

Package ‘`agena.ai`’

June 20, 2023

Title R Wrapper for 'agena.ai' API

Version 1.1.0

Description An R wrapper for 'agena.ai' <<https://www.agena.ai>> which provides users capabilities to work with 'agena.ai' using the R environment. Users can create Bayesian network models from scratch or import existing models in R and export to 'agena.ai' cloud or local API for calculations.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports rjson, httr, openxlsx, Rgraphviz, methods, utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Erhan Pisirir [aut],
Eugene Dementiev [aut, cre],
Martin Neil [aut, cph]

Maintainer Eugene Dementiev <support@agenarisk.com>

Repository CRAN

Date/Publication 2023-06-20 21:40:02 UTC

R topics documented:

calculate	2
create_batch_cases	3
create_csv_template	3
create_sensitivity_config	4
Dataset-class	4
from_cmpx	5
generate_cmpx	5
local_api_activate_license	6
local_api_batch_calculate	6

local_api_calculate	7
local_api_clone	7
local_api_compile	8
local_api_sensitivity	8
login	9
Model-class	9
Network-class	11
Node-class	11
sensitivity_analysis	13

Index	14
--------------	-----------

calculate	<i>Calculate posterior probabilities in a BN on agena.ai cloud</i>
-----------	--

Description

A function to send an input Bayesian network model to Agena AI Cloud servers. Once called, the function will check authentication status, if it has not expired, it will send the POST request with the model to the servers, and receive calculation results to update the Bayesian network model (filling the results field with calculation results).

Usage

```
calculate(input_model, login, dataSet = NULL, debug = FALSE)
```

Arguments

input_model	an R model object
login	an agena.ai cloud login
dataSet	a dataSet in the R model object
debug	boolean parameter to display debug messages or not

Value

BN inference results in the model

create_batch_cases *Create a json file of a BN model with observations*

Description

This function takes an R Model object and a CSV file of observations for cases and creates a batch of datasets / cases for each row in the input data and generates a .json file for the BN

Usage

```
create_batch_cases(inputModel, inputData)
```

Arguments

inputModel	an R model object
inputData	CSV file of observations

Value

json file

create_csv_template *Create a CSV template for a model for batch observations*

Description

This function creates an empty CSV file with the correct format so that it can be filled in and used for create_batch_cases.

Usage

```
create_csv_template(inputModel)
```

Arguments

inputModel	an R model object
------------	-------------------

Value

csv file

```
create_sensitivity_config
```

Create configuration object for sensitivity analysis

Description

See agena.ai manual for sensitivity analysis configuration

Usage

```
create_sensitivity_config(
  target,
  sensitivity_nodes,
  dataset = NULL,
  network = NULL,
  report_settings = NULL
)
```

Arguments

target	target node in the network
sensitivity_nodes	a set of sensitivity nodes in the network
dataset	a dataSet in the model
network	a network in the model
report_settings	a list of report settings

Value

sensitivity config object

Dataset-class

Dataset

Description

Dataset

Fields

id dataSet id
 observations list of observations.
 results list of inference results.

`from_cmpx`*Load a Model from CMPX*

Description

This function reads an input agena.ai model file with the .cmpx extension to create an R model object. The model object includes the networks and the nodes of the imported model.

Usage

```
from_cmpx(modelPath)
```

Arguments

`modelPath` Path to the input file

Value

An R model object

`generate_cmpx`*Generate a CMPX file from R Model*

Description

This function generates a .cmpx file for an agena.ai model based on an R model object

Usage

```
generate_cmpx(inputModel)
```

Arguments

`inputModel` an R model object

Value

cmpx file

`local_api_activate_license`*Local agena.ai developer license activation*

Description

sends the agena.ai developer license for activation

Usage

```
local_api_activate_license(key)
```

Arguments

key agena.ai developer license key

Value

no return value, activates user's local API license

`local_api_batch_calculate`*Local agena.ai API model calculation for batch cases*

Description

Runs inference for every dataSet in the model

Usage

```
local_api_batch_calculate(model)
```

Arguments

model an R model object

Value

a json file for the model with results

local_api_calculate *Local agena.ai API model calculation*

Description

Local agena.ai API model calculation

Usage

```
local_api_calculate(model, dataSet, output)
```

Arguments

model	an R model object
dataSet	a dataSet in the model
output	file name for the output json - just the filename using the working directory, not a full file path

Value

a json file for the model with results

local_api_clone *Local agena.ai API clone*

Description

clones the local agena.ai developer API in the working directory

Usage

```
local_api_clone()
```

Value

no return value, used to clone local API to directory

local_api_compile *Local agena.ai API compilation*

Description

sets the version and compiles the maven environment for the local agena.ai developer API in the working directory

Usage

```
local_api_compile()
```

Value

no return value, used to compile maven in the local API directory

local_api_sensitivity *Local agena.ai API sensitivity analysis*

Description

Local agena.ai API sensitivity analysis

Usage

```
local_api_sensitivity(model, sens_config, output)
```

Arguments

model	an R model object
sens_config	a sensitivity config object
output	file name for the output json - just the filename using the working directory, not a full file path

Value

a json file report of the sensitivity analysis results

login	<i>Login to agena.ai cloud</i>
-------	--------------------------------

Description

Login to agena.ai cloud

Usage

```
login(username, password)
```

Arguments

username	agenta.ai cloud username
password	agenta.ai cloud password

Value

no return value, used in to authenticate user credentials

Model-class	<i>BN Model object</i>
-------------	------------------------

Description

BN Model object

Fields

id	model id
networks	list of networks in the model
dataSets	list of dataSets in the model
networkLinks	list of network links in the model
settings	list of settings of the model

Methods

`add_network(newNetwork)` A method to add a new Network object to the networks field of a Model object. The input newNetwork is a Network object and it is added to the model if it is not already in it.

`add_network_link(source_network, source_node, target_network, target_node, link_type, pass_state = NULL)`
 This is the method to add links to a model between its networks. These links start from a "source node" in a network and go to a "target node" in another network. To create the link, the source and target nodes in the networks need to be specified together with the network they belong to (by the Node and Network ids). See README.md for details.

- `change_settings(settings)` A method to change model settings. The input parameter settings must be a list with the correctly named elements, see README.md for example.
- `clear_all_observations()` A method to clear all observations defined in a model. This function removes all observations from all datasets (scenarios).
- `clear_dataSet_observations(dataSet)` A method to clear all observations in a specific dataset (scenario) in the model.
- `create_dataSet(id)` It is possible to add multiple scenarios to a model. These scenarios are new DataSet objects added to the datasets field of a model. Initially these scenarios have no observations and are only defined by their ids. The scenarios are populated with the `enter_observation()` function.
- `default_settings()` A method to reset model settings back to default values.
- `enter_observation(dataSet = NULL, node, network, value, variable_input = FALSE, soft_evidence = FALSE)`
A method to enter observation to a model. To enter the observation to a specific dataset (scenario), the dataset id must be given as the input parameter dataSet. If dataSet is left NULL, the entered observation will by default go to "Scenario 1". This means that if there is no extra datasets created for a model (which by default comes with "Scenario 1"), any observation entered will be set for this dataset. See README.md for details and examples.
- `get_dataSets()` A method to list the ids of all existing scenarios in a model.
- `get_networks()` A method to see ids of all the networks in a model.
- `get_results(filename = NULL)` A method to generate a .csv file based on the calculation results a Model contains.
- `import_results(result_file)` A method to import results of a calculated dataSet from a json file. This correct format for the results json file for this method is the file generated with the local agena.ai developer API calculation (see README.md for details).
- `initialize(id = NULL, networks, from_cmpx = FALSE, datasets = NULL, networkLinks = NULL, settings = NULL)`
Creates a new Model object, a list of networks (at least one) is required. id and settings are set with defaults unless specified.
- `remove_all_network_links()` A method to remove all existing network links in a model.
- `remove_dataSet(olddataSet)` A method to remove an existing scenario from the model. Input parameter olddataSet is the string which is the id of a dataset (scenario).
- `remove_network(oldNetwork)` A method to remove an existing Network object from the model. Note that removing a Node from a network does not automatically remove its possible network links to other networks in the model. networkLinks field of a Model should be adjusted accordingly if needed.
- `remove_network_link(source_network, source_node, target_network, target_node)` A method to remove network links, given the ids of the source and target nodes (and the networks they belong to).
- `remove_observation(dataSet = NULL, node, network)` A method to remove a specific observation from the model. It requires the id of the node which has the observation to be removed and the id of the network the node belongs to.
- `to_cmpx(filename = NULL)` A method to export the Model to a .cmpx file. This method passes on all the information about the model, its datasets, its networks, their nodes, and model settings to a .cmpx file in the correct format readable by agena.ai. If the input parameter filename is not specified, it will use the Model\$id for the filename.

`to_json(filename = NULL)` A method to export the Model to a .json file. This method passes on all the information about the model, its datasets, its networks, their nodes, and model settings to a .json file in the correct format readable by agena.ai. If the input parameter filename is not specified, it will use the Model\$id for the filename.

Network-class *BN Network object*

Description

These represent each network in a BN. Networks consist of nodes and in a BN model there might be more than one network. These networks can also be linked to each other with the use of input and output nodes. For such links, see Model\$networkLinks.

Fields

id network id
name network display name
description network description
nodes list of nodes in the network

Methods

`add_node(newNode)` A method to add new Node objects to a Network. Note that adding a new Node to the network does not automatically add its parents to the network. You need to add all the parents separately too.

`initialize(id, name = NULL, description = NULL, nodes = NULL)` Creates a new Network object, a unique id is required, other fields are filled with defaults unless specified.

`plot()` A method to plot the graphical structure of a BN network.

`remove_node(oldNode)` A method to remove an existing Node object from the network. Note that removing a Node from a network does not automatically remove it from its previous parent-child relationships in the network. You need to adjust such relationships separately on the Node level.

Node-class *BN Node object*

Description

These represent the nodes in a BN.

Fields

id node id
 name node display name
 description node description
 type node type
 parents node parent nodes
 simulated whether node is simulated
 distr_type node distribution type
 states node states
 probabilities node probabilities
 expressions node expressions
 partitions node expression partitions
 variables node variables or constants

Methods

addParent_byID(newParentID, varList) This is a method to add parent Nodes by their ids for cmpx parser capabilities. To add parents to Node objects, please use \$add_parent(Node) method.

add_parent(newParent) Adds a Node object as a new parent node to the current Node object and resets/resizes the NPT values and expressions of the Node as needed. Parents list of a Node object is a list of other Node objects. The input parameter of the function is a Node object variable. A good practice is to use Node ids as their variable names.

get_parents() A method to list all the existing parent nodes of a Node.

initialize(id, name = NULL, description = NULL, type = NULL, simulated = FALSE, states = NULL)
 Creates a new Node object, a unique id is required, other fields are filled with defaults unless specified. Node id, name, description, type, states, and whether it is a simulation or regular node do not depend on its edges and parents in a network, a Node object can be defined with all this information outside a Network as well. To add/remove/modify parents, distr_type, probabilities, expressions, and partitions; use the corresponding method.

remove_parent(oldParent) Removes a Node object from parents of the current Node object and resets/resizes the NPT values and expressions of the Node as needed. The input parameter of the function is a Node object variable. A good practice is to use Node ids as their variable names.

remove_variable(variable_name) A method to remove one of the existing variables (constants) from a node, using the variable_name.

set_distribution_type(new_distr_type) A method to set the table type (distr_type) of a node. If a Node is simulated, its table type can be "Expression" or "Partitioned" - the latter is only if the node has parent nodes. If a Node is not simulated, its table type can be "Manual", "Expression", or "Partitioned Expression (if the node has parent nodes)".

`set_expressions(new_expr, partition_parents = NULL)` The method to set the probability values if the table type (`distr_type`) of a Node is "Expression" or "Partitioned". If the table type is "Expression", `new_expr` is a single string and `partition_parents` is left NULL. If the table type is "Partitioned", `new_expr` is a list of expressions for each parent state, and `partition_parents` is a list of strings for each partitioned parent node id.

`set_probabilities(new_probs, by_rows = TRUE)` The method to set the probability values if the table type (`distr_type`) of a Node is "Manual". `new_probs` is a list of numerical values, and the length of the input list depends on the number of the states of the node and of its parents. You can format the input list in two different orders. If the parameter `by_rows` is set to true, the method will read the input list to fill in the NPT row by row; if set to false, the method will read the input list to fill in the NPT column by column. See README.md for examples.

`set_variable(variable_name, variable_value)` A method to set variables (constants) for a node. Takes the `variable_name` and `variable_value` inputs which define a new variable (constant) for the node.

sensitivity_analysis *Sensitivity analysis on agena.ai cloud*

Description

Sensitivity analysis on agena.ai cloud

Usage

```
sensitivity_analysis(input_model, login, sensitivity_config, debug = FALSE)
```

Arguments

<code>input_model</code>	an R model object
<code>login</code>	an agena.ai cloud login
<code>sensitivity_config</code>	a sensitivity analysis config object
<code>debug</code>	boolean parameter to display debug messages or not

Value

sensitivity analysis report

Index

calculate, [2](#)
create_batch_cases, [3](#)
create_csv_template, [3](#)
create_sensitivity_config, [4](#)

Dataset (Dataset-class), [4](#)
Dataset-class, [4](#)

from_cmpx, [5](#)

generate_cmpx, [5](#)

local_api_activate_license, [6](#)
local_api_batch_calculate, [6](#)
local_api_calculate, [7](#)
local_api_clone, [7](#)
local_api_compile, [8](#)
local_api_sensitivity, [8](#)
login, [9](#)

Model (Model-class), [9](#)
Model-class, [9](#)

Network (Network-class), [11](#)
Network-class, [11](#)
Node (Node-class), [11](#)
Node-class, [11](#)

sensitivity_analysis, [13](#)