pybel-tools Documentation

Release 0.1.16-dev

Charles Tapley Hoyt

Jul 20, 2017

Getting Started

PyBEL Tools is a suite of tools built on top of PyBEL to facilitate data management, integration, and analysis. For further examples, see the PyBEL Notebooks repository.

CHAPTER 1

Links

- Documented on Read the Docs
- Versioned on GitHub
- Tested on Travis CI
- Distributed by PyPI
- Chat on Gitter

Installation

Cookbook

Pickling lots of BEL scripts

All of the BEL scripts in the current working directory (and sub-directories) can be pickled in-place with the following command (add `-d` to specify a different directory)

```
$ python3 -m pybel_tools io convert -d ~/bms/aetionomy/
```

Getting Data in to the Cache

Before running the service, some data can be pre-loaded in your cache.

Loading Selventa Corpra

The Selventa Small Corpus and Large Corpus are two example BEL documents distributed by the OpenBEL framework. They are good examples of many types of BEL statements and can be used immediately to begin exploring. Add `-v` for more logging information during compilation. This is highly suggested for the first run, since it takes a

while to cache all of the namespaces and annotations. This only has to be done once, and will be much faster the second time!

Small Corpus

```
$ python3 -m pybel_tools ensure small_corpus -v
```

Large Corpus

```
$ python3 -m pybel_tools ensure large_corpus -v
```

Loading Other Resources

Gene Families

```
$ python3 -m pybel_tools ensure gene_families -v
```

Named Protein Complexes

```
$ python3 -m pybel_tools ensure named_complexes -v
```

Orthology Relations

Coming soon!

Uploading Precompiled BEL

A single network stored as a PyBEL gpickle can quickly be uploaded using the following code:

```
$ python3 -m pybel_tools io upload -p /path/to/my_network.gpickle
```

Uploading Multiple Networks

Multiple networks in a given directory and sub-directories can be uploaded by adding the `-r` tag.

```
$ python3 -m pybel_tools io upload -p ~/bms/aetionomy/ -r
```

Summary Functions

Filters

Comparison Functions

Selection Functions

Integration Functions

Mutation Functions

Visualization

Orthology Resolution Workflow

Subgraph Expansion Workflow

Candidate Mechanism Generation

Heat Diffusion Workflow

Pipeline Builder

Database Service

Definition Utilities

Document Utilities

Utilities

Lexer

Get the Code

Get the pygment source code from bitbucket repo with `hg clone http://bitbucket.org/birkenfeld/pygments-main pygments`

Register Your Lexer

To make Pygments aware of your new lexer, you have to perform the following steps:

1. First, change to the current directory containing the Pygments source code:

   ```
   $ cd .../pygments
   ```

   Select a matching module under pygments/lexers, or create a new module for your lexer class. Copy the BELLexer.py in that module

2. The lexer can be made publicly known by rebuilding the lexer mapping:

   ```
   $ make mapfiles
   ```

Test

To test the new lexer, store an example file with the proper extension in tests/examplefiles. For example, to test the BELLexer, add a tests/examplefiles/Example.bel containing a sample bel code.

Now you can use pygmentize to render your example to HTML:

```
$ ./pygmentize -O full -f html -o /tmp/example.html tests/examplefiles/Example.bel
```

To view the result, open ./example.html in your browser.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search