

# G3viz: an R package to interactively visualize genetic mutation data using a lollipop-diagram

## 1 Introduction

Intuitively and effectively visualizing genetic mutation data can help researchers to better understand genomic data and validate findings. **G3viz** is an R package which provides an easy-to-use lollipop-diagram tool. It enables users to interactively visualize detailed translational effect of genetic mutations in RStudio or a web browser, without having to know any HTML5/JavaScript technologies.

The features of **g3viz** include

- Interactive (zoom & pan, tooltip, brush selection tool, and interactive legend)
- Highlight and label positional mutations
- 8 ready-to-use chart themes
- Highly customizable with over 50 chart options and over 35 color schemes
- Save charts in PNG or high-quality SVG format
- Built-in function to retrieve protein domain information and resolve gene isoforms
- Built-in function to map genetic mutation type (a.k.a, variant classification) to mutation class

## 2 Install g3viz

Install from R repository

```
# install package
install.packages("g3viz", repos = "http://cran.us.r-project.org")
```

or install development version from github

```
# Check if "devtools" installed
if("devtools" %in% rownames(installed.packages()) == FALSE){
  install.packages("devtools")
}

# install from github
devtools::install_github("g3viz/g3viz")
```

## 3 Quick Start

### 3.1 Example 1: Visualize genetic mutation data from MAF file

Mutation Annotation Format (MAF) is a commonly-used tab-delimited text file for storing aggregated mutation information. It could be generated from VCF file using tools like `vcf2maf`. Translational effect of variant alleles in MAF files are usually in the column named `Variant_Classification` or `Mutation_Type` (*i.e.*, `Frame_Shift_Del`, `Split_Site`). In this example, the somatic mutation data of the *TCGA-BRCA* study was originally downloaded from the GDC Data Portal.

```
# System file
maf.file <- system.file("extdata", "TCGA.BRCA.varscan.somatic.maf.gz", package = "g3viz")
```

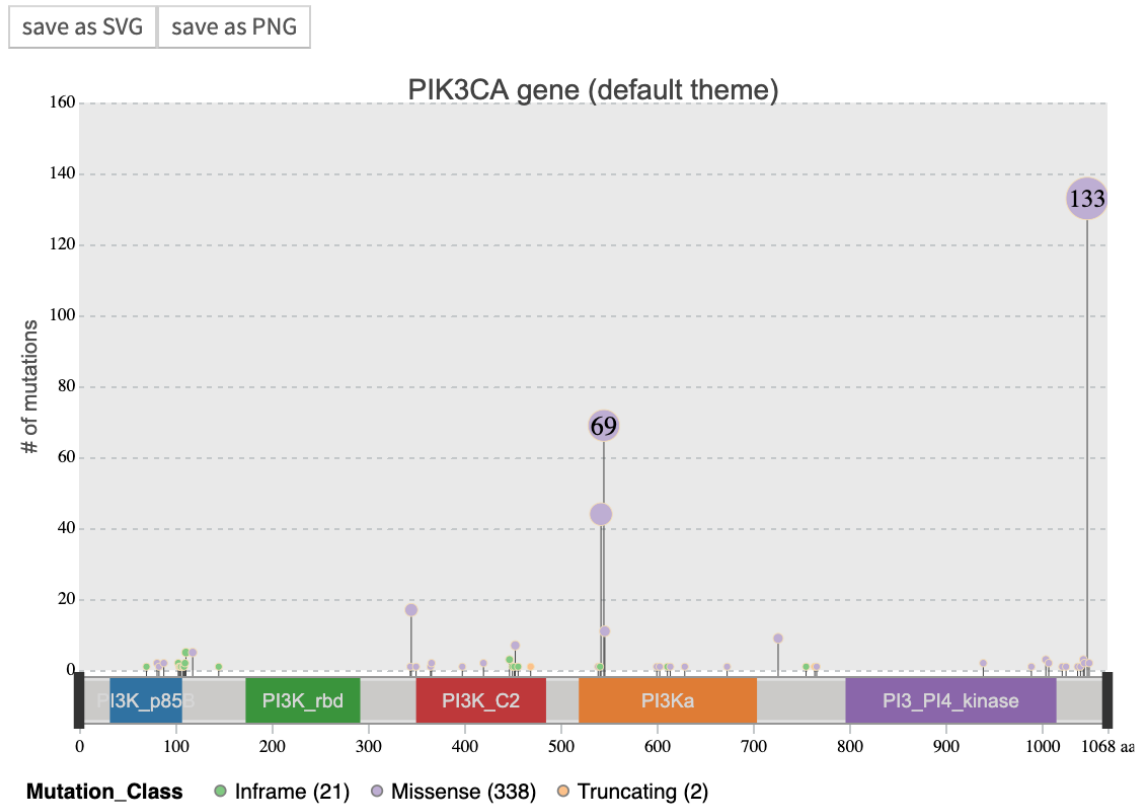


Figure 1: g3viz lollipop-diagram with default theme

```
# =====
# Read in MAF file
# In addition to read data in, g3viz::readMAF function does
# 1. parse "Mutation_Class" information from the "Variant_Classification"
#    column (also named "Mutation_Type" in some files)
# 2. parse "AA_position" (amino-acid position) from the "HGVS_Short" column
#    (also named "amino_acid_change" in some files) (e.g., p.Q136P)
# =====
mutation.dat <- readMAF(maf.file)

# =====
# Chart 1
# "default" chart theme
# =====
chart.options <- g3Lollipop.theme(theme.name = "default",
                                  title.text = "PIK3CA gene (default theme)")

g3Lollipop(mutation.dat,
            gene.symbol = "PIK3CA",
            plot.options = chart.options,
            output.filename = "default_theme")
#> Factor is set to Mutation_Class
#> legend title is set to Mutation_Class
```

## 3.2 Example 2: visualize genetic mutation data from CSV or TSV file

In this example, we read genetic mutation data from CSV or TSV files, and visualize it using some custom chart options. Note this is equivalent to *dark* chart theme.

```
# load data
mutation.csv <- system.file("extdata", "ccle.csv", package = "g3viz")

# =====
# read in data
# "gene.symbol.col" : column of gene symbol
# "variant.class.col" : column of variant class
# "protein.change.col" : column of protein change column
# =====
mutation.dat <- readMAF(mutation.csv,
                        gene.symbol.col = "Hugo_Symbol",
                        variant.class.col = "Variant_Classification",
                        protein.change.col = "amino_acid_change",
                        sep = ",") # column-separator of csv file

# set up chart options
plot.options <- g3Lollipop.options(
  # Chart settings
  chart.width = 600,
  chart.type = "pie",
  chart.margin = list(left = 30, right = 20, top = 20, bottom = 30),
  chart.background = "#d3d3d3",
  transition.time = 300,
  # Lollipop track settings
  lollipop.track.height = 200,
  lollipop.track.background = "#d3d3d3",
  lollipop.pop.min.size = 1,
  lollipop.pop.max.size = 8,
  lollipop.pop.info.limit = 5.5,
  lollipop.pop.info.dy = "0.24em",
  lollipop.pop.info.color = "white",
  lollipop.line.color = "#a9a9a9",
  lollipop.line.width = 3,
  lollipop.circle.color = "#ffdead",
  lollipop.circle.width = 0.4,
  lollipop.label.ratio = 2,
  lollipop.label.min.font.size = 12,
  lollipop.color.scheme = "dark2",
  highlight.text.angle = 60,
  # Domain annotation track settings
  anno.height = 16,
  anno.margin = list(top = 0, bottom = 0),
  anno.background = "#d3d3d3",
  anno.bar.fill = "#a9a9a9",
  anno.bar.margin = list(top = 4, bottom = 4),
  domain.color.scheme = "pie5",
  domain.margin = list(top = 2, bottom = 2),
  domain.text.color = "white",
  domain.text.font = "italic 8px Serif",
```

```

# Y-axis label
y.axis.label = "# of TP53 gene mutations",
axis.label.color = "#303030",
axis.label.alignment = "end",
axis.label.font = "italic 12px Serif",
axis.label.dy = "-1.5em",
y.axis.line.color = "#303030",
y.axis.line.width = 0.5,
y.axis.line.style = "line",
y.max.range.ratio = 1.1,
# Chart title settings
title.color = "#303030",
title.text = "TP53 gene (customized chart options)",
title.font = "bold 12px monospace",
title.alignment = "start",
# Chart legend settings
legend = TRUE,
legend.margin = list(left=20, right = 0, top = 10, bottom = 5),
legend.interactive = TRUE,
legend.title = "Variant classification",
# Brush selection tool
brush = TRUE,
brush.selection.background = "#F8F8FF",
brush.selection.opacity = 0.3,
brush.border.color = "#a9a9a9",
brush.border.width = 1,
brush.handler.color = "#303030",
# tooltip and zoom
tooltip = TRUE,
zoom = TRUE
)

g3Lollipop(mutation.dat,
  gene.symbol = "TP53",
  protein.change.col = "amino_acid_change",
  btn.style = "blue", # blue-style chart download buttons
  plot.options = plot.options,
  output.filename = "customized_plot")
#> Factor is set to Mutation_Class

```

### 3.3 Example 3: visualize genetic mutation data from cBioPortal

cBioPortal provides download for many cancer genomics data sets. g3viz has a convenient way to retrieve data directly from this portal.

In this example, we first retrieve genetic mutation data of TP53 gene for the msk\_impact\_2017 study, and then visualize the data using the built-in cbiportal theme, to mimic cBioPortal's mutation\_mapper.

```

# Retrieve mutation data of "msk_impact_2017" from cBioPortal
mutation.dat <- getMutationsFromCbiportal("msk_impact_2017", "TP53")

# "cbiportal" chart theme
plot.options <- g3Lollipop.theme(theme.name = "cbiportal",

```

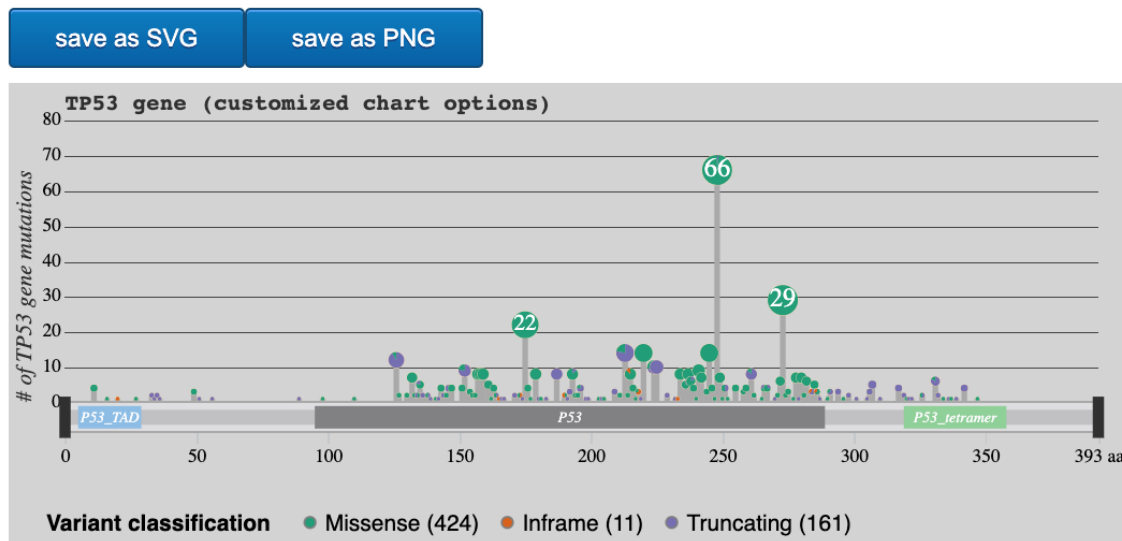


Figure 2: g3viz lollipop-diagram with custom chart options (equivalent to dark theme)

```

                                title.text = "TP53 gene (cbioportal theme)",
                                y.axis.label = "# of TP53 Mutations")

g3Lollipop(mutation.dat,
  gene.symbol = "TP53",
  btn.style = "gray", # gray-style chart download buttons
  plot.options = plot.options,
  output.filename = "cbioportal_theme")
#> Factor is set to Mutation_Class
#> legend title is set to Mutation_Class

```

### 3.3.1 Note:

- Internet access is required to download data from cBioPortal. This may take more than 10 seconds, or sometimes it may fail.
- To check what studies are available on cBioPortal

```

# Connect to CGDS (cancer Genomics Data Server)
cgds <- cgdsr::CGDS("http://www.cbioportal.org/")

# Test if connection is OK
cgdsr::test(cgds)

# To list all studies
all.studies <- cgdsr::getCancerStudies(cgds)

# Pick up a cancer study with mutation data
mutation.dat <- g3viz::getMutationsFromCbioportal("all_stjude_2016", "TP53")

```

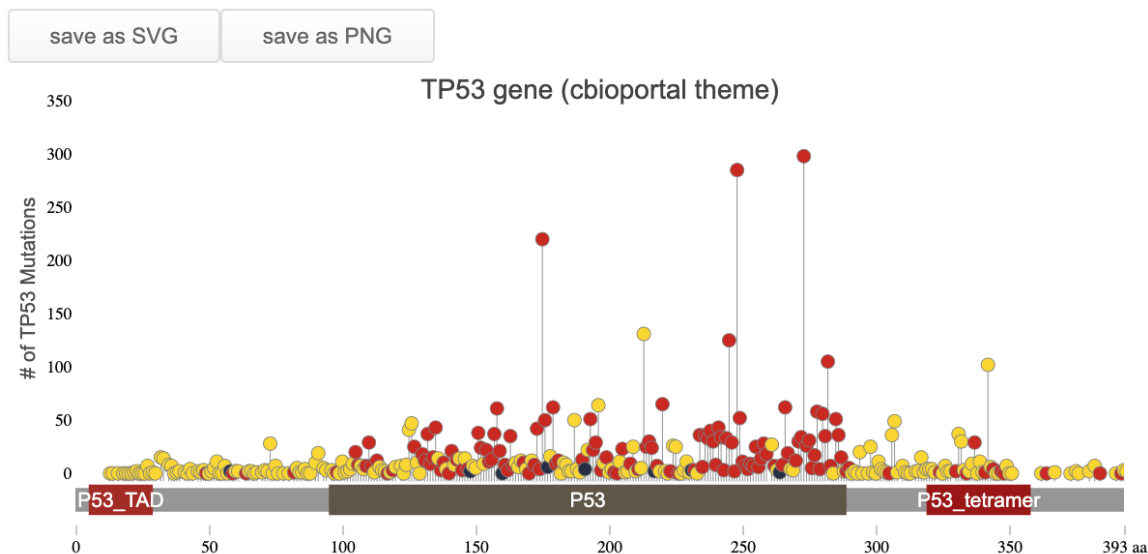


Figure 3: g3viz lollipop-diagram with cbioportal theme

## 4 Usage

### 4.1 Read data

In `g3viz`, annotated mutation data can be loaded in three ways

1. from MAF file, as in Example 1.
2. from CSV or TSV files, as in Example 2.
3. from cBioPortal (internet access required), as in Example 3.

### 4.2 Map mutation type to mutation class

In addition to reading mutation data, `readMAF` or `getMutationFromCbioportal` functions also map mutation type to mutation class and generate a `Mutation_Class` column by default. Mutation type is usually in the column of `Variant_Classification` or `Mutation_Type`. The default mapping table is,

Table 1: Variant classification to mutation class mapping table

Mutation_Type	Mutation_Class	Short_Name
In_Frame_Del	Inframe	IF del
In_Frame_Ins	Inframe	IF ins
Silent	Inframe	Silent
Targeted_Region	Inframe	IF
Missense_Mutation	Missense	Missense
Frame_Shift	Truncating	FS
Frame_Shift_Del	Truncating	FS del
Frame_Shift_Ins	Truncating	FS ins
Nonsense_Mutation	Truncating	Nonsense
Nonstop_Mutation	Truncating	Nonstop

Table 1: Variant classification to mutation class mapping table  
(continued)

Mutation_Type	Mutation_Class	Short_Name
Splice_Region	Truncating	Splice
Splice_Site	Truncating	Splice
3'Flank	Other	3'Flank
3'UTR	Other	3'UTR
5'Flank	Other	5'Flank
5'UTR	Other	5'UTR
De_novo_Start_InFrame	Other	de_novo_start_inframe
De_novo_Start_OutOfFrame	Other	de_novo_start_outofframe
Fusion	Other	Fusion
IGR	Other	IGR
Intron	Other	Intron
lincRNA	Other	lincRNA
RNA	Other	RNA
Start_Codon_Del	Other	Nonstart
Start_Codon_Ins	Other	start_codon_ins
Start_Codon_SNP	Other	Nonstart
Translation_Start_Site	Other	TSS
Unknown	Other	Unknown

### 4.3 Retrieve Pfam domain information

Given a HUGO gene symbol, users can either use `hgnc2pfam` function to retrieve Pfam protein domain information first or use all-in-one `g3Lollipop` function to directly create lollipop-diagram. In case that the given gene has multiple isoforms, `hgnc2pfam` returns all UniProt entries, and users can specify one using the corresponding UniProt entry. If attribute `guess` is `TRUE`, the Pfam domain information of the longest UniProt entry is returned.

```
# Example 1: TP53 has single UniProt entry
hgnc2pfam("TP53", output.format = "list")
#> $symbol
#> [1] "TP53"
#>
#> $uniprot
#> [1] "P04637"
#>
#> $length
#> [1] 393
#>
#> $pfam
#>      hmm.acc      hmm.name start end  type
#> 37815 PF08563      P53_TAD    5  29 Motif
#> 37816 PF00870        P53     95 289 Domain
#> 37817 PF07710 P53_tetramer 319 358 Motif

# Example 2: GNAS has multiple UniProt entries
# `guess = TRUE`: the Pfam domain information of the longest
#                  UniProt protein is returned
```

```

hgnc2pfam("GNAS", guess = TRUE)
#> GNAS maps to multiple UniProt entries:
#> symbol uniprot length
#> GNAS 095467 245
#> GNAS P63092 394
#> GNAS P84996 626
#> GNAS Q5JWF2 1037
#> Warning in hgnc2pfam("GNAS", guess = TRUE): Pick: Q5JWF2
#> {"symbol": "GNAS", "uniprot": "Q5JWF2", "length": 1037, "pfam": [{"hmm.acc": "PF00503", "hmm.name": "G-alpha",

```

## 4.4 Chart themes

The `g3viz` package contains 8 ready-to-use chart schemes: *default*, *blue*, *simple*, *cbiportal*, *nature*, *nature2*, *ggplot2*, and *dark*. Check online live demo or online pdf for usage and examples.

## 4.5 Color schemes

Figure 4 demonstrates all color schemes that `g3viz` supports for lollipop-pops and Pfam domains. More demos are available at demo 1, demo 2, and demo 3.

## 4.6 Chart options

Chart options can be specified using `g3Lollipop.options()` function (see example 2). Here is the full list of chart options,

Table 2: Chart options of `g3viz`

Option	Description
<code>chart.width</code>	chart width in px. Default '800'.
<code>chart.type</code>	pop type, 'pie' or 'circle'. Default 'pie'.
<code>chart.margin</code>	specify chart margin in <code>_list_</code> format. Default 'list(left = 40, right = 20, top = 15, bottom = 25)'.
<code>chart.background</code>	chart background. Default 'transparent'.
<code>transition.time</code>	chart animation transition time in millisecond. Default '600'.
<code>lollipop.track.height</code>	height of lollipop track. Default '420'.
<code>lollipop.track.background</code>	background of lollipop track. Default 'rgb(244,244,244)'.
<code>lollipop.pop.min.size</code>	lollipop pop minimal size in px. Default '2'.
<code>lollipop.pop.max.size</code>	lollipop pop maximal size in px. Default '12'.
<code>lollipop.pop.info.limit</code>	threshold of lollipop pop size to show count information in middle of pop. Default '8'.
<code>lollipop.pop.info.color</code>	lollipop pop information text color. Default '#EEE'.
<code>lollipop.pop.info.dy</code>	y-axis direction text adjustment of lollipop pop information. Default '-0.35em'.
<code>lollipop.line.color</code>	lollipop line color. Default 'rgb(42,42,42)'.
<code>lollipop.line.width</code>	lollipop line width. Default '0.5'.
<code>lollipop.circle.color</code>	lollipop circle border color. Default 'wheat'.



Table 2: Chart options of g3viz (*continued*)

Option	Description
lollipop.circle.width	lollipop circle border width. Default '0.5'.
lollipop.label.ratio	lollipop click-out label font size to circle size ratio. Default '1.4'.
lollipop.label.min.font.size	lollipop click-out label minimal font size. Default '10'.
lollipop.color.scheme	color scheme to fill lollipop pops. Default 'accent'. Check [color schemes](#schemes) for details.
highlight.text.angle	the rotation angle of on-click highlight text in degree. Default '90'.
anno.height	height of protein structure annotation track. Default '30'.
anno.margin	margin of protein structure annotation track. Default 'list(top = 4, bottom = 0)'.
anno.background	background of protein structure annotation track. Default 'transparent'.
anno.bar.fill	background of protein bar in protein structure annotation track. Default '#E5E3E1'.
anno.bar.margin	margin of protein bar in protein structure annotation track. Default 'list(top = 2, bottom = 2)'.
domain.color.scheme	color scheme of protein domains. Default 'category10'. Check [color schemes](#schemes) for details.
domain.margin	margin of protein domains. Default 'list(top = 0, bottom = 0)'.
domain.text.font	domain label text font in shorthand format. Default 'normal 11px Arial'.
domain.text.color	domain label text color. Default '#F2F2F2'.
y.axis.label	Y-axis label text. Default '# of mutations'.
axis.label.font	css font style shorthand (font-style font-variant font-weight font-size/line-height font-family). Default 'normal 12px Arial'.
axis.label.color	axis label text color. Default '#4f4f4f'.
axis.label.alignment	axis label text alignment (start/end/middle). Default 'middle'.
axis.label.dy	text adjustment of axis label text. Default '-2em'.
y.axis.line.color	color of y-axis in-chart lines (ticks). Default '#c4c8ca'.
y.axis.line.style	style of y-axis in-chart lines (ticks), 'dash' or 'line'. Default 'dash'.
y.axis.line.width	width of y-axis in-chart lines (ticks). Default '1'.
y.max.range.ratio	ratio of y-axis range to data value range. Default '1.1'.
title.text	title of chart. Default "".
title.font	font of chart title. Default 'normal 16px Arial'.
title.color	color of chart title. Default '#424242'.
title.alignment	text alignment of chart title (start/middle/end). Default 'middle'.
title.dy	text adjustment of chart title. Default '0.35em'.
legend	if show legend. Default 'TRUE'.

Table 2: Chart options of g3viz (*continued*)

Option	Description
legend.margin	legend margin in <code>_list_</code> format. Default <code>'list(left = 10, right = 0, top = 5, bottom = 5)'</code> .
legend.interactive	legend interactive mode. Default <code>'TRUE'</code> .
legend.title	legend title. If <code>'NA'</code> , use factor name as <code>'factor.col'</code> . Default is <code>'NA'</code> .
brush	if show brush. Default <code>'TRUE'</code> .
brush.selection.background	background color of selection brush. Default <code>'#666'</code> .
brush.selection.opacity	background opacity of selection brush. Default <code>'0.2'</code> .
brush.border.color	border color of selection brush. Default <code>'#969696'</code> .
brush.handler.color	color of left and right handlers of selection brush. Default <code>'#333'</code> .
brush.border.width	border width of selection brush. Default <code>'1'</code> .
tooltip	if show tooltip. Default <code>'TRUE'</code> .
zoom	if enable zoom feature. Default <code>'TRUE'</code> .

## 4.7 Save chart as HTML

`g3Lollipop` also renders two buttons over the lollipop-diagram, allowing to save the resulting chart in PNG or vector-based SVG file. To save chart programmatically as HTML, you can use `htmlwidgets::saveWidget` function.

```
chart <- g3Lollipop(mutation.dat,
  gene.symbol = "TP53",
  protein.change.col = "amino_acid_change",
  plot.options = plot.options)
htmlwidgets::saveWidget(chart, "g3lollipop_chart.html")
```

## 5 Session Info

```
sessionInfo()
#> R version 3.5.3 (2019-03-11)
#> Platform: x86_64-apple-darwin15.6.0 (64-bit)
#> Running under: macOS Mojave 10.14.5
#>
#> Matrix products: default
#> BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] stats      graphics  grDevices  utils      datasets  methods   base
#>
#> other attached packages:
#> [1] kableExtra_1.1.0 knitr_1.23      g3viz_1.1.2
#>
```

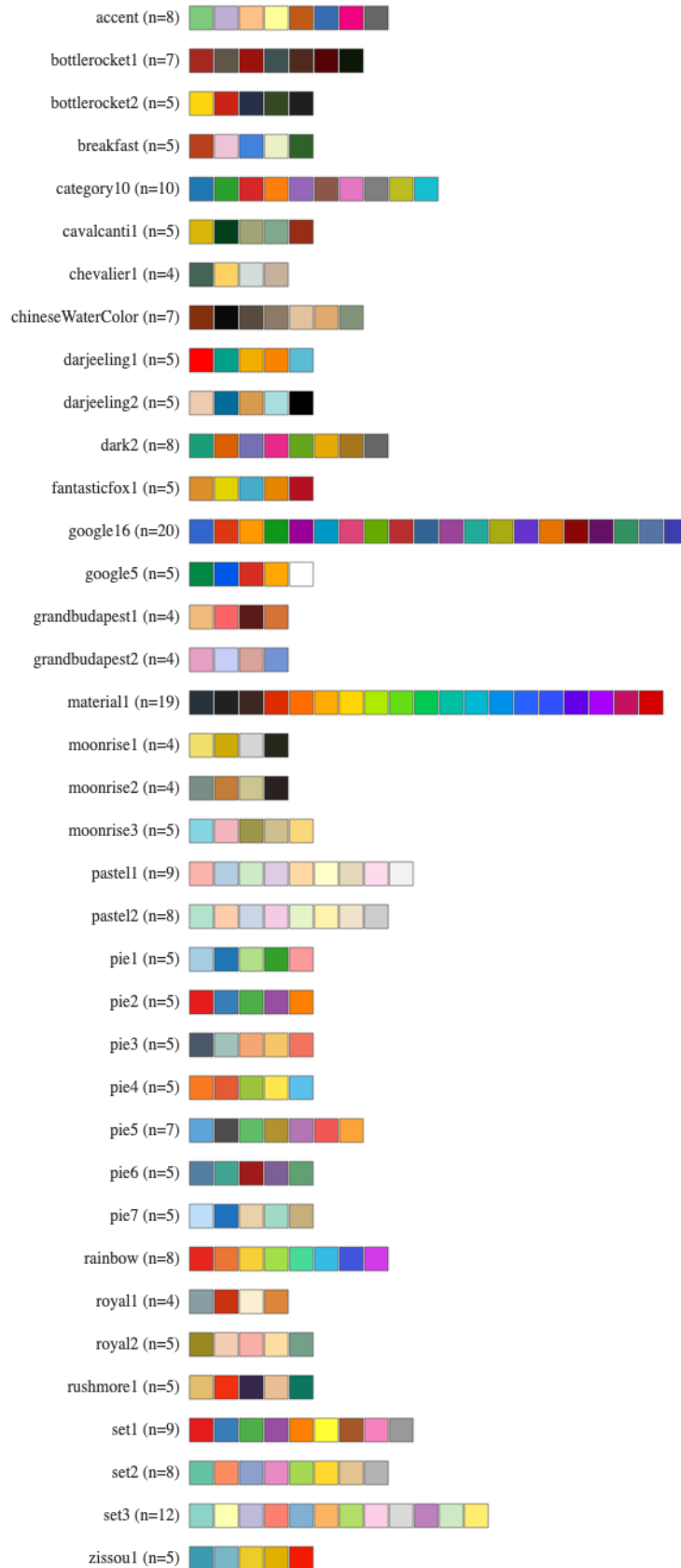


Figure 4: List of color schemes supported by g3viz

```

#> loaded via a namespace (and not attached):
#> [1] Rcpp_1.0.1      compiler_3.5.3  pillar_1.4.2
#> [4] R.methodsS3_1.7.1 tools_3.5.3     zeallot_0.1.0
#> [7] digest_0.6.20   jsonlite_1.6    evaluate_0.14
#> [10] tibble_2.1.3    viridisLite_0.3.0 pkgconfig_2.0.2
#> [13] rlang_0.4.0     rstudioapi_0.10 yaml_2.2.0
#> [16] xfun_0.8        cgdscr_1.3.0    stringr_1.4.0
#> [19] httr_1.4.0      xml2_1.2.0      vctrs_0.2.0
#> [22] htmlwidgets_1.3 hms_0.5.0       webshot_0.5.1
#> [25] glue_1.3.1      R6_2.4.0        rmarkdown_1.14
#> [28] readr_1.3.1     magrittr_1.5     backports_1.1.4
#> [31] scales_1.0.0    htmltools_0.3.6 rvest_0.3.4
#> [34] colorspace_1.4-1 stringi_1.4.3    munsell_0.5.0
#> [37] crayon_1.3.4    R.oo_1.22.0

```