

Package ‘enviPathR’

October 26, 2026

Title Client for the 'enviPath' 'RESTful' API

Version 1.0.2

Description The environmental pathway database ('enviPath') is a key resource for the functional annotation and pathway prediction of microbial communities, providing knowledge on biotransformation pathways along with the chemical reactions and produced compounds. This package enables the programmatic access to 'enviPath' by porting the database 'RESTful' API from R. This includes methods to both retrieve entries and predict pathways for custom compounds.

biocViews Software, DataImport, Cheminformatics, Annotation, Pathways, FunctionalPrediction

Depends R (>= 4.1.0)

Imports BiocParallel, httr2, stats, stringr, tools

Suggests BiocStyle, ggraph, httptest2, igraph, knitr, rmarkdown, testthat (>= 3.0.0)

License Artistic-2.0

Encoding UTF-8

RoxygenNote 8.0.0

URL <https://envipath.github.io/enviPathR/>

BugReports <https://github.com/enviPath/enviPathR/issues>

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Giulio Benedetti [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-8732-7692>>)

Maintainer Giulio Benedetti <giulio.benedetti@utu.fi>

Repository CRAN

Date/Publication 2026-10-26 09:20:02 UTC

Contents

enviPathR	2
epGet	2
epLink	4
epList	5
epLogin	6
epModel	7
epTypes	8
Index	10

enviPathR	<i>enviPathR: the R client for the enviPath RESTful API</i>
-----------	---

Description

The environmental pathway database (enviPath) is a key resource for the functional annotation and pathway prediction of microbial communities, providing knowledge on biotransformation pathways along with the chemical reactions and produced compounds. This package enables the programmatic access to enviPath by porting the database API from R.

Author(s)

Maintainer: Giulio Benedetti <giulio.benedetti@utu.fi> (**ORCID**) [copyright holder]

Authors:

- Giulio Benedetti <giulio.benedetti@utu.fi> (**ORCID**) [copyright holder]

See Also

- enviPath front page: <https://envipath.org/>
- legacy API docs: <https://envipath.org/api/legacy/docs#/>

epGet	<i>Get enviPath objects</i>
-------	-----------------------------

Description

epGet returns raw objects from enviPath.

Usage

```
epGet(  
  type,  
  init = NULL,  
  pkg = NULL,  
  property = NULL,  
  BPPARAM = bpparam(),  
  rate = 5  
)
```

Arguments

type	Character scalar. String specifying the object type to be fetched.
init	Character vector. Vector of strings specifying the initial values of type that should be fetched. When null, all objects belonging to type in pkg are used. (Default: NULL)
pkg	Character scalar. String specifying the unique identifier of the package from which objects should be fetched. When null, EAWAG-BBD is used. (Default: NULL)
property	Character scalar. String specifying the property to be fetched from the objects. When null, all elements of the objects are returned. (Default: NULL)
BPPARAM	A BiocParallelParam object specifying how the requests should be parallelised. (Default: bpparam())
rate	Numeric scalar. Integer number specifying the amount of requests per second to be sent to the <code>enviPath</code> API when BPPARAM is not serial. (Default: 5)

Value

A list of objects.

Examples

```
if( is_active ){  
  # Perform login with your credentials  
  epLogin(username, password)  
  
  # Define custom reaction identifier  
  rxn_id <- "2b6bbcc5-77f4-4bed-92a9-731cdc978f6a"  
  
  # Get custom reaction object  
  epGet("reaction", rxn_id)  
}
```

epLink	<i>Link enviPath objects</i>
--------	------------------------------

Description

epLink returns mappings between enviPath object types.

Usage

```
epLink(from, to, init = NULL, pkg = NULL, BPPARAM = bpparam(), rate = 5)
```

Arguments

from	Character scalar. String specifying the object type from which mapping is performed.
to	Character scalar. String specifying the object type to which mapping is performed.
init	Character vector. Vector of strings specifying the initial values of from that should be mapped to to. When null, all objects belonging to from in pkg are used. (Default: NULL)
pkg	Character scalar. String specifying the unique identifier of the package from which objects should be mapped. When null, EAWAG-BBD is used. (Default: NULL)
BPPARAM	A BiocParallelParam object specifying how the requests should be parallelised. (Default: bpparam())
rate	Numeric scalar. Integer number specifying the amount of requests per second to be sent to the enviPath API when BPPARAM is not serial. (Default: 5)

Value

A data frame with links between from and to.

Examples

```
if( is_active ){  
  # Perform login with your credentials  
  epLogin(username, password)  
  
  # Define custom reaction identifier  
  rxn_id <- "2b6bbcc5-77f4-4bed-92a9-731cdc978f6a"  
  
  # Link custom reaction to related compounds  
  epLink("reaction", "compound", rxn_id)  
}
```

epList	<i>List enviPath objects</i>
--------	------------------------------

Description

epList returns a list with info on objects available in enviPath.

Usage

```
epList(type, pkg = NULL)
```

Arguments

type	Character scalar. String specifying the object type to listed.
pkg	Character scalar. String specifying the unique identifier of the package from which objects should be listed. When null, all available packages are used. (Default: NULL)

Value

A data frame with all objects belonging to type in pkg.

Examples

```
if( is_active ){  
  # Perform login with your credentials  
  epLogin(username, password)  
  
  # List packages  
  pkg_df <- epList("package")  
  
  # Select reviewed packages  
  to_keep <- pkg_df$reviewStatus == "reviewed"  
  pkg_df <- pkg_df[to_keep, ]  
  
  # Remove lengthy description  
  pkg_df$description <- NULL  
  
  # View some packages  
  head(pkg_df)  
  
  # Select desired package  
  pkg_name <- "EAWAG-BBD"  
  pkg_id <- pkg_df$id[pkg_df$name == pkg_name]  
  
  # List pathways from desired package  
  path_df <- epList("pathway", pkg = pkg_id)  
  
  # View some pathways
```

```
head(path_df)
}
```

epLogin*Log into enviPath account*

Description

epLogin and epLogout let you log in and out of enviPath, respectively. An account is required to use the enviPath API.

Usage

```
epLogin(username, password)
```

```
epLogout()
```

Arguments

username	Character scalar. String specifying the username used to log into an enviPath personal account.
----------	---

password	Character scalar. String specifying the password used to log into an enviPath personal account.
----------	---

Value

A message upon successful login or logout.

Examples

```
if( is_active ){
# Perform login with your credentials
epLogin(username, password)

# Perform logout
epLogout()
}
```

epModel

Predict compound biotransformation pathways

Description

epModel predicts the biotransformation pathway for a compound expressed with smiles using the selected model setting.

Usage

```
epModel(smiles, setting = NULL)
```

Arguments

smiles	Character vector. String specifying the smiles of a compound, which does not necessarily have to be available in enviPath.
setting	Character scalar. String specifying the unique identifier of a model setting. When null, the enviFormer setting is used. (Default: NULL)

Value

A list with two data frames with information on nodes and edges, respectively.

Examples

```
if( is_active ){
  library(igraph)
  library(ggraph)

  # Perform login with your credentials
  epLogin(username, password)

  # Define smiles of interest
  smiles <- "ClC(Cl)=C(Cl)Cl"

  # Perform pathway prediction with enviFormer
  former_out <- epModel(smiles)

  # List available model settings
  set_df <- epList("setting")

  # View some model settings
  head(set_df)

  # Set id for PEPPER model setting
  set_name <- "Global Setting - ECC and App Domain - PEPPER"
  set_id <- set_df$id[set_df$name == set_name]

  # Perform pathway prediction with PEPPER
```

```
pepper_out <- epModel(smiles, set_id)

# Convert model output to igraph object
path_graph <- graph_from_data_frame(
  pepper_out$edges,
  vertices = pepper_out$nodes
)

# Visualise predicted pathway
plot(path_graph)

# Visualise with ggraph
ggraph(path_graph, layout = "sugiyama") +
  geom_edge_link(
    aes(colour = probability),
    arrow = arrow(type = "closed")
  ) +
  geom_node_point(size = 3) +
  geom_node_text(aes(label = name), vjust = 2) +
  scale_edge_colour_continuous(
    limits = c(0, 1), low = "white", high = "red"
  ) +
  theme_graph(base_family = "")
}
```

epTypes

Find available enviPath resources

Description

epTypes retrieves the set of listable and linkable enviPath object types

Usage

```
epTypes()
```

Value

A list with two elements:

- listable: a character vector of the object types supported by [eplist](#)
- linkable: a data frame of links between object types supported by [eplink](#)

Examples

```
# Retrieve available resources
types <- epTypes()

# View listable object types
types$listable
```

```
# View linkable object types
types$linkable
```

Index

BiocParallelParam, [3](#), [4](#)

bpparam(), [3](#), [4](#)

enviPathR, [2](#)

enviPathR-package (enviPathR), [2](#)

epGet, [2](#)

epLink, [4](#), [8](#)

epList, [5](#), [8](#)

epLogin, [6](#)

epLogout (epLogin), [6](#)

epModel, [7](#)

epTypes, [8](#)