

# Package ‘maths.genealogy’

July 25, 2025

**Title** Mathematics PhD Genealogy Data and Plotting

**Version** 0.1.4

**Description** Query, extract, and plot genealogical data from The Mathematics Genealogy Project <<https://mathgenealogy.org/>>. Data is gathered from the ‘WebSocket’ server run by the ‘geneagrapher-core’ project <<https://github.com/davidalber/geneagrapher-core>>.

**URL** <https://genealogy.louisaslett.com/>,  
<https://github.com/louisaslett/maths.genealogy>

**BugReports** <https://github.com/louisaslett/maths.genealogy/issues>

**Depends** R (>= 4.2.0)

**Imports** checkmate, cli, curl, httr2, jsonlite, later, rlang, rvest,  
stats, websocket

**Suggests** DiagrammeR, DiagrammeRsvg, ggenealogy, ggplot2, knitr,  
rmarkdown, rsvg, svgPanZoom

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Louis Aslett [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-2211-233X>>)

**Maintainer** Louis Aslett <louis.aslett@durham.ac.uk>

**Repository** CRAN

**Date/Publication** 2025-07-24 23:30:06 UTC

## Contents

disciplines . . . . .	2
get_genealogy . . . . .	3

plot_gg	4
plot_gg_path	6
plot_grviz	7
search_id	8
<b>Index</b>	<b>10</b>

---

disciplines	<i>Mathematical discipline IDs</i>
-------------	------------------------------------

---

### Description

Map mathematical disciplines to IDs for use in searching for mathematicians.

### Usage

```
disciplines(search = NULL)
```

### Arguments

`search` a character(1) string which will search within disciplines. This can be a regular expression search term if desired.

### Value

Data frame, with columns:

`id` the discipline ID, as required by `search_id()` when searching for a mathematician within a specific mathematical discipline;

`discipline` the name of the discipline classification, per the Mathematics Genealogy Project.

### Examples

```
# Lookup the ID of any discipline involving the partial word "stat"
disciplines("stat")

# Use a regular expression to only exactly match the whole word Statistics and nothing else
disciplines("^statistics$")

# Use the above to search only for statisticians with the first name Louis
search_id(given = "Louis", discipline = disciplines("^statistics$")$id)
```

---

get\_genealogy

*Retrieve genealogy tree by mathematician ID*


---

### Description

Queries the genealogy of a single or set of mathematicians by their ID in the [Mathematics Genealogy Project](#).

### Usage

```
get_genealogy(id, ancestors = TRUE, descendants = TRUE)
```

### Arguments

id	integer vector of IDs of mathematicians for whom the genealogy should be retrieved
ancestors	logical indicating whether to include the genealogy backward to include all ancestors, defaults to TRUE. This can be a single <code>logical(1)</code> which then applies to all mathematicians referenced in the <code>id</code> argument, or it can be a vector of the same length as <code>id</code> providing different selection for each individual.
descendants	logical indicating whether to include the genealogy forward to include all descendants, defaults to TRUE. This can be a single <code>logical(1)</code> which then applies to all mathematicians referenced in the <code>id</code> argument, or it can be a vector of the same length as <code>id</code> providing different selection for each individual.

### Value

A list object of class `genealogy`. Each element of the list represents a mathematician in the genealogical tree. The name of the element is the mathematician's ID in the [Mathematics Genealogy Project](#). Each element of the object is list with containing:

`id` `integer(1)` with Mathematician's ID;

`name` `character(1)` containing the full name of the mathematician;

`institution` `character(1)` containing the institution at which PhD was obtained;

`year` `integer(1)` with the year their PhD was completed;

`descendants` `integer` vector of IDs of any mathematicians who were supervised by this individual for their PhD;

`advisors` `integer` vector of IDs of any mathematicians who were supervisors of this individual for their PhD.

In addition, there is an attribute named `start_nodes` which contains an `integer` vector of IDs indicating the origin nodes used in the genealogical tree search that produced this object. In other words, the `id` argument as passed to this function.

## References

- Alber, D. (2024). “‘geneagrapher-core’ package”, <https://github.com/davidalber/geneagrapher-core>
- Jackson, A. (2007). “A Labor of Love: The Mathematics Genealogy Project”, *Notices of the AMS*, **54**(8), 1002-1003. <https://www.ams.org/notices/200708/tx070801002p.pdf>
- Mulcahy, C. (2017). “The Mathematics Genealogy Project Comes of Age at Twenty-one”, *Notices of the AMS*, **64**(5), 466-470. <https://www.ams.org/journals/notices/201705/rnoti-p466.pdf>

## Examples

```
# First, you need to use search_id() to find the mathematician ID for the
# individual(s) you wish to plot, or visit https://mathgenealogy.org/ to look
# up in the browser. Once you have these IDs the get_genealogy() function will
# retrieve the genealogical tree.

# For example, to find the package author would search for themselves using
search_id("Aslett", "Louis")

# Then, use the id to retrieve the genealogy
g <- get_genealogy(171971)

# With that genealogy, you can then plot using plot_grviz() or other plotting
# functions.
```

---

plot\_gg

*Plot genealogical tree with ggenealogy*

---

## Description

Plots a genealogical tree using the ggenealogy layout engine.

## Usage

```
plot_gg(g, max_anc = 3L, max_des = 3L, id = NULL, col = "red", expand = 0.15)
```

## Arguments

g	an object of class genealogy, as returned by <a href="#">get_genealogy()</a> .
max_anc	an integer(1) with the maximum number of generations of ancestors to be displayed.
max_des	an integer(1) with the maximum number of generations of descendants to be displayed.

id	an integer(1) or character(1) with the mathematician ID to highlight and centre the tree on. By default this is NULL which will use the first ID that was supplied to <code>get_genealogy()</code> when retrieving the genealogical tree. Note that the ID must be one of the IDs searched when calling <code>get_genealogy()</code> to construct <code>g</code> , since the search for ancestors/descendants only goes directly up/down branches reachable from the initial search ID.
col	a character(1) specifying the colour to highlight the mathematician one whom the graph is centred.
expand	a numeric(1) with the expansion factor for the graph. This defaults to 0.15, with larger values causing the x axis to expand, smaller values for it to shrink. This is useful if the nearest common ancestor has a long name, which may cause it to be clipped when plotting: increase this expansion factor to rectify this.

### Details

This function requires the `ggenealogy` package to be installed. It is only a "Suggests" dependency because this package supports multiple plotting approaches. The presence of this package will be verified when the function is actually called, providing an opportunity to install automatically if needed.

This function is not suitable for plotting very large whole genealogical trees. Consider using `plot_grviz()` if you want to see an entire tree.

### Value

An object of class ("`gg`", "`ggplot`") which can be displayed, or further manipulated using additional layers or aesthetic modifications from the `ggplot2` package.

### References

Rutter, L., VanderPlas, S., Cook, D. and Graham, M.A. (2019). "ggenealogy: An R Package for Visualizing Genealogical Data", *Journal of Statistical Software*, **89**(13), 1-31. doi:10.18637/jss.v089.i13.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

### Examples

```
# First, you need to use search_id() to find the mathematician ID for the
# individual(s) you wish to plot, or visit https://mathgenealogy.org/ to look
# up in the browser.

# For example, the package author would get their own tree using
g <- get_genealogy(171971)

# Then use the plot_gg() function to use the underlying ggenealogy package
plot_gg(g)
```

---

plot_gg_path	<i>Plot shortest path in genealogical tree with ggenealogy</i>
--------------	--

---

### Description

Plots a shortest path between two mathematicians in a genealogical tree using the ggenealogy layout engine.

### Usage

```
plot_gg_path(g, id1 = NULL, id2 = NULL, expand = 0.15)
```

### Arguments

g	an object of class genealogy, as returned by <code>get_genealogy()</code> .
id1	an integer(1) or character(1) with the ID of the first mathematician of interest.
id2	an integer(1) or character(1) with the ID of the second mathematician of interest.
expand	a numeric(1) with the expansion factor for the graph. This defaults to 0.15, with larger values causing the x axis to expand, smaller values for it to shrink. This is useful if the nearest common ancestor has a long name, which may cause it to be clipped when plotting: increase this expansion factor to rectify this.

### Details

This function requires the ggenealogy package to be installed. It is only a "Suggests" dependency because this package supports multiple plotting approaches. The presence of this package will be verified when the function is actually called, providing an opportunity to install automatically if needed.

The shortest path between the two mathematician IDs provided is plotted, with the x position of each label determined by the year of PhD award.

**NOTE:** if the name of the nearest common ancestor is long, it can be clipped by ggplot2. If this occurs, increase the expand argument greater than the default of 0.15.

### Value

An object of class ("gg", "ggplot") which can be displayed, or further manipulated using additional layers or aesthetic modifications from the ggplot2 package.

### References

Rutter, L., VanderPlas, S., Cook, D. and Graham, M.A. (2019). "ggenealogy: An R Package for Visualizing Genealogical Data", *Journal of Statistical Software*, **89**(13), 1-31. doi:10.18637/jss.v089.i13.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

## Examples

```
# First, you need to use search_id() to find the mathematician ID for the
# individual(s) you wish to plot, or visit https://mathgenealogy.org/ to look
# up in the browser.

# For example, to find the shortest genealogical path between the package
# author and my former postdoc supervisor, I would start by querying using
# both mathematician IDs
g <- get_genealogy(c(96119, 171971))

# Then use the plot_gg_path() function to use the underlying ggenealogy package
plot_gg_path(g)
```

---

plot\_grviz

*Plot genealogical tree with Graphviz*

---

## Description

Plots a genealogical tree either interactively or to PDF using the Graphviz layout engine.

## Usage

```
plot_grviz(g, file = "", max_zoom = 200)
```

## Arguments

g	an object of class <code>genealogy</code> , as returned by <code>get_genealogy()</code> .
file	an optional file name. If the file name is specified, then Graphviz will render the genealogical tree to PDF and save in this file. If the file name is not specified, then the plot will be rendered interactively in the RStudio Viewer panel.
max_zoom	a <code>numeric(1)</code> with the maximum zoom factor when plotting in the Viewer. If trees are particularly deep or wide, the default maximum zoom of 200x may be insufficient, in which case a value larger than 200 should be supplied. This option has no effect when plotting to a file.

## Details

This function requires the `DiagrammeR`, `DiagrammeRsvg` and either `svgPanZoom` (interactive) or `rsvg` (pdf output) packages to be installed. They are only "Suggests" dependencies as this package supports multiple plotting options. The presence of these packages will be verified when the function is actually called, providing an opportunity to install them automatically if needed.

## Value

If a filename was specified, the full path of the saved file is returned as a `character(1)` string. If no filename was specified, then an `htmlwidget` suitable for display in the RStudio Viewer is returned.

## References

- Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C. and Woodhull, G. (2004). “Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools”. In: Jünger, M., Mutzel, P. (eds) *Graph Drawing Software, Mathematics and Visualization*, 127-148. doi:10.1007/9783642186387\_6.
- Iannone, R. and Roy, O. (2024). *DiagrammeR: Graph/Network Visualization*. R package, <https://CRAN.R-project.org/package=DiagrammeR>.
- Iannone, R. (2016). *DiagrammeRsvg: Export DiagrammeR Graphviz Graphs as SVG*. R package, <https://CRAN.R-project.org/package=DiagrammeRsvg>.
- Ooms, J. (2024). *rsvg: Render SVG Images into PDF, PNG, (Encapsulated) PostScript, or Bitmap Arrays*. R package, <https://CRAN.R-project.org/package=rsvg>.
- Riutta, A., Tangelder, J., Russell, K., et al. (2020). *svgPanZoom: R 'Htmlwidget' to Add Pan and Zoom to Almost any R Graphic*. R package, <https://CRAN.R-project.org/package=svgPanZoom>.

## Examples

```
# First, you need to use search_id() to find the mathematician ID for the
# individual(s) you wish to plot, or visit https://mathgenealogy.org/ to look
# up in the browser.

# For example, the package author would get their own tree using
g <- get_genealogy(171971)

# Then use the plot_grviz() function to produce a full genealogical tree
plot_grviz(g)
```

---

search\_id

*Search for mathematician in Mathematics Genealogy Project*

---

## Description

Perform an online search using information about an individual mathematician to find their ID in the [Mathematics Genealogy Project](#).

## Usage

```
search_id(
  family = NULL,
  given = NULL,
  middle = NULL,
  university = NULL,
  year = NULL,
  thesis_keyword = NULL,
  country = NULL,
  discipline = NULL
)
```



**Arguments**

family	a character(1) string with the family names.
given	a character(1) string with the given names.
middle	a character(1) string with the collapsed middle name(s).
university	a character(1) string with the University at which PhD studied.
year	a character(1) string or integer(1) with the year of completion.
thesis_keyword	a character(1) string with keyword(s) in the PhD thesis title.
country	a character(1) string with the country of study.
discipline	an integer(1) with the mathematical sub-discipline code.

**Details**

Any one or more of the listed arguments can be provided. This will trigger an online search against the live **Mathematics Genealogy Project** database, so please be considerate and do not spam queries. All the information returned by a standard search on the website is gathered into a data frame and returned, enabling programmatic access to the data.

If you cannot find the individual you are looking for, it could be that they are not in the **Mathematics Genealogy Project** database. New data can be submitted by following the instructions in the "How to submit updates" section at <https://mathgenealogy.org/submit.php>.

**Value**

Data frame containing all matches against the provided search terms, with columns:

id Mathematician ID (as required by `get_genealogy()`);  
 name The full name (surname first) of the mathematician;  
 university The institution at which PhD was obtained;  
 year The year PhD was completed.

**References**

Jackson, A. (2007). "A Labor of Love: The Mathematics Genealogy Project", *Notices of the AMS*, **54**(8), 1002-1003. <https://www.ams.org/notices/200708/tx070801002p.pdf>

Mulcahy, C. (2017). "The Mathematics Genealogy Project Comes of Age at Twenty-one", *Notices of the AMS*, **64**(5), 466-470. <https://www.ams.org/journals/notices/201705/rnoti-p466.pdf>

**Examples**

```
# Search for the package author
search_id("Aslett", "Louis")

# You may find it easier to directly use the https://mathgenealogy.org/
# website, and extract the "id" from the URL on the page for the mathematician
# of interest.
```

# Index

`disciplines`, [2](#)

`get_genealogy`, [3](#)

`get_genealogy()`, [4-7](#), [9](#)

`ggplot2`, [5](#), [6](#)

`plot_gg`, [4](#)

`plot_gg_path`, [6](#)

`plot_grviz`, [7](#)

`plot_grviz()`, [5](#)

`search_id`, [8](#)

`search_id()`, [2](#)