
diffusive*distinguishability*Documentation

Release 0.1.0

Julie Cass

Jan 02, 2020

Contents:

1	diffusive_distinguishability	1
1.1	Getting Started	1
1.2	Installation	2
1.3	Support	2
1.4	Credits	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	diffusive_distinguishability	7
4.1	diffusive_distinguishability package	7
5	Allen Institute Contribution Agreement	9
5.1	Terms	9
5.2	Types of Contributions	10
5.3	Get Started!	10
5.4	Pull Request Guidelines	11
5.5	Tips	11
5.6	Deploying	11
6	Credits	13
6.1	Development Lead	13
6.2	Contributors	13
7	History	15
7.1	0.1.0 (2019-04-25)	15
8	Indices and tables	17
	Index	19

Simulation of homogeneous isotropic diffusion, bayesian estimation of underlying diffusion constant and analysis of distinguishability between diffusivities. The manuscript associated with this work is available on Biorxiv at the following link: <https://www.biorxiv.org/content/10.1101/740175v1>

1.1 Getting Started

The python package `ndim_homogeneous_distinguishability.py` contains the meat of this project, as a set of functions which can be used to:

1. Simulate diffusive trajectories (diffusion with a homogeneous isotropic diffusion constant)
2. Use Bayesian inference to estimate the diffusion constant used to generate a trajectory by producing a posterior diffusivity distribution
3. Analyze the dependence of diffusivity estimation error, and the ability to distinguish between trajectories with differing diffusivities, conditional on model parameters

This repo also includes:

1. `ndim_diffusion_analysis_tutorial.ipynb`: Jupyter notebook providing examples of how to use these functions, as well as some of our own analysis of diffusivity distinguishability. This includes the function calls used to generate some "Results" figures in our manuscript
2. `ndim_fbm_analysis_tutorial.ipynb`: Jupyter notebook similar to the one described above, but using trajectories generated using fractional brownian motion rather than pure homogeneous diffusion
3. `figure_production.ipynb`: Jupyter notebook used to generate the remainder of our computationally derived manuscript figures, provided for reproducibility
4. `test_overestimation.ipynb`: Jupyter notebook containing a toy model quantifying the relative impact of localization error on diffusion estimates conditional on number of spatial dimensions

5. `saved_data/`: Directory containing some example pre-calculated datasets (in the form of pickled dataframes), generated in our Jupyter notebook example analyses described above. Text files are included with the datasets, specifying the parameters used in their generation
6. `figures/`: Directory storing all of our computationally derived manuscript figures, stored as image files
 - Free software: Allen Institute Software License
 - Documentation: <https://diffusive-distinguishability.readthedocs.io>.

1.2 Installation

PyPI installation not available at this time, please install using git.:

```
pip install git+https://github.com/AllenCellModeling/diffusive_distinguishability.git
```

1.3 Support

We are not currently supporting this code, but simply releasing it to the community AS IS but are not able to provide any guarantees of support. The community is welcome to submit issues, but you should not expect an active response.

1.4 Credits

This package was created with [Cookiecutter](#).

2.1 Stable release

To install `diffusive_distinguishability`, run this command in your terminal:

```
$ pip install diffusive_distinguishability
```

This is the preferred method to install `diffusive_distinguishability`, 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 `diffusive_distinguishability` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/jcass11/diffusive_distinguishability
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jcass11/diffusive_distinguishability/tarball/master
```

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

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


CHAPTER 3

Usage

To use `diffusive_distinguishability` in a project:

```
import diffusive_distinguishability
```


4.1 diffusive_distinguishability package

4.1.1 Subpackages

diffusive_distinguishability.bin package

Module contents

Bin scripts package for diffusive_distinguishability.

diffusive_distinguishability.tests package

Submodules

diffusive_distinguishability.tests.test_ndim_homogeneous_distinguishability module

Module contents

Unit test package for diffusive_distinguishability.

4.1.2 Submodules

4.1.3 diffusive_distinguishability.ndim_homogeneous_distinguishability module

4.1.4 Module contents

Top-level package for diffusive_distinguishability.

```
diffusive_distinguishability.get_module_version()
```

Allen Institute Contribution Agreement

5.1 Terms

This document describes the terms under which you may make “Contributions” — which may include without limitation, software additions, revisions, bug fixes, configuration changes, documentation, or any other materials — to any of the projects owned or managed by the Allen Institute. If you have questions about these terms, please contact us at terms@alleninstitute.org.

You certify that:

- Your Contributions are either:
 1. Created in whole or in part by you and you have the right to submit them under the designated license (described below); or
 2. Based upon previous work that, to the best of your knowledge, is covered under an appropriate open source license and you have the right under that license to submit that work with modifications, whether created in whole or in part by you, under the designated license; or
 3. Provided directly to you by some other person who certified (1) or (2) and you have not modified them.
- You are granting your Contributions to the Allen Institute under the terms of the [2-Clause BSD license](<https://opensource.org/licenses/BSD-2-Clause>) (the “designated license”).
- You understand and agree that the Allen Institute projects and your Contributions are public and that a record of the Contributions (including all metadata and personal information you submit with them) is maintained indefinitely and may be redistributed consistent with the Allen Institute’s mission and the 2-Clause BSD license.

5.1.1 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:

5.2 Types of Contributions

Report bugs at https://github.com/jcass11/diffusive_distinguishability/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.

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

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

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

The best way to send feedback is to file an issue at https://github.com/jcass11/diffusive_distinguishability/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 :)

5.3 Get Started!

Ready to contribute? Here’s how to set up *diffusive_distinguishability* for local development.

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

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

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

```
$ mkvirtualenv diffusive_distinguishability
$ cd diffusive_distinguishability/
$ pip install -r requirements_dev.txt
$ pip install -e .
```

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 diffusive_distinguishability
$ make test-all
```

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 "Resolves gh-###. Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.4 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 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/jcass11/diffusive_distinguishability/pull_requests and make sure that the tests pass for all supported Python versions.

5.5 Tips

To run a subset of tests:

```
$ py.test tests.test_diffusive_distinguishability
```

5.6 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:

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

Travis will then deploy to PyPI if tests pass.

CHAPTER 6

Credits

6.1 Development Lead

- Julie Cass <juliec@alleninstitute.org>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.0 (2019-04-25)

- First release on PyPI.

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

D

`diffusive_distinguishability` (*module*), 7
`diffusive_distinguishability.bin` (*module*), 7
`diffusive_distinguishability.tests` (*module*), 7

G

`get_module_version()` (*in module diffusive_distinguishability*), 7