

# Package ‘g3viz’

October 13, 2022

**Type** Package

**Title** Interactively Visualize Genetic Mutation Data using a Lollipop-Diagram

**Version** 1.1.5

**Maintainer** Xin Guo <g3viz.group@gmail.com>

**Description** Interface for 'g3-lollipop' JavaScript library.  
Visualize genetic mutation data using an interactive lollipop diagram in Studio or your browser.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0)

**biocViews**

**Imports** jsonlite, stringr, cBioPortalData, htmlwidgets

**Suggests** shiny (>= 1.0.0), knitr, rmarkdown, kableExtra

**URL** <https://github.com/G3viz/g3viz>

**BugReports** <https://github.com/G3viz/g3viz/issues>

**RoxygenNote** 7.2.0

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Xin Guo [aut, cre],  
Rener Zhang [ctb]

**Repository** CRAN

**Date/Publication** 2022-08-19 13:20:02 UTC

## R topics documented:

g3Lollipop . . . . .	2
g3Lollipop-shiny . . . . .	4
g3Lollipop.options . . . . .	4
g3Lollipop.theme . . . . .	8
getMutationsFromCbioportal . . . . .	9
guessMAFColumnName . . . . .	10
hgnc2pfam . . . . .	11
hgnc2pfam.df . . . . .	12
hgnc2uniprot . . . . .	12
mapMutationTypeToMutationClass . . . . .	13
mutation.table.df . . . . .	15
parseProteinChange . . . . .	15
readMAF . . . . .	16
uniprot2pfam . . . . .	17
<b>Index</b>	<b>18</b>

---

g3Lollipop

*Render g3lollipop diagram for the given mutation data*

---

### Description

Render g3lollipop diagram for the given mutation data

### Usage

```
g3Lollipop(
  mutation.dat,
  gene.symbol,
  uniprot.id = NA,
  gene.symbol.col = "Hugo_Symbol",
  aa.pos.col = "AA_Position",
  protein.change.col = c("Protein_Change", "HGVS_Short"),
  factor.col = "Mutation_Class",
  plot.options = g3Lollipop.options(),
  save.png.btn = TRUE,
  save.svg.btn = TRUE,
  btn.style = NA,
  output.filename = "output"
)
```

### Arguments

mutation.dat	Input genomic mutation data frame
gene.symbol	HGNC primary gene symbol

uniprot.id	UniProt ID, in case that the specified gene symbol links to multiple UniProt entries (isoforms). For example, <i>AKAP7</i> gene has two isoforms in <a href="#">UniProt</a> , <a href="#">O43687</a> and <a href="#">Q9P0M2</a> .
gene.symbol.col	Column name of Hugo gene symbols (e.g., TP53). Default <i>Hugo_Symbol</i> .
aa.pos.col	Column name of the parsed amino-acid change position. Default <i>AA_Position</i> .
protein.change.col	Column name of protein change information (e.g., p.K960R, G658S, L14Sfs*15). Default is a list of <i>Protein_Change</i> , <i>HGVSp_Short</i> .
factor.col	column of classes in the plot legend. IF NA, use parsed <i>Mutation_Class</i> column, otherwise, use specified. Default NA.
plot.options	g3lollipop diagram options in list format. Check <a href="#">g3Lollipop.options</a>
save.png.btn	If add <i>save-as-png</i> button to the diagram. Default TRUE.
save.svg.btn	If add <i>save-as-svg</i> button to the diagram. Default TRUE.
btn.style	button style, including browser default button style, and two built-in styles, <i>blue</i> or <i>gray</i> . Default NA, indicating browser default.
output.filename	Specify output file name.

## Value

lollipop diagram for the given mutation data. The chart is interactive within either Shiny applications or Rmd documents under the bindings.

## Examples

```
# system mutation data
maf.file <- system.file("extdata", "TCGA.BRCA.varscan.somatic.maf.gz", package = "g3viz")
# read in MAF file
mutation.dat <- readMAF(maf.file)

# use built-in chart theme
chart.options <- g3Lollipop.theme(theme.name = "default",
                                  title.text = "PIK3CA gene (default theme)")

# generate chart
g3Lollipop(mutation.dat,
           gene.symbol = "PIK3CA",
           plot.options = chart.options,
           btn.style = "blue",
           output.filename = "default_theme")
```

---

g3Lollipop-shiny      *Shiny bindings for g3Lollipop*

---

### Description

Output and render functions for using g3viz lollipop diagram within Shiny applications and interactive Rmd documents.

### Usage

```
g3LollipopOutput(outputId, width = "100%", height = "520px")
renderG3Lollipop(expr, env = parent.frame(), quoted = FALSE)
```

### Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a g3-lollipop
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

### Value

No value returned. It is the binding which enables interactive functions within Shiny applications and Rmd documents.

---

g3Lollipop.options      *G3Lollipop plot options*

---

### Description

G3Lollipop plot options

### Usage

```
g3Lollipop.options(
  chart.width = 800,
  chart.type = "circle",
  chart.margin = list(left = 40, right = 20, top = 15, bottom = 25),
  chart.background = "transparent",
  transition.time = 600,
```

```
y.axis.label = "# of mutations",
axis.label.font = "normal 12px Arial",
axis.label.color = "#4f4f4f",
axis.label.alignment = "middle",
axis.label.dy = "-2em",
y.axis.line.color = "#c4c8ca",
y.axis.line.style = "dash",
y.axis.line.width = 1,
y.max.range.ratio = 1.1,
legend.margin = list(left = 10, right = 0, top = 5, bottom = 5),
legend.interactive = TRUE,
legend.title = NA,
lollipop.track.height = 420,
lollipop.track.background = "rgb(233,233,233)",
lollipop.pop.min.size = 2,
lollipop.pop.max.size = 12,
lollipop.pop.info.limit = 8,
lollipop.pop.info.color = "#EEE",
lollipop.pop.info.dy = "0.35em",
lollipop.line.color = "rgb(42,42,42)",
lollipop.line.width = 0.5,
lollipop.circle.color = "wheat",
lollipop.circle.width = 0.5,
lollipop.label.ratio = 1.4,
lollipop.label.min.font.size = 10,
lollipop.color.scheme = "accent",
highlight.text.angle = "90",
title.text = "",
title.font = "normal 16px Arial",
title.color = "#424242",
title.alignment = "middle",
title.dy = "0.35em",
anno.height = 30,
anno.margin = list(top = 4, bottom = 0),
anno.background = "transparent",
anno.bar.fill = "#e5e3e1",
anno.bar.margin = list(top = 2, bottom = 2),
domain.color.scheme = "category10",
domain.margin = list(top = 0, bottom = 0),
domain.text.font = "normal 11px Arial",
domain.text.color = "#f2f2f2",
brush = TRUE,
brush.selection.background = "#666",
brush.selection.opacity = 0.2,
brush.border.color = "#969696",
brush.handler.color = "#333",
brush.border.width = 1,
legend = TRUE,
```

```

    tooltip = TRUE,
    zoom = TRUE
  )

```

### Arguments

`chart.width` chart width. Default 800.

`chart.type` *pie* or *circle*. Default *circle*.

`chart.margin` specify chart margin in `_list_` format. Default `list(left = 40, right = 20, top = 15, bottom = 25)`.

`chart.background` chart background. Default *transparent*.

`transition.time` animation transition time when clicking lollipop pops to show labels (in millisecond). Default 600.

`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.

`legend.margin` legend margin in *list*. Default `list(left = 10, right = 0, top = 5, bottom = 5)`.

`legend.interactive` legend interactive mode. Default TRUE.

`legend.title` legend title. If NA, *factor.col* in `g3Lollipop` is used. Default is NA.

`lollipop.track.height` height of lollipop track. Default 420.

`lollipop.track.background` background of lollipop track. Default *rgb(244,244,244)*

`lollipop.pop.min.size` lollipop pop minimal size. Default 2.

`lollipop.pop.max.size` lollipop pop maximal size. Default 12.

`lollipop.pop.info.limit`  
threshold of lollipop pop size to show count information in middle of pop. Default 8.

`lollipop.pop.info.color`  
lollipop pop information text color. Default *#EEE*.

`lollipop.pop.info.dy`  
y-axis direction text adjustment of lollipop pop information. Default *-0.35em*.

`lollipop.line.color`  
lollipop line color. Default *rgb(42,42,42)*.

`lollipop.line.width`  
lollipop line width. Default 0.5.

`lollipop.circle.color`  
lollipop circle border color. Default *wheat*.

`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*.

`highlight.text.angle`  
pop-on-click highlight text angle. Default 90.

`title.text`  
title of chart. Default is empty.

`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*.

`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*.

`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.
brush	if show brush. Default TRUE.
brush.selection.background	background color of selection brush. Default #666.
brush.selection.opacity	background opacity of selection brush. Default 0.2.
brush.border.color	border color of selection brush. Default #969696.
brush.handler.color	color of left and right handlers of selection brush. Default #333.
brush.border.width	border width of selection brush. Default 1.
legend	if show legend. Default TRUE.
tooltip	if show tooltip. Default TRUE.
zoom	if enable zoom feature. Default TRUE.

**Value**

a list with g3Lollipop plot options

---

g3Lollipop.theme	<i>G3Lollipop chart options of built-in themes.</i>
------------------	---

---

**Description**

G3Lollipop chart options of built-in themes.

**Usage**

```
g3Lollipop.theme(
  theme.name = "default",
  title.text = "",
  y.axis.label = "# of mutations",
  legend.title = NA
)
```

**Arguments**

theme.name	theme name, including <i>default</i> , <i>cbiportal</i> , <i>nature</i> , <i>nature2</i> , <i>dark</i> , <i>blue</i> , <i>ggplot2</i> , and <i>simple</i> . Default <i>default</i> .
title.text	title of chart. Default is empty.
y.axis.label	Y-axis label text. Default " <i># of mutations</i> ".
legend.title	legend title. If NA, <i>factor.col</i> in <a href="#">g3Lollipop</a> is used. Default is NA.

**Value**

a list with g3Lollipop plot options



---

`getMutationsFromCbioportal`*Query cancer genomic mutation data from cBioPortal*

---

## Description

Retrieve and parse mutation data from cBioPortal by the given cBioPortal cancer study ID and the gene symbol.

## Usage

```
getMutationsFromCbioportal(  
  study.id,  
  gene.symbol,  
  output.file = NA,  
  mutation.type.to.class.df = NA  
)
```

## Arguments

`study.id`            cbioprotal study ID  
`gene.symbol`        HGNC gene symbol.  
`output.file`        if specified, output to a file in CSV format. Default is NA.  
`mutation.type.to.class.df`  
                     mapping table from mutation type to class. See [mapMutationTypeToMutationClass](#)  
                     for details. Default NA, which indicates to use default mappings.

## Value

a data frame with columns

**Hugo\_Symbol** Hugo gene symbol

**Protein\_Change** Protein change information (cBioportal uses *HGVS<sub>p</sub>* format)

**Sample\_ID** Sample ID

**Mutation\_Type** mutation type, aka, variant classification.

**Chromosome** chromosome

**Start\_Position** start position

**End\_Position** end position

**Reference\_Allele** reference allele

**Variant\_Allele** variant allele

**Mutation\_Class** mutation class (e.g., Truncating/Missense/Inframe/Other)

**AA\_Position** amino-acid position of the variant; if the variant is not in protein-coding region, NA

## Examples

```
## Not run:
# Usage:
# cBioPortalData has officially replaced the defunct cgdsr.
# Search online for cgdsrMigration.html if interested.
library(cBioPortalData)
cbio <- cBioPortal()

# list all studies of cBioPortal
all.studies <- getStudies(cbio, buildReport = FALSE)

# First, select a cancer study that contains mutation data set ("cancer_study_id")
# then, query genomic mutation data using a HGNC gene symbol,
# for example
mutation.dat <- getMutationsFromCbioportal("msk_impact_2017", "TP53")
mutation.dat <- getMutationsFromCbioportal("all_stjude_2016", "TP53")

## End(Not run)
```

---

guessMAFColumnName      *Guess column name for MAF file*

---

## Description

Guess column name for MAF file

## Usage

```
guessMAFColumnName(maf.df, alt.column.names)
```

## Arguments

maf.df                    MAF data frame  
alt.column.names        a vector of alternative column names

## Value

if hit one alternative column name, return the name; otherwise, return NA

---

`hgnc2pfam`*Map from Hugo symbol to Pfam domains*

---

### Description

Mapping from Hugo symbol to Pfam-A domain composition. If the given Hugo symbol has multiple UniProt ID mappings, and `guess == TRUE`, the longest UniProt protein is selected. Return is either a list or a JSON.

### Usage

```
hgnc2pfam(hgnc.symbol, guess = TRUE, uniprot.id = NA, output.format = "json")
```

### Arguments

<code>hgnc.symbol</code>	primary Hugo symbol
<code>guess</code>	if the given Hugo symbol links to multiple UniProt IDs, choose the longest one ( <code>guess == TRUE</code> ); otherwise NA ( <code>guess == FALSE</code> ). Default <code>TRUE</code> .
<code>uniprot.id</code>	UniProt ID, in case that gene symbol maps to multiple UniProt entries.
<code>output.format</code>	output format: JSON or list

### Value

A list or a JSON with attributes: *symbol*, *uniprot*, *length*, and a list of *Pfam* entries, including *hmm.acc*, *hmm.name*, *start*, *end*, and *type*.

### Examples

```
# general usage
hgnc2pfam("TP53")
hgnc2pfam("TP53", output.format = "json")
hgnc2pfam("TP53", output.format = "list")
hgnc2pfam("TP53", output.format = "json", uniprot.id = "P04637") # OK

# for gene mapping to multiple UniProt entries
hgnc2pfam("GNAS", guess = TRUE)
hgnc2pfam("GNAS", guess = FALSE)
hgnc2pfam("GNAS", output.format = "list")
hgnc2pfam("GNAS", output.format = "list", uniprot.id = "P84996")
## Not run:
hgnc2pfam("GNAS", output.format = "list", uniprot.id = "P84997") # not exists, returns FALSE

## End(Not run)

hgnc2pfam("PRAMEF9", output.format = "list") # no Pfam mappings
```

---

hgnc2pfam.df	<i>Mapping table between gene.symbol, uniprot.id, and pfam</i>
--------------	--

---

**Description**

A dataset containing the mapping table between Hugo symbol, UniProt ID, and Pfam ACC.

**Usage**

hgnc2pfam.df

**Format**

A data frame with columns:

**symbol** Gene symbol

**uniprot** UniProt ID

**length** protein length

**start** starting position of Pfam domain

**end** ending position of Pfam domain

**hmm.acc** Pfam accession number

**hmm.name** Pfam name

**type** Pfam type, i.e., domain/family/motif/repeat/disordered/coiled-coil

**Source**

Pfam (v31.0) and UniProt

**Examples**

hgnc2pfam.df

---

hgnc2uniprot	<i>Mapping from Hugo symbol to UniProt IDs</i>
--------------	--

---

**Description**

Mapping from Hugo Symbol to UniProt ID using internal mapping table. Return a data frame with columns *symbol* (Hugo symbol), *uniprot* (UniProt ID), and *length* (protein length).

**Usage**

hgnc2uniprot(hgnc.symbol)

**Arguments**

hgnc.symbol     primary HUGO symbol

**Value**

a data frame with columns *symbol* (Hugo symbol), *uniprot* (UniProt ID), and *length* (protein length).

**Examples**

```
# maps to single UniProt entry
hgnc2uniprot("TP53")

# maps to multiple UniProt entries
hgnc2uniprot("GNAS")
hgnc2uniprot("AKAP7")
```

---

mapMutationTypeToMutationClass

*Map from mutation type (aka, variant classification) to mutation class*

---

**Description**

Map from mutation type (aka, variant classification) to mutation class. Default mappings are as follows,

- Missense
  - *Missense\_Mutation* — a point mutation in which a single nucleotide change results in a codon that codes for a different amino acid See [https://en.wikipedia.org/wiki/Missense\\_mutation](https://en.wikipedia.org/wiki/Missense_mutation).
- Inframe
  - *In\_Frame\_Del* — a deletion that keeps the sequence in frame
  - *In\_Frame\_Ins* — an insertion that keeps the sequence in frame
  - *Silent* — variant is in coding region of the chosen transcript, but protein structure is identical (i.e., a synonymous mutation)
  - *Targeted\_Region* — targeted region
- Truncating
  - *Frame\_Shift* — a variant caused by indels of a number of nucleotides in a DNA sequence that is not divisible by three. See [https://en.wikipedia.org/wiki/Frameshift\\_mutation](https://en.wikipedia.org/wiki/Frameshift_mutation).
  - *Frame\_Shift\_Ins* — a variant caused by insertion that moves the coding sequence out of frame. See [https://en.wikipedia.org/wiki/Frameshift\\_mutation](https://en.wikipedia.org/wiki/Frameshift_mutation).
  - *Frame\_Shift\_Del* — a variant caused by deletion that moves the coding sequence out of frame. See [https://en.wikipedia.org/wiki/Frameshift\\_mutation](https://en.wikipedia.org/wiki/Frameshift_mutation).

- *Nonsense\_Mutation* — a premature stop codon that is created by the variant. See [https://en.wikipedia.org/wiki/Nonsense\\_mutation](https://en.wikipedia.org/wiki/Nonsense_mutation).
- *Nonstop\_Mutation* — a variant that removes stop codon.
- *Splice\_Site* — a variant that is within two bases of a splice site.
- *Splice\_Region* — a variant that is within splice region.
- Other
  - *5'UTR* — a variant that is on the 5'UTR for the chosen transcript.
  - *3'UTR* — a variant that is on the 3'UTR for the chosen transcript.
  - *5'Flank* — a variant that is upstream of the chosen transcript (generally within 3kb).
  - *3'Flank* — a variant that is downstream of the chosen transcript (generally within 3kb).
  - *Fusion* — a gene fusion.
  - *IGR* — an intergenic region. Does not overlap any transcript.
  - *Intron* — a variant that lies between exons within the bounds of the chosen transcript.
  - *Translation\_Start\_Site* — a variant that is in translation start site.
  - *De\_novo\_Start\_InFrame* — a novel start codon that is created by the given variant using the chosen transcript. However, it is in frame relative to the coded protein.
  - *De\_novo\_Start\_OutOfFrame* — a novel start codon that is created by the given variant using the chosen transcript. However, it is out of frame relative to the coded protein.
  - *Start\_Codon\_SNP* — a point mutation that overlaps the start codon.
  - *Start\_Codon\_Ins* — an insertion that overlaps the start codon.
  - *Start\_Codon\_Del* — a deletion that overlaps the start codon.
  - *RNA* — a variant that lies on one of the RNA transcripts.
  - *lincRNA* — a variant that lies on one of the lincRNAs.
  - *Unknown* — Unknown

## Usage

```
mapMutationTypeToMutationClass(
  mutation.type.vec,
  mutation.type.to.class.df = NA
)
```

## Arguments

```
mutation.type.vec
  a vector of mutation type information
mutation.type.to.class.df
  A mapping table from mutation type (header Mutation_Type) to mutation class
  (header Mutation_Class). Default NA, which indicates to use default mappings.
```

## Value

```
a vector of mapped mutation class information
```

---

mutation.table.df	<i>Default mapping table between mutation type (aka, variant classification) to mutation class</i>
-------------------	--

---

### Description

A dataset containing the mapping table between genomic mutation type (aka, variant classification) to mutation class. See [mapMutationTypeToMutationClass](#) for details.

### Usage

```
mutation.table.df
```

### Format

A data frame with three columns:

**Mutation\_Type** Mutation type, aka, variant classification

**Mutation\_Class** mutation class

**Short\_Name** short name of mutation type

### Examples

```
mutation.table.df
```

---

parseProteinChange	<i>Extract amino_acid_position from Protein_Change</i>
--------------------	--

---

### Description

Parse *amino\_acid\_position* according to HGVS<sub>p</sub>\_short format.

For example, *p.Q16Rfs\*28*, amino-acid position is 16. See <http://varnomen.hgvs.org/recommendations/protein/> or <https://www.hgvs.org/mutnomen/recs-prot.html>.

### Usage

```
parseProteinChange(protein.change.vec, mutation.class.vec)
```

### Arguments

protein.change.vec

a vector of strings with protein change information, usually in HGVS<sub>p</sub>\_short format.

mutation.class.vec

a vector of strings with mutation class (or so-called variant classification) information.

**Value**

a vector of parsed amino-acid position

---

readMAF	<i>Read MAF file</i>
---------	----------------------

---

**Description**

Read mutation information from MAF file. For MAF format specification, see [https://docs.gdc.cancer.gov/Data/File\\_Formats/MAF\\_Format/](https://docs.gdc.cancer.gov/Data/File_Formats/MAF_Format/).

**Usage**

```
readMAF(
  maf.file,
  gene.symbol.col = "Hugo_Symbol",
  variant.class.col = c("Variant_Classification", "Mutation_Type"),
  protein.change.col = c("Protein_Change", "HGVS_Short"),
  if.parse.aa.pos = TRUE,
  if.parse.mutation.class = TRUE,
  mutation.class.col = "Mutation_Class",
  aa.pos.col = "AA_Position",
  mutation.type.to.class.df = NA,
  sep = "\t",
  quote = "",
  ...
)
```

**Arguments**

maf.file	MAF file name. Gzipped input file allowed, with ".gz" file extension.
gene.symbol.col	Column name of Hugo gene symbols (e.g., TP53). Default <i>Hugo_Symbol</i> .
variant.class.col	Column name for variant class information (e.g., <i>Missense_Mutation</i> , <i>Nonsense_Mutation</i> ). Default is the first match of <i>Variant_Classification</i> or <i>Mutation_Type</i> .
protein.change.col	Column name for protein change information (e.g., p.K960R, G658S, L14Sfs*15). Default is the first match of <i>Protein_Change</i> or <i>HGVS_Short</i> .
if.parse.aa.pos	if parse amino-acid position of mutations. Default is TRUE.
if.parse.mutation.class	if parse mutation class from mutation type (variant classification) information. Default is TRUE.
mutation.class.col	Column name of the parsed mutation class. Default <i>Mutation_Class</i> .



`aa.pos.col` Column name of the parsed amino-acid change position. Default `AA_Position`.  
`mutation.type.to.class.df` mapping table from mutation type to class. [mapMutationTypeToMutationClass](#) for details. Default `NA`, which indicates to use default mappings.  
`sep` separator of columns. Default `sep = "\t"`.  
`quote` the set of quoting characters. To disable quoting altogether, use `quote = ""`. Default `quote = ""`.  
`...` additional parameters pass to [read.table](#).

**Value**

a data frame containing MAF information, plus optional columns of the parsed *Mutation\_Class* and *Protein\_Position*.

---

<code>uniprot2pfam</code>	<i>From UniProt ID to Pfam-A domain composition</i>
---------------------------	---

---

**Description**

Map from UniProt ID to Pfam-A domain composition.

**Usage**

```
uniprot2pfam(uniprot.id)
```

**Arguments**

`uniprot.id` UniProt ID

**Value**

a data frame with columns

- *uniprot* — UniProt ID
- *length* — protein length
- *hmm.acc* — accession number of Pfam HMM model, e.g., PF08563
- *hmm.name* — Pfam name, e.g., P53\_TAD
- *start* — Pfam domain start position
- *end* — Pfam domain end position
- *type* — Pfam type, including domain/motif/family

**Examples**

```
uniprot2pfam("Q5VWM5") # PRAMEF9; PRAMEF15
uniprot2pfam("P04637")
```

# Index

## \* datasets

hgnc2pfam.df, [12](#)  
mutation.table.df, [15](#)

g3Lollipop, [2](#), [6](#), [8](#)  
g3Lollipop-shiny, [4](#)  
g3Lollipop.options, [3](#), [4](#)  
g3Lollipop.theme, [8](#)  
g3LollipopOutput (g3Lollipop-shiny), [4](#)  
getMutationsFromCbioportal, [9](#)  
guessMAFColumnName, [10](#)

hgnc2pfam, [11](#)  
hgnc2pfam.df, [12](#)  
hgnc2uniprot, [12](#)

mapMutationTypeToMutationClass, [9](#), [13](#),  
[15](#), [17](#)  
mutation.table.df, [15](#)

parseProteinChange, [15](#)

read.table, [17](#)  
readMAF, [16](#)  
renderG3Lollipop (g3Lollipop-shiny), [4](#)  
uniprot2pfam, [17](#)