1 states, where 0 means reference genotype and 1 means any alternate allele is present. <code>string</code> converts genotypes to a single allele label; long InDel alleles are compressed as iN, where N is the displayed REF or ALT allele length, so the haplotype cell uses the same label as the corresponding REF/ALT row. |
| <code>missing_genotype</code>   | Missing genotype label. Default is <code>NA_character_</code> , which is displayed as NA in haplotype tables.                                                                                                                                                                                                                                                                                               |
| <code>min_variant_number</code> | Minimum number of non-missing variants required for a sample. If NULL, only samples with complete non-missing genotypes across all retained variants are kept.                                                                                                                                                                                                                                              |

**Value**

A HapVariant object.

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_gene_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  upstream = 1000, downstream = 500, min_variant_number = 1
)
hap
hap$haplotypes
hap_tx <- hap_gene_variant(
  vcf, annotation = anno, transcript_id = "TxA1",
  genotype_mode = "string", min_variant_number = 1
)
hap_tx$haplotypes
```

---

|                    |                                                           |
|--------------------|-----------------------------------------------------------|
| hap_region_variant | <i>Build haplotypes from variants in a genomic region</i> |
|--------------------|-----------------------------------------------------------|

---

## Description

Extracts variants from a user-defined genomic interval and converts sample genotype profiles into haplotypes.

## Usage

```
hap_region_variant(
  vcf,
  chrom,
  start,
  end,
  samples = NULL,
  variant_type = NULL,
  genotype_mode = c("code", "string"),
  missing_genotype = NA_character_,
  min_variant_number = NULL
)
```

## Arguments

|                    |                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| vcf                | A VariantTrack object or VCF file path.                                                                                                                                                                                                                                                                                                                                                                     |
| chrom              | Chromosome name.                                                                                                                                                                                                                                                                                                                                                                                            |
| start              | Region start in 1-based closed coordinates.                                                                                                                                                                                                                                                                                                                                                                 |
| end                | Region end in 1-based closed coordinates.                                                                                                                                                                                                                                                                                                                                                                   |
| samples            | Optional sample names to keep.                                                                                                                                                                                                                                                                                                                                                                              |
| variant_type       | Optional variant types to keep.                                                                                                                                                                                                                                                                                                                                                                             |
| genotype_mode      | Genotype representation. <code>code</code> converts genotypes to compact 0/1 states, where 0 means reference genotype and 1 means any alternate allele is present. <code>string</code> converts genotypes to a single allele label; long InDel alleles are compressed as iN, where N is the displayed REF or ALT allele length, so the haplotype cell uses the same label as the corresponding REF/ALT row. |
| missing_genotype   | Missing genotype label. Default is <code>NA_character_</code> , which is displayed as NA in haplotype tables.                                                                                                                                                                                                                                                                                               |
| min_variant_number | Minimum number of non-missing variants required for a sample. If <code>NULL</code> , only samples with complete non-missing genotypes across all retained variants are kept.                                                                                                                                                                                                                                |

## Value

A HapVariant object.

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
hap <- hap_region_variant(
  vcf, chrom = "chr1", start = 12339700, end = 12352000,
  genotype_mode = "code", min_variant_number = 1
)
hap
hap$haplotypes
```

---

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| hap_variant | <i>Build haplotypes from variants in a gene, transcript, or genomic region</i> |
|-------------|--------------------------------------------------------------------------------|

---

**Description**

Compatibility wrapper for haplotype construction. For new code, prefer `hap_gene_variant()` for gene/transcript queries and `hap_region_variant()` for direct genomic interval queries.

**Usage**

```
hap_variant(
  vcf,
  annotation = NULL,
  gene_id = NULL,
  transcript_id = NULL,
  chrom = NULL,
  start = NULL,
  end = NULL,
  upstream = 0L,
  downstream = 0L,
  strand_aware = TRUE,
  samples = NULL,
  variant_type = NULL,
  genotype_mode = c("code", "string"),
  missing_genotype = NA_character_,
  min_variant_number = NULL
)
```

**Arguments**

|               |                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------|
| vcf           | A VariantTrack object or VCF file path.                                                              |
| annotation    | Optional gene annotation object used for <code>gene_id</code> or <code>transcript_id</code> queries. |
| gene_id       | Optional gene ID.                                                                                    |
| transcript_id | Optional transcript ID.                                                                              |
| chrom         | Optional chromosome name.                                                                            |

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| start              | Optional region start in 1-based closed coordinates.                                                                                                                                                                                                                                                                                                                                                                      |
| end                | Optional region end in 1-based closed coordinates.                                                                                                                                                                                                                                                                                                                                                                        |
| upstream           | Upstream flanking length in bp for gene/transcript queries.                                                                                                                                                                                                                                                                                                                                                               |
| downstream         | Downstream flanking length in bp for gene/transcript queries.                                                                                                                                                                                                                                                                                                                                                             |
| strand_aware       | Logical. Whether upstream/downstream should follow strand direction. Default TRUE.                                                                                                                                                                                                                                                                                                                                        |
| samples            | Optional sample names to keep.                                                                                                                                                                                                                                                                                                                                                                                            |
| variant_type       | Optional variant types to keep.                                                                                                                                                                                                                                                                                                                                                                                           |
| genotype_mode      | Genotype representation. <code>code</code> converts genotypes to compact 0/1 states, where 0 means reference genotype and 1 means any alternate allele is present. <code>string</code> converts genotypes to a single allele label; long InDel alleles are compressed as <code>iN</code> , where N is the displayed REF or ALT allele length, so the haplotype cell uses the same label as the corresponding REF/ALT row. |
| missing_genotype   | Missing genotype label. Default is <code>NA_character_</code> , which is displayed as NA in haplotype tables.                                                                                                                                                                                                                                                                                                             |
| min_variant_number | Minimum number of non-missing variants required for a sample. If NULL, only samples with complete non-missing genotypes across all retained variants are kept.                                                                                                                                                                                                                                                            |

## Value

A HapVariant object.

## Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "code", min_variant_number = 1
)
hap
hap_region <- hap_variant(
  vcf, chrom = "chr1", start = 12339700, end = 12352000,
  min_variant_number = 1
)
hap_region$haplotypes
```

---

|               |                                                 |
|---------------|-------------------------------------------------|
| head.BwgTrack | <i>Show the first rows of a BwgTrack object</i> |
|---------------|-------------------------------------------------|

---

**Description**

Show the first rows of a BwgTrack object

**Usage**

```
## S3 method for class 'BwgTrack'
head(x, n = 6L, ...)
```

**Arguments**

|     |                         |
|-----|-------------------------|
| x   | A BwgTrack object.      |
| n   | Number of rows to show. |
| ... | Additional arguments.   |

**Value**

Invisibly returns the input object.

---

|         |                                 |
|---------|---------------------------------|
| LDTrack | <i>Create an LDTrack object</i> |
|---------|---------------------------------|

---

**Description**

LDTrack() stores pairwise linkage disequilibrium results, retained variant metadata, the selected genomic region, and optionally the dosage matrix and the generated figure. It follows the same lightweight S3 list-class style used by VariantTrack and HapVariant.

**Usage**

```
LDTrack(
  data = NULL,
  matrix = NULL,
  variants = NULL,
  region = list(),
  genotype = NULL,
  figure = NULL,
  meta = list(),
  plot = NULL
)
```

Arguments

|          |                                                                                           |
|----------|-------------------------------------------------------------------------------------------|
| data     | A data.frame/data.table of pairwise LD statistics.                                        |
| matrix   | A symmetric LD matrix. Diagonal values are usually 1.                                     |
| variants | Variant metadata used in the calculation.                                                 |
| region   | A list with chrom, start, and end.                                                        |
| genotype | Optional numeric genotype dosage matrix.                                                  |
| figure   | Optional figure object produced by plot_ld_block().                                       |
| meta     | A metadata list.                                                                          |
| plot     | Deprecated alias for figure, kept only for compatibility with older development versions. |

Value

An LDTrack object.

Examples

```
LDTrack(region = list(chrom = "chr1", start = 1L, end = 100L))
```

---

|           |                                   |
|-----------|-----------------------------------|
| merge_bwg | <i>Merge signal track objects</i> |
|-----------|-----------------------------------|

---

Description

Merge signal track objects

Usage

```
merge_bwg(  
  ...,  
  sample_conflict = c("error", "rename", "sum", "mean", "keep_first"),  
  require_same_norm = TRUE  
)
```

Arguments

|                   |                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ...               | BwgTrack objects.                                                                                                                                                                                |
| sample_conflict   | Conflict strategy for duplicated sample IDs. error stops, rename makes sample IDs unique, sum/mean combines records with identical genomic intervals, and keep_first keeps the first occurrence. |
| require_same_norm | Require identical normalization labels before merging.                                                                                                                                           |

**Value**

A merged BwgTrack object.

**Examples**

```
plus <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
minus <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_minus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_minus", strand = "-",
  mode = "memory", verbose = FALSE
)
signal_all <- merge_bwg(plus, minus)
signal_all
```

---

merge\_feature

---

Merge annotation feature objects

---

**Description**

merge\_feature() combines annotations read from GenePred, GTF, GFF, or BED files into one unified Feature object for downstream retrieval and plotting. Inputs may represent different genes, genomic regions, annotation formats, or independently retrieved subsets.

Duplicate identifiers are detected across input objects. For non-error conflict strategies, a warning reports duplicated gene, transcript, and feature IDs before the selected strategy is applied. With conflict = "error", the function stops directly without a preliminary warning.

With conflict = "deduplicate", input order defines precedence. The first complete copy of a duplicated transcript is retained, duplicated feature records are removed, and gene ranges are recalculated from all retained non-duplicated transcripts. With conflict = "rename", only conflicting identifiers in later inputs are renamed, and all hierarchical references are updated together.

**Usage**

```
merge_feature(
  ...,
  source_names = NULL,
  conflict = c("deduplicate", "rename", "keep_all", "error", "keep_first"),
  rename_prefix = NULL,
  sort = TRUE
)
```

Arguments

|               |                                                                                                                                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ...           | One or more Feature, FeatureTrack, or GenePred objects. A single list containing such objects is also accepted.                                                                                                                                                                                                             |
| source_names  | Optional unique source labels. If omitted, names from a supplied list or named arguments are used where available; remaining labels are generated as track1, track2, and so on. The labels are stored in the track_source column.                                                                                           |
| conflict      | Duplicate-ID strategy. "deduplicate" keeps the first compatible annotation model according to input order. "rename" renames conflicting IDs in later inputs. "keep_all" retains conflicts unchanged, and "error" stops after duplicate detection. "keep_first" is accepted as a backward-compatible alias of "deduplicate". |
| rename_prefix | Optional character vector with one prefix per input. Used only when conflict = "rename". By default, source_names followed by an underscore are used.                                                                                                                                                                       |
| sort          | Logical. Whether to sort the merged feature and hierarchy tables.                                                                                                                                                                                                                                                           |

Value

A merged Feature object. If transcript and exon tables are present, the result also inherits from GenePred.

Examples

```
annotation <- read_genepred(  
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),  
  format = "genePredExt", verbose = FALSE, progress = FALSE  
)  
gene_a <- retrieve_feature(annotation, gene_id = "GeneA")  
gene_b <- retrieve_feature(annotation, gene_id = "GeneB")  
merged_genes <- merge_feature(gene_a, gene_b)  
merged_genes
```

---

|          |                                |
|----------|--------------------------------|
| norm_bwg | <i>Normalize signal tracks</i> |
|----------|--------------------------------|

---

Description

Normalize signal tracks

Usage

```
norm_bwg(  
  object,  
  method = c("none", "RPM", "CPM", "scale", "custom"),  
  library_size = NULL,  
  scale_factor = NULL,  
  custom_factor = NULL  
)
```

**Arguments**

|               |                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------|
| object        | A BwgTrack object.                                                                                          |
| method        | Normalization method. Use RPM/CPM with library_size, scale with scale_factor, or custom with custom_factor. |
| library_size  | Optional named library sizes.                                                                               |
| scale_factor  | Optional named scale factors.                                                                               |
| custom_factor | Optional named custom factors.                                                                              |

**Value**

A normalized BwgTrack object.

**Examples**

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph",
  sample_names = "RNA_seq_plus",
  strand = "+",
  mode = "memory",
  verbose = FALSE
)
norm_bwg(rnaseq, method = "scale", scale_factor = c(RNA_seq_plus = 0.5))
```

---

|                    |                                   |
|--------------------|-----------------------------------|
| plot_feature_track | <i>Plot a FeatureTrack object</i> |
|--------------------|-----------------------------------|

---

**Description**

Draws BED/GFF/GTF-derived interval features in a genomic region. This function is designed to be compatible with plot\_tracks().

**Usage**

```
plot_feature_track(
  track,
  chrom,
  start,
  end,
  mode = c("overlap", "within", "trim"),
  color_by = "auto",
  label_by = c("none", "name", "feature_id", "type"),
  feature_palette = "Paired",
  feature_colors = NULL,
  feature_border_color = NA,
  max_legend_levels = 5,
```

```

    text_size = 14,
    plot_theme = c("bw", "classic", "light", "minimal"),
    show_panel_border = NULL
  )

```

### Arguments

|                      |                                                                                                                                                                         |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| track                | A FeatureTrack object.                                                                                                                                                  |
| chrom                | Chromosome name.                                                                                                                                                        |
| start                | Region start.                                                                                                                                                           |
| end                  | Region end.                                                                                                                                                             |
| mode                 | overlap, within, or trim.                                                                                                                                               |
| color_by             | Column used for fill colors. Use auto to choose a compact, informative grouping automatically. Common manual choices are feature_group, type, source, name, and strand. |
| label_by             | Column used for labels. Use none to hide labels.                                                                                                                        |
| feature_palette      | RColorBrewer palette name for feature fills.                                                                                                                            |
| feature_colors       | Optional named or unnamed feature fill color vector.                                                                                                                    |
| feature_border_color | Rectangle border color. Use NA to hide borders.                                                                                                                         |
| max_legend_levels    | Maximum number of displayed legend groups when the selected color attribute contains many categories.                                                                   |
| text_size            | Text size.                                                                                                                                                              |
| plot_theme           | Base ggplot2 theme. Use bw, classic, light, or minimal.                                                                                                                 |
| show_panel_border    | Whether to draw the panel border. NULL preserves the selected theme default.                                                                                            |

### Value

A ggplot object.

### Examples

```

bed_file <- system.file(
  "extdata", "gtr_demo_features.bed", package = "GeneTrackR"
)
features <- read_bed(bed_file, verbose = FALSE, progress = FALSE)
plot_feature_track(
  features, chrom = "chr1", start = 12339001L, end = 12352000L
)

```

---

|           |                              |
|-----------|------------------------------|
| plot_gene | <i>Plot a gene structure</i> |
|-----------|------------------------------|

---

## Description

Plot a gene structure

## Usage

```
plot_gene(
  object,
  gene_id,
  collapse = c("none", "union_exon", "longest"),
  coordinate = c("genomic", "transcript"),
  show_cds = TRUE,
  cds_height = 0.5,
  utr_height = 0.25,
  direction_mode = c("transcript", "gene", "end", "none"),
  highlight = NULL,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
  label_position = c("axis", "feature"),
  label_by = c("gene", "transcript"),
  text_size = 14,
  plot_theme = c("bw", "classic", "light", "minimal"),
  show_panel_border = NULL
)
```

## Arguments

|                |                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object         | A GenePred object.                                                                                                                                                                                                                                                                                                         |
| gene_id        | Gene ID.                                                                                                                                                                                                                                                                                                                   |
| collapse       | Transcript collapse mode.                                                                                                                                                                                                                                                                                                  |
| coordinate     | genomic or transcript.                                                                                                                                                                                                                                                                                                     |
| show_cds       | Whether to distinguish CDS and UTR segments.                                                                                                                                                                                                                                                                               |
| cds_height     | Vertical thickness of CDS rectangles.                                                                                                                                                                                                                                                                                      |
| utr_height     | Vertical thickness of UTR/non-coding exon rectangles.                                                                                                                                                                                                                                                                      |
| direction_mode | Direction-arrow style. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows.                                                                                                                           |
| highlight      | Optional data frame used to shade genomic or transcript intervals. It must contain start and end columns. Optional columns such as label or group are allowed but are not required. In coordinate = "genomic", start and end are genomic positions; in coordinate = "transcript", they are spliced transcript coordinates. |

|                   |                                                                                                                                                                                                                                                                                                                                |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gene_palette      | RColorBrewer palette name used for discrete fills. If the number of discrete groups exceeds the palette maximum, colors are automatically interpolated.                                                                                                                                                                        |
| gene_colors       | Optional custom fill colors for gene model features. Use a named vector such as <code>c(UTR = "#b2df8a", CDS = "#33a02c", exon = "#fb9a99")</code> to map colors explicitly. Unnamed colors are matched to the fixed gene-model levels UTR, CDS, and exon, so colors remain stable even when only one feature type is present. |
| gene_border_color | Optional rectangle border color. Use NA to hide borders, or a color such as "black" or "grey30" to draw feature outlines.                                                                                                                                                                                                      |
| label_position    | Where to draw feature labels. <code>axis</code> draws labels on the y axis and <code>feature</code> draws labels at the center of each gene/transcript structure.                                                                                                                                                              |
| label_by          | Which identifier to use for feature labels. Use <code>gene</code> for gene IDs or <code>transcript</code> for transcript IDs.                                                                                                                                                                                                  |
| text_size         | Text size in points for axis text, axis titles, and legends.                                                                                                                                                                                                                                                                   |
| plot_theme        | Base ggplot2 theme. Use <code>bw</code> , <code>classic</code> , <code>light</code> , or <code>minimal</code> .                                                                                                                                                                                                                |
| show_panel_border | Whether to draw the panel border. NULL preserves the selected theme default.                                                                                                                                                                                                                                                   |

## Details

Common feature names used by `gene_colors` are UTR, CDS, and exon. `highlight` must contain start and end columns. In genomic coordinate plots these are genomic positions; in transcript coordinate plots they are spliced transcript positions. See also [GeneTrackR-advanced-parameters](#).

## Value

A ggplot object.

## Examples

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
plot_gene(gp, gene_id = "GeneA")
```

---

plot\_gene\_length\_distribution

*Plot gene annotation feature length distributions*

---

## Description

Plot the length distribution of genes, transcripts, exons, CDS, UTRs, 5' UTRs, and 3' UTRs from any Feature/GenePred-compatible annotation object. The function is designed for annotation quality control and comparison of coding versus non-coding feature lengths.

**Usage**

```

plot_gene_length_distribution(
  object,
  feature = c("all", "gene", "transcript", "exon", "cds", "utr", "five_utr", "three_utr"),
  unit = c("auto", "transcript", "segment"),
  transcript_length = c("spliced", "genomic"),
  chrom = NULL,
  start = NULL,
  end = NULL,
  mode = c("overlap", "within", "trim"),
  group_by = c("gene_type", "feature", "strand", "chrom", "none"),
  plot_type = c("density", "histogram", "boxplot", "violin"),
  scale = c("log10", "linear"),
  bins = 60L,
  facet = TRUE,
  keep_zero = FALSE,
  fill_palette = "Paired",
  fill_colors = NULL,
  border_color = NA,
  return_data = FALSE
)

plot_genepred_length_distribution(...)

```

**Arguments**

|                   |                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------|
| object            | A Feature or GenePred-compatible annotation object.                                                       |
| feature           | Feature type to plot. Use all to facet multiple feature types.                                            |
| unit              | Output unit used for length extraction. See <a href="#">get_gene_length_distribution_table()</a> .        |
| transcript_length | Transcript length definition for transcript records.                                                      |
| chrom             | Optional chromosome filter.                                                                               |
| start             | Optional region start in 1-based closed coordinates.                                                      |
| end               | Optional region end in 1-based closed coordinates.                                                        |
| mode              | Region selection mode passed to <a href="#">retrieve_feature()</a> .                                      |
| group_by          | Grouping variable. Common choices are gene_type, feature, strand, and chrom.                              |
| plot_type         | Plot type. One of density, histogram, boxplot, or violin.                                                 |
| scale             | Length scale. log10 is recommended for genomic features because feature lengths are usually right-skewed. |
| bins              | Number of bins for histogram.                                                                             |
| facet             | Logical. Whether to facet by feature when multiple features are requested.                                |
| keep_zero         | Logical. Whether to keep zero-length CDS/UTR records.                                                     |
| fill_palette      | RColorBrewer palette name used for grouped fills. Default is Paired.                                      |

|              |                                                                                                                                                                                                                                                     |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fill_colors  | Optional custom fill colors. Named vectors are matched to the values of group_by, for example <code>c(coding = "#1f78b4", non-coding = "#a6cee3")</code> . Unnamed colors are matched in plotting order and are automatically extended when needed. |
| border_color | Optional border color for histograms, boxplots, and violin plots. Use NA to hide borders.                                                                                                                                                           |
| return_data  | Logical. If TRUE, return a list containing the plot and the underlying length table.                                                                                                                                                                |
| ...          | Deprecated compatibility arguments forwarded by <code>plot_genepred_length_distribution()</code> to <code>plot_gene_length_distribution()</code> .                                                                                                  |

### Details

fill\_colors follows the values of group\_by. For example, if group\_by = "gene\_type", names should match coding and non-coding; if group\_by = "feature", names can include gene, transcript, exon, cds, utr, five\_utr, and three\_utr. Use return\_data = TRUE to inspect the exact groups before assigning named colors. See also [GeneTrackR-advanced-parameters](#).

### Value

A ggplot object, or a list with plot and data when return\_data = TRUE.

### Examples

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
plot_gene_length_distribution(gp, feature = "gene", plot_type = "histogram")
```

---

|                |                                                   |
|----------------|---------------------------------------------------|
| plot_hap_pheno | <i>Plot phenotype values grouped by haplotype</i> |
|----------------|---------------------------------------------------|

---

### Description

Merges haplotype assignments and phenotype values, then draws phenotype distributions for each haplotype. Haplotype groups are ordered by sample number from left to right. Fill colors are mapped to the median phenotype value of each haplotype group.

### Usage

```
plot_hap_pheno(
  hap,
  phenotype,
  traits = NULL,
  sample_col = "sample_id",
  min_hap_samples = 2L,
  plot_type = c("violin_boxplot", "violin", "boxplot"),
  test_method = c("t.test", "wilcox.test", "ks.test"),
```

```

p_adjust = "BH",
p_label = c("stars", "number", "both"),
p_cutoff = 0.05,
p_value_type = c("raw", "adjusted"),
show_signif_only = TRUE,
show_points = FALSE,
show_outliers = FALSE,
fill_palette = "Paired",
fill_colors = NULL,
fill_alpha = 0.75,
violin_width = 0.9,
box_width = 0.18,
bracket_step = 0.08,
bracket_tip_fraction = 0.12,
x_text_angle = 90,
facet_ncol = 3L,
strip_label_width = 24,
strip_text_lineheight = 0.9,
strip_fill = "white",
strip_border_color = NULL,
text_size = 14
)

```

## Arguments

|                  |                                                                           |
|------------------|---------------------------------------------------------------------------|
| hap              | A HapVariant object from hap_variant().                                   |
| phenotype        | A phenotype table returned by read_pheno() or a compatible data.frame.    |
| traits           | Phenotype trait names. If NULL, all numeric traits are used.              |
| sample_col       | Sample column name in phenotype table.                                    |
| min_hap_samples  | Minimum sample number required for a haplotype group.                     |
| plot_type        | Plot type. One of violin, boxplot, or violin_boxplot.                     |
| test_method      | Pairwise test method. One of t.test, wilcox.test, or ks.test.             |
| p_adjust         | P-value adjustment method passed to p.adjust().                           |
| p_label          | P-value label style. One of stars, number, or both.                       |
| p_cutoff         | Significance cutoff used for displaying pairwise comparisons.             |
| p_value_type     | Which p-value is used for filtering and labeling. One of raw or adjusted. |
| show_signif_only | Logical. Whether to only display significant comparisons.                 |
| show_points      | Logical. Whether to show sample points.                                   |
| show_outliers    | Logical. Whether to show boxplot outliers.                                |
| fill_palette     | RColorBrewer palette name used for median-based haplotype fills.          |
| fill_colors      | Optional custom fill colors for haplotypes.                               |
| fill_alpha       | Alpha value for violin/boxplot fill colors.                               |

|                       |                                                                                |
|-----------------------|--------------------------------------------------------------------------------|
| violin_width          | Violin plot width.                                                             |
| box_width             | Boxplot width.                                                                 |
| bracket_step          | Fraction of y-range used to separate significance brackets.                    |
| bracket_tip_fraction  | Fraction of bracket vertical spacing used for the short downward bracket tips. |
| x_text_angle          | Rotation angle for haplotype labels on the x-axis.                             |
| facet_ncol            | Maximum number of facet columns when multiple traits are plotted.              |
| strip_label_width     | Maximum character width for wrapping long facet strip labels.                  |
| strip_text_lineheight | Line height for wrapped facet strip labels.                                    |
| strip_fill            | Facet strip background fill color.                                             |
| strip_border_color    | Facet strip border color. Use NULL to remove the border.                       |
| text_size             | Text size.                                                                     |

### Value

A list with figure and pvalue elements. Additional elements include summary, bracket, and plot\_data.

### Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo_genePredExt", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "code", min_variant_number = 1
)
pheno <- read_pheno(pheno_file, verbose = FALSE)
plot_hap_pheno(hap, phenotype = pheno, traits = "seed_weight", min_hap_samples = 3)
plot_hap_pheno(hap, phenotype = pheno, traits = "seed_weight", min_hap_samples = 3,
  test_method = "wilcox.test", p_label = "number")
```

---

|                  |                                                            |
|------------------|------------------------------------------------------------|
| plot_hap_refined | <i>Plot phenotype distributions for refined haplotypes</i> |
|------------------|------------------------------------------------------------|

---

### Description

Legacy compatibility alias of plot\_refined\_hap\_pheno().

### Usage

```
plot_hap_refined(refined_hap, phenotype, ...)
```

**Arguments**

refined\_hap      A HapRefined object returned by refine\_haplotype() or a refined HapVariant object.

phenotype        A phenotype table returned by read\_pheno() or a compatible data.frame.

...                Additional parameters passed to plot\_refined\_hap\_pheno().

**Value**

A GeneTrackRPhenoPlot object returned by plot\_hap\_pheno().

**Examples**

```
vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
vcf <- read_vcf(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
pheno_file <- system.file(
  "extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR"
)
pheno <- read_pheno(pheno_file, verbose = FALSE)
gp_file <- system.file(
  "extdata", "gtr_demo_genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
hap <- hap_variant(
  vcf, annotation = gp, gene_id = "GeneA",
  genotype_mode = "string", min_variant_number = 1
)
refined <- refine_haplotype(
  hap, phenotype = pheno, traits = "protein_content",
  min_hap_samples = 3, effect_threshold = 0.5
)
plot_hap_refined(
  refined, phenotype = pheno, traits = "protein_content",
  min_hap_samples = 3
)
```

---

plot\_hap\_variant

---

*Plot gene variants and haplotype table*


---

**Description**

Draws a haplotype-variant figure with a compact gene model track, variant markers, connector lines, and a haplotype genotype table. Genomic gaps are compressed by mapping variant positions to evenly spaced table columns while retaining the relative order of gene model features.

**Usage**

```

plot_hap_variant(
  hap,
  annotation = NULL,
  show_gene_model = TRUE,
  min_hap_samples = 5L,
  show_reference_row = TRUE,
  variant_label = c("variant_id", "pos"),
  show_gene_pos_axis = TRUE,
  gene_pos_axis_n = 5L,
  gene_pos_axis_label = NULL,
  gene_pos_x_angle = 0,
  gene_track_legend_position = c("right", "top", "none"),
  direction_mode = c("transcript", "gene", "end", "none"),
  text_size = 14,
  table_x_angle = 90,
  genotype_text_size = 3.2,
  gene_track_height = 1.25,
  connector_height = 0.35,
  table_height = NULL,
  exon_height = 0.22,
  cds_height = 0.44,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
  table_palette = "Paired",
  table_colors = NULL,
  table_alpha = 0.6,
  reference_fill = "white",
  variant_palette = "Paired",
  variant_colors = NULL,
  variant_alpha = 0.6,
  show_variant_marker = TRUE,
  variant_marker_size = 2.8
)

```

**Arguments**

|                    |                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------|
| hap                | A HapVariant object from hap_variant().                                                                              |
| annotation         | Optional Feature/GenePred annotation object. When supplied, a compact gene track is drawn above the haplotype table. |
| show_gene_model    | Logical. Whether to draw the gene track when annotation is supplied.                                                 |
| min_hap_samples    | Minimum sample number required for a haplotype group to be displayed.                                                |
| show_reference_row | Logical. Whether to add two reference rows showing REF and ALT alleles for each variant.                             |

|                            |                                                                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| variant_label              | Column used for variant labels. One of variant_id, pos, or an existing column in hap\$variants.                                                                                                                             |
| show_gene_pos_axis         | Logical. Whether to show genomic coordinate labels above the gene track.                                                                                                                                                    |
| gene_pos_axis_n            | Approximate number of genomic coordinate ticks above the gene track.                                                                                                                                                        |
| gene_pos_axis_label        | Optional x-axis title for genomic coordinate labels.                                                                                                                                                                        |
| gene_pos_x_angle           | Angle of gene-position x-axis labels. Default is 0.                                                                                                                                                                         |
| gene_track_legend_position | Legend position for variant-type markers in the gene track. One of right, top, or none.                                                                                                                                     |
| direction_mode             | Gene-strand arrow style for the compact gene track. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws a short arrow at the directional end of each gene, and none hides direction arrows. |
| text_size                  | Base text size for gene-track labels, axes, legends, and table axes.                                                                                                                                                        |
| table_x_angle              | Angle of haplotype table x-axis labels. Default is 90.                                                                                                                                                                      |
| genotype_text_size         | Genotype cell text size only. Default is 3.2.                                                                                                                                                                               |
| gene_track_height          | Relative height of the gene track panel.                                                                                                                                                                                    |
| connector_height           | Relative height of the connector panel.                                                                                                                                                                                     |
| table_height               | Relative height of the haplotype table panel.                                                                                                                                                                               |
| exon_height                | Height of exon boxes in the compact gene track.                                                                                                                                                                             |
| cds_height                 | Height of CDS boxes in the compact gene track.                                                                                                                                                                              |
| gene_palette               | RColorBrewer palette name used for gene model feature fills.                                                                                                                                                                |
| gene_colors                | Optional custom fill colors for gene model features.                                                                                                                                                                        |
| gene_border_color          | Border color for gene model feature boxes.                                                                                                                                                                                  |
| table_palette              | RColorBrewer palette name used for haplotype table fills. For allele-string haplotypes, the first five colors are always assigned in the fixed order A, T, C, G, and indel.                                                 |
| table_colors               | Optional custom fill colors for haplotype table genotypes. For allele-string haplotypes, use a vector named A, T, C, G, and indel, or supply five unnamed colors in that order.                                             |
| table_alpha                | Alpha value for haplotype table fill colors.                                                                                                                                                                                |
| reference_fill             | Background fill color for the REF and ALT reference rows.                                                                                                                                                                   |
| variant_palette            | RColorBrewer palette name used for solid variant-type triangle marker colors.                                                                                                                                               |
| variant_colors             | Optional custom colors for solid variant-type triangle markers.                                                                                                                                                             |

variant\_alpha    Alpha value for solid variant-type triangle marker colors.

show\_variant\_marker  
                 Logical. Whether to draw natural-variant triangle markers.

variant\_marker\_size  
                 Size of natural-variant triangle markers. Set to 0 to hide markers.

## Value

A patchwork object with attributes plot\_data, variant\_data, and gene\_data.

## Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "string", min_variant_number = 1
)
plot_hap_variant(hap, annotation = anno, min_hap_samples = 1)
plot_ld_block(
  hap,
  annotation = anno,
  min_hap_samples = 1,
  table_x_angle = 90,
  table_palette = "Paired"
)
```

---

|               |                                     |
|---------------|-------------------------------------|
| plot_ld_block | <i>Plot a triangular LD heatmap</i> |
|---------------|-------------------------------------|

---

## Description

Draws an inverted triangular LD heatmap using ggplot2. For exactly two variants, the single pairwise LD value is drawn as one diamond-shaped heatmap cell, matching the 45-degree geometry used by the triangular LD matrix. Interior grid lines are suppressed; only the outside frame is drawn. The function accepts either an LDTrack object or a VariantTrack/VCF path, in which case LD is computed first by compute\_ld\_block(). The default return value is an updated LDTrack object: LD calculation results remain in the object and the generated plot is stored in LDTrack\$figure. When show\_region = TRUE, the region track follows the compact gene-model style used by plot\_hap\_variant() in GeneTrackR 0.3.15 and connects genomic variant markers to LD heatmap columns with a shared x scale, so line endpoints remain aligned after resizing.

**Usage**

```

plot_ld_block(
  object,
  chrom = NULL,
  start = NULL,
  end = NULL,
  variant_type = c("both", "snp", "ind"),
  method = c("r2", "Dprime"),
  color_palette = "Reds",
  font = 14,
  title = NULL,
  label_by = c("pos", "variant_id"),
  show_variant_labels = TRUE,
  show_region = FALSE,
  show_variant_marker = TRUE,
  variant_marker_size = 2.8,
  annotation = NULL,
  connect_region = TRUE,
  region_height = 1.25,
  connector_height = 0.35,
  heatmap_height = NULL,
  samples = NULL,
  min_pair_samples = 3L,
  ploidy = 2L,
  verbose = TRUE,
  return_object = TRUE
)

```

**Arguments**

|                     |                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------|
| object              | An LDTrack object, a VariantTrack object, or a VCF path.                                     |
| chrom, start, end   | Optional region used when object is not an LDTrack.                                          |
| variant_type        | One of both, snp, or ind when computing from VCF.                                            |
| method              | LD method used when computing from VCF.                                                      |
| color_palette       | RColorBrewer palette name used to generate the continuous LD heatmap gradient. Default Reds. |
| font                | Base font size. Font color is always black.                                                  |
| title               | Plot title. Default is chrom:start-end LD.                                                   |
| label_by            | Variant label column, either pos, variant_id, or another column in LDTrack\$variants.        |
| show_variant_labels | Logical. Whether to show labels above the heatmap.                                           |
| show_region         | Logical. Whether to add a compact region/variant track above the LD heatmap.                 |
| show_variant_marker | Logical. Whether to draw natural-variant triangle markers in the region track.               |

variant\_marker\_size      Size of natural-variant triangle markers. Set to 0 to hide markers.

annotation      Optional gene annotation used by the compact region track.

connect\_region      Logical. Whether to connect region-track variant positions to heatmap columns.

region\_height      Relative height of the compact region track.

connector\_height      Relative height of connector lines.

heatmap\_height      Relative height of the heatmap. If NULL, it is chosen from the number of variants.

samples, min\_pair\_samples, ploidy, verbose      Passed to compute\_ld\_block() when object is not already an LDTrack.

return\_object      Logical. Default TRUE. If TRUE, return an updated LDTrack containing the figure in \$figure; if FALSE, return the figure directly for compatibility with older scripts.

### Value

By default, an updated LDTrack object with the generated figure stored in \$figure. If return\_object = FALSE, the ggplot/patchwork figure is returned directly.

### Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo_genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
anno <- read_genePred(anno_file, format = "genePredExt", verbose = FALSE)
ld <- compute_ld_block(
  vcf,
  chrom = "chr1",
  start = 12342620,
  end = 12355500,
  variant_type = "snp",
  verbose = FALSE
)
ld <- plot_ld_block(ld, show_region = TRUE, annotation = anno, show_variant_labels = FALSE)
ld$figure
```

---

plot\_pheno

*Plot phenotype distributions*

---

### Description

Plot phenotype distributions

**Usage**

```
plot_pheno(
  pheno,
  traits = NULL,
  sample_col = "sample_id",
  bins = 30L,
  text_size = 14
)
```

**Arguments**

|            |                                                        |
|------------|--------------------------------------------------------|
| pheno      | A phenotype data.frame/data.table.                     |
| traits     | Optional trait names. If NULL, all traits are plotted. |
| sample_col | Sample column name.                                    |
| bins       | Histogram bins for numeric traits.                     |
| text_size  | Text size.                                             |

**Value**

A ggplot object.

**Examples**

```
pheno_file <- system.file(
  "extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR"
)
pheno <- read_pheno(pheno_file, verbose = FALSE)
plot_pheno(pheno, traits = "protein_content")
```

---

plot\_refined\_hap\_pheno

*Plot phenotype distributions for refined haplotypes*

---

**Description**

Draw phenotype distributions for refined haplotype groups generated by `refine_haplotype()`. This function is a refined-haplotype wrapper around `plot_hap_pheno()`, but all commonly used phenotype-plot parameters are exposed explicitly so that the help page shows the complete usage.

**Usage**

```
plot_refined_hap_pheno(
  refined_hap,
  phenotype,
  traits = NULL,
  sample_col = "sample_id",
```

```

min_hap_samples = 2L,
plot_type = c("violin_boxplot", "violin", "boxplot"),
test_method = c("t.test", "wilcox.test", "ks.test"),
p_adjust = "BH",
p_label = c("stars", "number", "both"),
p_cutoff = 0.05,
p_value_type = c("raw", "adjusted"),
show_signif_only = TRUE,
show_points = FALSE,
show_outliers = FALSE,
fill_palette = "Paired",
fill_colors = NULL,
fill_alpha = 0.75,
violin_width = 0.9,
box_width = 0.18,
bracket_step = 0.08,
bracket_tip_fraction = 0.12,
x_text_angle = 90,
strip_label_width = 24,
strip_text_lineheight = 0.9,
strip_fill = "white",
strip_border_color = NULL,
text_size = 14
)

```

### Arguments

|                  |                                                                                     |
|------------------|-------------------------------------------------------------------------------------|
| refined_hap      | A HapRefined object returned by refine_haploptype() or a refined HapVariant object. |
| phenotype        | A phenotype table returned by read_pheno() or a compatible data.frame.              |
| traits           | Phenotype trait names. If NULL, all numeric traits are used.                        |
| sample_col       | Sample column name in phenotype table.                                              |
| min_hap_samples  | Minimum sample number required for a refined haplotype group.                       |
| plot_type        | Plot type. One of violin, boxplot, or violin_boxplot.                               |
| test_method      | Pairwise test method. One of t.test, wilcox.test, or ks.test.                       |
| p_adjust         | P-value adjustment method passed to p.adjust().                                     |
| p_label          | P-value label style. One of stars, number, or both.                                 |
| p_cutoff         | Significance cutoff used for displaying pairwise comparisons.                       |
| p_value_type     | Which p-value is used for filtering and labeling. One of raw or adjusted.           |
| show_signif_only | Logical. Whether to only display significant comparisons.                           |
| show_points      | Logical. Whether to show sample points.                                             |
| show_outliers    | Logical. Whether to show boxplot outliers.                                          |
| fill_palette     | RColorBrewer palette name used for median-based refined haplotype fills.            |

|                       |                                                                                |
|-----------------------|--------------------------------------------------------------------------------|
| fill_colors           | Optional custom fill colors for refined haplotypes.                            |
| fill_alpha            | Alpha value for violin/boxplot fill colors.                                    |
| violin_width          | Violin plot width.                                                             |
| box_width             | Boxplot width.                                                                 |
| bracket_step          | Fraction of y-range used to separate significance brackets.                    |
| bracket_tip_fraction  | Fraction of bracket vertical spacing used for the short downward bracket tips. |
| x_text_angle          | Rotation angle for refined haplotype labels on the x-axis.                     |
| strip_label_width     | Maximum character width for wrapping long facet strip labels.                  |
| strip_text_lineheight | Line height for wrapped facet strip labels.                                    |
| strip_fill            | Facet strip background fill color.                                             |
| strip_border_color    | Facet strip border color. Use NULL to remove the border.                       |
| text_size             | Text size.                                                                     |

**Value**

A GeneTrackRPhenoPlot object returned by plot\_hap\_pheno().

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo_genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
pheno <- read_pheno(pheno_file, verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "string", min_variant_number = 1
)
refined <- refine_haplotype(
  hap,
  phenotype = pheno,
  traits = "protein_content",
  min_hap_samples = 3,
  effect_threshold = 0.5
)
plot_refined_hap_pheno(
  refined, phenotype = pheno, traits = "protein_content",
  min_hap_samples = 3
)
```

---

plot\_refined\_hap\_variant

*Plot grouped variant table for refined haplotypes*


---

## Description

Draw a refined haplotype-variant table. By default, when the input is a HapRefined object, the original haplotype rows are retained and arranged into refined haplotype blocks. For example, if ten original haplotypes are refined into three groups, the table shows the ten original haplotypes split into three vertically separated blocks. This makes it easier to inspect which variants are shared or variable within each refined group.

If `collapse_refined = TRUE`, the function falls back to a collapsed refined table in which each refined haplotype is shown as one row, using the same visual grammar as `plot_hap_variant()`.

## Usage

```
plot_refined_hap_variant(
  refined_hap,
  annotation = NULL,
  show_gene_model = TRUE,
  min_hap_samples = 5L,
  show_reference_row = TRUE,
  collapse_refined = FALSE,
  group_gap = 0.8,
  show_group_label = TRUE,
  group_label_prefix = "",
  group_label_hjust = 3.4,
  group_label_size = NULL,
  group_label_left_margin = NULL,
  variant_label = c("variant_id", "pos"),
  show_gene_pos_axis = TRUE,
  gene_pos_axis_n = 5L,
  gene_pos_axis_label = NULL,
  gene_pos_x_angle = 0,
  gene_track_legend_position = c("right", "top", "none"),
  text_size = 14,
  table_x_angle = 90,
  genotype_text_size = 3.2,
  gene_track_height = 1.25,
  connector_height = 0.35,
  table_height = NULL,
  exon_height = 0.22,
  cds_height = 0.44,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
```

```

    table_palette = "Paired",
    table_colors = NULL,
    table_alpha = 0.6,
    reference_fill = "white",
    variant_palette = "Paired",
    variant_colors = NULL
)

```

## Arguments

|                                      |                                                                                                                                                                                                                        |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>refined_hap</code>             | A HapRefined object returned by <code>refine_haplotype()</code> or a refined HapVariant object. The grouped display requires a HapRefined object generated by the current version of <code>refine_haplotype()</code> . |
| <code>annotation</code>              | Optional Feature/GenePred annotation object. When supplied, a compact gene track is drawn above the refined haplotype table.                                                                                           |
| <code>show_gene_model</code>         | Logical. Whether to draw the gene track when annotation is supplied.                                                                                                                                                   |
| <code>min_hap_samples</code>         | Minimum sample number required for an original haplotype row.                                                                                                                                                          |
| <code>show_reference_row</code>      | Logical. Whether to add two reference rows showing REF and ALT alleles for each variant.                                                                                                                               |
| <code>collapse_refined</code>        | Logical. Whether to plot one collapsed row per refined haplotype group.                                                                                                                                                |
| <code>group_gap</code>               | Numeric vertical gap inserted between refined haplotype blocks.                                                                                                                                                        |
| <code>show_group_label</code>        | Logical. Whether to show refined group labels on the left side of the table.                                                                                                                                           |
| <code>group_label_prefix</code>      | Prefix used for left-side group labels.                                                                                                                                                                                |
| <code>group_label_hjust</code>       | Horizontal adjustment for refined group labels. Larger values move the group labels further left and reduce overlap with original haplotype row labels.                                                                |
| <code>group_label_size</code>        | Text size for refined group labels. If NULL, it is derived from <code>text_size</code> .                                                                                                                               |
| <code>group_label_left_margin</code> | Left plot margin used when refined group labels are shown. Increase this value when group labels are clipped.                                                                                                          |
| <code>variant_label</code>           | Column used for variant labels. One of <code>variant_id</code> , <code>pos</code> , or an existing column in <code>hap\$variants</code> .                                                                              |
| <code>show_gene_pos_axis</code>      | Logical. Whether to show genomic coordinate labels above the gene track.                                                                                                                                               |
| <code>gene_pos_axis_n</code>         | Approximate number of genomic coordinate ticks above the gene track.                                                                                                                                                   |
| <code>gene_pos_axis_label</code>     | Optional x-axis title for genomic coordinate labels.                                                                                                                                                                   |

|                                         |                                                                                                                                                                                 |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>gene_pos_x_angle</code>           | Angle of gene-position x-axis labels.                                                                                                                                           |
| <code>gene_track_legend_position</code> | Legend position for variant-type markers in the gene track. One of right, top, or none.                                                                                         |
| <code>text_size</code>                  | Base text size for gene-track labels, axes, legends, and table axes.                                                                                                            |
| <code>table_x_angle</code>              | Angle of haplotype table x-axis labels.                                                                                                                                         |
| <code>genotype_text_size</code>         | Genotype cell text size only.                                                                                                                                                   |
| <code>gene_track_height</code>          | Relative height of the gene track panel.                                                                                                                                        |
| <code>connector_height</code>           | Relative height of the connector panel.                                                                                                                                         |
| <code>table_height</code>               | Relative height of the haplotype table panel. If NULL, it is inferred from the number of displayed rows and group gaps.                                                         |
| <code>exon_height</code>                | Height of exon boxes in the compact gene track.                                                                                                                                 |
| <code>cds_height</code>                 | Height of CDS boxes in the compact gene track.                                                                                                                                  |
| <code>gene_palette</code>               | RColorBrewer palette name used for gene model feature fills.                                                                                                                    |
| <code>gene_colors</code>                | Optional custom fill colors for gene model features.                                                                                                                            |
| <code>gene_border_color</code>          | Border color for gene model feature boxes.                                                                                                                                      |
| <code>table_palette</code>              | RColorBrewer palette name used for haplotype table fills. For allele-string haplotypes, the first five colors are always assigned in the fixed order A, T, C, G, and indel.     |
| <code>table_colors</code>               | Optional custom fill colors for haplotype table genotypes. For allele-string haplotypes, use a vector named A, T, C, G, and indel, or supply five unnamed colors in that order. |
| <code>table_alpha</code>                | Alpha value for haplotype table fill colors.                                                                                                                                    |
| <code>reference_fill</code>             | Background fill color for the REF and ALT reference rows.                                                                                                                       |
| <code>variant_palette</code>            | RColorBrewer palette name used for solid variant-type triangle marker colors.                                                                                                   |
| <code>variant_colors</code>             | Optional custom colors for solid variant-type triangle markers.                                                                                                                 |

**Value**

A patchwork object with attributes `plot_data`, `variant_data`, `gene_data`, `refined_haplotype_map`, and `refined_haplotypes` when available.

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
pheno <- read_pheno(pheno_file, verbose = FALSE)
```

```

anno <- read_genePred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "string", min_variant_number = 1
)
refined <- refine_haplotypes(
  hap,
  phenotype = pheno,
  traits = "protein_content",
  min_hap_samples = 3,
  effect_threshold = 0.5
)
plot_refined_hap_variant(refined, annotation = anno, min_hap_samples = 1)

```

---

plot\_region

---

*Plot gene structures in a genomic region*


---

## Description

Plot gene structures in a genomic region

## Usage

```

plot_region(
  object,
  chrom,
  start,
  end,
  mode = c("within", "overlap", "trim"),
  collapse = c("none", "union_exon", "longest"),
  show_cds = TRUE,
  cds_height = 0.5,
  utr_height = 0.25,
  direction_mode = c("transcript", "gene", "end", "none"),
  highlight = NULL,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
  label_position = c("axis", "feature"),
  label_by = c("gene", "transcript"),
  text_size = 14,
  plot_theme = c("bw", "classic", "light", "minimal"),
  show_panel_border = NULL
)

```

## Arguments

object            A GenePred object.

|                   |                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| chrom             | Chromosome name.                                                                                                                                                                                                                                                                                                           |
| start             | Region start.                                                                                                                                                                                                                                                                                                              |
| end               | Region end.                                                                                                                                                                                                                                                                                                                |
| mode              | within, overlap, or trim.                                                                                                                                                                                                                                                                                                  |
| collapse          | Transcript collapse mode.                                                                                                                                                                                                                                                                                                  |
| show_cds          | Whether to distinguish CDS and UTR segments.                                                                                                                                                                                                                                                                               |
| cds_height        | Vertical thickness of CDS rectangles.                                                                                                                                                                                                                                                                                      |
| utr_height        | Vertical thickness of UTR/non-coding exon rectangles.                                                                                                                                                                                                                                                                      |
| direction_mode    | Direction-arrow style. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows.                                                                                                                           |
| highlight         | Optional data frame used to shade genomic or transcript intervals. It must contain start and end columns. Optional columns such as label or group are allowed but are not required. In coordinate = "genomic", start and end are genomic positions; in coordinate = "transcript", they are spliced transcript coordinates. |
| gene_palette      | RColorBrewer palette name used for discrete fills. If the number of discrete groups exceeds the palette maximum, colors are automatically interpolated.                                                                                                                                                                    |
| gene_colors       | Optional custom fill colors for gene model features. Use a named vector such as c(UTR = "#b2df8a", CDS = "#33a02c", exon = "#fb9a99") to map colors explicitly. Unnamed colors are matched to the fixed gene-model levels UTR, CDS, and exon, so colors remain stable even when only one feature type is present.          |
| gene_border_color | Optional rectangle border color. Use NA to hide borders, or a color such as "black" or "grey30" to draw feature outlines.                                                                                                                                                                                                  |
| label_position    | Where to draw feature labels. axis draws labels on the y axis and feature draws labels at the center of each gene/transcript structure.                                                                                                                                                                                    |
| label_by          | Which identifier to use for feature labels. Use gene for gene IDs or transcript for transcript IDs.                                                                                                                                                                                                                        |
| text_size         | Text size in points for axis text, axis titles, and legends.                                                                                                                                                                                                                                                               |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                                                                                                                                                                                                                                                                    |
| show_panel_border | Whether to draw the panel border. NULL preserves the selected theme default.                                                                                                                                                                                                                                               |

## Details

Common feature names used by gene\_colors are UTR, CDS, and exon. highlight must contain start and end columns. In genomic coordinate plots these are genomic positions; in transcript coordinate plots they are spliced transcript positions. See also [GeneTrackR-advanced-parameters](#).

## Value

A ggplot object.

**Examples**

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
plot_region(gp, chrom = "chr1", start = 12339001, end = 12352000)
```

---

|                  |                                |
|------------------|--------------------------------|
| plot_signal_gene | <i>Plot signal over a gene</i> |
|------------------|--------------------------------|

---

**Description**

Plot signal over a gene

**Usage**

```
plot_signal_gene(
  signal,
  annotation,
  gene_id,
  samples = NULL,
  sample_groups = NULL,
  signal_color_by = c("sample", "group"),
  signal_summary = c("none", "mean", "median", "sum"),
  plot_type = c("bar", "line", "heatmap"),
  strand = c("auto", "+", "-", "both", "ignore"),
  bin_size = NULL,
  highlight = NULL,
  show_gene_model = TRUE,
  signal_track_height = 3,
  gene_track_height = 1,
  signal_palette = "Paired",
  signal_palette_direction = 1,
  signal_colors = NULL,
  signal_transform = c("none", "log2", "log10", "sqrt"),
  signal_y_scale = c("free", "fixed"),
  signal_y_ticks = c("range", "pretty"),
  heatmap_bin_size = NULL,
  heatmap_max_bins = 800L,
  heatmap_summary = c("mean", "max", "sum", "median"),
  cds_height = 0.5,
  utr_height = 0.25,
  direction_mode = c("transcript", "gene", "end", "none"),
  label_position = c("axis", "feature"),
  label_by = c("gene", "transcript"),
  text_size = 14,
  signal_y_limits = NULL,
```

```

    signal_alpha = 0.85,
    signal_bar_width = 1,
    plot_theme = c("bw", "classic", "light", "minimal"),
    show_panel_border = NULL
)

```

## Arguments

|                          |                                                                                                                                                                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal                   | A BwgTrack object.                                                                                                                                                                                                                                                                                  |
| annotation               | A GenePred object.                                                                                                                                                                                                                                                                                  |
| gene_id                  | Gene ID.                                                                                                                                                                                                                                                                                            |
| samples                  | Optional sample IDs to plot. If NULL, all samples are used.                                                                                                                                                                                                                                         |
| sample_groups            | Optional sample group mapping for group-level coloring or replicate summaries. Use a named character vector, a data frame with sample_id and group, or an unnamed vector with one group per selected sample.                                                                                        |
| signal_color_by          | Color signal tracks by sample or group.                                                                                                                                                                                                                                                             |
| signal_summary           | Replicate summary mode. Use none to plot individual samples, or mean, median, or sum to summarize samples within each group. Summary is performed on the current signal intervals, so using bin_size is recommended when raw interval boundaries differ among samples.                              |
| plot_type                | Signal plot type: bar, line, or heatmap.                                                                                                                                                                                                                                                            |
| strand                   | Strand selector. Use auto to use the gene strand.                                                                                                                                                                                                                                                   |
| bin_size                 | Optional bin size for signal aggregation.                                                                                                                                                                                                                                                           |
| highlight                | Optional data frame used to shade intervals on the signal and gene model tracks. It must contain start and end columns in genomic coordinates.                                                                                                                                                      |
| show_gene_model          | Whether to append the gene model track.                                                                                                                                                                                                                                                             |
| signal_track_height      | Relative height of the signal panel when the gene model is shown. Default 3.                                                                                                                                                                                                                        |
| gene_track_height        | Relative height of the gene model panel when it is shown. Default 1.                                                                                                                                                                                                                                |
| signal_palette           | Signal color palette. Any palette name from RColorBrewer::brewer.pal.info can be used, such as Blues, Reds, RdBu, Paired, Set1, Dark2, YlGnBu, or Spectral. Discrete sample/group colors are assigned in the standard RColorBrewer class order; heatmaps use the corresponding continuous gradient. |
| signal_palette_direction | Direction for generated signal colors. Use 1 for the standard palette order and -1 for the reversed palette order. Discrete sample/group colors preserve level-to-color order; heatmap gradients reverse continuously.                                                                              |
| signal_colors            | Optional named or unnamed vector of explicit colors for samples. If supplied, it overrides signal_palette; explicit colors are not modified by signal_palette_direction.                                                                                                                            |
| signal_transform         | Signal-axis transformation. Use none, log2, log10, or sqrt. Log transforms use signed log1p-style transformation to tolerate zero values.                                                                                                                                                           |

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal_y_scale    | Signal y-axis scale mode. Use free for each sample to have its own y-axis range, or fixed to force all samples to share the same y-axis range.                                                                            |
| signal_y_ticks    | Signal y-axis tick mode. Use range to show only integer axis limits as the minimum and maximum ticks, or pretty to use ggplot2 default-style breaks.                                                                      |
| heatmap_bin_size  | Optional bin size used only for heatmap rendering. NULL uses adaptive binning.                                                                                                                                            |
| heatmap_max_bins  | Maximum number of heatmap bins when heatmap_bin_size = NULL.                                                                                                                                                              |
| heatmap_summary   | Summary statistic used to aggregate signal into heatmap bins: mean, max, sum, or median.                                                                                                                                  |
| cds_height        | Vertical thickness of CDS rectangles in the gene model track.                                                                                                                                                             |
| utr_height        | Vertical thickness of UTR/non-coding exon rectangles in the gene model track.                                                                                                                                             |
| direction_mode    | Direction-arrow style for the gene model track. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows. |
| label_position    | Where to draw gene model labels. axis draws labels on the y axis and feature draws labels at the center of each gene/transcript structure.                                                                                |
| label_by          | Which identifier to use for gene model labels. Use gene for gene IDs or transcript for transcript IDs.                                                                                                                    |
| text_size         | Text size in points for signal and gene model axis text, axis titles, facet strips, and legends.                                                                                                                          |
| signal_y_limits   | Optional two-element numeric vector giving the plotted y-axis limits after signal_transform. Supplying limits changes signal_y_scale to fixed.                                                                            |
| signal_alpha      | Signal geometry transparency from 0 to 1.                                                                                                                                                                                 |
| signal_bar_width  | Relative width of bar intervals from greater than 0 to 1. A value below 1 creates proportional gaps without changing interval centers.                                                                                    |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                                                                                                                                                                   |
| show_panel_border | Whether to draw panel borders. NULL preserves the selected theme default.                                                                                                                                                 |

## Details

samples selects the samples to draw. sample\_groups can be used for group-level coloring and replicate summaries. When raw intervals differ among samples, set bin\_size before using signal\_summary so the summary is made on comparable bins. signal\_transform changes the plotted y value only; the queried signal table is not modified. See also [GeneTrackR-advanced-parameters](#).

## Value

A ggplot or patchwork object.

**Examples**

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
plot_signal_gene(signal = rnaseq, annotation = gp, gene_id = "GeneA")
```

---

|                    |                                          |
|--------------------|------------------------------------------|
| plot_signal_region | <i>Plot signal over a genomic region</i> |
|--------------------|------------------------------------------|

---

**Description**

Plot signal over a genomic region

**Usage**

```
plot_signal_region(
  signal,
  chrom,
  start,
  end,
  samples = NULL,
  sample_groups = NULL,
  signal_color_by = c("sample", "group"),
  signal_summary = c("none", "mean", "median", "sum"),
  plot_type = c("bar", "line", "heatmap"),
  strand = c("ignore", "+", "-", "both"),
  bin_size = NULL,
  highlight = NULL,
  annotation = NULL,
  show_gene_model = TRUE,
  signal_track_height = 3,
  gene_track_height = 1,
  signal_palette = "Paired",
  signal_palette_direction = 1,
  signal_colors = NULL,
  signal_transform = c("none", "log2", "log10", "sqrt"),
  signal_y_scale = c("free", "fixed"),
  signal_y_ticks = c("range", "pretty"),
  heatmap_bin_size = NULL,
  heatmap_max_bins = 800L,
  heatmap_summary = c("mean", "max", "sum", "median"),
```

```

    cds_height = 0.5,
    utr_height = 0.25,
    direction_mode = c("transcript", "gene", "end", "none"),
    label_position = c("axis", "feature"),
    label_by = c("gene", "transcript"),
    text_size = 14,
    signal_y_limits = NULL,
    signal_alpha = 0.85,
    signal_bar_width = 1,
    plot_theme = c("bw", "classic", "light", "minimal"),
    show_panel_border = NULL
)

```

### Arguments

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal              | A BwgTrack object.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| chrom               | Chromosome name.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| start               | Region start.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| end                 | Region end.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| samples             | Optional sample IDs to plot. If NULL, all samples are used.                                                                                                                                                                                                                                                                                                                                                                                   |
| sample_groups       | Optional sample group mapping for group-level coloring or replicate summaries. Use a named character vector, a data frame with <code>sample_id</code> and <code>group</code> , or an unnamed vector with one group per selected sample.                                                                                                                                                                                                       |
| signal_color_by     | Color signal tracks by sample or group.                                                                                                                                                                                                                                                                                                                                                                                                       |
| signal_summary      | Replicate summary mode. Use <code>none</code> to plot individual samples, or <code>mean</code> , <code>median</code> , or <code>sum</code> to summarize samples within each group. Summary is performed on the current signal intervals, so using <code>bin_size</code> is recommended when raw interval boundaries differ among samples.                                                                                                     |
| plot_type           | Signal plot type: <code>bar</code> , <code>line</code> , or <code>heatmap</code> .                                                                                                                                                                                                                                                                                                                                                            |
| strand              | Strand selector.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| bin_size            | Optional bin size for signal aggregation.                                                                                                                                                                                                                                                                                                                                                                                                     |
| highlight           | Optional data frame used to shade intervals on the signal and gene model tracks. It must contain <code>start</code> and <code>end</code> columns in genomic coordinates.                                                                                                                                                                                                                                                                      |
| annotation          | Optional GenePred object.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| show_gene_model     | Whether to append a gene model track. Default TRUE.                                                                                                                                                                                                                                                                                                                                                                                           |
| signal_track_height | Relative height of the signal panel when the gene model is shown. Default 3.                                                                                                                                                                                                                                                                                                                                                                  |
| gene_track_height   | Relative height of the gene model panel when it is shown. Default 1.                                                                                                                                                                                                                                                                                                                                                                          |
| signal_palette      | Signal color palette. Any palette name from <code>RColorBrewer::brewer.pal.info</code> can be used, such as <code>Blues</code> , <code>Reds</code> , <code>RdBu</code> , <code>Paired</code> , <code>Set1</code> , <code>Dark2</code> , <code>YlGnBu</code> , or <code>Spectral</code> . Discrete sample/group colors are assigned in the standard <code>RColorBrewer</code> class order; heatmaps use the corresponding continuous gradient. |

|                                       |                                                                                                                                                                                                                                                                               |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>signal_palette_direction</code> | Direction for generated signal colors. Use 1 for the standard palette order and -1 for the reversed palette order. Discrete sample/group colors preserve level-to-color order; heatmap gradients reverse continuously.                                                        |
| <code>signal_colors</code>            | Optional named or unnamed vector of explicit colors for samples. If supplied, it overrides <code>signal_palette</code> ; explicit colors are not modified by <code>signal_palette_direction</code> .                                                                          |
| <code>signal_transform</code>         | Signal-axis transformation. Use <code>none</code> , <code>log2</code> , <code>log10</code> , or <code>sqrt</code> . Log transforms use signed <code>log1p</code> -style transformation to tolerate zero values.                                                               |
| <code>signal_y_scale</code>           | Signal y-axis scale mode. Use <code>free</code> for each sample to have its own y-axis range, or <code>fixed</code> to force all samples to share the same y-axis range.                                                                                                      |
| <code>signal_y_ticks</code>           | Signal y-axis tick mode. Use <code>range</code> to show only integer axis limits as the minimum and maximum ticks, or <code>pretty</code> to use <code>ggplot2</code> default-style breaks.                                                                                   |
| <code>heatmap_bin_size</code>         | Optional bin size used only for heatmap rendering. <code>NULL</code> uses adaptive binning.                                                                                                                                                                                   |
| <code>heatmap_max_bins</code>         | Maximum number of heatmap bins when <code>heatmap_bin_size = NULL</code> .                                                                                                                                                                                                    |
| <code>heatmap_summary</code>          | Summary statistic used to aggregate signal into heatmap bins: <code>mean</code> , <code>max</code> , <code>sum</code> , or <code>median</code> .                                                                                                                              |
| <code>cds_height</code>               | Vertical thickness of CDS rectangles in the gene model track.                                                                                                                                                                                                                 |
| <code>utr_height</code>               | Vertical thickness of UTR/non-coding exon rectangles in the gene model track.                                                                                                                                                                                                 |
| <code>direction_mode</code>           | Direction-arrow style for the gene model track. <code>transcript</code> draws one arrow per transcript, <code>gene</code> draws one arrow per gene, <code>end</code> draws one short arrow at the directional end of each gene, and <code>none</code> hides direction arrows. |
| <code>label_position</code>           | Where to draw gene model labels. <code>axis</code> draws labels on the y axis and <code>feature</code> draws labels at the center of each gene/transcript structure.                                                                                                          |
| <code>label_by</code>                 | Which identifier to use for gene model labels. Use <code>gene</code> for gene IDs or <code>transcript</code> for transcript IDs.                                                                                                                                              |
| <code>text_size</code>                | Text size in points for signal and gene model axis text, axis titles, facet strips, and legends.                                                                                                                                                                              |
| <code>signal_y_limits</code>          | Optional two-element numeric vector giving the plotted y-axis limits after <code>signal_transform</code> . Supplying limits changes <code>signal_y_scale</code> to <code>fixed</code> .                                                                                       |
| <code>signal_alpha</code>             | Signal geometry transparency from 0 to 1.                                                                                                                                                                                                                                     |
| <code>signal_bar_width</code>         | Relative width of bar intervals from greater than 0 to 1. A value below 1 creates proportional gaps without changing interval centers.                                                                                                                                        |
| <code>plot_theme</code>               | Base <code>ggplot2</code> theme. Use <code>bw</code> , <code>classic</code> , <code>light</code> , or <code>minimal</code> .                                                                                                                                                  |
| <code>show_panel_border</code>        | Whether to draw panel borders. <code>NULL</code> preserves the selected theme default.                                                                                                                                                                                        |

**Details**

If annotation is supplied and `show_gene_model = TRUE`, a gene model track is appended below the signal panel. `signal_colors` can be an unnamed vector or a named vector. Named values are matched to sample IDs when `signal_color_by = "sample"`, and to group names when `signal_color_by = "group"`. See also [GeneTrackR-advanced-parameters](#).

**Value**

A ggplot or patchwork object.

**Examples**

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
rnaseq <- read_bwg(
  system.file(
    "extdata", c("gtr_demo_rnaseq_plus.bedgraph", "gtr_demo_rnaseq_minus.bedgraph"),
    package = "GeneTrackR"
  ),
  format = "bedgraph", sample_names = c("RNA_seq_plus", "RNA_seq_minus"),
  strand = c("+", "-"), mode = "memory", verbose = FALSE
)
plot_signal_region(
  signal = rnaseq, annotation = gp, chrom = "chr1",
  start = 12339001, end = 12352000, strand = "both", plot_type = "bar"
)
```

---

plot\_signal\_transcript

*Plot signal over a transcript*


---

**Description**

Plot signal over a transcript

**Usage**

```
plot_signal_transcript(
  signal,
  annotation,
  transcript_id,
  samples = NULL,
  sample_groups = NULL,
  signal_color_by = c("sample", "group"),
  signal_summary = c("none", "mean", "median", "sum"),
  coordinate = c("transcript", "genomic"),
```

```

plot_type = c("bar", "line", "heatmap", "frame"),
strand = c("auto", "+", "-", "both", "ignore"),
bin_size = NULL,
highlight = NULL,
show_gene_model = TRUE,
signal_track_height = 3,
gene_track_height = 1,
signal_palette = "Paired",
signal_palette_direction = 1,
signal_colors = NULL,
frame_palette = "Paired",
frame_colors = NULL,
signal_transform = c("none", "log2", "log10", "sqrt"),
signal_y_scale = c("free", "fixed"),
signal_y_ticks = c("range", "pretty"),
heatmap_bin_size = NULL,
heatmap_max_bins = 800L,
heatmap_summary = c("mean", "max", "sum", "median"),
cds_height = 0.5,
utr_height = 0.25,
direction_mode = c("transcript", "gene", "end", "none"),
label_position = c("axis", "feature"),
label_by = c("gene", "transcript"),
text_size = 14,
signal_y_limits = NULL,
signal_alpha = 0.85,
signal_bar_width = 1,
plot_theme = c("bw", "classic", "light", "minimal"),
show_panel_border = NULL
)

```

## Arguments

|                 |                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal          | A BwgTrack object.                                                                                                                                                                                                                                                     |
| annotation      | A GenePred object.                                                                                                                                                                                                                                                     |
| transcript_id   | Transcript ID.                                                                                                                                                                                                                                                         |
| samples         | Optional sample IDs to plot. If NULL, all samples are used.                                                                                                                                                                                                            |
| sample_groups   | Optional sample group mapping for group-level coloring or replicate summaries. Use a named character vector, a data frame with sample_id and group, or an unnamed vector with one group per selected sample.                                                           |
| signal_color_by | Color signal tracks by sample or group.                                                                                                                                                                                                                                |
| signal_summary  | Replicate summary mode. Use none to plot individual samples, or mean, median, or sum to summarize samples within each group. Summary is performed on the current signal intervals, so using bin_size is recommended when raw interval boundaries differ among samples. |

|                          |                                                                                                                                                                                                                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coordinate               | Coordinate mode. transcript removes introns and displays spliced transcript coordinates, while genomic keeps genomic coordinates.                                                                                                                                                                   |
| plot_type                | Signal plot type: bar, line, heatmap, or frame. frame is designed for Ribo-seq style transcript plots and colors CDS positions by frame0/frame1/frame2.                                                                                                                                             |
| strand                   | Strand selector. Use auto to use the transcript strand.                                                                                                                                                                                                                                             |
| bin_size                 | Optional bin size for signal aggregation.                                                                                                                                                                                                                                                           |
| highlight                | Optional data frame used to shade intervals on the signal and gene model tracks. It must contain start and end columns. For coordinate = "genomic", these are genomic coordinates; for coordinate = "transcript", these are spliced transcript coordinates.                                         |
| show_gene_model          | Whether to append the transcript gene model track. Default TRUE.                                                                                                                                                                                                                                    |
| signal_track_height      | Relative height of the signal panel when the gene model is shown. Default 3.                                                                                                                                                                                                                        |
| gene_track_height        | Relative height of the gene model panel when it is shown. Default 1.                                                                                                                                                                                                                                |
| signal_palette           | Signal color palette. Any palette name from RColorBrewer::brewer.pal.info can be used, such as Blues, Reds, RdBu, Paired, Set1, Dark2, YlGnBu, or Spectral. Discrete sample/group colors are assigned in the standard RColorBrewer class order; heatmaps use the corresponding continuous gradient. |
| signal_palette_direction | Direction for generated signal colors. Use 1 for the standard palette order and -1 for the reversed palette order. Discrete sample/group colors preserve level-to-color order; heatmap gradients reverse continuously.                                                                              |
| signal_colors            | Optional named or unnamed vector of explicit colors for samples. If supplied, it overrides signal_palette; explicit colors are not modified by signal_palette_direction.                                                                                                                            |
| frame_palette            | RColorBrewer palette for CDS frame colors when plot_type = "frame". Default is Paired. Colors are assigned in order to frame0, frame1, and frame2.                                                                                                                                                  |
| frame_colors             | Optional named colors for frame0, frame1, and frame2.                                                                                                                                                                                                                                               |
| signal_transform         | Signal-axis transformation. Use none, log2, log10, or sqrt. Log transforms use signed log1p-style transformation to tolerate zero values.                                                                                                                                                           |
| signal_y_scale           | Signal y-axis scale mode. Use free for each sample to have its own y-axis range, or fixed to force all samples to share the same y-axis range.                                                                                                                                                      |
| signal_y_ticks           | Signal y-axis tick mode. Use range to show only integer axis limits as the minimum and maximum ticks, or pretty to use ggplot2 default-style breaks.                                                                                                                                                |
| heatmap_bin_size         | Optional bin size used only for heatmap rendering. NULL uses adaptive binning.                                                                                                                                                                                                                      |
| heatmap_max_bins         | Maximum number of heatmap bins when heatmap_bin_size = NULL.                                                                                                                                                                                                                                        |
| heatmap_summary          | Summary statistic used to aggregate signal into heatmap bins: mean, max, sum, or median.                                                                                                                                                                                                            |
| cds_height               | Vertical thickness of CDS rectangles in the gene model track.                                                                                                                                                                                                                                       |

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| utr_height        | Vertical thickness of UTR/non-coding exon rectangles in the gene model track.                                                                                                                                             |
| direction_mode    | Direction-arrow style for the gene model track. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows. |
| label_position    | Where to draw gene model labels. axis draws labels on the y axis and feature draws labels on the model.                                                                                                                   |
| label_by          | Which identifier to use for gene model labels.                                                                                                                                                                            |
| text_size         | Text size in points for signal and gene model axis text, axis titles, facet strips, and legends.                                                                                                                          |
| signal_y_limits   | Optional two-element numeric vector giving the plotted y-axis limits after signal_transform. Supplying limits changes signal_y_scale to fixed.                                                                            |
| signal_alpha      | Signal geometry transparency from 0 to 1.                                                                                                                                                                                 |
| signal_bar_width  | Relative width of bar intervals from greater than 0 to 1. A value below 1 creates proportional gaps without changing interval centers.                                                                                    |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                                                                                                                                                                   |
| show_panel_border | Whether to draw panel borders. NULL preserves the selected theme default.                                                                                                                                                 |

## Details

samples selects the samples to draw. sample\_groups can be a named vector or a data frame with sample\_id and group. Use signal\_color\_by = "group" to color by group, and signal\_summary = "mean", "median", or "sum" to collapse replicates within groups. signal\_colors overrides signal\_palette and may be named by sample ID or group name. See also [GeneTrackR-advanced-parameters](#).

## Value

A ggplot or patchwork object.

## Examples

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
plot_signal_transcript(
  signal = rnaseq, annotation = gp, transcript_id = "TxA1",
  coordinate = "transcript", plot_type = "bar"
)
```

---

plot\_tracks

---

*Plot combined signal and gene model tracks*


---

## Description

Draw a genome-browser-like figure containing a gene model track alone or a signal track combined with gene, feature, and variant tracks. The genomic interval can be specified in three ways: by gene\_id, by transcript\_id, or by explicit chrom, start, and end coordinates.

## Usage

```
plot_tracks(
  annotation,
  signal = NULL,
  features = NULL,
  variants = NULL,
  chrom = NULL,
  start = NULL,
  end = NULL,
  gene_id = NULL,
  transcript_id = NULL,
  samples = NULL,
  sample_groups = NULL,
  signal_color_by = c("sample", "group"),
  signal_summary = c("none", "mean", "median", "sum"),
  signal_type = c("bar", "line", "heatmap"),
  signal_palette = "Paired",
  signal_palette_direction = 1,
  signal_colors = NULL,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
  feature_color_by = "auto",
  feature_palette = "Paired",
  feature_colors = NULL,
  feature_border_color = NA,
  feature_max_legend_levels = 5,
  variant_palette = "Paired",
  variant_colors = NULL,
  ribo_signal_type = c("auto", "bar", "frame"),
  frame_palette = "Paired",
  frame_colors = NULL,
  signal_transform = c("none", "log2", "log10", "sqrt"),
  signal_y_scale = c("free", "fixed"),
  signal_y_ticks = c("range", "pretty"),
  heatmap_bin_size = NULL,
  heatmap_max_bins = 800L,
```

```

heatmap_summary = c("mean", "max", "sum", "median"),
collapse = c("none", "union_exon", "longest"),
strand = c("ignore", "+", "-", "both"),
bin_size = NULL,
highlight = NULL,
layout = c("signal_top", "gene_top"),
heights = c(signal = 3, gene = 1, feature = 0.8, variant = 0.7),
cds_height = 0.5,
utr_height = 0.25,
direction_mode = c("transcript", "gene", "end", "none"),
label_position = c("axis", "feature"),
label_by = c("gene", "transcript"),
text_size = 14,
signal_y_limits = NULL,
signal_alpha = 0.85,
signal_bar_width = 1,
plot_theme = c("bw", "classic", "light", "minimal"),
show_panel_border = NULL
)

```

### Arguments

|                 |                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| annotation      | A GenePred object or a standardized Feature object with transcript/exon records.                                                                                                                             |
| signal          | Optional BwgTrack object. If NULL, only annotation/feature/variant tracks are drawn.                                                                                                                         |
| features        | Optional FeatureTrack object or named list of FeatureTrack objects from read_bed(), read_gff(), or read_gtf().                                                                                               |
| variants        | Optional VariantTrack object or named list of VariantTrack objects from read_vcf().                                                                                                                          |
| chrom           | Chromosome name. Required when gene_id and transcript_id are not supplied.                                                                                                                                   |
| start           | Region start in 1-based closed coordinates. Required with chrom/end.                                                                                                                                         |
| end             | Region end in 1-based closed coordinates. Required with chrom/start.                                                                                                                                         |
| gene_id         | Optional gene ID. If supplied, the plotting interval is inferred from the gene locus.                                                                                                                        |
| transcript_id   | Optional transcript ID. If supplied, the plotting interval is inferred from the transcript locus.                                                                                                            |
| samples         | Optional signal sample IDs.                                                                                                                                                                                  |
| sample_groups   | Optional sample group mapping for group-level coloring or replicate summaries. Use a named character vector, a data frame with sample_id and group, or an unnamed vector with one group per selected sample. |
| signal_color_by | Color signal tracks by sample or group.                                                                                                                                                                      |
| signal_summary  | Replicate summary mode. Use none to plot individual samples, or mean, median, or sum to summarize samples within each group.                                                                                 |
| signal_type     | Signal plot type for non-Ribo-seq browser signal tracks.                                                                                                                                                     |

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| signal_palette            | Signal color palette for signal tracks. Any palette name from <code>RColorBrewer::brewer.pal.info</code> can be used. Discrete signal sample/group colors follow the standard <code>RColorBrewer</code> class order; heatmaps use the corresponding continuous gradient.                                                                                                                                                                     |
| signal_palette_direction  | Direction for generated signal colors. Use 1 for the standard palette order and -1 for the reversed palette order. Discrete sample/group colors preserve level-to-color order; heatmap gradients reverse continuously.                                                                                                                                                                                                                       |
| signal_colors             | Optional named or unnamed vector of explicit colors for signal samples. Explicit colors override <code>signal_palette</code> and are not modified by <code>signal_palette_direction</code> .                                                                                                                                                                                                                                                 |
| gene_palette              | <code>RColorBrewer</code> palette name used for gene model feature fills.                                                                                                                                                                                                                                                                                                                                                                    |
| gene_colors               | Optional custom fill colors for gene model features. Use a named vector such as <code>c(UTR = "#b2df8a", CDS = "#33a02c", exon = "#fb9a99")</code> .                                                                                                                                                                                                                                                                                         |
| gene_border_color         | Optional rectangle border color for gene model features. Use NA to hide borders.                                                                                                                                                                                                                                                                                                                                                             |
| feature_color_by          | Feature-track color grouping. <code>auto</code> chooses a compact grouping automatically; other common choices are <code>feature_group</code> , <code>type</code> , <code>source</code> , <code>name</code> , and <code>strand</code> .                                                                                                                                                                                                      |
| feature_palette           | <code>RColorBrewer</code> palette name used for feature-track fills.                                                                                                                                                                                                                                                                                                                                                                         |
| feature_colors            | Optional explicit feature-track fill colors.                                                                                                                                                                                                                                                                                                                                                                                                 |
| feature_border_color      | Optional rectangle border color for feature tracks. Use NA to hide borders.                                                                                                                                                                                                                                                                                                                                                                  |
| feature_max_legend_levels | Maximum number of legend groups shown for each feature track.                                                                                                                                                                                                                                                                                                                                                                                |
| variant_palette           | <code>RColorBrewer</code> palette name used for variant-track colors.                                                                                                                                                                                                                                                                                                                                                                        |
| variant_colors            | Optional named or unnamed explicit colors passed to <code>plot_variant()</code> . Named vectors are recommended when stable SNP/INS/DEL colors are required.                                                                                                                                                                                                                                                                                 |
| ribo_signal_type          | How Ribo-seq samples should be displayed in transcript-centered browser plots. <code>auto</code> uses <code>frame</code> for samples whose IDs look like Ribo-seq/RPF tracks when <code>transcript_id</code> is supplied, otherwise it falls back to standard genomic signal tracks. <code>bar</code> always uses the standard signal geometry, and <code>frame</code> forces frame rendering when transcript-centered plotting is possible. |
| frame_palette             | <code>RColorBrewer</code> palette name used for Ribo-seq frame plots.                                                                                                                                                                                                                                                                                                                                                                        |
| frame_colors              | Optional explicit colors for <code>frame0</code> , <code>frame1</code> , and <code>frame2</code> .                                                                                                                                                                                                                                                                                                                                           |
| signal_transform          | Signal-axis transformation. Use <code>none</code> , <code>log2</code> , <code>log10</code> , or <code>sqrt</code> .                                                                                                                                                                                                                                                                                                                          |
| signal_y_scale            | Signal y-axis scale mode. Use <code>free</code> for independent sample-specific y-axis ranges or <code>fixed</code> for a shared y-axis range across samples.                                                                                                                                                                                                                                                                                |
| signal_y_ticks            | Signal y-axis tick mode. Use <code>range</code> to show only integer minimum and maximum limits or <code>pretty</code> for default-style breaks.                                                                                                                                                                                                                                                                                             |
| heatmap_bin_size          | Optional bin size used only for heatmap rendering. NULL uses adaptive binning.                                                                                                                                                                                                                                                                                                                                                               |

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| heatmap_max_bins  | Maximum number of heatmap bins when heatmap_bin_size = NULL.                                                                                                                                                              |
| heatmap_summary   | Summary statistic used to aggregate signal into heatmap bins: mean, max, sum, or median.                                                                                                                                  |
| collapse          | Gene model collapse mode for region-level plotting.                                                                                                                                                                       |
| strand            | Signal strand selector.                                                                                                                                                                                                   |
| bin_size          | Optional signal bin size.                                                                                                                                                                                                 |
| highlight         | Optional data frame used to shade intervals on signal and gene model tracks. It must contain start and end columns in genomic coordinates. Optional columns are allowed but ignored by the default renderer.              |
| layout            | Track layout. Use signal_top to place signal above gene model, or gene_top to place gene model above signal.                                                                                                              |
| heights           | Relative panel heights. Must contain at least signal, gene, feature, and variant names when those tracks are used.                                                                                                        |
| cds_height        | Vertical thickness of CDS rectangles in the gene model track.                                                                                                                                                             |
| utr_height        | Vertical thickness of UTR/non-coding exon rectangles in the gene model track.                                                                                                                                             |
| direction_mode    | Direction-arrow style for the gene model track. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows. |
| label_position    | Where to draw gene model labels. axis draws labels on the y axis and feature draws labels near the feature.                                                                                                               |
| label_by          | Which identifier to use for gene model labels. Use gene for gene IDs or transcript for transcript IDs.                                                                                                                    |
| text_size         | Text size in points for axis text, axis titles, legends, and facet labels.                                                                                                                                                |
| signal_y_limits   | Optional two-element numeric vector giving the plotted y-axis limits after signal_transform. Supplying limits changes signal_y_scale to fixed.                                                                            |
| signal_alpha      | Signal geometry transparency from 0 to 1.                                                                                                                                                                                 |
| signal_bar_width  | Relative width of bar intervals from greater than 0 to 1. A value below 1 creates proportional gaps without changing interval centers.                                                                                    |
| plot_theme        | Base ggplot2 theme used by all standard track panels. Use bw, classic, light, or minimal.                                                                                                                                 |
| show_panel_border | Whether to draw panel borders. NULL preserves the selected theme default.                                                                                                                                                 |

**Value**

A patchwork object or ggplot object.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
plot_tracks(annotation = gp, gene_id = "GeneA")
```

---

|                 |                                    |
|-----------------|------------------------------------|
| plot_transcript | <i>Plot a transcript structure</i> |
|-----------------|------------------------------------|

---

**Description**

Plot a transcript structure

**Usage**

```
plot_transcript(
  object,
  transcript_id,
  coordinate = c("transcript", "genomic"),
  show_cds = TRUE,
  cds_height = 0.5,
  utr_height = 0.25,
  direction_mode = c("transcript", "gene", "end", "none"),
  highlight = NULL,
  gene_palette = "Paired",
  gene_colors = NULL,
  gene_border_color = NA,
  label_position = c("axis", "feature"),
  label_by = c("gene", "transcript"),
  text_size = 14,
  plot_theme = c("bw", "classic", "light", "minimal"),
  show_panel_border = NULL
)
```

**Arguments**

|               |                                                          |
|---------------|----------------------------------------------------------|
| object        | A GenePred object.                                       |
| transcript_id | Transcript ID.                                           |
| coordinate    | transcript removes introns, genomic keeps genomic scale. |
| show_cds      | Whether to distinguish CDS and UTR segments.             |
| cds_height    | Vertical thickness of CDS rectangles.                    |
| utr_height    | Vertical thickness of UTR/non-coding exon rectangles.    |

|                   |                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| direction_mode    | Direction-arrow style. transcript draws one arrow per transcript, gene draws one arrow per gene, end draws one short arrow at the directional end of each gene, and none hides direction arrows.                                                                                                                           |
| highlight         | Optional data frame used to shade genomic or transcript intervals. It must contain start and end columns. Optional columns such as label or group are allowed but are not required. In coordinate = "genomic", start and end are genomic positions; in coordinate = "transcript", they are spliced transcript coordinates. |
| gene_palette      | RColorBrewer palette name used for discrete fills. If the number of discrete groups exceeds the palette maximum, colors are automatically interpolated.                                                                                                                                                                    |
| gene_colors       | Optional custom fill colors for gene model features. Use a named vector such as c(UTR = "#b2df8a", CDS = "#33a02c", exon = "#fb9a99") to map colors explicitly. Unnamed colors are matched to the fixed gene-model levels UTR, CDS, and exon, so colors remain stable even when only one feature type is present.          |
| gene_border_color | Optional rectangle border color. Use NA to hide borders, or a color such as "black" or "grey30" to draw feature outlines.                                                                                                                                                                                                  |
| label_position    | Where to draw feature labels. axis draws labels on the y axis and feature draws labels at the center of each gene/transcript structure.                                                                                                                                                                                    |
| label_by          | Which identifier to use for feature labels. Use gene for gene IDs or transcript for transcript IDs.                                                                                                                                                                                                                        |
| text_size         | Text size in points for axis text, axis titles, and legends.                                                                                                                                                                                                                                                               |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                                                                                                                                                                                                                                                                    |
| show_panel_border | Whether to draw the panel border. NULL preserves the selected theme default.                                                                                                                                                                                                                                               |

## Details

Common feature names used by gene\_colors are UTR, CDS, and exon. highlight must contain start and end columns. In genomic coordinate plots these are genomic positions; in transcript coordinate plots they are spliced transcript positions. See also [GeneTrackR-advanced-parameters](#).

## Value

A ggplot object.

## Examples

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
plot_transcript(gp, transcript_id = "TxA1", coordinate = "transcript")
```

---

|              |                                          |
|--------------|------------------------------------------|
| plot_variant | <i>Plot variants in a genomic region</i> |
|--------------|------------------------------------------|

---

## Description

Draws VCF-derived variant records in a genomic region. This function is dedicated to variant visualization and can be used directly or through `plot_tracks()`.

## Usage

```
plot_variant(
  variant,
  chrom = NULL,
  start = NULL,
  end = NULL,
  variant_type = NULL,
  color_by = c("variant_type", "filter", "none"),
  label_by = c("none", "variant_id", "variant_type", "ref", "alt"),
  variant_shape = c("lollipop", "point", "rug"),
  variant_palette = "Paired",
  variant_colors = NULL,
  point_size = 2,
  line_width = 0.35,
  text_size = 14,
  plot_theme = c("bw", "classic", "light", "minimal"),
  show_panel_border = NULL
)
```

## Arguments

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| variant         | A VariantTrack object.                                                                                        |
| chrom           | Chromosome name. If NULL, all chromosomes in variant are used.                                                |
| start           | Region start in 1-based coordinates. If NULL, the minimum variant position is used.                           |
| end             | Region end in 1-based coordinates. If NULL, the maximum variant position is used.                             |
| variant_type    | Optional variant type filter, such as SNP, INS, DEL, or MNV.                                                  |
| color_by        | Column used for variant colors. Use none to disable grouping.                                                 |
| label_by        | Column used for labels. Use none to hide labels.                                                              |
| variant_shape   | Plot geometry. lollipop draws vertical stems and points, point draws points only, and rug draws bottom ticks. |
| variant_palette | RColorBrewer palette name for variant colors.                                                                 |
| variant_colors  | Optional named or unnamed variant color vector.                                                               |

|                   |                                                                              |
|-------------------|------------------------------------------------------------------------------|
| point_size        | Point size.                                                                  |
| line_width        | Line width for lollipop/rug stems.                                           |
| text_size         | Text size.                                                                   |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                      |
| show_panel_border | Whether to draw the panel border. NULL preserves the selected theme default. |

**Value**

A ggplot object.

**Examples**

```
vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
vcf <- read_vcf(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
plot_variant(
  vcf, chrom = "chr1", start = 12339701L, end = 12352000L
)
```

---

|                     |                                                           |
|---------------------|-----------------------------------------------------------|
| plot_variant_effect | <i>Plot phenotype effect size of each natural variant</i> |
|---------------------|-----------------------------------------------------------|

---

**Description**

Calculate per-variant phenotype effect sizes from a HapVariant object and draw an ordered effect-size plot. For a two-genotype variant, the effect is the mean difference between the second and first genotype groups. For variants with more than two genotype groups, the effect is the maximum group mean minus the minimum group mean. Point color encodes the signed effect direction and magnitude, with blue for negative effects and red for positive effects.

**Usage**

```
plot_variant_effect(
  hap,
  phenotype,
  traits = NULL,
  sample_col = "sample_id",
  min_group_samples = 2L,
  test_method = c("t.test", "wilcox.test", "anova", "kruskal.test"),
  p_adjust = "BH",
  effect_type = c("absolute", "signed"),
  top_n = 10L,
  variant_label = c("variant_id", "pos"),
```

```

    x_axis = c("index", "position"),
    point_size = 2.2,
    label_size = 3.2,
    text_size = 14
  )

```

### Arguments

|                   |                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| hap               | A HapVariant object from hap_variant().                                                                                                             |
| phenotype         | A phenotype table returned by read_pheno() or a compatible data.frame.                                                                              |
| traits            | Numeric phenotype trait names. Integer and double storage modes are accepted and normalized before reshaping. If NULL, all numeric traits are used. |
| sample_col        | Sample column name in phenotype table.                                                                                                              |
| min_group_samples | Minimum sample number required for a genotype group.                                                                                                |
| test_method       | Significance test method. One of t.test, wilcox.test, anova, or kruskal.test.                                                                       |
| p_adjust          | P-value adjustment method passed to p.adjust().                                                                                                     |
| effect_type       | Which effect value is plotted. One of absolute or signed.                                                                                           |
| top_n             | Optional number of top variants to label by absolute effect size.                                                                                   |
| variant_label     | Column used for variant labels. One of variant_id, pos, or an existing column in hap\$variants.                                                     |
| x_axis            | X-axis type. One of index or position.                                                                                                              |
| point_size        | Point size.                                                                                                                                         |
| label_size        | Text size for top-variant labels.                                                                                                                   |
| text_size         | Base text size.                                                                                                                                     |

### Value

A list with figure, effect, and plot\_data elements.

### Examples

```

vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
pheno <- read_pheno(pheno_file, verbose = FALSE)
anno <- read_genepred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "code", min_variant_number = 1
)
plot_variant_effect(
  hap, phenotype = pheno, traits = "protein_content",
  min_group_samples = 3
)

```

---

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| plot_variant_pheno | <i>Plot phenotype values grouped by a single variant genotype</i> |
|--------------------|-------------------------------------------------------------------|

---

### Description

Draws phenotype distributions for genotype groups at one variant site. The function is designed as the single-variant counterpart of `plot_hap_pheno()`. Groups are ordered by sample number from left to right, and fill colors are mapped to the median phenotype value of each genotype group.

### Usage

```
plot_variant_pheno(
  variant,
  phenotype,
  traits = NULL,
  sample_col = "sample_id",
  variant_id = NULL,
  chrom = NULL,
  pos = NULL,
  start = NULL,
  end = NULL,
  samples = NULL,
  genotype_mode = c("code", "string"),
  missing_genotype = NA_character_,
  min_group_samples = 2L,
  plot_type = c("violin_boxplot", "violin", "boxplot"),
  test_method = c("t.test", "wilcox.test", "ks.test"),
  p_adjust = "BH",
  p_label = c("stars", "number", "both"),
  p_cutoff = 0.05,
  p_value_type = c("raw", "adjusted"),
  show_signif_only = TRUE,
  show_points = FALSE,
  show_outliers = FALSE,
  fill_palette = "Paired",
  fill_colors = NULL,
  fill_alpha = 0.75,
  violin_width = 0.9,
  box_width = 0.18,
  bracket_step = 0.08,
  bracket_tip_fraction = 0.12,
  x_text_angle = 90,
  strip_label_width = 24,
  strip_fill = "white",
  strip_border_color = NULL,
  strip_text_lineheight = 0.9,
  text_size = 14
```

)

**Arguments**

|                   |                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| variant           | A VariantTrack object, a VCF file path, a HapVariant object, or a VCF-like data.frame/data.table containing one or more variants with genotype sample columns.                                                                                                       |
| phenotype         | A phenotype table returned by read_pheno() or a compatible data.frame.                                                                                                                                                                                               |
| traits            | Phenotype trait names. If NULL, all numeric traits are used.                                                                                                                                                                                                         |
| sample_col        | Sample column name in phenotype table.                                                                                                                                                                                                                               |
| variant_id        | Optional variant ID to select.                                                                                                                                                                                                                                       |
| chrom             | Optional chromosome name used to select a variant.                                                                                                                                                                                                                   |
| pos               | Optional 1-based variant position. This is equivalent to setting both start and end to the same value.                                                                                                                                                               |
| start             | Optional 1-based start position for region-based selection.                                                                                                                                                                                                          |
| end               | Optional 1-based end position for region-based selection.                                                                                                                                                                                                            |
| samples           | Optional sample names to keep.                                                                                                                                                                                                                                       |
| genotype_mode     | Genotype representation. code converts genotypes to 0/1 states, where 0 means reference genotype and 1 means any alternate allele is present. string converts genotypes to compact allele labels; long InDel alleles are compressed as iN, where N is allele length. |
| missing_genotype  | Missing genotype label. Default is NA_character_.                                                                                                                                                                                                                    |
| min_group_samples | Minimum sample number required for a genotype group.                                                                                                                                                                                                                 |
| plot_type         | Plot type. One of violin, boxplot, or violin_boxplot.                                                                                                                                                                                                                |
| test_method       | Pairwise test method. One of t.test, wilcox.test, or ks.test.                                                                                                                                                                                                        |
| p_adjust          | P-value adjustment method passed to p.adjust().                                                                                                                                                                                                                      |
| p_label           | P-value label style. One of stars, number, or both.                                                                                                                                                                                                                  |
| p_cutoff          | Significance cutoff used for displaying pairwise comparisons.                                                                                                                                                                                                        |
| p_value_type      | Which p-value is used for filtering and labeling. One of raw or adjusted.                                                                                                                                                                                            |
| show_signif_only  | Logical. Whether to only display significant comparisons.                                                                                                                                                                                                            |
| show_points       | Logical. Whether to show sample points.                                                                                                                                                                                                                              |
| show_outliers     | Logical. Whether to show boxplot outliers.                                                                                                                                                                                                                           |
| fill_palette      | RColorBrewer palette name used for median-based genotype fills.                                                                                                                                                                                                      |
| fill_colors       | Optional custom fill colors for genotype groups.                                                                                                                                                                                                                     |
| fill_alpha        | Alpha value for violin/boxplot fill colors.                                                                                                                                                                                                                          |
| violin_width      | Violin plot width.                                                                                                                                                                                                                                                   |
| box_width         | Boxplot width.                                                                                                                                                                                                                                                       |
| bracket_step      | Fraction of y-range used to separate significance brackets.                                                                                                                                                                                                          |

bracket\_tip\_fraction      Fraction of bracket vertical spacing used for the short downward bracket tips.  
 x\_text\_angle      Rotation angle for genotype labels on the x-axis.  
 strip\_label\_width      Maximum character width for wrapping long facet strip labels.  
 strip\_fill      Strip background fill color. Default is white.  
 strip\_border\_color      Strip border color. Default NULL removes the strip border.  
 strip\_text\_lineheight      Line height used for wrapped facet-strip labels.  
 text\_size      Text size.

### Value

A list with figure and pvalue elements. Additional elements include summary, bracket, plot\_data, and variant\_data.

### Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
pheno <- read_pheno(pheno_file, verbose = FALSE)
plot_variant_pheno(vcf, phenotype = pheno, variant_id = "varA03",
  traits = "protein_content", min_group_samples = 3)
plot_variant_pheno(vcf, phenotype = pheno, chrom = "chr1", pos = 12342550,
  traits = "protein_content", genotype_mode = "string",
  min_group_samples = 3, test_method = "wilcox.test")
```

---

|                    |                                   |
|--------------------|-----------------------------------|
| plot_variant_track | <i>Plot a VariantTrack object</i> |
|--------------------|-----------------------------------|

---

### Description

Backward-compatible alias of plot\_variant().

### Usage

```
plot_variant_track(
  track,
  chrom,
  start,
  end,
  color_by = c("variant_type", "filter", "none"),
  label_by = c("none", "variant_id", "variant_type", "ref", "alt"),
  variant_palette = "Paired",
  variant_colors = NULL,
```

```

    text_size = 14,
    plot_theme = c("bw", "classic", "light", "minimal"),
    show_panel_border = NULL
  )

```

## Arguments

|                   |                                                                                     |
|-------------------|-------------------------------------------------------------------------------------|
| track             | A VariantTrack object.                                                              |
| chrom             | Chromosome name. If NULL, all chromosomes in variant are used.                      |
| start             | Region start in 1-based coordinates. If NULL, the minimum variant position is used. |
| end               | Region end in 1-based coordinates. If NULL, the maximum variant position is used.   |
| color_by          | Column used for variant colors. Use none to disable grouping.                       |
| label_by          | Column used for labels. Use none to hide labels.                                    |
| variant_palette   | RColorBrewer palette name for variant colors.                                       |
| variant_colors    | Optional named or unnamed variant color vector.                                     |
| text_size         | Text size.                                                                          |
| plot_theme        | Base ggplot2 theme. Use bw, classic, light, or minimal.                             |
| show_panel_border | Whether to draw the panel border. NULL preserves the selected theme default.        |

## Value

A ggplot object.

## Examples

```

vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
vcf <- read_vcf(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
plot_variant_track(
  vcf, chrom = "chr1", start = 12339701L, end = 12352000L
)

```

---

```
print.GeneTrackRPhenoPlot
```

*Print GeneTrackR phenotype plot result*

---

### Description

Print GeneTrackR phenotype plot result

### Usage

```
## S3 method for class 'GeneTrackRPhenoPlot'
print(x, ...)
```

### Arguments

**x** A GeneTrackRPhenoPlot object returned by plot\_hap\_pheno() or plot\_variant\_pheno().

**...** Additional arguments.

### Value

Invisibly returns x.

---

```
read_bed
```

*Read a BED file as a FeatureTrack*

---

### Description

Reads BED3-BED12 style files and converts BED coordinates from 0-based half-open to 1-based closed intervals by default.

### Usage

```
read_bed(
  file,
  coordinate = c("bed", "granges"),
  name_col = 4L,
  type = "BED",
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

**Arguments**

|            |                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------|
| file       | BED file path. Gzip-compressed files are supported by <code>data.table::fread()</code> .                      |
| coordinate | Input coordinate system. <code>bed</code> means 0-based half-open; <code>granges</code> means 1-based closed. |
| name_col   | BED column used as feature name. Default 4.                                                                   |
| type       | Feature type assigned to BED records.                                                                         |
| verbose    | Whether to print progress messages.                                                                           |
| progress   | Whether to show a stage progress bar.                                                                         |

**Value**

A `FeatureTrack` object.

**Examples**

```
bed_file <- system.file("extdata", "gtr_demo_features.bed", package = "GeneTrackR")
features <- read_bed(bed_file, verbose = FALSE, progress = FALSE)
features
plot_feature_track(features, chrom = "chr1", start = 12338201, end = 12374500)
```

---

|          |                                                    |
|----------|----------------------------------------------------|
| read_bwg | <i>Read bedGraph, wig, or bigWig signal tracks</i> |
|----------|----------------------------------------------------|

---

**Description**

Read `bedGraph`, `wig`, or `bigWig` signal tracks

**Usage**

```
read_bwg(
  files,
  format = c("auto", "bedgraph", "bigwig", "wig"),
  sample_names = NULL,
  strand = NULL,
  mode = c("lazy", "memory"),
  genome = NULL,
  check_chrom = TRUE,
  use_tabix = c("auto", "yes", "no"),
  tabix_empty_fallback = FALSE,
  verbose = TRUE
)
```

## Arguments

|                                   |                                                                                                                                                                                                                                           |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>files</code>                | Signal file paths.                                                                                                                                                                                                                        |
| <code>format</code>               | Input format. Use auto, bedgraph, bigwig, or wig.                                                                                                                                                                                         |
| <code>sample_names</code>         | Optional sample names. If NULL, names are inferred from file basenames after removing compression suffixes and signal-track suffixes such as <code>.bedgraph.gz</code> , <code>.bigwig</code> , <code>.bw</code> , or <code>.wig</code> . |
| <code>strand</code>               | Optional strand labels for files.                                                                                                                                                                                                         |
| <code>mode</code>                 | lazy stores file paths, memory reads data into R.                                                                                                                                                                                         |
| <code>genome</code>               | Optional genome label.                                                                                                                                                                                                                    |
| <code>check_chrom</code>          | Whether to check chromosome names for in-memory files.                                                                                                                                                                                    |
| <code>use_tabix</code>            | Whether to use tabix/Rsamtools for indexed bedGraph-like files in lazy mode. Accepts "auto", "yes", "no", TRUE, or FALSE. TRUE is equivalent to "yes", and FALSE is equivalent to "no".                                                   |
| <code>tabix_empty_fallback</code> | Logical. Whether an empty tabix result should be verified by a full-file fread query. Default FALSE for performance. Set TRUE only when you suspect the tabix index coordinate convention is incompatible with the queried region.        |
| <code>verbose</code>              | Logical. Whether to print read/setup progress messages.                                                                                                                                                                                   |

## Details

`format = "auto"` infers the input type from file extensions after removing compression suffixes such as `.gz` or `.bgz`. `sample_names = NULL` infers names from file basenames, removing suffixes such as `.bedgraph.gz`, `.bw`, `.bigwig`, and `.wig`.

For bigWig and wig files, strand information is not stored in the file, so the sample is treated as unstranded. For paired plus/minus bedGraph files, provide `strand = c("+", "-")` and matching `sample_names` if strand-specific filtering is required.

`use_tabix = "auto"` uses indexed querying only when a `.tbi` index and an available backend are detected. GeneTrackR first checks the system tabix command and then the R package Rsamtools. Otherwise, lazy bedGraph queries fall back to full-file reading.

BigWig metadata, full-memory loading, and lazy regional queries use the built-in pure-R BigWig reader. No compiled library or external BigWig executable is required.

`BwgTrack$seqinfo` follows the schema-v2 sequence metadata contract. bigWig inputs record chromosome lengths from the file header. In-memory bedGraph and wig inputs record observed chromosome names with unknown lengths (NA).

bedGraph, WIG, and BigWig memory reads are dispatched through one internal pure-R I/O layer so all formats return the same canonical signal schema.

## Value

A BwgTrack object.

## Examples

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
rnaseq
```

---

|               |                                        |
|---------------|----------------------------------------|
| read_genepred | <i>Read a GenePred annotation file</i> |
|---------------|----------------------------------------|

---

## Description

Read a GenePred annotation file

## Usage

```
read_genepred(
  file,
  format = c("auto", "genePred", "genePredExt"),
  coordinate = c("ucsc", "granges"),
  remove_invalid = TRUE,
  report_invalid = TRUE,
  gene_col = c("name2", "name"),
  transcript_col = "name",
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

## Arguments

|                |                                                                                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| file           | GenePred or GenePredExt file path.                                                                                                         |
| format         | Input format. Use auto, genePred, or genePredExt.                                                                                          |
| coordinate     | Input coordinate system. ucsc means 0-based half-open.                                                                                     |
| remove_invalid | Remove invalid records after validation.                                                                                                   |
| report_invalid | Store invalid records in the object validation slot.                                                                                       |
| gene_col       | Gene identifier source. name2 is recommended for GenePredExt.                                                                              |
| transcript_col | Transcript identifier source.                                                                                                              |
| verbose        | Logical. Whether to print step-level progress messages. Default TRUE.                                                                      |
| progress       | Logical. Whether to show a stage-level progress bar. By default, a progress bar is shown only in interactive sessions when verbose = TRUE. |

## Details

Standard GenePred uses name as transcript ID. GenePredExt commonly uses name as transcript ID and name2 as gene ID, so the default gene\_col tries name2 first and falls back to name when name2 is unavailable. coordinate = "ucsc" converts 0-based half-open GenePred coordinates to the package's internal 1-based closed coordinate system.

For large files, verbose = TRUE prints major processing stages, while progress = TRUE additionally shows a stage-level text progress bar.

## Value

A GenePred object.

## Examples

```
gp_file <- system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR")
gp <- read_genepred(gp_file, format = "genePredExt", verbose = FALSE)
gp
head(gp$genes)
head(gp$transcripts)
```

---

read\_gff

*Read a GFF3 file as a FeatureTrack*


---

## Description

Read a GFF3 file as a FeatureTrack

## Usage

```
read_gff(
  file,
  feature_types = NULL,
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

## Arguments

|               |                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------|
| file          | GFF3 file path.                                                                                                |
| feature_types | Optional feature types to keep, such as gene, mRNA, exon, or CDS.                                              |
| verbose       | Whether to print progress messages.                                                                            |
| progress      | Whether to show a stage progress bar. The parser reports major stages using the same style as read_genepred(). |

## Value

A FeatureTrack object.

**Examples**

```
gff_file <- system.file("extdata", "gtr_demo.gff3", package = "GeneTrackR")
gff <- read_gff(gff_file, verbose = FALSE, progress = FALSE)
gff
head(gff$genes)
```

read\_gtf

*Read a GTF file as a FeatureTrack***Description**

Read a GTF file as a FeatureTrack

**Usage**

```
read_gtf(
  file,
  feature_types = NULL,
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

**Arguments**

|               |                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------|
| file          | GTF file path.                                                                                                 |
| feature_types | Optional feature types to keep, such as gene, transcript, exon, or CDS.                                        |
| verbose       | Whether to print progress messages.                                                                            |
| progress      | Whether to show a stage progress bar. The parser reports major stages using the same style as read_genepred(). |

**Value**

A FeatureTrack object.

**Examples**

```
gtf_file <- system.file("extdata", "gtr_demo.gtf", package = "GeneTrackR")
gtf <- read_gtf(gtf_file, verbose = FALSE, progress = FALSE)
gtf
head(gtf$genes)
```

---

|            |                               |
|------------|-------------------------------|
| read_pheno | <i>Read a phenotype table</i> |
|------------|-------------------------------|

---

### Description

Reads a phenotype table where the first column contains taxa/sample IDs and all remaining columns are phenotype traits.

### Usage

```
read_pheno(  
  file,  
  sep = "auto",  
  sample_col = NULL,  
  na_strings = c("NA", "NaN", "", ".", "null", "NULL"),  
  verbose = TRUE,  
  progress = interactive() && isTRUE(verbose)  
)
```

### Arguments

|            |                                                                               |
|------------|-------------------------------------------------------------------------------|
| file       | Phenotype file path.                                                          |
| sep        | Field separator. Use auto for automatic detection by <code>fread()</code> .   |
| sample_col | Optional sample column name. If <code>NULL</code> , the first column is used. |
| na_strings | Strings treated as missing values.                                            |
| verbose    | Logical. Whether to print reading messages.                                   |
| progress   | Logical. Reserved for future compact progress display.                        |

### Value

A `data.table` with the first column standardized as `sample_id`.

### Examples

```
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")  
pheno <- read_pheno(pheno_file, verbose = FALSE)  
head(pheno)  
summary_pheno(pheno)  
plot_pheno(pheno, traits = c("plant_height", "seed_weight"))
```

---

|          |                                          |
|----------|------------------------------------------|
| read_vcf | <i>Read a VCF file as a VariantTrack</i> |
|----------|------------------------------------------|

---

## Description

Reads VCF records and stores them as genomic point variants. Standard VCF columns are always imported. If genotype sample columns are present, they are preserved for haplotype analysis. For bgzip-compressed VCF files with a tabix index, chrom, start, and end can be supplied to load only a genomic interval instead of parsing the whole VCF file.

## Usage

```
read_vcf(
  file,
  keep_genotype = TRUE,
  mode = c("auto", "memory", "lazy"),
  chrom = NULL,
  start = NULL,
  end = NULL,
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

## Arguments

|               |                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file          | VCF file path. Plain VCF, gzip-compressed VCF, and bgzip VCF are supported.                                                                                       |
| keep_genotype | Logical. Whether to keep FORMAT and sample genotype columns.                                                                                                      |
| mode          | Reading mode. memory parses records immediately, lazy stores indexed VCF metadata only, and auto uses lazy mode for indexed VCF files when no region is supplied. |
| chrom         | Optional chromosome name for indexed regional reading.                                                                                                            |
| start         | Optional 1-based region start for indexed regional reading.                                                                                                       |
| end           | Optional 1-based region end for indexed regional reading.                                                                                                         |
| verbose       | Logical. Whether to print progress messages.                                                                                                                      |
| progress      | Logical. Whether to print a compact stage-level progress indicator.                                                                                               |

## Value

A VariantTrack object.

## Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
variants <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
region_variants <- retrieve_vcf(
```

```

variants,
chrom = "chr1",
start = 12339700,
end = 12343200,
as = "VariantTrack",
verbose = FALSE
)
plot_variant(region_variants, chrom = "chr1", start = 12339700, end = 12343200)

```

---

|                |                                          |
|----------------|------------------------------------------|
| read_vcf_track | <i>Read a VCF file as a VariantTrack</i> |
|----------------|------------------------------------------|

---

### Description

Backward-compatible alias of read\_vcf().

### Usage

```

read_vcf_track(
  file,
  keep_genotype = TRUE,
  mode = c("auto", "memory", "lazy"),
  chrom = NULL,
  start = NULL,
  end = NULL,
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)

```

### Arguments

|               |                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file          | VCF file path. Plain VCF, gzip-compressed VCF, and bgzip VCF are supported.                                                                                       |
| keep_genotype | Logical. Whether to keep FORMAT and sample genotype columns.                                                                                                      |
| mode          | Reading mode. memory parses records immediately, lazy stores indexed VCF metadata only, and auto uses lazy mode for indexed VCF files when no region is supplied. |
| chrom         | Optional chromosome name for indexed regional reading.                                                                                                            |
| start         | Optional 1-based region start for indexed regional reading.                                                                                                       |
| end           | Optional 1-based region end for indexed regional reading.                                                                                                         |
| verbose       | Logical. Whether to print progress messages.                                                                                                                      |
| progress      | Logical. Whether to print a compact stage-level progress indicator.                                                                                               |

### Value

A VariantTrack object.

**Examples**

```
vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
read_vcf_track(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
```

---

|                  |                                                  |
|------------------|--------------------------------------------------|
| refine_haplotype | <i>Refine haplotypes by phenotype similarity</i> |
|------------------|--------------------------------------------------|

---

**Description**

Collapse original haplotypes into refined haplotype groups according to phenotype similarity. Haplotypes are merged only when their pairwise phenotype difference is not significant and, optionally, the absolute mean difference is smaller than `effect_threshold`.

**Usage**

```
refine_haplotype(
  hap,
  phenotype,
  traits = NULL,
  sample_col = "sample_id",
  min_hap_samples = 2L,
  test_method = c("t.test", "wilcox.test", "ks.test"),
  p_adjust = "BH",
  alpha = 0.05,
  effect_threshold = NULL,
  group_prefix = "RHap",
  mixed_label = "mixed"
)
```

**Arguments**

|                              |                                                                                                                                     |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <code>hap</code>             | A <code>HapVariant</code> object from <code>hap_variant()</code> .                                                                  |
| <code>phenotype</code>       | A phenotype table returned by <code>read_pheno()</code> or a compatible <code>data.frame</code> .                                   |
| <code>traits</code>          | Phenotype trait names. If <code>NULL</code> , all numeric traits are used.                                                          |
| <code>sample_col</code>      | Sample column name in phenotype table.                                                                                              |
| <code>min_hap_samples</code> | Minimum sample number required for an original haplotype.                                                                           |
| <code>test_method</code>     | Pairwise test method. One of <code>t.test</code> , <code>wilcox.test</code> , or <code>ks.test</code> .                             |
| <code>p_adjust</code>        | P-value adjustment method passed to <code>p.adjust()</code> .                                                                       |
| <code>alpha</code>           | Adjusted p-value cutoff. Pairs with adjusted p-value larger than <code>alpha</code> are considered statistically indistinguishable. |

|                  |                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------|
| effect_threshold | Optional maximum absolute mean difference for merging haplotypes. If NULL, only the significance criterion is used. |
| group_prefix     | Prefix for refined haplotype IDs.                                                                                   |
| mixed_label      | Label used for variant states that are heterogeneous within a refined haplotype group.                              |

### Value

A HapRefined object. The object contains refined\_hap, which is a HapVariant-compatible object and can be passed to plot\_hap\_pheno() and plot\_hap\_variant().

### Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
pheno_file <- system.file("extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR")
anno_file <- system.file("extdata", "gtr_demo_genePredExt", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
pheno <- read_pheno(pheno_file, verbose = FALSE)
anno <- read_genePred(anno_file, format = "genePredExt", verbose = FALSE)
hap <- hap_variant(
  vcf, annotation = anno, gene_id = "GeneA",
  genotype_mode = "string", min_variant_number = 1
)
refined <- refine_haplotype(
  hap,
  phenotype = pheno,
  traits = "protein_content",
  min_hap_samples = 3,
  effect_threshold = 0.5
)
refined$refined_haplotypes
plot_hap_pheno(
  refined$refined_hap, phenotype = pheno,
  traits = "protein_content", min_hap_samples = 3
)
```

---

retrieve\_bwg

*Retrieve signal records from a BwgTrack object*

---

### Description

retrieve\_bwg() is the unified signal retrieval API in GeneTrackR. It retrieves signal intervals from bedGraph, wig, or bigWig tracks by genomic region.

**Usage**

```

retrieve_bwg(
  object,
  chrom = NULL,
  start = NULL,
  end = NULL,
  annotation = NULL,
  gene_id = NULL,
  transcript_id = NULL,
  upstream = 0L,
  downstream = 0L,
  strand_aware = TRUE,
  samples = NULL,
  strand = c("ignore", "+", "-", "both", "auto"),
  strand_policy = c("ignore_unstranded", "strict"),
  as = c("data.table", "BwgTrack", "GRanges"),
  verbose = FALSE,
  progress = interactive() && isTRUE(verbose),
  keep_empty_samples = FALSE,
  tabix_empty_fallback = NULL
)

```

**Arguments**

|                            |                                                                                                                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code>        | A BwgTrack object returned by <code>read_bwg()</code> .                                                                                                                                      |
| <code>chrom</code>         | Chromosome name. Required for direct region queries.                                                                                                                                         |
| <code>start</code>         | Region start in 1-based closed coordinates. Required for direct region queries.                                                                                                              |
| <code>end</code>           | Region end in 1-based closed coordinates. Required for direct region queries.                                                                                                                |
| <code>annotation</code>    | Optional GenePred/Feature annotation object used for gene/transcript-aware retrieval.                                                                                                        |
| <code>gene_id</code>       | Optional gene ID. When supplied, annotation is used to resolve the gene range.                                                                                                               |
| <code>transcript_id</code> | Optional transcript ID. When supplied, annotation is used to resolve the transcript range.                                                                                                   |
| <code>upstream</code>      | Upstream flanking length in bp for gene/transcript queries.                                                                                                                                  |
| <code>downstream</code>    | Downstream flanking length in bp for gene/transcript queries.                                                                                                                                |
| <code>strand_aware</code>  | Logical. Whether upstream/downstream should follow gene/transcript strand direction.                                                                                                         |
| <code>samples</code>       | Optional character vector of sample IDs to retrieve. If NULL, all samples are queried.                                                                                                       |
| <code>strand</code>        | Strand selector. Use "ignore" for unstranded retrieval, "+" or "-" for strand-specific tracks, "both" for both strands, or "auto" when called by higher-level gene-aware plotting functions. |
| <code>strand_policy</code> | How to handle strand filtering for unstranded signal files. "ignore_unstranded" returns unstranded tracks for any strand request; "strict" only returns explicitly matching strand records.  |

|                      |                                                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| as                   | Output type. "data.table" returns a signal table, "BwgTrack" returns an in-memory BwgTrack sub-object, and "GRanges" returns a GRanges object.                           |
| verbose              | Logical. Whether to print progress messages.                                                                                                                             |
| progress             | Logical. Whether to show a text progress bar for multi-sample queries.                                                                                                   |
| keep_empty_samples   | Logical. If TRUE, samples without signal in the requested region are returned as zero-valued placeholder intervals. This is useful for preserving empty facets in plots. |
| tabix_empty_fallback | Optional logical. If TRUE, empty tabix results are verified by full-file fread. Keep FALSE for speed unless debugging tabix coordinate/index issues.                     |

### Value

A data.table, BwgTrack object, or GRanges object. BwgTrack subsets retain sequence metadata for the selected samples and chromosome when available.

### Examples

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
retrieve_bwg(
  rnaseq, chrom = "chr1", start = 12339001, end = 12352000, strand = "+"
)
```

---

|                  |                                                    |
|------------------|----------------------------------------------------|
| retrieve_feature | <i>Retrieve annotation features or sub-objects</i> |
|------------------|----------------------------------------------------|

---

### Description

retrieve\_feature() extracts a true sub-object from a Feature/GenePred-compatible annotation object. By default it returns a Feature/GenePred-compatible object. Use as = "data.table" only when a plain table is required.

### Usage

```
retrieve_feature(
  object,
  pattern = NULL,
  level = c("feature", "gene", "transcript", "exon"),
  chrom = NULL,
  start = NULL,
  end = NULL,
```

```

mode = c("overlap", "within", "trim"),
gene_id = NULL,
transcript_id = NULL,
type = NULL,
ignore_case = TRUE,
fixed = FALSE,
as = c("Feature", "data.table")
)

```

### Arguments

|                            |                                                                                                                                                                                                             |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code>        | A Feature-compatible annotation object.                                                                                                                                                                     |
| <code>pattern</code>       | Optional pattern used for ID/name matching.                                                                                                                                                                 |
| <code>level</code>         | Output table level when <code>as = "data.table"</code> . One of feature, gene, transcript, or exon.                                                                                                         |
| <code>chrom</code>         | Optional chromosome.                                                                                                                                                                                        |
| <code>start</code>         | Optional start coordinate.                                                                                                                                                                                  |
| <code>end</code>           | Optional end coordinate.                                                                                                                                                                                    |
| <code>mode</code>          | Region matching mode. <code>overlap</code> keeps overlapping transcripts/features, <code>within</code> keeps fully contained records, and <code>trim</code> clips gene-model records to the query interval. |
| <code>gene_id</code>       | Optional gene ID filter. Exact matching is used.                                                                                                                                                            |
| <code>transcript_id</code> | Optional transcript ID filter. Exact matching is used.                                                                                                                                                      |
| <code>type</code>          | Optional feature type filter for feature-table retrieval.                                                                                                                                                   |
| <code>ignore_case</code>   | Logical. Whether pattern matching ignores case.                                                                                                                                                             |
| <code>fixed</code>         | Logical. Whether pattern is fixed text.                                                                                                                                                                     |
| <code>as</code>            | Output type. Default Feature returns a sub-object. Use <code>data.table</code> for a plain table.                                                                                                           |

### Value

A Feature/GenePred-compatible object or a `data.table`.

### Examples

```

gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
retrieve_feature(gp, gene_id = "GeneA")

```

---

retrieve\_vcf

---

Retrieve variants from a VariantTrack or an indexed VCF file

---

## Description

Retrieves VCF records by optional genomic region, gene/transcript locator, variant ID, type, or text pattern. With an in-memory VariantTrack, all location arguments may be omitted to query the full object. For large bgzip-compressed VCF files with a tabix index, a complete genomic region is queried with `Rsamtools::scanTabix()`; non-regional filters require reading the full file before filtering.

## Usage

```
retrieve_vcf(
  object,
  pattern = NULL,
  chrom = NULL,
  start = NULL,
  end = NULL,
  annotation = NULL,
  gene_id = NULL,
  transcript_id = NULL,
  upstream = 0L,
  downstream = 0L,
  strand_aware = TRUE,
  variant_id = NULL,
  variant_type = NULL,
  keep_genotype = TRUE,
  ignore_case = TRUE,
  fixed = FALSE,
  as = c("data.table", "VariantTrack", "GRanges"),
  verbose = TRUE,
  progress = interactive() && isTRUE(verbose)
)
```

## Arguments

|            |                                                                                                                        |
|------------|------------------------------------------------------------------------------------------------------------------------|
| object     | A VariantTrack object or a path to a VCF/VCF.GZ file.                                                                  |
| pattern    | Optional text pattern matched against variant ID, REF, ALT, INFO, and variant type.                                    |
| chrom      | Optional chromosome name or names. May be used alone as a chromosome filter. Required when start and end are supplied. |
| start      | Optional 1-based region start. Must be supplied together with end.                                                     |
| end        | Optional 1-based region end. Must be supplied together with start.                                                     |
| annotation | Optional GenePred/Feature annotation object used for gene/transcript-aware retrieval.                                  |

|               |                                                                                             |
|---------------|---------------------------------------------------------------------------------------------|
| gene_id       | Optional gene ID. When supplied, annotation is used to resolve the gene range.              |
| transcript_id | Optional transcript ID. When supplied, annotation is used to resolve the transcript range.  |
| upstream      | Upstream flanking length in bp for gene/transcript queries.                                 |
| downstream    | Downstream flanking length in bp for gene/transcript queries.                               |
| strand_aware  | Logical. Whether upstream/downstream should follow gene/transcript strand direction.        |
| variant_id    | Optional variant ID vector.                                                                 |
| variant_type  | Optional variant type vector, such as SNP, INS, DEL, or MNV.                                |
| keep_genotype | Logical. Whether to keep FORMAT and sample genotype columns when object is a VCF file path. |
| ignore_case   | Logical. Whether pattern matching ignores case.                                             |
| fixed         | Logical. Whether pattern is matched as a fixed string.                                      |
| as            | Output type: data.table, VariantTrack, or GRanges.                                          |
| verbose       | Logical. Whether to print progress messages.                                                |
| progress      | Logical. Whether to print a compact stage-level progress indicator.                         |

**Value**

A data.table, VariantTrack, or GenomicRanges::GRanges object.

**Examples**

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
retrieve_vcf(vcf, chrom = "chr1", start = 12339700, end = 12343200)
retrieve_vcf(vcf, variant_type = "SNP")
retrieve_vcf(vcf, pattern = "varA", fixed = TRUE)
vt <- retrieve_vcf(vcf, chrom = "chr1", start = 12339700, end = 12343200, as = "VariantTrack")
vt
retrieve_vcf(vcf, chrom = "chr1", start = 12339700, end = 12343200, as = "GRanges")
```

---

seqinfo\_bwg

---

*Get chromosome information from BwgTrack objects*


---

**Description**

Get chromosome information from BwgTrack objects

**Usage**

```
seqinfo_bwg(object, samples = NULL)
```

**Arguments**

|         |                      |
|---------|----------------------|
| object  | A BwgTrack object.   |
| samples | Optional sample IDs. |

**Details**

Schema-v2 objects return stored sequence metadata directly. Legacy objects without a seqinfo slot query chromosome metadata from bigWig source files through the built-in pure-R backend.

**Value**

A data.table containing chromosome names and lengths for each sample.

**Examples**

```
signal_file <- system.file(
  "extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"
)
signal <- read_bwg(
  signal_file, format = "bedgraph", sample_names = "RNA_plus",
  strand = "+", mode = "memory", verbose = FALSE
)
seqinfo_bwg(signal)
```

---

slice\_bwg

*Slice signal tracks by genomic region*

---

**Description**

Slice signal tracks by genomic region

**Usage**

```
slice_bwg(
  object,
  chrom,
  start,
  end,
  samples = NULL,
  strand = "ignore",
  as = c("BwgTrack", "data.frame", "GRanges")
)
```

**Arguments**

|         |                                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------|
| object  | A BwgTrack object.                                                                                                                |
| chrom   | Chromosome name.                                                                                                                  |
| start   | Region start.                                                                                                                     |
| end     | Region end.                                                                                                                       |
| samples | Optional sample IDs.                                                                                                              |
| strand  | Strand selector. For unstranded bigWig/wig tracks, '+' and '-' do not filter records.                                             |
| as      | Return type. Use BwgTrack to keep the object structure, data.frame for a plain table, or GRanges for genomic interval operations. |

**Value**

A BwgTrack, data.table, or GRanges object.

**Examples**

```
rnaseq <- read_bwg(
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",
  mode = "memory", verbose = FALSE
)
slice_bwg(
  rnaseq, chrom = "chr1", start = 12339001, end = 12352000,
  strand = "+", as = "BwgTrack"
)
```

---

|             |                                                |
|-------------|------------------------------------------------|
| summary_bwg | <i>Summarize signal tracks within a region</i> |
|-------------|------------------------------------------------|

---

**Description**

Summarize signal tracks within a region

**Usage**

```
summary_bwg(object, chrom = NULL, start = NULL, end = NULL, samples = NULL)
```

**Arguments**

|         |                           |
|---------|---------------------------|
| object  | A BwgTrack object.        |
| chrom   | Optional chromosome name. |
| start   | Optional region start.    |
| end     | Optional region end.      |
| samples | Optional sample IDs.      |

**Value**

A data.table summary.

**Examples**

```
signal_file <- system.file(
  "extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"
)
signal <- read_bwg(
  signal_file, format = "bedgraph", sample_names = "RNA_plus",
  strand = "+", mode = "memory", verbose = FALSE
)
summary_bwg(signal)
```

---

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| summary_feature | <i>Summarize a Feature-compatible annotation object</i> |
|-----------------|---------------------------------------------------------|

---

**Description**

summary\_feature() is the unified summary API for GenePred, GTF, GFF, and BED-derived annotation objects.

**Usage**

```
summary_feature(
  object,
  chrom = NULL,
  start = NULL,
  end = NULL,
  level = c("feature", "gene", "transcript", "exon"),
  by = c("chrom", "type")
)
```

**Arguments**

|        |                                                        |
|--------|--------------------------------------------------------|
| object | A Feature-compatible annotation object.                |
| chrom  | Optional chromosome filter.                            |
| start  | Optional region start.                                 |
| end    | Optional region end.                                   |
| level  | Summary level. Use feature, gene, transcript, or exon. |
| by     | Grouping columns used when level = "feature".          |

**Value**

A data.table summary.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
summary_feature(gp, level = "gene")
```

---

|               |                                   |
|---------------|-----------------------------------|
| summary_pheno | <i>Summarize phenotype traits</i> |
|---------------|-----------------------------------|

---

**Description**

Summarize phenotype traits

**Usage**

```
summary_pheno(pheno, sample_col = "sample_id")
```

**Arguments**

- pheno            A phenotype data.frame/data.table returned by read\_pheno().
- sample\_col      Sample column name.

**Value**

A data.table summarizing trait type, missing values, and basic statistics.

**Examples**

```
pheno_file <- system.file(
  "extdata", "gtr_demo_pheno.tsv", package = "GeneTrackR"
)
pheno <- read_pheno(pheno_file, verbose = FALSE)
summary_pheno(pheno)
```

---

|             |                                        |
|-------------|----------------------------------------|
| summary_vcf | <i>Summarize a VariantTrack object</i> |
|-------------|----------------------------------------|

---

## Description

summary\_vcf() is the unified summary API for VCF-derived VariantTrack objects. When no genomic range is supplied, the full in-memory object is summarized. For lazy tracks, an omitted range causes the source VCF to be read before summarization.

## Usage

```
summary_vcf(
  object,
  chrom = NULL,
  start = NULL,
  end = NULL,
  by = c("chrom", "variant_type")
)
```

## Arguments

|        |                                                             |
|--------|-------------------------------------------------------------|
| object | A VariantTrack object.                                      |
| chrom  | Optional chromosome filter. May be used without start/end.  |
| start  | Optional 1-based start coordinate. Must be paired with end. |
| end    | Optional 1-based end coordinate. Must be paired with start. |
| by     | Grouping columns. Default c("chrom", "variant_type").       |

## Value

A data.table summary.

## Examples

```
vcf_file <- system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR")
vcf <- read_vcf(vcf_file, mode = "memory", verbose = FALSE)
summary_vcf(vcf)
summary_vcf(vcf, chrom = "chr1")
```

---

|              |                                   |
|--------------|-----------------------------------|
| validate_bwg | <i>Validate a BwgTrack object</i> |
|--------------|-----------------------------------|

---

**Description**

Validate a BwgTrack object

**Usage**

```
validate_bwg(object)
```

**Arguments**

object            A BwgTrack object.

**Value**

A validation list.

**Examples**

```
signal_file <- system.file(
  "extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"
)
signal <- read_bwg(
  signal_file, format = "bedgraph", sample_names = "RNA_plus",
  strand = "+", mode = "memory", verbose = FALSE
)
validate_bwg(signal)
```

---

|                  |                                             |
|------------------|---------------------------------------------|
| validate_feature | <i>Validate a Feature annotation object</i> |
|------------------|---------------------------------------------|

---

**Description**

validate\_feature() is the unified validation entry for annotation data. It validates BED/GFF/GTF/GenePred-derived Feature objects. If gene-model tables are present, transcript/exon consistency is checked in addition to the flat standardized feature table.

**Usage**

```
validate_feature(object, check_gene_model = TRUE)
```

**Arguments**

`object`            A Feature-compatible annotation object.  
`check_gene_model`       Logical. Whether to check derived gene-model tables when transcripts and exons are available.

**Value**

A validation list with `invalid_records`, `invalid_summary`, and `warnings`.

**Examples**

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
validate_feature(gp)
```

---

|                           |                                       |
|---------------------------|---------------------------------------|
| <code>validate_vcf</code> | <i>Validate a VariantTrack object</i> |
|---------------------------|---------------------------------------|

---

**Description**

`validate_vcf()` validates genome-level variant records stored in a `VariantTrack` object. It checks chromosome and position fields, allele fields, duplicated variant IDs, and consistency between `pos`, `start`, and `end` when those columns are available.

**Usage**

```
validate_vcf(object)
```

**Arguments**

`object`            A `VariantTrack` object.

**Value**

A validation list with `invalid_records`, `invalid_summary`, and `warnings`.

**Examples**

```
vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
vcf <- read_vcf(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
validate_vcf(vcf)
```

---

|              |                                     |
|--------------|-------------------------------------|
| VariantTrack | <i>Create a VariantTrack object</i> |
|--------------|-------------------------------------|

---

### Description

VariantTrack() stores VCF-like variant records as genomic point features. Coordinates are stored internally as 1-based positions with `start == end`.

### Usage

```
VariantTrack(data = NULL, meta = list())
```

### Arguments

|                   |                                                                                          |
|-------------------|------------------------------------------------------------------------------------------|
| <code>data</code> | A data.frame or data.table containing at least <code>chrom</code> and <code>pos</code> . |
| <code>meta</code> | A list of metadata.                                                                      |

### Value

A VariantTrack object.

### Examples

```
VariantTrack(data.frame(chrom = "chr1", pos = 10L, ref = "A", alt = "G"))
```

---

|                        |                                   |
|------------------------|-----------------------------------|
| <code>write_bwg</code> | <i>Write BwgTrack signal data</i> |
|------------------------|-----------------------------------|

---

### Description

Writes a BwgTrack object to bedGraph, wig, or bigWig files. All three formats are written directly in pure R. bigWig output uses GeneTrackR's built-in binary writer and requires chromosome sizes; no external conversion program or compiled library is required. All in-memory formats are dispatched through the same internal pure-R I/O layer.

For in-memory BwgTrack objects, signal records are written from `object$data`. For lazy objects, direct conversion is not possible because signal records are not loaded; if the requested format matches the original file format, the original files are copied to the output directory.

Usage

```
write_bwg(  
  object,  
  outdir,  
  format = c("bedgraph", "wig", "bigwig"),  
  samples = NULL,  
  chrom_sizes = NULL,  
  overwrite = FALSE,  
  compress = FALSE  
)
```

Arguments

|             |                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------|
| object      | A BwgTrack object.                                                                                                       |
| outdir      | Output directory.                                                                                                        |
| format      | Output format. One of bedgraph, wig, or bigwig.                                                                          |
| samples     | Optional sample IDs to write. Default writes all samples.                                                                |
| chrom_sizes | Chromosome sizes for bigWig output. Can be a file path or a data.frame/data.table with two columns: chromosome and size. |
| overwrite   | Whether to overwrite existing files.                                                                                     |
| compress    | Whether to gzip-compress bedGraph or wig output.                                                                         |

Value

Invisibly returns a data.table containing sample IDs and output files.

Examples

```
rnaseq <- read_bwg(  
  system.file("extdata", "gtr_demo_rnaseq_plus.bedgraph", package = "GeneTrackR"),  
  format = "bedgraph", sample_names = "RNA_seq_plus", strand = "+",  
  mode = "memory", verbose = FALSE  
)  
outdir <- tempfile("gtr_signal_")  
write_bwg(rnaseq, outdir = outdir, format = "bedgraph", overwrite = TRUE)
```

---

|               |                                      |
|---------------|--------------------------------------|
| write_feature | <i>Write Feature annotation data</i> |
|---------------|--------------------------------------|

---

Description

Writes a unified Feature annotation object to GenePred, GenePredExt, GFF3, GTF, or BED format. This is the main output function for annotation tracks produced by read\_genepred(), read\_gff(), read\_gtf(), or read\_bed().

**Usage**

```
write_feature(
  object,
  file,
  format = c("auto", "genepred", "genepredext", "gff", "gtf", "bed6", "bed12"),
  coordinate = c("ucsc", "granges"),
  overwrite = FALSE,
  keep_attributes = TRUE,
  sort_output = TRUE,
  chrom_order = NULL
)
```

**Arguments**

|                 |                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object          | A Feature, FeatureTrack, or GenePred-compatible object.                                                                                                                                                                                                                                  |
| file            | Output file path.                                                                                                                                                                                                                                                                        |
| format          | Output format. Use auto, genepred, genepredext, gff, gtf, bed6, or bed12. bed6 writes six-column BED intervals. For gene-model objects, BED6 is transcript-level. For generic interval objects, BED6 is feature-level. bed12 writes transcript-level BED12 gene models with exon blocks. |
| coordinate      | Output coordinate system for GenePred/BED-like formats. ucsc means 0-based half-open where applicable; granges means 1-based closed.                                                                                                                                                     |
| overwrite       | Whether to overwrite an existing file.                                                                                                                                                                                                                                                   |
| keep_attributes | Whether to reuse existing attribute strings for GFF/GTF output when available.                                                                                                                                                                                                           |
| sort_output     | Whether to sort records before writing. Default TRUE. GenePred output is sorted by chromosome and transcript start. GFF/GTF output is sorted by chromosome, start, and feature hierarchy.                                                                                                |
| chrom_order     | Optional chromosome order used for sorting. It can be NULL, a character vector of chromosome names, a data frame whose first column contains chromosome names, or a FASTA index .fai file path. If NULL, a natural chromosome order is used.                                             |

**Value**

Invisibly returns the output file path.

**Examples**

```
gp <- read_genepred(
  system.file("extdata", "gtr_demo.genePredExt", package = "GeneTrackR"),
  format = "genePredExt", verbose = FALSE, progress = FALSE
)
outfile <- tempfile(fileext = ".bed")
write_feature(gp, outfile, format = "bed12", overwrite = TRUE)
```

---

|                     |                                    |
|---------------------|------------------------------------|
| write_feature_track | <i>Write a FeatureTrack object</i> |
|---------------------|------------------------------------|

---

## Description

Backward-compatible wrapper around write\_feature().

## Usage

```
write_feature_track(
  object,
  file,
  format = c("bed12", "bed6", "gff", "gtf", "genepred", "genepredext"),
  sort_output = TRUE,
  chrom_order = NULL
)
```

## Arguments

|             |                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object      | A Feature, FeatureTrack, or GenePred-compatible object.                                                                                                                                                                                                                                  |
| file        | Output file path.                                                                                                                                                                                                                                                                        |
| format      | Output format. Use auto, genepred, genepredext, gff, gtf, bed6, or bed12. bed6 writes six-column BED intervals. For gene-model objects, BED6 is transcript-level. For generic interval objects, BED6 is feature-level. bed12 writes transcript-level BED12 gene models with exon blocks. |
| sort_output | Whether to sort records before writing. Default TRUE. GenePred output is sorted by chromosome and transcript start. GFF/GTF output is sorted by chromosome, start, and feature hierarchy.                                                                                                |
| chrom_order | Optional chromosome order used for sorting. It can be NULL, a character vector of chromosome names, a data frame whose first column contains chromosome names, or a FASTA index . fai file path. If NULL, a natural chromosome order is used.                                            |

## Value

Invisibly returns the output file path.

## Examples

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
  gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
)
features <- as_feature(gp)
outfile <- tempfile(fileext = ".bed")
write_feature_track(features, outfile)
```

---

|                |                                                      |
|----------------|------------------------------------------------------|
| write_genepred | <i>Write a GenePred-compatible annotation object</i> |
|----------------|------------------------------------------------------|

---

## Description

Backward-compatible wrapper around write\_feature().

## Usage

```
write_genepred(
  object,
  file,
  format = c("genePred", "genePredExt"),
  coordinate = c("ucsc", "granges"),
  sort_output = TRUE,
  chrom_order = NULL
)
```

## Arguments

|             |                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object      | A Feature, FeatureTrack, or GenePred-compatible object.                                                                                                                                                                                                                                  |
| file        | Output file path.                                                                                                                                                                                                                                                                        |
| format      | Output format. Use auto, genePred, genePredExt, gff, gtf, bed6, or bed12. bed6 writes six-column BED intervals. For gene-model objects, BED6 is transcript-level. For generic interval objects, BED6 is feature-level. bed12 writes transcript-level BED12 gene models with exon blocks. |
| coordinate  | Output coordinate system for GenePred/BED-like formats. ucsc means 0-based half-open where applicable; granges means 1-based closed.                                                                                                                                                     |
| sort_output | Whether to sort records before writing. Default TRUE. GenePred output is sorted by chromosome and transcript start. GFF/GTF output is sorted by chromosome, start, and feature hierarchy.                                                                                                |
| chrom_order | Optional chromosome order used for sorting. It can be NULL, a character vector of chromosome names, a data frame whose first column contains chromosome names, or a FASTA index .fai file path. If NULL, a natural chromosome order is used.                                             |

## Value

Invisibly returns the output file path.

## Examples

```
gp_file <- system.file(
  "extdata", "gtr_demo.genePredExt", package = "GeneTrackR"
)
gp <- read_genepred(
```

```

    gp_file, format = "genePredExt", verbose = FALSE, progress = FALSE
  )
  outfile <- tempfile(fileext = ".genePred")
  write_genepred(gp, outfile)

```

---

|                     |                                    |
|---------------------|------------------------------------|
| write_variant_track | <i>Write a VariantTrack object</i> |
|---------------------|------------------------------------|

---

## Description

Backward-compatible wrapper around write\_vcf().

## Usage

```
write_variant_track(object, file, overwrite = FALSE, include_header = TRUE)
```

## Arguments

|                |                                              |
|----------------|----------------------------------------------|
| object         | A VariantTrack object created by read_vcf(). |
| file           | Output VCF file path.                        |
| overwrite      | Whether to overwrite an existing file.       |
| include_header | Whether to include minimal VCF header lines. |

## Value

Invisibly returns the output file path.

## Examples

```

vcf_file <- system.file(
  "extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"
)
vcf <- read_vcf(
  vcf_file, mode = "memory", verbose = FALSE, progress = FALSE
)
outfile <- tempfile(fileext = ".vcf")
write_variant_track(vcf, outfile, overwrite = TRUE)

```

---

`write_vcf`*Write a VariantTrack object to VCF*

---

**Description**

Write a VariantTrack object to VCF

**Usage**

```
write_vcf(object, file, overwrite = FALSE, include_header = TRUE)
```

**Arguments**

|                             |                                                            |
|-----------------------------|------------------------------------------------------------|
| <code>object</code>         | A VariantTrack object created by <code>read_vcf()</code> . |
| <code>file</code>           | Output VCF file path.                                      |
| <code>overwrite</code>      | Whether to overwrite an existing file.                     |
| <code>include_header</code> | Whether to include minimal VCF header lines.               |

**Value**

Invisibly returns the output file path.

**Examples**

```
vcf <- read_vcf(  
  system.file("extdata", "gtr_demo_variants.vcf", package = "GeneTrackR"),  
  mode = "memory", verbose = FALSE, progress = FALSE  
)  
outfile <- tempfile(fileext = ".vcf")  
write_vcf(vcf, outfile, overwrite = TRUE)
```

# Index

as\_exon\_table, [4](#)  
as\_feature, [4](#)  
as\_feature\_table, [5](#)  
as\_gene\_table, [6](#)  
as\_genepred, [6](#)  
as\_granges, [7](#)  
as\_transcript\_table, [8](#)  
  
benchmark\_tabix\_query, [8](#)  
bin\_bwg, [9](#)  
BwgTrack, [10](#)  
  
compute\_ld\_block, [11](#)  
  
diagnose\_tabix, [12](#)  
  
Feature, [13](#)  
FeatureTrack, [14](#)  
  
GenePred, [15](#)  
GeneTrackR-advanced-parameters, [16](#), [32](#),  
[34](#), [50](#), [53](#), [57](#), [60](#), [66](#)  
GeneTrackR-demo-data, [18](#)  
get\_gene\_length\_distribution\_table, [19](#)  
get\_gene\_length\_distribution\_table(),  
[33](#)  
get\_genepred\_length\_table  
(get\_gene\_length\_distribution\_table),  
[19](#)  
  
hap\_gene\_variant, [20](#)  
hap\_region\_variant, [22](#)  
hap\_variant, [23](#)  
head.BwgTrack, [25](#)  
  
LDTrack, [25](#)  
  
merge\_bwg, [26](#)  
merge\_feature, [27](#)  
  
norm\_bwg, [28](#)  
  
plot\_feature\_track, [29](#)  
plot\_gene, [31](#)  
plot\_gene(), [16](#), [17](#)  
plot\_gene\_length\_distribution, [32](#)  
plot\_genepred\_length\_distribution  
(plot\_gene\_length\_distribution),  
[32](#)  
plot\_hap\_pheno, [34](#)  
plot\_hap\_refined, [36](#)  
plot\_hap\_variant, [37](#)  
plot\_ld\_block, [40](#)  
plot\_pheno, [42](#)  
plot\_refined\_hap\_pheno, [43](#)  
plot\_refined\_hap\_variant, [46](#)  
plot\_region, [49](#)  
plot\_region(), [16](#), [17](#)  
plot\_signal\_gene, [51](#)  
plot\_signal\_gene(), [17](#)  
plot\_signal\_region, [54](#)  
plot\_signal\_region(), [17](#)  
plot\_signal\_transcript, [57](#)  
plot\_signal\_transcript(), [17](#)  
plot\_tracks, [61](#)  
plot\_tracks(), [17](#)  
plot\_transcript, [65](#)  
plot\_transcript(), [16](#), [17](#)  
plot\_variant, [67](#)  
plot\_variant\_effect, [68](#)  
plot\_variant\_pheno, [70](#)  
plot\_variant\_track, [72](#)  
print.GeneTrackRPhenoPlot, [74](#)  
  
read\_bed, [74](#)  
read\_bwg, [75](#)  
read\_bwg(), [17](#)  
read\_genepred, [77](#)  
read\_gff, [78](#)  
read\_gtf, [79](#)  
read\_pheno, [80](#)  
read\_vcf, [81](#)

read\_vcf\_track, [82](#)  
refine\_haploptype, [83](#)  
retrieve\_bwg, [84](#)  
retrieve\_bwg(), [17](#)  
retrieve\_feature, [86](#)  
retrieve\_feature(), [19](#), [33](#)  
retrieve\_vcf, [88](#)  
  
seqinfo\_bwg, [89](#)  
slice\_bwg, [90](#)  
summary\_bwg, [91](#)  
summary\_feature, [92](#)  
summary\_pheno, [93](#)  
summary\_vcf, [94](#)  
  
validate\_bwg, [95](#)  
validate\_feature, [95](#)  
validate\_vcf, [96](#)  
VariantTrack, [97](#)  
  
write\_bwg, [97](#)  
write\_feature, [98](#)  
write\_feature\_track, [100](#)  
write\_genepred, [101](#)  
write\_variant\_track, [102](#)  
write\_vcf, [103](#)