Package 'ggvariant'

May 8, 2026

Title Tidy, 'ggplot2'-Native Visualization for Genomic Variants

Version 0.1.0

Description A simple, opinionated toolkit for visualizing genomic variant data using a 'ggplot2'-native grammar. Accepts VCF files or plain data frames and produces publication-ready lollipop plots, consequence summaries, mutational spectrum charts, and cohort-level comparisons with minimal code. Designed for both wet-lab biologists and experienced bioinformaticians.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports ggplot2 (>= 3.4.0), cli (>= 3.6.0), scales (>= 1.3.0)

Suggests plotly (>= 4.10.0), testthat (>= 3.0.0), covr, knitr, rmarkdown

Config/testthat/edition 3

URL https://github.com/josh45-source/ggvariant

BugReports https://github.com/josh45-source/ggvariant/issues

VignetteBuilder knitr

NeedsCompilation no

Author Joash Joshua Ayo [aut, cre] (ORCID: <https://orcid.org/0009-0007-1642-0172>)

Maintainer Joash Joshua Ayo <joashjoshua789@gmail.com>

Repository CRAN

Date/Publication 2026-02-27 20:40:02 UTC

Contents

`ggvariant-package` *ggvariant: Tidy, ggplot2-Native Visualization for Genomic Variants*

Description

A simple, opinionated toolkit for visualizing genomic variant data using a 'ggplot2'-native grammar. Accepts VCF files or plain data frames and produces publication-ready lollipop plots, consequence summaries, mutational spectrum charts, and cohort-level comparisons with minimal code. Designed for both wet-lab biologists and experienced bioinformaticians.

Author(s)

Maintainer: Joash Joshua Ayo <joashjoshua789@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/josh45-source/ggvariant
- Report bugs at https://github.com/josh45-source/ggvariant/issues

---

`coerce_variants` *Coerce a plain data frame to a* `gvf` *object*

Description

If you already have variant data in a `data.frame` (e.g. exported from Excel, a database, or another tool), use this function to prepare it for use with `ggvariant` plotting functions.

Usage

```
coerce_variants(
  x,
  chrom = "chrom",
  pos = "pos",
  ref = "ref",
  alt = "alt",
  consequence = "consequence",
  gene = "gene",
  sample = "sample"
)
```

Arguments

x	A data.frame or tibble.
chrom	Column name containing chromosome (default "chrom").
pos	Column name containing position (default "pos").
ref	Column name containing reference allele (default "ref").
alt	Column name containing alternate allele (default "alt").
consequence	Column name containing variant consequence annotation, e.g. "Missense_Mutation". If NULL, consequence is inferred from REF/ALT lengths.
gene	Column name containing gene symbol (default "gene").
sample	Column name containing sample identifier (default "sample").

Value

A gvf object.

Examples

```
df <- data.frame(
  chromosome = c("chr1", "chr1", "chr7"),
  position   = c(100200, 100350, 55249071),
  ref_allele = c("A", "G", "C"),
  alt_allele = c("T", "A", "T"),
  variant_class = c("missense_variant", "synonymous_variant", "missense_variant"),
  hugo_symbol = c("GENE1", "GENE1", "EGFR"),
  tumor_sample = c("S1", "S2", "S2")
)

variants <- coerce_variants(df,
  chrom       = "chromosome",
  pos         = "position",
  ref         = "ref_allele",
  alt         = "alt_allele",
  consequence = "variant_class",
  gene        = "hugo_symbol",
  sample      = "tumor_sample"
```

```
)
```

`gv_palette` *ggvariant colour palettes*

Description

Access the built-in colour palettes used by `ggvariant` plot functions.

Usage

```
gv_palette(type = c("consequence", "spectrum", "domain"), n = 8L)
```

Arguments

`type`	One of `"consequence"` (default), `"spectrum"`, or `"domain"`.
`n`	Integer. For `"domain"`, the number of colours to generate.

Value

A named character vector of hex colour codes.

Examples

```
gv_palette("consequence")
gv_palette("spectrum")
```

`plot_consequence_summary`
Consequence summary bar chart

Description

Summarises variant consequences (e.g. missense, frameshift, synonymous) across one or more samples, producing a stacked or grouped bar chart.

Usage

```
plot_consequence_summary(
  variants,
  samples = NULL,
  group_by = c("consequence", "gene"),
  top_n = 10L,
  position = c("stack", "fill", "dodge"),
  palette = NULL,
  flip = FALSE,
  interactive = FALSE
)
```

Arguments

variants	A gvf object or compatible data.frame.
samples	Character vector of sample names to include. NULL (default) uses all samples. Ignored if there is no sample column.
group_by	"consequence" (default) stacks bars by consequence per sample; "gene" stacks by gene per consequence.
top_n	Integer. For group_by = "gene", show only the top N genes by total variant count. Default 10.
position	"stack" (default) or "fill" (proportional) or "dodge".
palette	Named character vector of colours. NULL uses built-in.
flip	Logical. If TRUE, flips coordinates for horizontal bars. Default FALSE.
interactive	Logical. Returns a plotly object if TRUE.

Value

A ggplot object.

Examples

```
vcf_file <- system.file("extdata", "example.vcf", package = "ggvariant")
variants <- read_vcf(vcf_file)

# Consequence counts per sample
plot_consequence_summary(variants)

# Proportional bars
plot_consequence_summary(variants, position = "fill")

# Top 10 genes coloured by consequence
plot_consequence_summary(variants, group_by = "gene", top_n = 10)
```

plot_lollipop *Lollipop plot of variants along a gene*

Description

Draws a lollipop (stem-and-dot) diagram showing variant positions along a gene, coloured by consequence. Optionally overlays protein domain annotations when domain boundaries are supplied.

Usage

```
plot_lollipop(
  variants,
  gene = NULL,
  domains = NULL,
  color_by = "consequence",
  palette = NULL,
  protein_length = NULL,
  stack_dots = TRUE,
  title = NULL,
  interactive = FALSE
)
```

Arguments

variants	A gvf object from read_vcf() or coerce_variants(), or any data.frame with columns pos, consequence, and optionally gene and sample.
gene	Character. Gene to filter on. If NULL and variants contains a gene column, the most-mutated gene is chosen automatically.
domains	A data.frame with columns name, start, end (amino acid positions) for domain annotation. NULL (default) omits domains.
color_by	Column name to use for dot colour. Default "consequence". Set to "sample" to colour by sample instead.
palette	Named character vector of colours for each consequence/sample category. NULL uses the built-in ggvariant palette.
protein_length	Integer. Total length of the protein in amino acids, used to scale the x-axis. If NULL, inferred from max(pos).
stack_dots	Logical. If TRUE (default), dots at the same position are stacked vertically (beeswarm-style) rather than overlapping.
title	Character. Plot title. Defaults to the gene name.
interactive	Logical. If TRUE, returns a plotly interactive plot (requires the plotly package).

Value

A ggplot object (or a plotly object when interactive = TRUE).

Examples

```
vcf_file <- system.file("extdata", "example.vcf", package = "ggvariant")
variants <- read_vcf(vcf_file)

# Basic lollipop for the most-mutated gene
plot_lollipop(variants)

# Specific gene
plot_lollipop(variants, gene = "TP53")

# With domain annotation
tp53_domains <- data.frame(
  name  = c("Transactivation", "DNA-binding", "Tetramerization"),
  start = c(1, 102, 323),
  end   = c(67, 292, 356)
)
plot_lollipop(variants, gene = "TP53", domains = tp53_domains)
```

---

plot_variant_spectrum *Mutational spectrum (SBS) bar chart*

Description

Plots the single-base substitution (SBS) spectrum — the relative frequency of each of the 6 substitution classes (C>A, C>G, C>T, T>A, T>C, T>G) — optionally broken down by trinucleotide context.

Usage

```
plot_variant_spectrum(
  variants,
  sample = NULL,
  context = FALSE,
  genome = NULL,
  facet_by_sample = FALSE,
  palette = NULL,
  normalize = TRUE,
  interactive = FALSE
)
```

Arguments

variants	A gvf object or compatible data.frame containing SNVs. Indels are automatically excluded.
sample	Character. Sample name to filter on. NULL uses all variants pooled (or facets by sample if facet_by_sample = TRUE).

context	Logical. If TRUE, shows 96-trinucleotide context bars (requires a context column or a reference genome via genome). Default FALSE.
genome	A BSgenome object or genome abbreviation string (e.g. "hg38") used to extract trinucleotide context when context = TRUE and no context column is present. Requires the BSgenome and Biostrings packages.
facet_by_sample	Logical. If TRUE, facets the plot by sample. Default FALSE.
palette	Named character vector with names matching substitution classes ("C>A", "C>G", etc.). NULL uses COSMIC-style colours.
normalize	Logical. If TRUE (default), shows relative proportions. If FALSE, shows raw counts.
interactive	Logical. Returns a plotly object if TRUE.

Value

A ggplot object.

Examples

```
vcf_file <- system.file("extdata", "example.vcf", package = "ggvariant")
variants <- read_vcf(vcf_file)

# Basic 6-class SBS spectrum
plot_variant_spectrum(variants)

# Faceted by sample
plot_variant_spectrum(variants, facet_by_sample = TRUE)
```

---

read_vcf *Read a VCF file into a tidy variant data frame*

Description

Parses a standard VCF (v4.x) file and returns a tidy data.frame (a gvf object) that all ggvariant plotting functions accept. For users who already have variant data in a plain data.frame or tibble, see coerce_variants().

Usage

```
read_vcf(path, samples = NULL, pass_only = TRUE, info_fields = NULL)
```

Arguments

path	Path to a .vcf or .vcf.gz file.
samples	Character vector of sample names to retain. NULL (default) keeps all samples.
pass_only	Logical. If TRUE (default), only variants with FILTER equal to "PASS" or "." are retained.
info_fields	Character vector of INFO field names to expand into columns. NULL keeps none. Use "all" to expand everything (may be slow for large files).

Value

A gvf (genomic variant frame) — a data.frame with columns:

chrom Chromosome (character)

pos Position (integer)

ref Reference allele

alt Alternate allele (multi-allelic sites are split into rows)

qual QUAL score (numeric)

filter FILTER field

sample Sample name (NA for single-sample VCFs without GT field)

consequence Variant consequence if ANN/CSQ INFO field is present

gene Gene symbol if ANN/CSQ INFO field is present

See Also

coerce_variants(), plot_lollipop(), plot_consequence_summary()

Examples

```
vcf_file <- system.file("extdata", "example.vcf", package = "ggvariant")
variants <- read_vcf(vcf_file)
head(variants)
```

---

theme_ggvariant *ggvariant ggplot2 theme*

Description

A clean, publication-ready theme based on theme_minimal. Applied automatically by all ggvariant plot functions; export it to customise further.

Usage

```
theme_ggvariant(base_size = 12, base_family = "")
```

Arguments

base_size	Base font size in pt. Default 12.
base_family	Base font family. Default "" (system sans-serif).

Value

A ggplot2 theme object.

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg, wt)) + geom_point() + theme_ggvariant()
```

Index