pyGFE Documentation

Release 1.1.5

Mike Halagan

Mar 31, 2021

Contents

1 py-gfe **3**
- 1.1 Docker . . . 3
- 1.2 Example . . . 3
- 1.3 Credits . . . 4

2 Installation **5**
- 2.1 Stable release . . . 5
- 2.2 From sources . . . 5

3 pygfe package **7**
- 3.1 pygfe . . . 7

4 Usage **11**

5 Contributing **13**
- 5.1 Types of Contributions . . . 13
- 5.2 Get Started! . . . 14
- 5.3 Pull Request Guidelines . . . 15
- 5.4 Tips . . . 15

6 History **17**
- 6.1 0.0.1 (2017-11-09) . . . 17

7 Indices and tables **19**

Python Module Index **21**

Index **23**

Copyright (c) 2017 Be The Match operated by National Marrow Donor Program. All Rights Reserved.

Contents:

CHAPTER 1

py-gfe

Python Boilerplate contains all the boilerplate you need to create a Python package.

- Free software: LGPL 3.0
- Documentation: https://pygfe.readthedocs.io.

1.1 Docker

```
docker pull nmdpbioinformatics/py-gfe
```

1.2 Example

```
>>> from Bio import SeqIO
>>> from BioSQL import BioSeqDatabase
>>> from seqann.sequence_annotation import BioSeqAnn
>>> import pygfe
>>> seq_file = 'test_dq.fasta'
>>> gfe = pygfe.pyGFE()
>>> server = BioSeqDatabase.open_database(driver="pymysql", user="root",
...                                        passwd="", host="localhost",
...                                       db="bioseqdb")
>>> seqann = BioSeqAnn(server=server)
>>> seq_rec = list(SeqIO.parse(seq_file, 'fasta'))[0]
>>> annotation = seqann.annotate(seq_rec, "HLA-DQB1")
>>> gfe = gfe.get_gfe(annotation, "HLA-DQB1")
```

(continues on next page)

(continued from previous page)

```
>>> print(gfe)
HLA-DQB1w0-4-0-141-0-12-0-4-0-0-0-0-0
```

1.3 Credits

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

CHAPTER 2

Installation

2.1 Stable release

To install pyGFE, run this command in your terminal:

```
$ pip install pygfe
```

This is the preferred method to install pyGFE, 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.

2.2 From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/mhalagan-nmdp/pygfe
```

Or download the tarball:

```
$ curl  -OL https://github.com/mhalagan-nmdp/pygfe/tarball/master
```

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

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

CHAPTER 3

pygfe package

3.1 pygfe

Created on Feb 8, 2017

@author: mhalagan

exception pygfe.pygfe.**SeqAnnException**(*inputtype=None*, *reason=None*)

Bases: Exception

pygfe.pygfe.**flatten**(*l*)

pygfe.pygfe.**is_classI**(*x*)

pygfe.pygfe.**is_gfe**(*x*)

pygfe.pygfe.**lc**(*x*)

class pygfe.pygfe.**pyGFE**(*url='http://feature.nmdp-bioinformatics.org', loci=['HLA-A', 'HLA-B', 'HLA-C', 'HLA-DRB1', 'HLA-DQB1', 'HLA-DRB4', 'HLA-DRB5', 'HLA-DPB1', 'HLA-DPA1', 'HLA-DQA1', 'HLA-DRB3'], graph: py2neo.database.Graph = None, seqann: Any = {}, features: Dict[KT, VT] = None, verbose: bool = False, kir: bool = False, pid: str = 'NA', gfe2hla: Dict[KT, VT] = None, gfe_feats: pandas.core.frame.DataFrame = None, seq2hla: pandas.core.frame.DataFrame = None, load_gfe2hla: bool = False, load_seq2hla: bool = False, load_gfe2feat: bool = False, verbosity=1*)

Bases: object

classdocs

breakup_gfe(*gfe*)

creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.

- **sequence** – string containing sequence data.

Returns GFEobject.

calcDiff(*gfe1*, *gfe2*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

calcSim(*gfe1*, *gfe2*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

create_typing(*similar_data*, *gfe*, *features*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

diff_seq(*ref_allele*, *annotation*, *imgtdb_version*)

find_gfe_kir(*gfe*, *features*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

find_similar(*gfe*, *features*, *imgtdb_version*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

gfe_create(*locus*, *sequence*, *imgtdb_version*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns Dict.

list_all_db_releases()

map_structures(*gfe_structs*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

matching_features(*gfe1*, *gfe2*, *structures*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

rename_feat(*f*)

sequence_lookup(*locus*, *sequence*, *imgtdb_version*)
Looks up sequence from

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

type_from_seq(*locus: str = None*, *sequence: str = None*, *imgtdb_version: str = '3.31.0'*, *nseqs: int = 20*, *alignseqs: int = 10*, *skip: List[T] = []*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

unique_features(*features*, *locus*, *imgtdb_version*)
creates GFE from HLA sequence and locus

Parameters

- **locus** – string containing HLA locus.
- **sequence** – string containing sequence data.

Returns GFEobject.

CHAPTER 4

Usage

To use pyGFE in a project:

```
import pygfe
```

CHAPTER 5

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/mhalagan-nmdp/pygfe/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/mhalagan-nmdp/pygfe/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 :)

5.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pygfe.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 pygfe
$ cd pygfe/
$ 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 pygfe tests
$ python setup.py test or py.test
$ 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.

5.3 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/mhalagan-nmdp/pygfe/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_pygfe
```

CHAPTER 6

History

6.1 0.0.1 (2017-11-09)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pygfe.pygfe, 7

Index

B

`breakup_gfe()` (*pygfe.pygfe.pyGFE method*), 7

C

`calcDiff()` (*pygfe.pygfe.pyGFE method*), 8
`calcSim()` (*pygfe.pygfe.pyGFE method*), 8
`create_typing()` (*pygfe.pygfe.pyGFE method*), 8

D

`diff_seq()` (*pygfe.pygfe.pyGFE method*), 8

F

`find_gfe_kir()` (*pygfe.pygfe.pyGFE method*), 8
`find_similar()` (*pygfe.pygfe.pyGFE method*), 8
`flatten()` (*in module pygfe.pygfe*), 7

G

`gfe_create()` (*pygfe.pygfe.pyGFE method*), 8

I

`is_classI()` (*in module pygfe.pygfe*), 7
`is_gfe()` (*in module pygfe.pygfe*), 7

L

`lc()` (*in module pygfe.pygfe*), 7
`list_all_db_releases()` (*pygfe.pygfe.pyGFE method*), 9

M

`map_structures()` (*pygfe.pygfe.pyGFE method*), 9
`matching_features()` (*pygfe.pygfe.pyGFE method*), 9

P

`pyGFE` (*class in pygfe.pygfe*), 7
`pygfe.pygfe` (*module*), 7

R

`rename_feat()` (*pygfe.pygfe.pyGFE method*), 9

S

`SeqAnnException`, 7
`sequence_lookup()` (*pygfe.pygfe.pyGFE method*), 9

T

`type_from_seq()` (*pygfe.pygfe.pyGFE method*), 9

U

`unique_features()` (*pygfe.pygfe.pyGFE method*), 9