C3Pred Documentation

Release iGEMtuebingen

iGEM Tuebingen

Oct 21, 2019

Contents:

CHAPTER 1

C3Pred - Cell-penetrating peptide activity prediction tool

Software project of the iGEM Team Tübingen 2019

C3Pred is an easy-to-use software tool, to predict the transport effectivity of cell-penetrating peptides (CPP). CPPs are short 4-30 amino acids long peptides, which possess the ability to transport different cargo over the cell membrane. These cargos include proteins, nanobodies, DNA molecules, and small molecule drugs. In recent years, numerous promising clinical and pre-clinical trials have been launched, with CPPs as a carrier for pharmacologically active small molecules. C3Pred allows scientists to make design choices for their CPP-utilizing system based on quantitative transport activity scores.

C3Pred accepts three possible input formats for protein data:

- FASTA-formatted sequences
- UniProtKB Accession Number
- iGEM Part ID

C3Pred automatically fetches and parses the information about the given identifiers using the UniProt website REST API or using the iGEM Registry API, respectively.

C3Pred was released as:

- Web application with an intuitive browser-based graphical user interface (https://github.com/igemsoftware2019/Tuebingen_c3pred_webapp)
- Python package / Command-line tool (https://github.com/igemsoftware2019/Tuebingen_c3pred)

1.1 Installation and Usage

Documentation: https://c3pred.readthedocs.io.

1.2 License

Free software: MIT license

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation Web App

You can either clone the public repository:

```
git clone https://github.com/igemsoftware2019/Tuebingen_c3pred_webapp.git

pip install -r requirements.txt

flask run
```

Or use the use the live version hosted on our server:

https://igem-tuebingen.com/c3pred/

CHAPTER 3

Usage Web App

3.1 Demonstration

CHAPTER 4

Installation Python Package

4.1 Stable release

To install C3Pred, run this command in your terminal:

```
$ pip install c3pred
```

This is the preferred method to install C3Pred, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

4.2 From sources

The sources for C3Pred can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone https://github.com/igemsoftware2019/Tuebingen_c3pred.git
```

Or download the tarball:

```
$ curl -OJL https://github.com/igemsoftware2019/c3pred/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 5

Usage Python Package

5.1 The CLI - Command Line Interface

```
Usage: c3pred [OPTIONS]

  Console script for c3pred.

Options:
  -s, --sequence TEXT  If the input is a FASTA protein string, please use this flag
  -u, --uniprot TEXT   If the input is a UniProtKB accession number, please use this␣
↪flag
  -g, --igem TEXT      If the input is a iGEM Registry ID, please use this flag
  --help               Show this message and exit.
```

5.2 How to use C3Pred in a Python script

Example Python script:

```
from c3pred.c3pred import *

# predict using sequence string
example_1 = predict_fasta("AGYLLGKINLKALAALAKKIL")

# predict using sequence string
example_2 = predict_uniprot("Q86FU0")

# predict using sequence string
example_3 = predict_igem("BBa_K2660000")
```

The functions **predict_fasta**(), :**predict_uniprot**(), **predict_igem**() return a "Results" object:

Results object attributes:

sequence str - sequence string

activity float - activity score

activity_class str - classification for activity (“low/none”/”medium”/”high”)

description str - description of the sequence (if available for Uniprot ID, iGEM ID)

error boolean - describes whether the prediction was successful

error_type str - if an error occured, the error message is stored here

CHAPTER 6

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/igemsoftware2019/Tuebingen_c3pred/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

6.1.4 Write Documentation

C3Pred could always use more documentation, whether as part of the official C3Pred docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/igemsoftware2019/Tuebingen_c3pred/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *c3pred* for local development.

1. Fork the *c3pred* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/c3pred.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv c3pred
$ cd c3pred/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 c3pred tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.5, 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/steffenlem/c3pred/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ pytest tests.test_c3pred
```

6.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 7

Credits

7.1 Development Lead

- iGEM Tuebingen <info@igem-tuebingen.de>

7.2 Contributors

None yet. Why not be the first?

CHAPTER 8

History

8.1 iGEMtuebingen (2019-10-16)

- First release on PyPI.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search