

Package ‘avidaR’

June 21, 2024

Title A Computational Biologist’s Toolkit To Get Data From 'avidaDB'

Version 1.2.1

Description Easy-to-use tools for performing complex queries on 'avidaDB', a semantic database that stores genomic and transcriptomic data of self-replicating computer programs (known as digital organisms) that mutate and evolve within a user-defined computational environment.

License MIT + file LICENSE

URL <https://gitlab.com/fortunallab/avidaR>

Encoding UTF-8

Language en-US

RoxygenNote 7.3.1

Depends R (>= 3.6.0)

Imports base64enc (>= 0.1-3), xml2 (>= 1.3.2), httr (>= 1.4.2), dplyr (>= 1.0.6), readr (>= 1.4.0), tidyr (>= 1.1.2), tibble (>= 3.0.6), circlize (>= 0.4.11), RColorBrewer, R6, curl (>= 5.0.2)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Miguel A. Fortuna [aut] (<<https://orcid.org/0000-0002-8374-1941>>),
Raúl Ortega [cre] (<<https://orcid.org/0000-0002-1306-5378>>)

Maintainer Raúl Ortega <raul.ortega@ebd.csic.es>

Repository CRAN

Date/Publication 2024-06-21 17:50:02 UTC

Contents

| | |
|---|---|
| convert_org_into_seq | 2 |
| convert_seq_into_org | 3 |
| get_db_summary | 5 |
| get_docker_image_from_experiment_id | 5 |

| | |
|---|----|
| get_doi_from_experiment_id | 6 |
| get_experiment_id_from_organism_id | 7 |
| get_genome_id_from_genome_seq | 8 |
| get_genome_id_from_logic_operation | 9 |
| get_genome_id_from_phenotype_id | 11 |
| get_genome_id_from_transcriptome_id | 12 |
| get_genome_id_of_wild_type_organisms | 13 |
| get_genome_seq_from_genome_id | 14 |
| get_logic_operation_from_phenotype_id | 15 |
| get_mutant_at_pos | 16 |
| get_phenotype_id_from_genome_id | 18 |
| get_phenotype_id_from_genome_seq | 19 |
| get_phenotype_id_from_logic_operation | 21 |
| get_phenotype_id_from_transcriptome_id | 22 |
| get_tandem_id_from_genome_id | 23 |
| get_tandem_id_from_genome_seq | 25 |
| get_tandem_id_from_logic_operation | 26 |
| get_tandem_id_from_phenotype_id | 28 |
| get_tandem_seq_from_tandem_id | 29 |
| get_transcriptome_id_from_genome_id | 31 |
| get_transcriptome_id_from_genome_seq | 32 |
| get_transcriptome_id_from_logic_operation | 34 |
| get_transcriptome_id_from_phenotype_id | 36 |
| get_transcriptome_seq_from_transcriptome_id | 37 |
| instruction_set | 39 |
| logic_operation | 39 |
| plot_transcriptome | 40 |
| triplestore_access | 41 |

Index **44**

convert_org_into_seq *Converts a genome instruction sequence into a digital organism file*

Description

Converts a digital organism instruction set into a digital genome instruction sequence.

Usage

```
convert_org_into_seq(
    org,
    inst_set = "heads",
    save = FALSE,
    file_name = NULL,
    save_path = getwd(),
    format = "seq",
    silent = FALSE
)
```

Arguments

| | |
|-----------|--|
| org | Data Frame. Column names: instruction. |
| inst_set | Name of the instruction set. It must be one of the following: heads (default), heads-sex", or transsmt. The names correspond to the instruction set configuration files (e.g., instset-heads.cfg for heads). |
| save | Logical value (TRUE/FALSE) indicating whether the output should or should not be saved to a file ("FALSE" by default). |
| file_name | String of characters representing the name of the file without any extension ("genome.seq" by default). |
| save_path | String of characters representing the name of the folder where the digital organism file will be saved. |
| format | String of characters representing the format of the file ("seq" by default). |
| silent | Logical value (TRUE/FALSE) to show/hide messages ("FALSE" by default). |

Value

String of characters.

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

sequence <- get_genome_seq_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)$genome_seq[[1]]

convert_seq_into_org(genome_seq = sequence)
```

convert_seq_into_org *Converts a genome instruction sequence into a digital organism file*

Description

Converts a genome instruction sequence into a digital organism file.

Usage

```
convert_seq_into_org(
  genome_seq,
  inst_set = "heads",
  save = FALSE,
  file_name = NULL,
  save_path = getwd(),
  format = "org",
  silent = FALSE
)
```

Arguments

| | |
|------------|---|
| genome_seq | String of letters. |
| inst_set | Name of the instruction set. It must be one of the following: heads (default), heads-sex, or transsmt. The names correspond to the instruction set configuration files (e.g., instset-heads.cfg for heads). |
| save | Logical value (TRUE/FALSE) indicating whether the output should or should not be saved to a file ("FALSE" by default). |
| file_name | String of characters representing the name of the file without any extension ("organism.org" by default). |
| save_path | String of characters representing the name of the folder where the digital organism file will be saved. |
| format | String of characters representing the format of the file ("org" by default). |
| silent | Logical value (TRUE/FALSE) to show/hide messages ("FALSE" by default). |

Value

Data frame. Column names: "instruction".

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

sequence <- get_genome_seq_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)$genome_seq[[1]]
```

```
convert_seq_into_org(genome_seq = sequence)
```

| | |
|-----------------------------|-----------------------------|
| <code>get_db_summary</code> | <i>Get database summary</i> |
|-----------------------------|-----------------------------|

Description

Get a summary of the data stored.

Usage

```
get_db_summary(triplestore)
```

Arguments

`triplestore` Object of class `triplestore_access` which manages database access.

Value

Data frame: Columns: "data type", "value".

Examples

```
avidaDB <- triplestore_access$new()

avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

get_db_summary(triplestore = avidaDB)
```

| | |
|--|---|
| <code>get_docker_image_from_experiment_id</code> | <i>Get docker image from experiment</i> |
|--|---|

Description

get the docker image built to run the experiment, which guarantees reproducibility .

Usage

```
get_docker_image_from_experiment_id(avida_experiment_id, triplestore)
```

Arguments

`avida_experiment_id` Integer or a vector of integer values.
`triplestore` Object of class `triplestore_access` which manages database access.

Value

Data frame. Columns: "avida_experiment_id" "docker_image_id" "id" "repo_digest"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single paper
get_docker_image_from_experiment_id(
  avida_experiment_id = 1,
  triplestore = avidaDB
)

# More than one experiment
get_docker_image_from_experiment_id(
  avida_experiment_id = c(1, 2, 3, 4),
  triplestore = avidaDB
)
```

`get_doi_from_experiment_id`
Get doi from experiment

Description

Get the DOI of the scientific publication that documents the experiment carried out.

Usage

```
get_doi_from_experiment_id(avida_experiment_id, triplestore)
```

Arguments

`avida_experiment_id` Integer or a list of integer values.
`triplestore` Object of class `triplestore_access` which manages database access.

Value

Data frame. Columns: "doi" "avida_experiment_id"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single paper
get_doi_from_experiment_id(
  avida_experiment_id = 1,
  triplestore = avidaDB
)

# More than one experiment
get_doi_from_experiment_id(
  avida_experiment_id = c(1, 2, 4),
  triplestore = avidaDB
)
```

`get_experiment_id_from_organism_id`
Get experiment from organism

Description

Get the experiment identifier and description from which a digital organism is derived.

Usage

```
get_experiment_id_from_organism_id(
  organism_id,
  triplestore,
  description = FALSE
)
```

Arguments

| | |
|-------------|---|
| organism_id | Integer or a list of integer values. |
| triplestore | Object of class triplestore_access which manages database access. |
| description | Logical value (TRUE/FALSE) to show/hide the description of the experiment (FALSE by default). |

Value

Data frame. Columns: "organism_id" "avida_experiment_id"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single organism
get_experiment_id_from_organism_id(organism_id = 1, triplestore = avidaDB)

# More than one organism
get_experiment_id_from_organism_id(
  organism_id = c(1, 2, 3),
  triplestore = avidaDB
)
```

```
get_genome_id_from_genome_seq
```

Get genome from genome sequence

Description

Get the genome of a digital organism from the linear string of letters representing the instruction codes that make up its genome.

Usage

```
get_genome_id_from_genome_seq(genome_seq, triplestore)
```


Arguments

genome_seq String of letters or a list of strings
triplestore Object of class triplestore_access which manages database access.

Value

Data frame. Columns: "genome_id" "genome_seq"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Get sequence for genome_1
sequence <- get_genome_seq_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)$genome_seq[1]

# Get genome id from sequence
get_genome_id_from_genome_seq(
  genome_seq = sequence,
  triplestore = avidaDB
)
```

get_genome_id_from_logic_operation
Get genome from logic operations

Description

Get the genome of a digital organism that encodes a unique combination of logic operations for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_genome_id_from_logic_operation(
  logic_operation,
  seed_id = sample(1:1000, 1),
```

```

    genome_seq = FALSE,
    triplestore
  )

```

Arguments

| | |
|------------------------------|---|
| <code>logic_operation</code> | List of logical operations from the following set: "equals", "exclusive-or", "not-or", "and-not", "or", "orn-not", "and", "not-and", "not". |
| <code>seed_id</code> | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If <code>seed_id</code> value is not specified, it returns data for a single randomly chosen <code>seed_id</code> value (between 1 and 1000). |
| <code>genome_seq</code> | Logical value (TRUE/FALSE) to show/hide this column ("FALSE" by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_id", "genome_seq" (optional).

Examples

```

# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single logic operation
get_genome_id_from_logic_operation(
  logic_operation = "not-or",
  triplestore = avidaDB
)

# More than one logic operation
get_genome_id_from_logic_operation(
  logic_operation = c("not", "not-and"),
  triplestore = avidaDB
)

# At seed_1
get_genome_id_from_logic_operation(
  logic_operation = c("or", "equals", "and"),
  seed_id = 1,
  genome_seq = TRUE,
  triplestore = avidaDB
)

```

```
)
```

```
get_genome_id_from_phenotype_id
      Get genome from phenotype
```

Description

Get the genome of a digital organism that encodes a specific phenotype for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_genome_id_from_phenotype_id(
  phenotype_id,
  seed_id = sample(1:1000, 1),
  genome_seq = FALSE,
  triplestore
)
```

Arguments

| | |
|--------------|---|
| phenotype_id | Integer or a list of integer values. |
| seed_id | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If seed_id value is not specified, it returns data for a single randomly chosen seed_id value (between 1 and 1000). |
| genome_seq | Logical value (TRUE/FALSE) to show/hide this column ("FALSE" by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "phenotype_id", "genome_seq" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)
```

```

# Single phenotype
get_genome_id_from_phenotype_id(phenotype_id = 1, triplestore = avidaDB)

# More than one phenotype
get_genome_id_from_phenotype_id(
  phenotype_id = c(1, 2),
  genome_seq = TRUE,
  triplestore = avidaDB
)

# At seeds_4 and seed_5
get_genome_id_from_phenotype_id(
  phenotype_id = c(1, 2),
  seed_id = c(4, 5),
  triplestore = avidaDB
)

```

```

get_genome_id_from_transcriptome_id
      Get genome from transcriptome

```

Description

Get the genome of a digital organism that executes a specific transcriptome for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_genome_id_from_transcriptome_id(
  transcriptome_id,
  seed_id = FALSE,
  genome_seq = FALSE,
  triplestore
)

```

Arguments

| | |
|-------------------------------|---|
| <code>transcriptome_id</code> | Integer or a list of integer values. |
| <code>seed_id</code> | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| <code>genome_seq</code> | Logical value (TRUE/FALSE) to show/hide this column ("FALSE" by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "transcriptome_id", "genome_seq" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single transcriptome
get_genome_id_from_transcriptome_id(
  transcriptome_id = 1,
  triplestore = avidaDB
)

# More than one transcriptome
get_genome_id_from_transcriptome_id(
  transcriptome_id = c(1, 2, 3),
  genome_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_genome_id_from_transcriptome_id(
  transcriptome_id = 1,
  seed_id = c(1, 2),
  triplestore = avidaDB
)
```

get_genome_id_of_wild_type_organisms

Get genomes of wild-type organisms

Description

Get the genome of the digital organisms that were used as wild-type organisms to get their single-point mutants by calling the function `get_mutant_at_pos`.

Usage

```
get_genome_id_of_wild_type_organisms(triplestore)
```

Arguments

triplestore Object of class triplestore_access which manages database access.

Value

Data frame: "genome_id_wild_type".

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

get_genome_id_of_wild_type_organisms(triplestore = avidaDB)
```

`get_genome_seq_from_genome_id`

Get genome sequence from genome

Description

Get the linear string of letters representing the instruction codes that make up the genome of a digital organism from the id of the genome of a digital organism.

Usage

```
get_genome_seq_from_genome_id(genome_id, triplestore)
```

Arguments

genome_id Integer or a list of integer values.

triplestore Object of class triplestore_access which manages database access.

Value

Data frame. Columns: "genome_id" "genome_seq"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single genome
get_genome_seq_from_genome_id(1, triplestore = avidaDB)

# More than one genome
get_genome_seq_from_genome_id(
  genome_id = c(1, 2, 3, 4),
  triplestore = avidaDB
)
```

```
get_logic_operation_from_phenotype_id
```

Get the logic operations computed by a digital organism whose genome encodes a specific phenotype

Description

Get the logic operations encoded by a digital organism having the requested phenotype.

Usage

```
get_logic_operation_from_phenotype_id(
  phenotype_id = FALSE,
  phenotype_binary = FALSE,
  triplestore
)
```

Arguments

| | |
|------------------|--|
| phenotype_id | Integer, a vector of integers (from 0 to 511), or a logical value (if FALSE, the function returns the entire phenotype space). |
| phenotype_binary | Logical value (TRUE/FALSE) to show/hide phenotype_id in binary notation (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "phenotype_id", "phenotype_binary" (optional), "equals", "exclusive-or", "not-or", "and-not", "or", "orn-not", "and", "not-and", "not"

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# One phenotype
get_logic_operation_from_phenotype_id(
  phenotype_id = 1,
  phenotype_binary = TRUE,
  triplestore = avidaDB
)

# More than one phenotype
get_logic_operation_from_phenotype_id(
  phenotype_id = c(1,2,3),
  phenotype_binary = TRUE,
  triplestore = avidaDB
)

# All phenotypes
get_logic_operation_from_phenotype_id(triplestore = avidaDB)
```

get_mutant_at_pos *Get single-point mutants of wild-type organisms*

Description

Get the genome sequence of a digital organism (i.e., wild-type) and its single-point mutants.

Usage

```
get_mutant_at_pos(
  genome_id = NULL,
  inst_replaced = NULL,
  inst_replaced_by = NULL,
  pos = NULL,
```



```

    triplestore
  )

```

Arguments

| | |
|------------------|--|
| genome_id | Integer or a list of integer values. If not specified, the function will return the single-point mutants of a randomly chosen wild-type organism. |
| inst_replaced | A letter representing the instruction of the genome sequence of the wild-type organism to be mutated. If not specified, the function will return the single-point mutants that have replaced the letter that the genome of the wild-type organism carried at that position (if the position is specified, otherwise it will return the mutations located on all positions) by the letter indicated in the argument inst_replaced_by (if specified, otherwise it will return all the mutants at that position on the genome of the wild type organism). |
| inst_replaced_by | A letter representing the instruction of the genome of the single-point mutant that have replaced the instruction of the genome of the wild-type organism. If not specified, the function will return all single-point mutants that have replaced the letter indicated in the argument inst_replaced (if specified, otherwise it will return all the mutants at that position on the genome of the wild type organism) of the genome of the wild-type organism at that position (if the position is specified, otherwise it will return the mutations located on all positions). |
| pos | Integer representing the position of the single-point mutation along the genome of a digital organism (from 1 to 100 for a genome length of 100 instructions). If not specified, the function will return all single-point mutants of the genome of the wild-type organism. |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame: Columns: "genome_id_wild_type", "genome_seq_wild_type", "genome_id_mutant", "genome_seq_mutant", "pos".

Examples

```

# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

get_mutant_at_pos(
  genome_id = 582,
  inst_replaced = 'o',

```

```
inst_replaced_by = 'a',
pos = 1,
triplestore = avidaDB)
```

```
get_phenotype_id_from_genome_id
```

Get phenotype from genome

Description

Get the phenotype encoded by the genome of a digital organism for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_phenotype_id_from_genome_id(
  genome_id,
  seed_id = FALSE,
  phenotype_binary = FALSE,
  triplestore
)
```

Arguments

| | |
|-------------------------------|---|
| <code>genome_id</code> | Integer or a list of integer values. |
| <code>seed_id</code> | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| <code>phenotype_binary</code> | Logical value (TRUE/FALSE) to show/hide <code>phenotype_id</code> in binary notation (FALSE by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_id", "phenotype_id" "phenotype_binary" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
```

```
avidaDB$set_access_options(  
  url = "https://graphdb.fortunalab.org",  
  user = "public_avida",  
  password = "public_avida",  
  repository = "avidaDB_test"  
)  
  
# Single genome  
get_phenotype_id_from_genome_id(genome_id = 1, triplestore = avidaDB)  
  
# More than one genome at seed_1  
get_phenotype_id_from_genome_id(  
  genome_id = c(1, 2, 3),  
  seed_id = 1,  
  triplestore = avidaDB  
)  
  
# More than one genome at more than one seed (e.g., seed_3 and seed_4)  
get_phenotype_id_from_genome_id(  
  genome_id = 1,  
  seed_id = c(3, 4),  
  phenotype_binary = TRUE,  
  triplestore = avidaDB  
)
```

get_phenotype_id_from_genome_seq

Get phenotype from genome sequence

Description

Get the phenotype encoded by the instruction sequence constituting the genome of a digital organism for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_phenotype_id_from_genome_seq(  
  genome_seq,  
  seed_id = FALSE,  
  genome_id = FALSE,  
  phenotype_binary = FALSE,  
  triplestore  
)
```

Arguments

genome_seq String of letters or a list of strings.

| | |
|------------------|---|
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| genome_id | Logical value (TRUE/FALSE) to show/hide genome_id (FALSE by default). |
| phenotype_binary | Logical value (TRUE/FALSE) to show/hide phenotype_id in binary notation (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_id" (optional), "genome_seq", "phenotype_id", "phenotype_binary" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Get sequences for genomes_1 and genome_2
sequence1 <- get_genome_seq_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)$genome_seq[1]

sequence2 <- get_genome_seq_from_genome_id(
  genome_id = 2,
  triplestore = avidaDB
)$genome_seq[1]

# Single genome
get_phenotype_id_from_genome_seq(
  genome_seq = sequence1,
  triplestore = avidaDB
)

# More than one genome
get_phenotype_id_from_genome_seq(
  genome_seq = c(sequence1, sequence2),
  genome_id = TRUE,
  phenotype_binary = TRUE,
```

```
triplestore = avidaDB
)

# At seed_1 and seed_2
get_phenotype_id_from_genome_seq(
  genome_seq = sequence2,
  seed_id = c(1, 2),
  triplestore = avidaDB
)
```

get_phenotype_id_from_logic_operation

Get phenotype from logic operations

Description

Get the phenotype encoded by the genome of a digital organism that is specified by a unique combination of logic operations.

Usage

```
get_phenotype_id_from_logic_operation(
  logic_operation,
  phenotype_binary = FALSE,
  triplestore
)
```

Arguments

logic_operation List of logical functions from the following set: "equals", "exclusive-or", "not-or", "and-not", "or", "orn-not", "and", "not-and", "not".

phenotype_binary Logical value (TRUE/FALSE) to show/hide the phenotype in binary notation (FALSE by default).

triplestore Object of class triplestore_access which manages database access.

Value

Data frame. Columns: "phenotype_id", "phenotype_binary" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
```

```

avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single logic operation
get_phenotype_id_from_logic_operation(
  logic_operation = "not-or",
  triplestore = avidaDB
)

# More than one logic operation
ops <- c("equals", "exclusive or", "not-or", "and-not", "or", "orn-not")

get_phenotype_id_from_logic_operation(
  logic_operation = ops,
  phenotype_binary = TRUE,
  triplestore = avidaDB
)

```

```

get_phenotype_id_from_transcriptome_id
  Get phenotype from transcriptome

```

Description

Get the phenotype encoded by the genome of a digital organism that executes a specific transcriptome for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_phenotype_id_from_transcriptome_id(
  transcriptome_id,
  seed_id = FALSE,
  phenotype_binary = FALSE,
  triplestore
)

```

Arguments

| | |
|------------------|---|
| transcriptome_id | Integer or list of integer values. |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, |

TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed.

phenotype_binary Logical value (TRUE/FALSE) to show/hide phenotype in binary notation (FALSE by default).

triplestore Object of class triplestore_access which manages database access.

Value

Data frame. Columns: "seed_id" (optional), "transcriptome_id", "phenotype_id", "phenotype_binary" (optional)

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single transcriptome
get_phenotype_id_from_transcriptome_id(
  transcriptome_id = 53674,
  triplestore = avidaDB
)

# More than one transcriptome
get_phenotype_id_from_transcriptome_id(
  transcriptome_id = c(53674, 1666099),
  phenotype_binary = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_3
get_phenotype_id_from_transcriptome_id(
  transcriptome_id = 53674, seed_id = c(1,3),
  triplestore = avidaDB
)
```

get_tandem_id_from_genome_id

Get tandem repeat from genome

Description

Get the tandem repeat contained in the transcriptome of a digital organism having a specific genome.

Usage

```
get_tandem_id_from_genome_id(
  genome_id,
  seed_id = FALSE,
  tandem_seq = FALSE,
  tandem_pos = FALSE,
  triplestore
)
```

Arguments

| | |
|--------------------------|---|
| <code>genome_id</code> | Integer or a list of integer values. |
| <code>seed_id</code> | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| <code>tandem_seq</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>tandem_pos</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_id", "tandem_id", "tandem_seq" (optional), "tandem_pos" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single genome
get_tandem_id_from_genome_id(genome_id = 1, triplestore = avidaDB)

# More than one genome
get_tandem_id_from_genome_id(
  genome_id = c(1, 2, 3),
```



```

    tandem_seq = TRUE,
    triplestore = avidaDB
  )

# At seed_1, seed_3 and seed_5
get_tandem_id_from_genome_id(
  genome_id = 2,
  seed_id = c(1, 3, 5),
  tandem_pos = TRUE,
  triplestore = avidaDB
)

```

```
get_tandem_id_from_genome_seq
```

Get tandem repeat from genome sequence

Description

Get the tandem repeat contained in the transcriptome of a digital organism having a specific linear string of letters representing the instruction codes that make up its genome.

Usage

```

get_tandem_id_from_genome_seq(
  genome_seq,
  seed_id = FALSE,
  tandem_seq = FALSE,
  tandem_pos = FALSE,
  triplestore
)

```

Arguments

| | |
|-------------|---|
| genome_seq | String of letters or a list of strings. |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| tandem_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| tandem_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Column: "seed_id" (optional), "genome_id", "tandem_id", "tandem_seq" (optional), "tandem_pos" (optional)."

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Get sequences for genomes 1 and 2
sequence1 <- get_genome_seq_from_genome_id(
  1,
  triplestore = avidaDB
)$genome_seq

sequence2 <- get_genome_seq_from_genome_id(
  2,
  triplestore = avidaDB
)$genome_seq

# Single genome
get_tandem_id_from_genome_seq(
  genome_seq = sequence1,
  triplestore = avidaDB
)

# More than one genome
get_tandem_id_from_genome_seq(
  genome_seq = c(sequence1, sequence2),
  tandem_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_tandem_id_from_genome_seq(
  genome_seq = sequence2,
  seed_id = c(1,2),
  tandem_seq = TRUE,
  tandem_pos = TRUE,
  triplestore = avidaDB
)
```

`get_tandem_id_from_logic_operation`

Get tandem repeat from logic operations

Description

Get the tandem repeat contained in the transcriptome of a digital organism that executes a specific combination of logic operations.

Usage

```
get_tandem_id_from_logic_operation(
  logic_operation,
  seed_id = sample(1:1000, 1),
  tandem_seq = FALSE,
  tandem_pos = FALSE,
  triplestore
)
```

Arguments

| | |
|-----------------|---|
| logic_operation | List of logical functions from the following set: "equals", "exclusive or", "not-or", "and-not", "or", "orn-not", "and", "not-and", "not". |
| seed_id | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If seed_id value is not specified, it returns data for a single randomly chosen seed_id value (between 1 and 1000). |
| tandem_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| tandem_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "tandem_id", "tandem_seq" (optional), "tandem_pos" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single logic operation
get_tandem_id_from_logic_operation(
  logic_operation = "not",
  triplestore = avidaDB
```

```

)

# More than one logic operation
get_tandem_id_from_logic_operation(
  logic_operation = c("not", "and"),
  tandem_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_tandem_id_from_logic_operation(
  logic_operation = c("not", "and"),
  tandem_seq = TRUE,
  tandem_pos = TRUE,
  seed_id = c(1,2),
  triplestore = avidaDB
)

```

```
get_tandem_id_from_phenotype_id
```

Get tandem repeat from phenotype

Description

Get the tandem repeat contained in the transcriptome of a digital organism that encodes a specific phenotype for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_tandem_id_from_phenotype_id(
  phenotype_id,
  seed_id = sample(1:1000, 1),
  tandem_seq = FALSE,
  tandem_pos = FALSE,
  triplestore
)

```

Arguments

| | |
|--------------|---|
| phenotype_id | Integer or list of integer values. |
| seed_id | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If seed_id value is not specified, it returns data for a single randomly chosen seed_id value (between 1 and 1000). |
| tandem_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| tandem_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Column: "seed_id" (optional), "phenotype_id", "tandem_id", "tandem_seq" (optional), "tandem_pos" (optional)."

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single phenotype
get_tandem_id_from_phenotype_id(
  phenotype_id = 8, tandem_seq = TRUE,
  triplestore = avidaDB
)

# More than one phenotype at seed_1
get_tandem_id_from_phenotype_id(
  phenotype_id = c(2, 4, 8), seed_id = 1,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_tandem_id_from_phenotype_id(
  phenotype_id = 1,
  seed_id = c(1, 2),
  tandem_pos = TRUE,
  triplestore = avidaDB
)
```

get_tandem_seq_from_tandem_id

Get the tandem repeat sequence from tandem repeat

Description

Get the tandem sequence from the tandem repeat contained in the transcriptome of a digital organism for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_tandem_seq_from_tandem_id(
  tandem_id,
  seed_id = FALSE,
  genome_id = FALSE,
  tandem_pos = FALSE,
  triplestore
)
```

Arguments

| | |
|-------------|---|
| tandem_id | Integer or a list of integer values. |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| genome_id | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| tandem_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "tandem_id", "tandem_seq", "tandem_pos" (optional), "genome_id" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single tandem
get_tandem_seq_from_tandem_id(
  tandem_id = 6336945,
  triplestore = avidaDB
)

# More than one tandem at seed_1
get_tandem_seq_from_tandem_id(
  tandem_id = c(6336945, 2520963, 2520963),
  seed_id = 1,
```

```

    triplestore = avidaDB
  )

# At seed_3 and seed_5
get_tandem_seq_from_tandem_id(
  tandem_id = 6336945,
  seed_id = c(1, 3),
  tandem_pos = TRUE,
  genome_id = TRUE,
  triplestore = avidaDB
)

```

```
get_transcriptome_id_from_genome_id
```

Get transcriptome from genome

Description

Get the transcriptome of a digital organism having a specific genome for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_transcriptome_id_from_genome_id(
  genome_id,
  seed_id = FALSE,
  transcriptome_seq = FALSE,
  transcriptome_pos = FALSE,
  genome_seq = FALSE,
  triplestore
)

```

Arguments

| | |
|-------------------|---|
| genome_id | Integer or list of integer values. |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| transcriptome_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| transcriptome_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| genome_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_id", "genome_seq" (optional), "transcriptome_id", "transcriptome_seq" (optional), "transcriptome_pos" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Singel genome
get_transcriptome_id_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)

# More than one genome
get_transcriptome_id_from_genome_id(
  genome_id = c(1, 2),
  transcriptome_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_3
get_transcriptome_id_from_genome_id(
  genome_id = 2,
  seed_id = c(1, 3),
  transcriptome_pos = TRUE,
  triplestore = avidaDB
)
```

```
get_transcriptome_id_from_genome_seq
```

Get transcriptome from genome sequence

Description

Get the transcriptome of a digital organism having a specific linear string of letters representing the instruction codes that make up its genome for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_transcriptome_id_from_genome_seq(
  genome_seq,
  seed_id = FALSE,
  transcriptome_seq = FALSE,
  transcriptome_pos = FALSE,
  genome_id = FALSE,
  triplestore
)
```

Arguments

| | |
|-------------------|---|
| genome_seq | String of letters or a list of strings. |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| transcriptome_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| transcriptome_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| genome_id | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "genome_seq", "transcriptome_id", "transcriptome_seq" (optional), "transcriptome_pos" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Get sequences for genome_1 and genome_2
sequence1 <- get_genome_seq_from_genome_id(
  genome_id = 1,
  triplestore = avidaDB
)$genome_seq[1]
```

```

sequence2 <- get_genome_seq_from_genome_id(
  genome_id = 2,
  triplestore = avidaDB
)$genome_seq[1]

# Single genome
get_transcriptome_id_from_genome_seq(
  genome_seq = sequence1,
  triplestore = avidaDB
)

# More than one genome
get_transcriptome_id_from_genome_seq(
  genome_seq = c(sequence1, sequence2),
  transcriptome_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_transcriptome_id_from_genome_seq(
  genome_seq = sequence2,
  seed_id = c(1,2),
  transcriptome_seq = TRUE,
  transcriptome_pos = TRUE,
  triplestore = avidaDB
)

```

```

get_transcriptome_id_from_logic_operation
  Get transcriptome from logic operations

```

Description

Get the transcriptome of a digital organism that executes a specific combination of logic operations for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_transcriptome_id_from_logic_operation(
  logic_operation,
  seed_id = sample(1:1000, 1),
  transcriptome_seq = FALSE,
  transcriptome_pos = FALSE,
  triplestore
)

```

Arguments

| | |
|-------------------|---|
| logic_operation | List of logical functions from the following set: "equals", "exclusive or", "not-or", "and-not", "or", "orn-not", "and", "not-and", "not". |
| seed_id | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If seed_id value is not specified, it returns data for a single randomly chosen seed_id value (between 1 and 1000). |
| transcriptome_seq | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| transcriptome_pos | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "transcriptome_id", "transcriptome_seq" (optional), "transcriptome_pos" (optional)

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single logic operation
get_transcriptome_id_from_logic_operation(
  logic_operation = "not",
  triplestore = avidaDB
)

# More than one logic operation
get_transcriptome_id_from_logic_operation(
  logic_operation = c("not", "and"),
  transcriptome_seq = TRUE,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_transcriptome_id_from_logic_operation(
  logic_operation = c("not", "and"),
  seed_id = c(1,2),
  transcriptome_seq = TRUE,
```

```

transcriptome_pos = TRUE,
triplestore = avidaDB
)

```

```
get_transcriptome_id_from_phenotype_id
```

Get transcriptome from phenotype

Description

Get the transcriptome of a digital organism whose genome encodes a specific phenotype for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```

get_transcriptome_id_from_phenotype_id(
  phenotype_id,
  seed_id = sample(1:1000, 1),
  transcriptome_seq = FALSE,
  transcriptome_pos = FALSE,
  phenotype_binary = FALSE,
  triplestore
)

```

Arguments

| | |
|--------------------------------|---|
| <code>phenotype_id</code> | Integer or list of integer values. |
| <code>seed_id</code> | Integer (from 1 to 1000) or a vector of integer values. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If <code>seed_id</code> value is not specified, it returns data for a single randomly chosen <code>seed_id</code> value (between 1 and 1000). |
| <code>transcriptome_seq</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>transcriptome_pos</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>phenotype_binary</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

Data frame. Columns: "seed_id" (optional), "transcriptome_id", "transcriptome_seq" (optional), "transcriptome_pos" (optional), "phenotype_id", "phenotype_binary" (optional).

Examples

```
# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single phenotype
get_transcriptome_id_from_phenotype_id(
  phenotype_id = 1,
  transcriptome_seq = TRUE,
  triplestore = avidaDB
)

# More than one phenotype at seed_1
get_transcriptome_id_from_phenotype_id(
  phenotype_id = c(1, 2), seed_id = 1,
  triplestore = avidaDB
)

# At seed_1 and seed_2
get_transcriptome_id_from_phenotype_id(
  phenotype_id = 1,
  seed_id = c(1, 2),
  transcriptome_pos = TRUE,
  triplestore = avidaDB
)
```

get_transcriptome_seq_from_transcriptome_id

Get transcriptome sequence from transcriptome

Description

Get the transcriptome sequence of the transcriptome executed by a digital organism for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
get_transcriptome_seq_from_transcriptome_id(
  transcriptome_id,
  seed_id = FALSE,
  transcriptome_pos = FALSE,
```

```

    genome_seq = FALSE,
    triplestore
  )

```

Arguments

| | |
|--------------------------------|---|
| <code>transcriptome_id</code> | Integer or list of integer values. |
| <code>seed_id</code> | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| <code>transcriptome_pos</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>genome_seq</code> | Logical value (TRUE/FALSE) to show/hide this column (FALSE by default). |
| <code>triplestore</code> | Object of class <code>triplestore_access</code> which manages database access. |

Value

data frame. Columns: "seed_id" (optional), "transcriptome_id", "transcriptome_seq", "transcriptome_pos" (optional), "genome_seq" (optional).

Examples

```

# Create triplestore object
avidaDB <- triplestore_access$new()

# Set access options
avidaDB$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_avida",
  password = "public_avida",
  repository = "avidaDB_test"
)

# Single transcriptome
get_transcriptome_seq_from_transcriptome_id(
  transcriptome_id = 53674,
  triplestore = avidaDB
)

# More than one transcriptome at seed_1
get_transcriptome_seq_from_transcriptome_id(
  transcriptome_id = c(53674, 1666099),
  seed_id = 1,
  triplestore = avidaDB
)

# At seed_1 and seed_3

```

```

get_transcriptome_seq_from_transcriptome_id(
  transcriptome_id = 2,
  seed_id = c(1, 3),
  transcriptome_pos = TRUE,
  triplestore = avidaDB
)

```

| | |
|-----------------|--|
| instruction_set | <i>Get the genetic language of Avida</i> |
|-----------------|--|

Description

List of the instruction codes comprising the genetic language of digital organisms in Avida.

Usage

```
instruction_set(inst_set = "heads")
```

Arguments

| | |
|----------|---|
| inst_set | Name of the instruction set. It must be one of the following: "heads" (default), "heads-sex", or transsmt. The names correspond to the instruction set configuration files (e.g., instset-heads.cfg for "heads"). |
|----------|---|

Value

Data frame. Columns: "instruction", "letter", "color"

| | |
|-----------------|---|
| logic_operation | <i>Get the list of logic operations that a digital organism can compute</i> |
|-----------------|---|

Description

List of the logic operations that a digital organism can execute.

Usage

```
logic_operation()
```

Value

Vector of character.

plot_transcriptome *Get a plot of the transcriptome as a chord diagram*

Description

Get a plot of the transcriptome executed by a digital organism for a list of seeds used for starting the pseudo-random number generator (i.e., a set of environments).

Usage

```
plot_transcriptome(
  transcriptome_id,
  seed_id = NULL,
  inst_set = "heads",
  save = FALSE,
  file_name = NULL,
  save_path = "~/transcriptome@chords",
  format = "svg",
  silent = FALSE,
  triplestore
)
```

Arguments

| | |
|------------------|---|
| transcriptome_id | Integer |
| seed_id | Integer (from 1 to 1000), a vector of integer values, or a logical value. This integer is used for starting the pseudo-random number generator that represents the environment experiencing a digital organism. If a logical value is used, TRUE returns data found in all environments and FALSE (by default) returns only distinct data regardless of the seed. |
| inst_set | Name of the instruction set. It must be one of the following: heads (default), heads-sex, or transsmt. The names correspond to the instruction set configuration files (e.g., instset-heads.cfg for heads). |
| save | Logical value (TRUE/FALSE) to save the plot (FALSE by default). |
| file_name | String of characters indicating the name of the file to be saved (without extension). |
| save_path | String of characters indicating the name of the folder where the file will be saved. |
| format | String of characters indicating the format of the file ("pdf" and "svg" are currently supported). |
| silent | Logical value (TRUE/FALSE) to show/hide messages (TRUE by default). |
| triplestore | Object of class triplestore_access which manages database access. |

Examples

```
# Create triplestore object
triplestore <- triplestore_access$new()

# Set access options
triplestore$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_aida",
  password = "public_aida",
  repository = "aidaDB_test"
)

# plot transcriptome 53674 at seed_1 and save to disk in pdf format
plot_transcriptome(
  transcriptome_id = 53674,
  seed_id = 1,
  save = FALSE,
  save_path = getwd(),
  format = "pdf",
  triplestore = triplestore
)
```

triplestore_access *Class to manage triplestore access options*

Description

Class to manage triplestore access options

Methods

Public methods:

- `triplestore_access$new()`
- `triplestore_access$get_access_options()`
- `triplestore_access$set_access_options()`
- `triplestore_access$submit_query()`
- `triplestore_access$ontology()`
- `triplestore_access$clone()`

Method `new()`: Create and initialize the object.

Usage:

```
triplestore_access$new()
```

Returns: Object of class `triplestore_access`.

Method `get_access_options()`: Get access options

Usage:

```
triplestore_access$get_access_options()
```

Returns: list containing URL of the API server, user credentials, repository name, authentication status, and SPARQL protocol version

Method `set_access_options()`: Set authentication access options for graphdb triplestore

Usage:

```
triplestore_access$set_access_options(
  url = NULL,
  user = NULL,
  password = NULL,
  repository = NULL,
  timeout = 100
)
```

Arguments:

`url` String containing the URL of the triplestore server

`user` String containing the username if authentication is needed

`password` String containing the password if authentication is needed

`repository` String containing the ID of the repository to which you want to connect to

`timeout` Connection timeout limit in seconds used for queries

Method `submit_query()`: Submit a SPARQL query to the triplestore to obtain data

Usage:

```
triplestore_access$submit_query(query)
```

Arguments:

`query` String containing the SPARQL query to retrieve data

Method `ontology()`: Show ontology information

Usage:

```
triplestore_access$ontology()
```

Returns: List containing title, description and versionIRI of the ontology

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
triplestore_access$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# Create object triplestore
```

```
triplestore <- triplestore_access$new()
```

```
# Set options to access a specific triple-store implemented in GraphDB

triplestore$set_access_options(
  url = "https://graphdb.fortunalab.org",
  user = "public_aida",
  password = "public_aida",
  repository = "aidaDB_test"
)

# Show current access options

triplestore$get_access_options()

# Querying data with SPARQL

triplestore$submit_query('PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
  select ?tandem_id where {
    ?digital_tandem_repeat rdfs:label "digital tandem repeat"@en .
    ?tandem_id a ?digital_tandem_repeat .
  } limit 10')
```

```
# Show ontology info

triplestore$ontology()
```

Index

convert_org_into_seq, [2](#)
convert_seq_into_org, [3](#)

get_db_summary, [5](#)
get_docker_image_from_experiment_id, [5](#)
get_doi_from_experiment_id, [6](#)
get_experiment_id_from_organism_id, [7](#)
get_genome_id_from_genome_seq, [8](#)
get_genome_id_from_logic_operation, [9](#)
get_genome_id_from_phenotype_id, [11](#)
get_genome_id_from_transcriptome_id,
[12](#)
get_genome_id_of_wild_type_organisms,
[13](#)
get_genome_seq_from_genome_id, [14](#)
get_logic_operation_from_phenotype_id,
[15](#)
get_mutant_at_pos, [16](#)
get_phenotype_id_from_genome_id, [18](#)
get_phenotype_id_from_genome_seq, [19](#)
get_phenotype_id_from_logic_operation,
[21](#)
get_phenotype_id_from_transcriptome_id,
[22](#)
get_tandem_id_from_genome_id, [23](#)
get_tandem_id_from_genome_seq, [25](#)
get_tandem_id_from_logic_operation, [26](#)
get_tandem_id_from_phenotype_id, [28](#)
get_tandem_seq_from_tandem_id, [29](#)
get_transcriptome_id_from_genome_id,
[31](#)
get_transcriptome_id_from_genome_seq,
[32](#)
get_transcriptome_id_from_logic_operation,
[34](#)
get_transcriptome_id_from_phenotype_id,
[36](#)
get_transcriptome_seq_from_transcriptome_id,
[37](#)

instruction_set, [39](#)
logic_operation, [39](#)
plot_transcriptome, [40](#)
triplestore_access, [41](#)