PyEMMA / Deeptime package development

Moritz Hoffmann

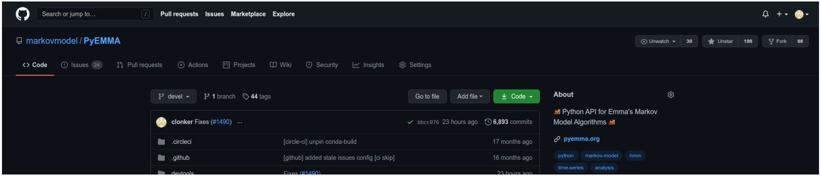
February 27, 2021

Principles

- Use Python for quick prototyping and to easily glue faster implementations (e.g. C++) into performance-critical components of the code
- Open source
- GitHub as frontend (collect issues/bugs, discuss proposed changes, plan new features, …)
- Continuous integration via TravisCI, Appveyor, Azure pipelines
- Unit-testing for large portions of the code

PyEMMA

PyEMMA on GitHub



markovmodel / PyEMMA

clonker Fixes (#1490) … ✓ bbcc076 23 hours ago 6,893 commits

.circleci	[circle-ci] unpin conda-build	17 months ago
.github	[github] added stale issues config [ci skip]	16 months ago
devtools	Fixes (#1490)	23 hours ago

About

Python API for Emma's Markov Model Algorithms

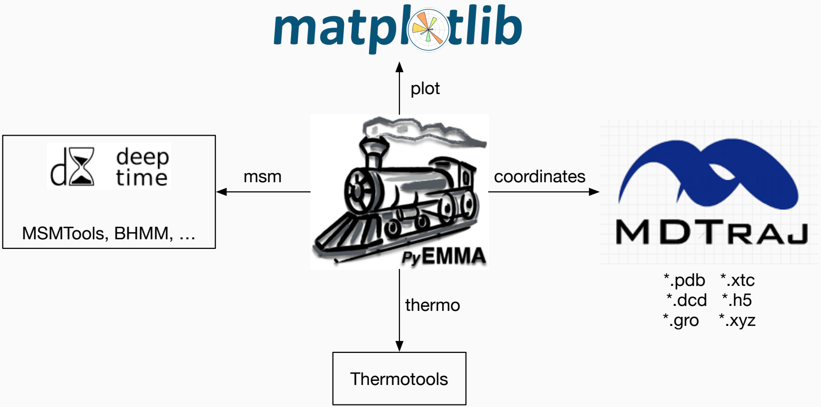
pyemma.org

python markov-model hmm time-series analysis

https://github.com/markovmodel/pyemma

http://www.emma-project.org

Main dependencies and packages



plot

msm

coordinates

thermo

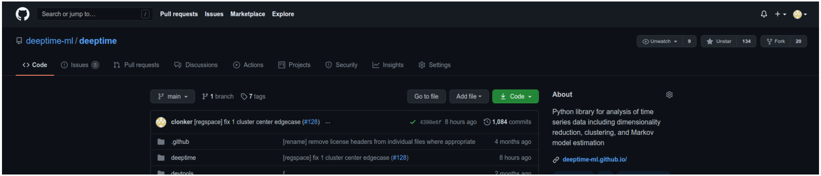
MSMTools, BHMM, ...

Thermotools

*.pdb *.xtc
*.dcd *.h5
*.gro *.xyz

deeptime

deeptime on GitHub



deeptime-ml / deeptime

clonker [regspace] fix 1 cluster center edgecase (#128) … ✓ 4390e6f 8 hours ago 1,084 commits

.github	[rename] remove license headers from individual files where appropriate	4 months ago
deeptime	[regspace] fix 1 cluster center edgecase (#128)	8 hours ago
devtools	f	2 months ago

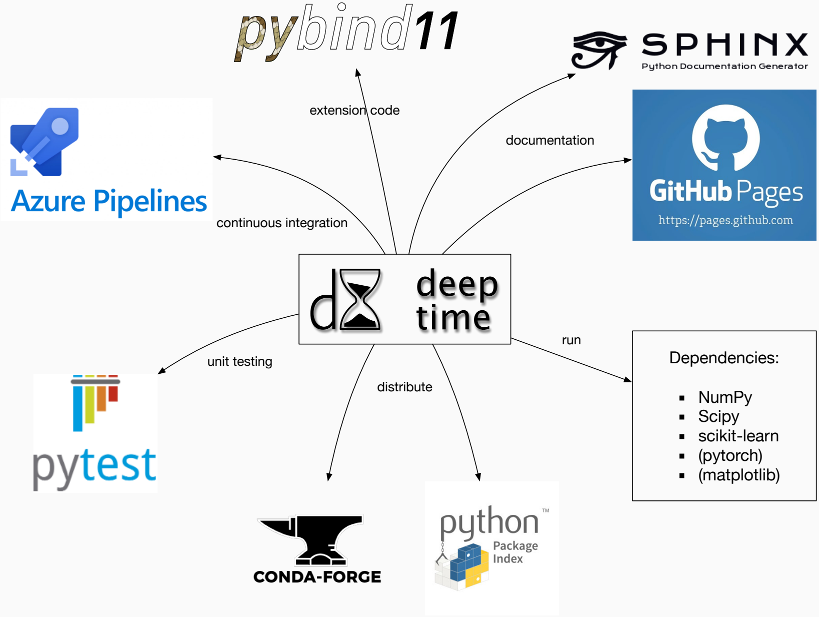
About

Python library for analysis of time series data including dimensionality reduction, clustering, and Markov model estimation

deeptime-ml.github.io/

https://github.com/deeptime-ml/deeptime

https://deeptime-ml.github.io



extension code

documentation

continuous integration

unit testing

distribute

run

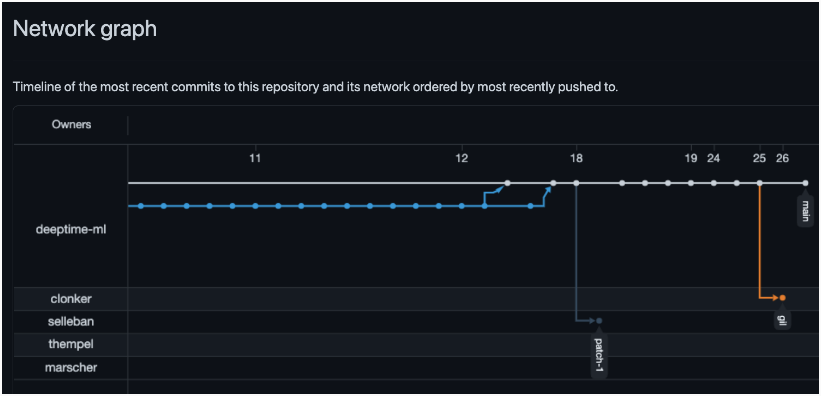
Dependencies:

- NumPy
- Scipy
- scikit-learn
- (pytorch)
- (matplotlib)

Participate!

Forks

Forks are your gateway to contributing

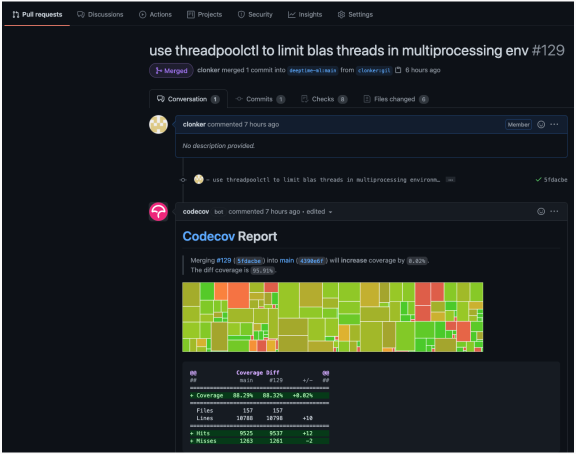


Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

Pull requests

Sets of changes can be admitted into the codebase via pull requests



use threadpoolctl to limit blas threads in multiprocessing env #129

clonker merged 1 commit into `deeptime-ml:main` from `clonker:gil` 6 hours ago

clonker commented 7 hours ago

No description provided.

`- use threadpoolctl to limit blas threads in multiprocessing environm…` ✓ 5fdacbe

codecov bot commented 7 hours ago • edited

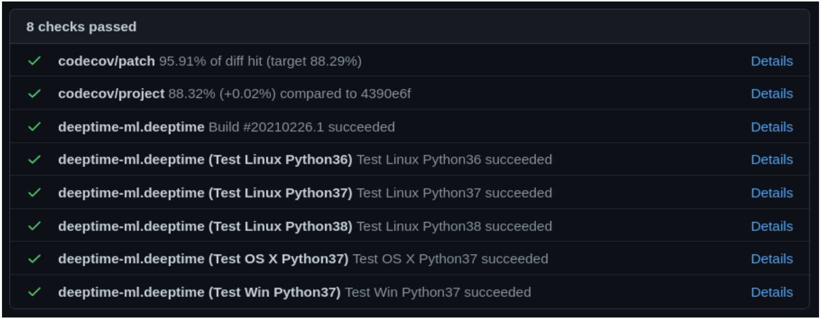
Codecov Report

> Merging #129 (`5fdacbe`) into main (`4390e6f`) will **increase** coverage by `0.02%`.
> The diff coverage is `95.91%`.

```
@@            Coverage Diff             @@
##             main     #129      +/-   ##
==========================================
+ Coverage   88.29%   88.32%   +0.02%     
==========================================
  Files         157      157              
  Lines       10788    10798      +10     
==========================================
+ Hits         9525     9537      +12     
+ Misses       1263     1261       -2     
```

Automated testing

Each pull request goes through automated testing



8 checks passed

✓ **codecov/patch** 95.91% of diff hit (target 88.29%) Details

✓ **codecov/project** 88.32% (+0.02%) compared to 4390e6f Details

✓ **deeptime-ml.deeptime** Build #20210226.1 succeeded Details

✓ **deeptime-ml.deeptime (Test Linux Python36)** Test Linux Python36 succeeded Details

✓ **deeptime-ml.deeptime (Test Linux Python37)** Test Linux Python37 succeeded Details

✓ **deeptime-ml.deeptime (Test Linux Python38)** Test Linux Python38 succeeded Details

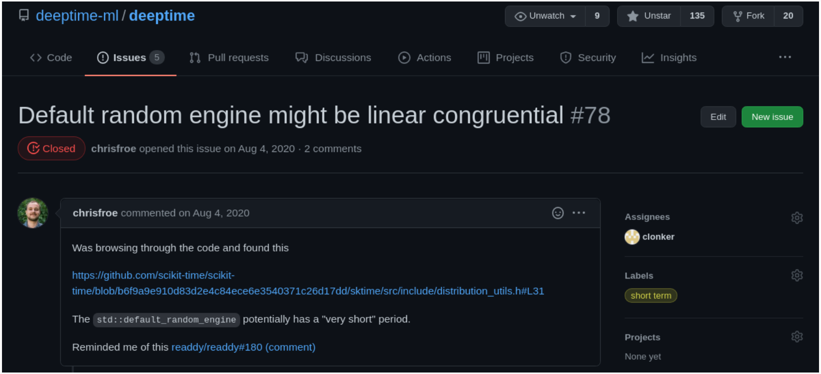
✓ **deeptime-ml.deeptime (Test OS X Python37)** Test OS X Python37 succeeded Details

✓ **deeptime-ml.deeptime (Test Win Python37)** Test Win Python37 succeeded Details

under a selection of operating systems and python versions

Found a bug? Need a feature?

Create a GitHub issue to discuss the matter with us! We are always happy about feedback.



Default random engine might be linear congruential #78

Closed **chrisfroe** opened this issue on Aug 4, 2020 · 2 comments

chrisfroe commented on Aug 4, 2020

Was browsing through the code and found this

https://github.com/scikit-time/scikit-time/blob/b6f9a9e910d83d2e4c84ece6e3540371c26d17dd/sktime/src/include/distribution_utils.h#L31

The `std::default_random_engine` potentially has a "very short" period.

Reminded me of this readdy/readdy#180 (comment)