
insilico Documentation

felipe zapata

Dec 05, 2018

Contents:

1	Installation	1
2	Virtual Screening	3
3	Tutorial	5
4	Library API	7
4.1	Input validation	7
4.2	Tasks	7
4.3	Graph of Dependencies	8
5	Indices and tables	9

CHAPTER 1

Installation

1. Install `miniconda`.
2. Install `rdkit` and `dask` using the following commands:
 - `conda install -c rdkit rdkit`
 - `conda install dask`
3. Install the `insilico` library using the following command:
 - `pip install git+https://github.com/SCM-NV/filterInsilico@master#egg=insilico-0.0.1`

CHAPTER 2

Virtual Screening

Virtual screening allows to filter a set of molecules based on their physico-chemical properties from a potentially large number of candidates.

CHAPTER 3

Tutorial

4.1 Input validation

`insilico.validate_input.validate_input` (*input_file*: *str*) → Dict

Read the *input_file* in YAML format, validate it against the corresponding schema and return a nested dictionary with the input.

Parameters `input_file` (*str*) – path to the input

Returns Input as dictionary

Raises `SchemaError` – If the input is not valid

4.2 Tasks

`insilico.filters.apply_filter` (*filters*: Dict, *molecules*: *pandas.core.frame.DataFrame*, *dependencies*: Dict = None) → *pandas.core.frame.DataFrame*

Apply a different set of *filters* to a molecular set.

Parameters

- **filters** (*dict*) – Set of predicates to filter the molecules
- **molecules** – Pandas DataFrame containing the properties
- **dependencies** (*dict*) – Current task parent

Returns Pandas Dataframe

`insilico.properties.compute_property` (*molecular_properties*: Dict, *molecules*: *pandas.core.frame.DataFrame*, *dependencies*: Dict = None) → *pandas.core.frame.DataFrame*

Calculate a set of *molecular_properties*.

Parameters

- **molecular_properties** (*dict*) – Properties to compute
- **molecules** – Pandas DataFrame containing the properties
- **dependencies** (*dict*) – Current task parent

Returns Pandas Dataframe

`insilico.properties.search_property` (*molecular_properties: List, molecules: pandas.core.frame.DataFrame, dependencies: Dict = None*) → `pandas.core.frame.DataFrame`

Search for a set of *molecular_properties* in the pubchem database.

Parameters

- **molecular_properties** (*dict*) – Properties to look at
- **molecules** – Pandas DataFrame containing the properties
- **dependencies** (*dict*) – Current task parent

Returns Pandas Dataframe

4.3 Graph of Dependecies

`insilico.runner.build_graph` (*steps: Dict, state: pandas.core.frame.DataFrame*) → `Dict`

Create a Direct Acyclic Graph containing all the dependencies between the filters and the properties to compute.

Parameters

- **steps** (*dict*) – Task to perform
- **state** – Current DataFrame used as state

Returns Dictionary representing the graph of dependencies

Raises *DependencyError* – if the dependencies are incongruent

`insilico.runner.runner` (*dag: object*) → `pandas.core.frame.DataFrame`

Run the Direct Acyclic Graph containing all the filters and properties.

Returns Pandas DataFrame containing the results

class `insilico.runner.DependencyError`

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

A

`apply_filter()` (in module `insilico.filters`), 7

B

`build_graph()` (in module `insilico.runner`), 8

C

`compute_property()` (in module `insilico.properties`), 7

D

`DependencyError` (class in `insilico.runner`), 8

R

`runner()` (in module `insilico.runner`), 8

S

`search_property()` (in module `insilico.properties`), 8

V

`validate_input()` (in module `insilico.validate_input`), 7