Package: rsynthbio (via r-universe)

May 6, 2026

Type Package

Title Synthesize Bio API Wrapper

Version 4.1.0

Description Access Synthesize Bio models from their API <https://app.synthesize.bio/> using this wrapper that provides a convenient interface to the Synthesize Bio API, allowing users to generate realistic gene expression data based on specified biological conditions. This package enables researchers to easily access AI-generated transcriptomic data for various modalities including bulk RNA-seq, single-cell RNA-seq, microarray data, and more.

URL https://github.com/synthesizebio/rsynthbio

BugReports https://github.com/synthesizebio/rsynthbio/issues

Imports getPass, keyring, jsonlite, httr

Suggests rmarkdown, knitr, testthat (>= 3.0.2), mockery

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

LazyData true

VignetteBuilder knitr

License MIT + file LICENSE

Config/pak/sysreqs libsecret-1-dev libssl-dev

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

Date/Publication 2026-05-06 17:00:13 UTC

RemoteUrl https://github.com/synthesizebio/rsynthbio

RemoteRef HEAD

RemoteSha 9fecbcfdb46f19536864e55e671a64fd72ad85e7

Contents

`API_BASE_URL` *API Base URL*

Description

Base URL for the Synthesize Bio API

Usage

```
API_BASE_URL
```

Format

An object of class `character` of length 1.

`clear_synthesize_token` *Clear Synthesize Bio API Token*

Description

Clears the Synthesize Bio API token from the environment for the current R session. This is useful for security purposes when you've finished working with the API or when switching between different accounts.

Usage

```
clear_synthesize_token(remove_from_keyring = FALSE)
```

Arguments

`remove_from_keyring`	Logical, whether to also remove the token from the system keyring if it's stored there. Defaults to FALSE.

Value

Invisibly returns TRUE.

Examples

```
## Not run:
# Clear token from current session only
clear_synthesize_token()

# Clear token from both session and keyring
clear_synthesize_token(remove_from_keyring = TRUE)

## End(Not run)
```

`DEFAULT_POLL_INTERVAL_SECONDS`
Default Poll Interval

Description

Default polling interval (seconds) for async model queries

Usage

```
DEFAULT_POLL_INTERVAL_SECONDS
```

Format

An object of class `numeric` of length 1.

`DEFAULT_POLL_TIMEOUT_SECONDS`
Default Poll Timeout

Description

Default maximum timeout (seconds) for async model queries

Usage

```
DEFAULT_POLL_TIMEOUT_SECONDS
```

Format

An object of class `numeric` of length 1.

`DEFAULT_TIMEOUT` *Default Timeout*

Description

Default timeout (seconds) for outbound HTTP requests

Usage

```
DEFAULT_TIMEOUT
```

Format

An object of class `numeric` of length 1.

`get_example_query` *Get Example Query for Model*

Description

Retrieves an example query structure for a specific model. This provides a template that can be modified for your specific needs.

Usage

```
get_example_query(model_id, api_base_url = API_BASE_URL)
```

Arguments

`model_id`	Character string specifying the model ID (e.g., "gem-1-bulk", "gem-1-sc").
`api_base_url`	The base URL for the API server. Default is API_BASE_URL.

Value

A list representing a valid query structure for the specified model.

Examples

```
## Not run:
# Get example query for bulk RNA-seq model
query <- get_example_query(model_id = "gem-1-bulk")$example_query

# Get example query for single-cell model
query_sc <- get_example_query(model_id = "gem-1-sc")$example_query

# Modify the query structure
query$inputs[[1]]$num_samples <- 10

## End(Not run)
```

`has_synthesize_token` *Check if Synthesize Bio API Token is Set*

Description

Checks whether a Synthesize Bio API token is currently set in the environment. Useful for conditional code that requires an API token.

Usage

```
has_synthesize_token()
```

Value

Logical, TRUE if token is set, FALSE otherwise.

Examples

```
 ## Not run:
# Check if token is set
if (!has_synthesize_token()) {
  # Prompt for token if not set
  set_synthesize_token()
}

## End(Not run)
```

`list_models` *List Available Models*

Description

Returns a list of all models available in the Synthesize Bio API. Each model has a unique ID that can be used with predict_query() and get_example_query().

Usage

```
list_models(api_base_url = API_BASE_URL)
```

Arguments

`api_base_url`	The base URL for the API server. Default is API_BASE_URL.

Value

A list or data frame containing available models with their IDs and metadata.

Examples

```
## Not run:
# Get all available models
models <- list_models()
print(models)

## End(Not run)
```

`load_synthesize_token_from_keyring` *Load Synthesize Bio API Token from Keyring*

Description

Loads the previously stored Synthesize Bio API token from the system keyring and sets it in the environment for the current session.

Usage

```
load_synthesize_token_from_keyring()
```

Value

Invisibly returns TRUE if successful, FALSE if token not found in keyring.

Examples

```
## Not run:
# Load token from keyring
load_synthesize_token_from_keyring()

## End(Not run)
```

`predict_query` *Predict Gene Expression*

Description

Sends a query to the Synthesize Bio API for prediction and retrieves gene expression samples. This function sends the query to the API and processes the response into usable data frames.

Usage

```
predict_query(
  query,
  model_id,
  api_base_url = API_BASE_URL,
  poll_interval_seconds = DEFAULT_POLL_INTERVAL_SECONDS,
  poll_timeout_seconds = DEFAULT_POLL_TIMEOUT_SECONDS,
  return_download_url = FALSE,
  raw_response = FALSE,
  ...
)
```

Arguments

`query`	A list representing the query data to send to the API. Use ‘get_example_query()‘ to generate an example. The query supports additional optional fields: • ‘total_count‘ (integer): Library size used when converting predicted log CPM back to raw counts. Higher values scale counts up proportionally. • ‘deterministic_latents‘ (logical): If TRUE, the model uses the mean of each latent distribution instead of sampling, producing deterministic outputs for the same inputs. Useful for reproducibility. • ‘seed‘ (integer): Random seed for reproducibility.
`model_id`	Character string specifying the model ID (e.g., "gem-1-bulk", "gem-1-sc"). Use ‘list_models()‘ to see available models.
`api_base_url`	The base URL for the API server. Default is API_BASE_URL.
`poll_interval_seconds`	Seconds between polling attempts of the status endpoint. Default is DEFAULT_POLL_INTERVAL_SEC (2).

`poll_timeout_seconds`	Maximum total seconds to wait before timing out. Default is DEFAULT_POLL_TIMEOUT_SECONDS (900 = 15 minutes).
`return_download_url`	Logical, if TRUE, returns a list containing the signed download URL instead of parsing into data frames. Default is FALSE.
`raw_response`	Logical, if TRUE, returns the raw (unformatted) JSON response from the API without applying any output transformers. Default is FALSE.
`...`	Additional parameters to include in the query body. These are passed directly to the API and validated server-side.

Value

A list. If ‘return_download_url‘ is ‘FALSE‘ (default), the list contains two data frames: ‘metadata‘ and ‘expression‘. If ‘TRUE‘, the list contains ‘download_url‘ and empty ‘metadata‘ and ‘expression‘ data frames.

Examples

```
# Set your API key (in practice, use a more secure method)
## Not run:

# To start using rsynthbio, first you need to have an account with synthesize.bio.
# Go here to create one: https://app.synthesize.bio/

set_synthesize_token()

# Get available models
models <- list_models()

# Create a query for a specific model
query <- get_example_query(model_id = "gem-1-bulk")$example_query

# Request raw counts
result <- predict_query(query, model_id = "gem-1-bulk")

# Access the results
metadata <- result$metadata
expression <- result$expression

# Explore the top expressed genes in the first sample
head(sort(expression[1, ], decreasing = TRUE))

# Use deterministic latents for reproducible results
query$deterministic_latents <- TRUE
result_det <- predict_query(query, model_id = "gem-1-bulk")

# Specify a custom total count (library size)
query$total_count <- 5000000
result_custom <- predict_query(query, model_id = "gem-1-bulk")
```

```
## End(Not run)
```

`set_synthesize_token` *Set Synthesize Bio API Token*

Description

Securely prompts for and stores the Synthesize Bio API token in the environment. This function uses getPass to securely handle the token input without displaying it in the console. The token is stored in the SYNTHESIZE_API_KEY environment variable for the current R session.

Usage

```
set_synthesize_token(use_keyring = FALSE, token = NULL)
```

Arguments

`use_keyring`	Logical, whether to also store the token securely in the system keyring for future sessions. Defaults to FALSE.
`token`	Character, optional. If provided, uses this token instead of prompting. This parameter should only be used in non-interactive scripts.

Value

Invisibly returns TRUE if successful.

Examples

```
# Interactive prompt for token
## Not run:
set_synthesize_token()

# Provide token directly (less secure, not recommended for interactive use)
set_synthesize_token(token = "your-token-here")

# Store in system keyring for future sessions
set_synthesize_token(use_keyring = TRUE)

## End(Not run)
```

Index