mermithid Documentation

Release v1.1.10-0-ga1d7d3f

The Project 8 Collaboration

Aug 04, 2019

Contents

Contents:

CHAPTER 1

Introduction to mermithid

Mermithid is a python package for Project 8 analyses. Since morpho is a very generic package, mermithid is built to extend morpho, adding new functions and processors for Project 8 specific needs. It also interfaces with external C++ libraries (Cicada and Phylloxera) used in these new processors.

The core functioning of mermithid is similar to morpho. Mermithid can be used from morpho config files or inside a python script thanks to its python interface.

CHAPTER 2

Installation

These are two possible ways of installing and working with mermithid.

2.1 Virtual environment installation

One can install mermithid on one's machine using a virtual environment: this allows to keep the system relatively clean. However, mermithid uses C++ librairies (Cicada and Phylloxera) that need to be built beforehand.

Cicada and Phylloxera need to be installed in a sub directory:

```
mkdir build
cd build
cmake ..
make -j3
make -j3 install
```

These libraries need to be added to your PYTHONPATH:

```
echo "export PYTHONPATH=${PWD}/build:$PYTHONPATH" >> ~/.bash_profile
```

Inside your virtual environement, install mermithid:

```
source ~/path/to/the/virtual/environment/bin/activate # activate the virtual␣
↪environement
echo $PYTHONPATH # make sure the build folder above is in this path
pip install . --process-dependency-links
```

(The *–process-dependency-links* is here to install the right morpho version from github.)

2.2 Docker installation

Docker provides a uniform test bed for development and bug testing. Please use this environment to testing/resolving bugs.

1. Install Docker (Desktop version): https://docs.docker.com/engine/installation/
2. Clone and pull the latest master version of morpho
3. Inside the morpho folder, execute `` `docker-compose run mermithid` ``. The container prompter should appear at the end of the installation. A directory (`` `mermithid_share` ``) should be created in your home and mounted under the `` `/host` `` folder: you can modify this by editing the docker-compose file. Once inside the container, run *source /setup.sh* to be able to access morpho and mermithid libraries.
4. When reinstalling, you can remove the image using `` `docker rmi mermithid_mermithid` ``

CHAPTER 3

Contribute

3.1 Branching Model

Mermithid uses the git flow branching model, as described here. In summary, the master branch is reserved for numbered releases of mermithid. The only branches that may branch off of master are hotfixes. All development should branch off of the develop branch, and merge back into the develop branch when complete. Once the develop branch is ready to go into a numbered release, a release branch is created where any final testing and bug fixing is carried out. This release branch is then merged into master, and the resulting commit is tagged with the number of the new release.

3.2 Style

Mermithid loosely follows the style suggested in the Style Guide for Python (PEP 8).

Every package, module, class, and function should contain a docstring, that is, a comment beginning and ending with three double quotes. We use the Google format, because the docstrings can then be automatically formatted by sphinx and shown in the API.

Every docstring should start with a single line (<=72 characters) summary of the code. This is followed by a blank line, then further description is in paragraphs separated by blank lines. Functions should contain "Args:", "Returns:", and if necessary, "Raises" sections to specify the inputs, outputs, and exceptions for the function. All text should be wrapped to around 72 characters to improve readability.

3.3 Other Conventions

- __init__.py files:

In mermithid 1, __init__.py files are set up such that

```
from package import *
```

will import all functions from all subpackages and modules into the namespace. If a package contains the subpackages "subpackage1" and "subpackage2", and the modules "module1" and "module2", then the __init__.py file should include imports of the form:

```
from . import subpackage1
from . import subpackage2
from ./module1 import *
from ./module2 import *
```

In mermithid 2, __init__.py files are set up such that

```
from package import *
```

will import all modules into the namespace, but it will not directly import the functions into the namespace. For our package containing "subpackage1", "subpackage2", "module1", and "module2", __init__.py should be of the form:

```
__all__ = ["module1", "module2"]
```

In this case, functions would be called via module1.function_name(). If one wants all of the functions from module1 in the namespace, then they can include "from package.module1 import *" at the top of their code. This change to more explicit imports should prevent any issues with function names clashing as mermithid grows.

CHAPTER 4

Validation Log

4.1 Log

4.1.1 Version: v1.1.10

Release Date: Sat Aug 3 2019

New features:

- Documentation update
- ReadTheDocs repair

4.1.2 Version: v1.1.9

Release Date: Mon Jul 22 2019

New features:

- P8 Compute Dependencies image update to v0.7.0

4.1.3 Version: v1.1.8

Release Date: Thur Apr 18 2019

New features:

- Morpho update to v2.3.2

4.1.4 Version: v1.1.7

Release Date: Thur Apr 4 2019

New features:

- Morpho update to v2.3.1
- Cicada update to v1.3.3

4.1.5 Version: v1.1.6

Release Date: Mon Feb 11 2019

New features:

- Update docker-compose.yaml

4.1.6 Version: v1.1.5

Release Date: Wed Dec 21 2018

New features:

- Update Dockerfile

4.1.7 Version: v1.1.4

Release Date: Wed Dec 6 2018

New features:

- Update Dockerfile

4.1.8 Version: v1.1.3

Release Date: Wed Dec 5 2018

New features:

- Update to Phylloxera v1.2.4

4.1.9 Version: v1.1.2

Release Date: Wed Dec 5 2018

New features:

- Update to Phylloxera v1.2.3

4.1.10 Version: v1.1.1

Release Date: Wed Dec 5 2018

Fixes:

- Changing base processor for TritiumLikelihoodSampler
- Upgrade of docker image build

4.1.11 Version: v1.1.0

Release Date: Mon Nov 19 2018

New Features:

- Documentation update (RTD and source code)
- morpho update to v2.3.0
- Kurie plot generator and fitter have been merged

Fixes:

- Various comments from users