Package: archaeacentre (via r-universe)

November 3, 2026

Title Default plots and functions for the German Archaea Centre

Version 1.1.0

Date 2025-02-03

Description Default plots and functions for the German Archaea Centre.

License BSL-1.0

URL https://github.com/richardstoeckl/archaeacentre

BugReports https://support.bioconductor.org/tag/archaeacentre

biocViews Software

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests BiocStyle, knitr, RefManageR, rmarkdown, sessioninfo, testthat

Config/testthat/edition 3

VignetteBuilder knitr

Imports dplyr, ggplot2, ggpubr, httr, magrittr, purrr, scales, stringr, tibble

Depends R (>= 3.5)

LazyData true

Config/pak/sysreqs cmake make libicu-dev libssl-dev

Repository https://richardstoeckl.r-universe.dev

Date/Publication 2025-02-12 16:24:14 UTC

RemoteUrl https://github.com/richardstoeckl/archaeacentre

RemoteRef v1.1.0

RemoteSha c27de77fd92972946a7500b5b855be77b00bad75

Contents

get_pfam_annotation_for_targets 2
get_polymer_info 3
growthData 4
plotGrowthCurve 4

Index **6**

`get_pfam_annotation_for_targets`
Retrieve Pfam Annotations for PDB Targets from RCSB API

Description

This function retrieves Pfam domain annotations for a given set of PDB target structures. The targets are expected to be results from a local Foldseek search against the PDB database. The function extracts the relevant PDB assembly and chain identifiers, queries the RCSB API, and returns the Pfam descriptions associated with the identified polymer entities.

Usage

```
get_pfam_annotation_for_targets(targets, batch_size = 200)
```

Arguments

`targets`	A character vector of PDB filenames with assembly and chain information as returned by Foldseek in the "target" column.
`batch_size`	An integer specifying the number of targets to process in one batch (default: 200). The API calls to the RCSB server are made in batches to avoid overloading the server.

Details

The function operates in the following steps:

1. Extracts PDB assembly and chain IDs from the target names.
2. Queries the RCSB API to retrieve corresponding polymer entity identifiers.
3. Filters results to match Foldseek output.
4. Retrieves Pfam annotations for the identified entities.
5. Merges results into a structured data frame.

Internally, the function calls:

- `get_assembly_id_from_target()`: Extracts assembly ID from PDB target name.
- `get_polymer_info()`: Queries the RCSB API for Pfam domain information.

Value

A data frame with columns:

- `target`: The original target name from Foldseek output.
- `rcsb_id`: The RCSB identifier for the matched polymer entity. Note: This uses the "label_asym_id" instead of the "auth_asym_id" used in the target name.
- `title`: The title of the PDB entry associated with the entity.
- `pfam_description`: The description of the Pfam family associated with the entity.

Background

The Research Collaboratory for Structural Bioinformatics (RCSB) provides structural and functional annotations for macromolecules stored in the Protein Data Bank (PDB). Foldseek is a fast search tool for comparing protein structures. When searching for similar structures in the PDB, Foldseek returns a table which contains the "target" column. In the case of searches against the PDB, this target column contains the PDB filename of the hit, which is not easily interpretable.

This function automates the extraction of Pfam domain annotations for these target PDB filenames returned by Foldseek, using the RCSB GraphQL API.

Examples

```
## Not run:
targets <- c("1ABC_assembly1.cif_A", "2XYZ_assembly2.cif_B")
pfam_results <- get_pfam_annotation_for_targets(targets, batch_size = 400)
head(pfam_results)

## End(Not run)
```

---

get_polymer_info *Retrieve Polymer Entity Identifiers and Pfam Annotations from RCSB API (batched)*

Description

This function queries the RCSB GraphQL API to retrieve polymer entity identifiers (auth_asym_id and rcsb_id) along with Pfam descriptions for a given assembly ID.

Usage

```
get_polymer_info(assembly_ids, batch_size = 200)
```

Arguments

`assembly_ids`	A character vector of assembly IDs.
`batch_size`	An integer specifying the number of assembly IDs to process in one batch (default: 400).

Value

A data frame with columns:

- `entry_id`: The PDB entry ID.
- `assembly_id`: The corresponding assembly ID.
- `auth_asym_id`: The author-specified asymmetry ID.
- `rcsb_id`: The RCSB polymer entity instance ID.
- `title`: The title of the PDB entry.
- `pfam_description`: The Pfam description for the entity.

`growthData` *Example Dataset for microbial Growth Curves*

Description

Example Dataset for microbial Growth Curves

`plotGrowthCurve` *Plot a microbial growth curve*

Description

This functions plots a microbial growth curve with the option to add a confidence interval. Under the hood, it is a simple ggplot2 wrapper, with defaults set depending on the type of plot. As such, it can be expanded upon using the usual ggplot2 syntax.

Usage

```
plotGrowthCurve(data, x, y, grouping, color, type = "robert")
```

Arguments

`data`	A data frame containing the data to plot.
`x`	The x-axis variable. Usually the time in hours after starting the experiment.
`y`	The y-axis variable. Usually the concentration of microbes per mL as a `double`
`grouping`	A character vector containing one or more column names for the grouping variables. Usually "c("timepoint", "organism")".
`color`	The color variable. Usually the strain or condition.
`type`	The type of plot to create. Currently only 'robert' is supported. This sets the default look for the plot.

See Also

The first steps chapter of the online ggplot2 book.

Examples

```
## Not run:
# load example data from this package
growthData <- archaeacentre::growthData
# plot the growth curve
plotGrowthCurve(growthData, timepoint, concentration, grouping = c("timepoint", "organism"), organism, type = "rob

## End(Not run)
```

Index

∗ **data**
growthData, 4

get_pfam_annotation_for_targets, 2
get_polymer_info, 3
growthData, 4

plotGrowthCurve, 4