Bio2BEL Reactome Documentation

Release 0.2.4-dev

Daniel Domingo-Fernández and Charles Tapley Hoyt

Nov 18, 2020

CONTENTS:

Bio2BEL Reactome is a package for enriching BEL networks with Reactome information by wrapping its RESTful API.

Reactome is a pathway database comprising established pathways between different species that contain genetic as well as chemical information. This package downloads pathway information from Reactome's API and store it in template data model relating genes and chemical to pathways. Moreover, the hierarchy of the pathways is maintained enabling pathway comparison and exploration in the ComPath environment.

CHAPTER

ONE

CITATION

- Fabregat, Antonio et al. The Reactome Pathway Knowledgebase. Nucleic Acids Research 44.Database issue (2016): D481–D487. PMC. Web. 6 Oct. 2017.
- Croft, David et al. The Reactome Pathway Knowledgebase. Nucleic Acids Research 42.Database issue (2014): D472–D477. PMC. Web. 6 Oct. 2017.

CHAPTER

TWO

MANAGER

Database Manager and query functions

This module populates the tables of bio2bel_reactome.

class bio2bel_reactome.manager.**Manager**(**args*, ***kwargs*)

Protein-pathway and chemical-pathway memberships.

Doesn't let this class get instantiated if the pathway_model.

protein_model

alias of *bio2bel_reactome.models.Protein*

namespace_model

alias of *bio2bel_reactome.models.Pathway*

pathway_model

alias of *bio2bel_reactome.models.Pathway*

summarize() → Mapping[str, int]

Summarize the database.

count_chemicals() → int

Count the chemicals in the database.

count_species() → int

Count the species in the database.

get_gene_sets(*only_human: bool = False*) → Mapping[str, Set[str]]

Return pathway - genesets mapping.

get_or_create_pathway(*, *reactome_id: str*, *name: str*, *species:* bio2bel_reactome.models.Species, *chemicals: Optional[List[*bio2bel_reactome.models.Chemical*]]*) → *bio2bel_reactome.models.Pathway*

Get a pathway from the database or creates it.

Parameters

- **reactome_id** – pathway identifier
- **name** – name of the pathway
- **species** – Species object
- **chemicals** – An optional list of chemicals that belong too this pathway

get_or_create_chemical(*, *chebi_id: str*, *chebi_name: str*) → *bio2bel_reactome.models.Chemical*

Get a Chemical from the database or creates it.

Parameters

- **chebi_id** – ChEBI identifier
- **chebi_name** – ChEBI name

get_or_create_species(*, *taxonomy_id: str*, *name: str*) → *bio2bel_reactome.models.Species*
Get a Species from the database or creates it.

get_or_create_protein(*uniprot_id: str*, *hgnc_symbol: Optional[str] = None*, *hgnc_id: Optional[str] = None*) → *bio2bel_reactome.models.Protein*
Get a protein from the database or creates it.

Parameters

- **uniprot_id** – pathway identifier
- **hgnc_symbol** – name of the pathway
- **hgnc_id** – Species object

get_species_by_name(*species_name: str*) → Optional[*bio2bel_reactome.models.Species*]
Get a Species by its species_name.

Parameters **species_name** – name

get_pathway_names_to_ids(*only_human: bool = False*)
Return a dictionary of pathway names to ids.

Return type dict[str,str]

get_pathway_parent_by_id(*reactome_id: str*) → Optional[*bio2bel_reactome.models.Pathway*]
Get parent pathway by its reactome id.

Parameters **reactome_id** – reactome identifier

get_top_hiearchy_parent_by_id(*reactome_id: str*) → Optional[*bio2bel_reactome.models.Pathway*]
Get the oldest pathway at the top of the hierarchy a pathway by its reactome id.

Parameters **reactome_id** – reactome identifier

get_all_top_hierarchy_pathways() → List[*bio2bel_reactome.models.Pathway*]
Get all pathways without a parent (top hierarchy).

get_human_pathways() → List[*bio2bel_reactome.models.Pathway*]
Get human pathways.

get_pathways_by_species(*species_name: str*) → Optional[List[*bio2bel_reactome.models.Pathway*]]
Get pathways by species.

get_chemical_by_chebi_id(*chebi_id: str*) → Optional[*bio2bel_reactome.models.Chemical*]
Get chemical by ChEBI id.

get_protein_by_uniprot_id(*uniprot_id: str*) → Optional[*bio2bel_reactome.models.Protein*]
Get protein by UniProt id.

populate(*pathways_path: Optional[str] = None*, *pathways_hierarchy_path: Optional[str] = None*, *pathways_proteins_path: Optional[str] = None*, *pathways_chemicals_path: Optional[str] = None*) → None
Populate all tables.

Parameters

- **pathways_path** – url from pathway table file
- **pathways_hierarchy_path** – url from pathway hierarchy file

- **pathways_proteins_path** – url from pathway protein file
- **pathways_chemicals_path** – url from pathway chemical file

CHAPTER

THREE

COMMAND LINE INTERFACE

The command line interface allows you to communicate with the package and perform basic functions such as:

- Populate the database: `python3 -m bio2bel_reactome populate`. By default this command populates the database only with human information. In order to populate all species pathway information you can add the "–not-only-human" argument. By default the database is reset every time is populated. However, another optional parameter "–reset-db=False", allows you to avoid the reset. More logging can be activated by added "-vv" or "-v" as an argument.
- Drop the database: `python3 -m bio2bel_reactome drop`. More logging can be activated by added "-vv" or "-v" as an argument.
- Export gene sets as an excel file: `python3 -m bio2bel_reactome export`. By default, the excel will contain all pathways from all species. However, you can add the argument "species" and type the name of a particular one to get only those pathways (e.g., "–species='Homo sapiens'""). Since Reactome has a hierarchy pathway structure, you can get only the major pathways with the optional parameter "–top-hierarchy".

CHAPTER

FOUR

CONSTANTS

This module contains all the constants used in this package.

This module contains all the constants used in bio2bel Reactome project.

CHAPTER

FIVE

MODELS

Database models.

Reactome database model.

class bio2bel_reactome.models.**Species**(***kwargs*)
Species Table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

name
NCBI taxonomy label

taxonomy_id
NCBI taxonomy identifier

class bio2bel_reactome.models.**Protein**(***kwargs*)
Protein Table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

to_pybel() → pybel.dsl.node_classes.Protein
Serialize to PyBEL node data dictionary.

class bio2bel_reactome.models.**Chemical**(***kwargs*)
Chemical Table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

to_pybel() → pybel.dsl.node_classes.Abundance
Serialize to PyBEL node data dictionary.

class bio2bel_reactome.models.**Pathway**(***kwargs*)
A reactome pathway.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

CHAPTER

SIX

WEB

This module contains the web application to explore the database

This module contains the flask-admin application to visualize the db.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

INDEX

S

T