bcforms documentation

Release 0.0.9

Karr Lab

Jul 09, 2020

Contents

1 Contents **3**

1.1 Installation . . . 3
1.1.1 Prerequisites . . . 3
1.1.2 Latest release From PyPI . . . 3
1.1.3 Latest revision from GitHub . . . 4
1.1.4 Installing the optional features . . . 4
1.2 Command line interface . . . 4
1.3 About . . . 5
1.3.1 License . . . 5
1.3.2 Development team . . . 5
1.3.3 Acknowledgements . . . 6
1.3.4 Questions and comments . . . 6
1.4 References . . . 6

BcForms is a toolkit for concretely describing the molecular structure (atoms and bonds) of macromolecular complexes, including non-canonical monomeric forms, circular topologies, and crosslinks. *BcForms* was developed to help describe the semantic meaning of whole-cell computational models (see https://wholecell.org).

BcForms includes a grammar for describing forms of macromolecular complexes composed of DNA, RNA, protein, and small molecular subunits and crosslinks between the subunits. The DNA, RNA, and protein subunits can be described using BpForms and the small molecule subunits can be described using SMILES. *BcForms* also includes four software tools for verifying descriptions of complexes and calculating physical properties of complexes such as their molecular structure, formula, molecular weight, and charge: this website, a JSON REST API, a command line interface, and a Python API. *BcForms* is available open-source under the MIT license.

CHAPTER 1

Contents

1.1 Installation

The following is a brief guide to installing *BcForms*. The Dockerfile in the *BpForms* repository contains detailed instructions for how to install *BcForms* in Ubuntu Linux.

1.1.1 Prerequisites

First, install the third-party packages listed below.

- ChemAxon Marvin: optional to calculate major protonation and tautomerization states and draw molecules
  - Java >= 1.8
- Open Babel
- Pip >= 18.0
- Python >= 3.6

To use ChemAxon Marvin to calculate major protonation and tautomerization states, set `JAVA_HOME` to the path to your Java virtual machine (JVM) and add Marvin to the Java class path:

```
export JAVA_HOME=/usr/lib/jvm/default-java
export CLASSPATH=$CLASSPATH:/opt/chemaxon/marvinsuite/lib/MarvinBeans.jar
```

1.1.2 Latest release From PyPI

Run the following command to install the latest release from PyPI.:

```
pip install bcforms[all]
```

1.1.3 Latest revision from GitHub

Run the following command to install the latest version from GitHub.:

```
pip install git+https://github.com/KarrLab/pkg_utils.git#egg=pkg_utils
pip install git+https://github.com/KarrLab/wc_utils.git#egg=wc_utils[chem,␣
↪protonation]
pip install git+https://github.com/KarrLab/bpforms.git#egg=bpforms
pip install git+https://github.com/KarrLab/bcforms.git#egg=bcforms[all]
```

1.1.4 Installing the optional features

To draw molecules, *BcForms* must be installed with the *[draw]* option::

```
pip install bcforms[draw]
pip install git+https://github.com/KarrLab/bcforms.git#egg=bcforms[draw]
```

To install the REST API, *BcForms* must be installed with the *[rest_api option]*::

```
pip install bcforms[rest_api]
pip install git+https://github.com/KarrLab/bcforms.git#egg=bcforms[rest_api]
```

1.2 Command line interface

The command line interface provides five functions to easily manipulate *BcForms*-encoded descriptions of complexes.

- **Get help with the 'BcForms' command line interface.** The following commands return inline help information about the command line interface:

  ```
  bcforms
  bcforms -h
  bcforms --help
  ```

- **Validate a 'BcForms'-encoded description of a form of a complex.** The following command can be used to verify if description of a complex is syntactically and semantically valid. The command line interface will print any errors to the standard error:

  ```
  bcforms validate <bcform>

  bcforms validate '2 * a + 3 * b'
  # Form is valid
  ```

- **Calculate the formula of a complex.** The following command can be used to calculate the formula of a complex:

  ```
  bcforms get-formula --help

  bcforms get-formula <bcform> <dictionary of formulae of subunits>

  # Calculate the formula of a complex
  bcforms get-formula '2 * a + 3 * b' '{a: CHO, b: C2H2O2}'
  # C8H8O8
  ```

- **Calculate the charge of a complex.** The following command can be used to calculate the charge of a complex:

```
bcforms get-charge --help

bcforms get-charge <bcform> <dictionary of charges of subunits>

# Calculate the charge of a complex
bcforms get-charge '2 * a + 3 * b' '{a: 1, b: 2}'
# 8
```

- **Calculate the molecular weight of a complex.** The following command can be used to calculate the molecular weight of a complex:

```
bcforms get-molwt --help
bcforms get-molwt <bcform> <dictionary of molecular weights of subunits>

# Calculate the molecular weight of a complex
bcforms get-molwt '2 * a + 3 * b' '{a: 1, b: 2}'
# 8
```

1.3 About

1.3.1 License

The software is released under the MIT license

```
The MIT License (MIT)

Copyright (c) 2019 Karr Lab

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

1.3.2 Development team

This package was developed by the Karr Lab at the Icahn School of Medicine at Mount Sinai in New York, USA.

1.3.3 Acknowledgements

1.3.4 Questions and comments

Please contact the Karr Lab with any questions or comments.

1.4 References