biothings_explorer Documentation

Release 0.0.1

Jiwen Xin

Aug 18, 2022

Contents

Obsolete repo

Note that this repo is for the python client for BioThings Explorer. It is not currently being actively developed. The primary development effort on the BioThings Explorer project is at https://github.com/biothings/BioThings_Explorer_TRAPI.

CHAPTER 2

Introduction

BioThings Explorer is an engine for autonomously querying a distributed knowledge graph. The distributed knowledge graph is made up of biomedical APIs that have been annotated with semantically-precise descriptions of their inputs and outputs. The knowledge graph is currently comprised by the APIs in this figure:
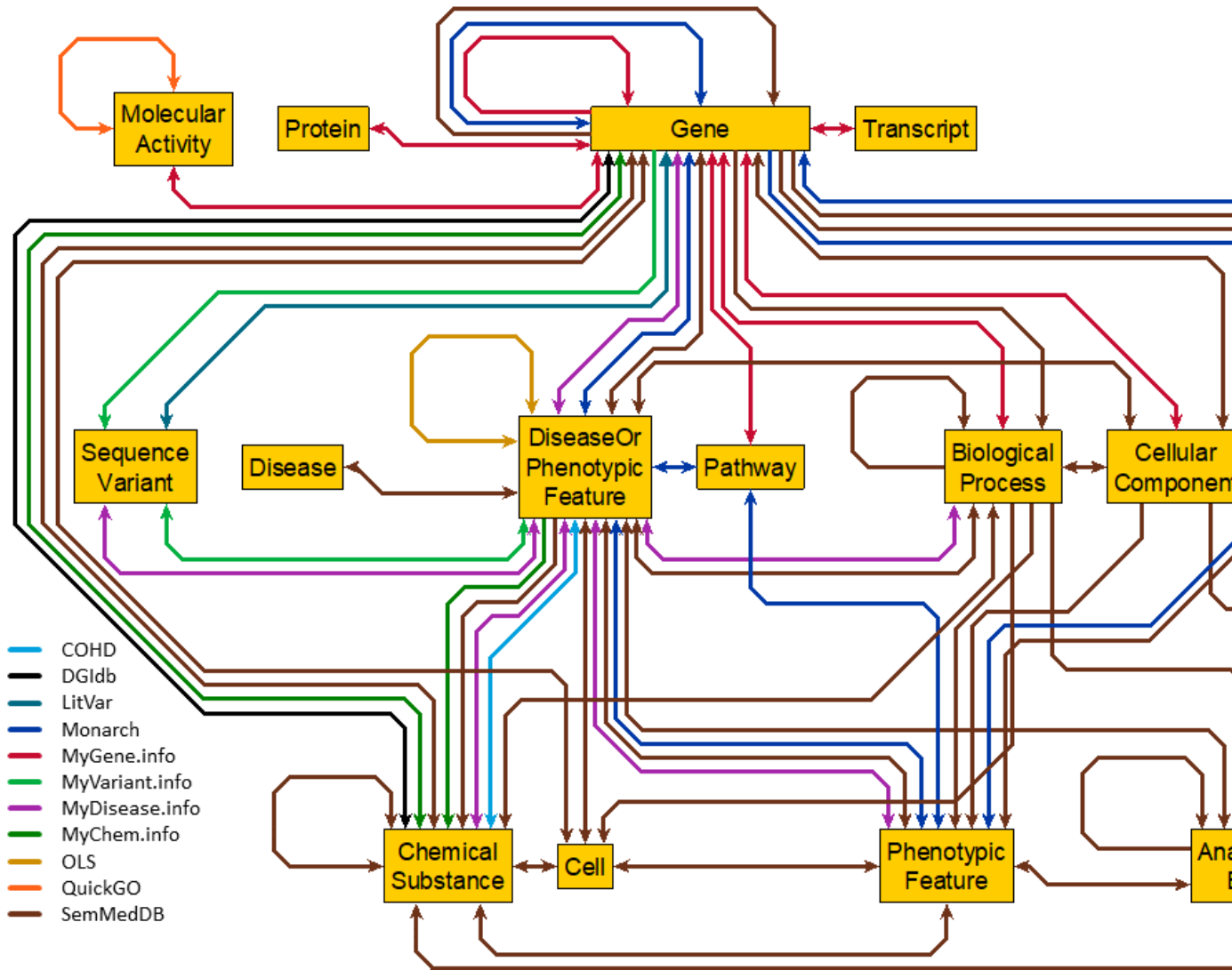
Installation

Install using pip:

```
pip install git+https://github.com/biothings/biothings_explorer#egg=biothings_explorer
```

QuickStart

4.1 Quickstart

Eager to get started? This page gives a good introduction in how to get started with BioThings Explorer.

First, make sure that: biothings_explorer is installed.

Let's get started with some simple examples.

4.1.1 Find all chemicals related to a gene

- Begin by importing the biothings_explorer modules

```
>>> from biothings_explorer.hint import Hint
>>> from biothings_explorer.user_query_dispatcher import FindConnection
```

- Find representation of "CXCR4" Gene in BioThings Explorer

```
>>> ht = Hint()
>>> cxcr4_hint = ht.query("cxcr4")
>>> cxcr4 = cxcr4_hint['Gene'][0]
>>> cxcr4
```

- Now we have the representation of cxcr4 in BTE, let's go ahead and find all chemicals that are related to cxcr4

```
>>> fc = FindConnection(input_obj=cxcr4, output_obj='ChemicalSubstance', intermediate_
→nodes=None)
>>> fc.connect(verbose=True)
```

- So far, BTE has queried all APIs which could connects from CXCR4 to chemical substances. Now we could go ahead and explore the results, e.g. as a pandas data frame, as a graphml file, a reasoner API standard JSON file

```
>>> fc.display_table_view()
>>> fc.to_graphml('cxcr4_chemical.graphml')
>>> fc.to_reasoner_std()
```

4.1.2 Find all chemicals that target genes that are associated with gene PRDX1

This is a more complicated example with intermediate nodes involved. It requires:

1. First, find all genes that associate with PRDX1

2. Second, find all chemicals that target these genes

- Begin by importing the biothings_explorer modules

```
>>> from biothings_explorer.hint import Hint
>>> from biothings_explorer.user_query_dispatcher import FindConnection
```

- Find representation of "PRDX1" Gene in BioThings Explorer

```
>>> ht = Hint()
>>> prdx1_hint = ht.query("PRDX1")
>>> prdx1 = prdx1_hint['Gene'][0]
>>> prdx1
```

- Now we have the representation of PRDX1 in BTE, let's go ahead and find all chemicals that target genes which associate with PRDX1

```
>>> fc = FindConnection(input_obj=prdx1, output_obj='ChemicalSubstance', intermediate_
→nodes=['Gene'])
>>> fc.connect(verbose=True)
```

- So far, BTE has queried all APIs which could connects from PRDX1 to genes, then to chemical substances. Now we could go ahead and explore the results, e.g. as a pandas data frame, as a graphml file, a reasoner API standard JSON file

```
>>> fc.display_table_view()
>>> fc.to_graphml('prdx1_chemical.graphml')
>>> fc.to_reasoner_std()
```

- full jupyer notebook demo is available at the link below

Finding Marketed Drugs that Might Treat an Unknown Syndrome by Perturbing the Disease Mechanism Pathway

4.1.3 More advanced usages

- For more complicated examples, please check out jupyter notebooks at:

1. Finding New Uses for Existing Drugs to Treat Parkinson's Disease.ipynb

2. What drugs might be used to treat hyperphenylalaninemia?

3. Why does imatinib have an effect on the treatment of chronic myelogenous leukemia (CML)?

The API Documentation / Guide

If you are looking for information on a specific function, class, or method, this part of the documentation is for you.

5.1 Developer Interface

This part of the documentation covers all the interfaces of BioThings Explorer. For parts where Requests depends on external libraries, we document the most important right here and provide links to the canonical documentation.

5.1.1 Query Module

class biothings_explorer.user_query_dispatcher.**FindConnection**(*input_obj*, *output_obj*, *intermediate_nodes*, *registry=None*)

Find relationships between one specific entity and another specific entity or other classes of entity types.

Args:

input_obj (required): must be an object returned from Hint corresponding to a specific biomedical entity.

Examples:

Hint().query("Fanconi anemia")['DiseaseOrPhenotypicFeature'][0] Hint().query("acetaminophen")['ChemicalSubstance'][0]

output_obj (required): must EITHER be an object returned from Hint corresponding to a specific biomedical

entity, OR be a string or list of strings corresponding to Biolink Entity classes. Examples:

Hint().query("acetaminophen")['ChemicalSubstance'][0] 'Gene' ['Gene','ChemicalSubstance']

intermediate_nodes (required): the semantic type(s) of the intermediate node(s). Examples:
None : no intermediate node, find direct connections only [] : no intermediate node, find direct connections only ['BiologicalEntity'] : one intermediate node of any semantic type ['Gene'] : one intermediate node that must be a Gene [('Gene','Pathway')] : one intermediate node that must be a Gene or a Pathway ['Gene','Pathway'] : two intermediate nodes, first must be a Gene, second must be a Pathway. ['Gene',('Pathway','Gene')] : two intermediate nodes, first must be a Gene, second must be a Pathway or Gene.

NOTE: queries with more than one intermediate node are currently not supported

NOTE: queries with more than one intermediate node are currently not supported

display_edge_info(*start_node*, *end_node*)
Display detailed edge info between start node and end node.

Parameters

- start_node (str): start node id

- end_node (str): end node id

display_node_info(*node*)
Show detailed node information.

Parameters

- node (str): node id

display_table_view()
Display the query results as a pandas table.

Examples

```
>>> df = fc.display_table_view()
>>> df
```

show_all_edges()
Show all edges in the graph.

show_all_nodes()
Show all nodes in the graph.

to_graphml(*path*)
Convert the output to graphml format.

parameters

- path (str): the file path to store the graphml file

to_json()
Convert the graph into JSON through networkx.

to_reasoner_std()
Convert the output to reasoner api standard.

5.1.2 Hint Module

class biothings_explorer.hint.**Hint**
Resolving Biomedical Identifiers through BioThings APIs.

query(*_input*, *semantic_type=None*)
Main function to resolve identifiers.

Param _input: any string input

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

b

Index