

Package ‘ggvariant’

August 21, 2026

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

Version 0.2.0

Description A simple, opinionated toolkit for visualizing genomic variant data using a 'ggplot2'-native grammar. Accepts VCF files or plain data frames and produces lollipop plots, consequence summaries, mutational spectrum charts, and cohort-level comparisons as standard 'ggplot2' objects. 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, vdiff

Config/testthat/edition 3

Config/Needs/website pkgdown, rmarkdown

URL <https://github.com/josh45-source/ggvariant>,
<https://josh45-source.github.io/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 2026-08-21 05:43:32 UTC

Contents

ggvariant-package	2
coerce_variants	3
gvf-methods	4
gv_palette	5
plot_consequence_summary	6
plot_lollipop	7
plot_oncoprint	9
plot_tmb	11
plot_variant_spectrum	13
read_vcf	14
theme_ggvariant	16
Index	17

ggvariant-package	<i>ggvariant: Tidy, ggplot2-Native Visualization for Genomic Variants</i>
-------------------	---

Description

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

Author(s)

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

References

Danecek P, Auton A, Abecasis G, et al.; 1000 Genomes Project Analysis Group (2011). The variant call format and VCFtools. *Bioinformatics*, 27(15), 2156-2158. doi:10.1093/bioinformatics/btr330

Alexandrov LB, Kim J, Haradhvala NJ, et al.; PCAWG Consortium (2020). The repertoire of mutational signatures in human cancer. *Nature*, 578(7793), 94-101. doi:10.1038/s4158602019433

See Also

- Useful links:
- <https://github.com/josh45-source/ggvariant>
 - <https://josh45-source.github.io/ggvariant/>
 - Report bugs at <https://github.com/josh45-source/ggvariant/issues>

coerce_variants	<i>Coerce a plain data frame to a gvf object</i>
-----------------	--

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 <code>data.frame</code> 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.

See Also

[read_vcf\(\)](#)

Other ggvariant input: [read_vcf\(\)](#)

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"
)
```

gvf-methods

Print and summarise gvf objects

Description

`print.gvf()` shows a compact header (variant / sample / chromosome / gene counts) followed by a truncated preview, rather than dumping the full underlying `data.frame`. `summary.gvf()` returns consequence, per-sample, and per-chromosome breakdowns as a `summary.gvf` object, printed via its own method.

Usage

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

## S3 method for class 'gvf'
summary(object, ...)

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

Arguments

<code>x</code> , object	A gvf object.
<code>...</code>	Passed to further methods; currently unused.
<code>n</code>	Integer. Number of rows to preview in <code>print.gvf()</code> . Default 6.

Value

`print.gvf()` returns `x`, invisibly. `summary.gvf()` returns a `summary.gvf` object.

Examples

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

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

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

Details

The "consequence" palette covers the most common VEP/SnpEff/MAF consequence terms; any other term is standardised to an "Other" bucket (its own entry in this palette) by plot functions rather than being dropped.

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 from read_vcf() or coerce_variants() , or any data.frame with columns pos, consequence, and optionally gene and sample.
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 for each consequence/sample category. NULL uses the built-in ggvariant palette.
flip	Logical. If TRUE, flips coordinates for horizontal bars. Default FALSE.
interactive	Logical. If TRUE, returns a plotly interactive plot (requires the plotly package).

Value

A ggplot object.

See Also

[plot_lollipop\(\)](#), [plot_variant_spectrum\(\)](#), [gv_palette\(\)](#)

Other ggvariant plots: [plot_lollipop\(\)](#), [plot_oncoprint\(\)](#), [plot_tmb\(\)](#), [plot_variant_spectrum\(\)](#)

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	<i>Lollipop plot of variants along a gene</i>
---------------	---

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 (default), inferred from max(pos) and the x-axis is labelled "Genomic position", since pos is assumed to be a raw genomic coordinate. Supplying protein_length is taken as a signal that pos has already been rescaled to protein coordinates (as in the @examples below), and labels the x-axis "Amino acid position" instead.
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).

Details

domains accepts any protein domain boundaries you supply. Pfam (Paysan-Lafosse et al. 2025) is one source for real domain coordinates, rather than typing them by hand as in the @examples below.

Value

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

References

Paysan-Lafosse T, Andreeva A, Blum M, et al. (2025). The Pfam protein families database: embracing AI/ML. *Nucleic Acids Research*, 53(D1), D523-D534. doi:10.1093/nar/gkae997

See Also

[plot_consequence_summary\(\)](#), [plot_variant_spectrum\(\)](#), [gv_palette\(\)](#)

Other ggvariant plots: [plot_consequence_summary\(\)](#), [plot_oncoprint\(\)](#), [plot_tmb\(\)](#), [plot_variant_spectrum\(\)](#)

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_oncoprint	<i>Oncoprint / waterfall plot of variants across samples</i>
----------------	--

Description

Draws a gene-by-sample mutation matrix ("oncoprint" in the ComplexHeatmap/cBioPortal vocabulary, "waterfall plot" in the GenVisR/maftools vocabulary — the same visualisation under two names). `plot_waterfall()` is an alias for `plot_oncoprint()`; both produce identical output.

Usage

```
plot_oncoprint(
  variants,
  top_n = NULL,
  genes = NULL,
  samples = NULL,
  annotation = NULL,
  palette = NULL,
  interactive = FALSE
)

plot_waterfall(
  variants,
  top_n = NULL,
  genes = NULL,
  samples = NULL,
  annotation = NULL,
  palette = NULL,
  interactive = FALSE
)
```

Arguments

<code>variants</code>	A gvf object from <code>read_vcf()</code> or <code>coerce_variants()</code> , or any <code>data.frame</code> with gene, sample, and consequence columns.
<code>top_n</code>	Integer. Show the <code>top_n</code> most frequently altered genes. Mutually exclusive with <code>genes</code> ; if both are <code>NULL</code> , defaults to 10.

genes	Character vector of specific genes to show, in place of top_n. Mutually exclusive with top_n. Genes not present in variants are shown as fully unaltered rows.
samples	Character vector of sample names to include as columns. NULL (default) uses every sample present in variants.
annotation	Optional data.frame of sample-level metadata drawn as annotation tracks below the mutation matrix. Must contain a sample column matching sample identifiers in variants, plus one or more additional columns to display, one per track. NULL (default) omits annotation tracks.
palette	Named character vector of colours keyed by consequence. NULL uses the built-in <code>gv_palette("consequence")</code> . If it does not already contain "Multi_Hit" or "Other" entries, they are added automatically. Unlike <code>plot_consequence_summary()</code> , which shows the full breakdown of consequence types, <code>plot_oncoprint()</code> only distinguishes missense_variant, stop_gained, frameshift_variant, synonymous_variant, and Multi_Hit by colour (plus their common MAF/SnpEff aliases, e.g. Missense_Mutation); every other consequence term is shown under "Other" rather than being dropped.
interactive	Logical. Returns a plotly object if TRUE.

Details

Gene order. Genes are ranked by the number of distinct samples carrying at least one mutation in that gene (ties broken by total mutation count, then alphabetically), and the top_n most frequently altered genes are shown, most-altered at the top.

Sample order. Samples are ordered using the memo-sort ("cascade") algorithm: the displayed genes, already ranked most-to-least altered, are treated as bits of a binary number (the most-altered gene the most significant bit). Each sample's mutation pattern across the displayed genes is scored as that binary number, and samples are sorted by descending score (ties broken alphabetically by sample name). This greedily groups samples that share mutations in the top genes together, producing the characteristic left-to-right staircase pattern. Only the displayed genes contribute to the score; genes excluded by top_n/ genes have no effect on sample order.

Multi-hit cells. A gene mutated more than once in the same sample cannot be represented by a single consequence colour, so it is shown as a distinct "Multi_Hit" category instead of either consequence.

Value

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

References

- Gao J, Aksoy BA, Dogrusoz U, et al. (2013). Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Science Signaling*, 6(269), pl1. doi:10.1126/scisignal.2004088
- Skidmore ZL, Wagner AH, Lesurf R, et al. (2016). GenVisR: Genomic Visualizations in R. *Bioinformatics*, 32(19), 3012-3014. doi:10.1093/bioinformatics/btw325
- Gu Z, Eils R, Schlesner M (2016). Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics*, 32(18), 2847-2849. doi:10.1093/bioinformatics/btw313

Mayakonda A, Lin DC, Assenov Y, Plass C, Koeffler HP (2018). Maftools: efficient and comprehensive analysis of somatic variants in cancer. *Genome Research*, 28(11), 1747-1756. doi:[10.1101/gr.239244.118](https://doi.org/10.1101/gr.239244.118)

See Also

`plot_tmb()`, `plot_consequence_summary()`, `gv_palette()`

Other ggvariant plots: `plot_consequence_summary()`, `plot_lollipop()`, `plot_tmb()`, `plot_variant_spectrum()`

Examples

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

# Top 5 most-altered genes
plot_oncoprint(variants, top_n = 5)

# A specific gene panel instead of top_n
plot_oncoprint(variants, genes = c("TP53", "BRCA1", "BRCA2"))

# With a clinical annotation track
clinical <- data.frame(
  sample = c("TUMOR_S1", "TUMOR_S2"),
  stage = c("III", "IV")
)
plot_oncoprint(variants, top_n = 5, annotation = clinical)
```

plot_tmb

Tumour mutational burden bar chart

Description

Plots per-sample tumour mutational burden (TMB): the number of mutations carried by each sample, optionally normalised to mutations per megabase.

Usage

```
plot_tmb(
  variants,
  samples = NULL,
  mb_size = NULL,
  sample_order = NULL,
  palette = NULL,
  interactive = FALSE
)
```

Arguments

variants	A gvf object from <code>read_vcf()</code> or <code>coerce_variants()</code> , or any <code>data.frame</code> with <code>sample</code> and <code>pos</code> columns.
samples	Character vector of sample names to include. NULL (default) uses every sample present in <code>variants</code> .
mb_size	Numeric. Size, in megabases, of the sequenced region used to normalise mutation counts into mutations/Mb. NULL (default) plots raw mutation counts instead.
sample_order	Character vector giving an explicit left-to-right sample order, e.g. to align with <code>plot_oncoprint()</code> 's memo-sorted column order. NULL (default) orders samples by descending TMB.
palette	Single hex colour string for the bars. NULL uses a built-in default.
interactive	Logical. Returns a plotly object if TRUE.

Details

The per-sample mutation count is computed by an internal helper shared with any future `plot_oncoprint()` TMB marginal, so the two always agree. Pass `mb_size` (the size, in megabases, of the sequenced region) to convert raw counts into mutations/Mb, the conventional TMB unit; leave it NULL to plot raw counts.

Value

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

References

Chalmers ZR, Connelly CF, Fabrizio D, et al. (2017). Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Medicine*, 9(1), 34. doi:10.1186/s1307301704242

See Also

`plot_oncoprint()`, `gv_palette()`

Other ggvariant plots: `plot_consequence_summary()`, `plot_lollipop()`, `plot_oncoprint()`, `plot_variant_spectrum()`

Examples

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

# Raw mutation counts, samples ordered by descending TMB
plot_tmb(variants)

# Normalised to mutations/Mb for a 38 Mb exome
plot_tmb(variants, mb_size = 38)

# Aligned to a specific sample order, e.g. from plot_oncoprint()
```

```
plot_tmb(variants, sample_order = c("TUMOR_S1", "TUMOR_S2"))
```

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. 96-trinucleotide context bars are not yet implemented; passing TRUE aborts with an error. Default FALSE, which produces the 6-class SBS spectrum. See https://github.com/josh45-source/ggvariant/issues/1 .
genome	Not yet implemented; passing a non-NULL value aborts with an error. Reserved for future BSgenome-based trinucleotide context extraction. See https://github.com/josh45-source/ggvariant/issues/1 .
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. If TRUE, returns a plotly interactive plot (requires the plotly package).

Value

A ggplot object.

References

Alexandrov LB, Kim J, Haradhvala NJ, et al.; PCAWG Consortium (2020). The repertoire of mutational signatures in human cancer. *Nature*, 578(7793), 94-101. doi:[10.1038/s4158602019433](https://doi.org/10.1038/s4158602019433)

Blokzijl F, Janssen R, van Boxtel R, Cuppen E (2018). MutationalPatterns: comprehensive genome-wide analysis of mutational processes. *Genome Medicine*, 10(1), 33. doi:[10.1186/s1307301805390](https://doi.org/10.1186/s1307301805390)

See Also

[plot_lollipop\(\)](#), [plot_consequence_summary\(\)](#), [gv_palette\(\)](#)

Other ggvariant plots: [plot_consequence_summary\(\)](#), [plot_lollipop\(\)](#), [plot_oncoprint\(\)](#), [plot_tmb\(\)](#)

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 Not yet implemented; passing a non-NULL value aborts with an error. Reserved for future INFO field expansion. See <https://github.com/josh45-source/ggvariant/issues/2>.

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

References

Danecek P, Auton A, Abecasis G, et al.; 1000 Genomes Project Analysis Group (2011). The variant call format and VCFtools. *Bioinformatics*, 27(15), 2156-2158. doi:10.1093/bioinformatics/btr330

Cingolani P, Platts A, Wang LL, et al. (2012). A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly*, 6(2), 80-92. doi:10.4161/fly.19695

McLaren W, Gil L, Hunt SE, et al. (2016). The Ensembl Variant Effect Predictor. *Genome Biology*, 17(1), 122. doi:10.1186/s1305901609744

See Also

[coerce_variants\(\)](#), [plot_lollipop\(\)](#), [plot_consequence_summary\(\)](#)

Other ggvariant input: [coerce_variants\(\)](#)

Examples

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

theme_ggvariant	<i>ggvariant ggplot2 theme</i>
-----------------	--------------------------------

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

- * **ggvariant input**
 - coerce_variants, [3](#)
 - read_vcf, [14](#)
- * **ggvariant plots**
 - plot_consequence_summary, [6](#)
 - plot_lollipop, [7](#)
 - plot_oncoprint, [9](#)
 - plot_tmb, [11](#)
 - plot_variant_spectrum, [13](#)
- coerce_variants, [3](#), [15](#)
- coerce_variants(), [6](#), [7](#), [9](#), [12](#), [14](#), [15](#)
- ggvariant (ggvariant-package), [2](#)
- ggvariant-package, [2](#)
- gv_palette, [5](#)
- gv_palette(), [7](#), [8](#), [11](#), [12](#), [14](#)
- gvf-methods, [4](#)
- plot_consequence_summary, [6](#), [8](#), [11](#), [12](#), [14](#)
- plot_consequence_summary(), [8](#), [10](#), [11](#), [14](#), [15](#)
- plot_lollipop, [7](#), [7](#), [11](#), [12](#), [14](#)
- plot_lollipop(), [7](#), [14](#), [15](#)
- plot_oncoprint, [7](#), [8](#), [9](#), [12](#), [14](#)
- plot_oncoprint(), [12](#)
- plot_tmb, [7](#), [8](#), [11](#), [11](#), [14](#)
- plot_tmb(), [11](#)
- plot_variant_spectrum, [7](#), [8](#), [11](#), [12](#), [13](#)
- plot_variant_spectrum(), [7](#), [8](#)
- plot_waterfall (plot_oncoprint), [9](#)
- print.gvf (gvf-methods), [4](#)
- print.summary.gvf (gvf-methods), [4](#)
- read_vcf, [3](#), [14](#)
- read_vcf(), [3](#), [6](#), [7](#), [9](#), [12](#)
- summary.gvf (gvf-methods), [4](#)
- theme_ggvariant, [16](#)