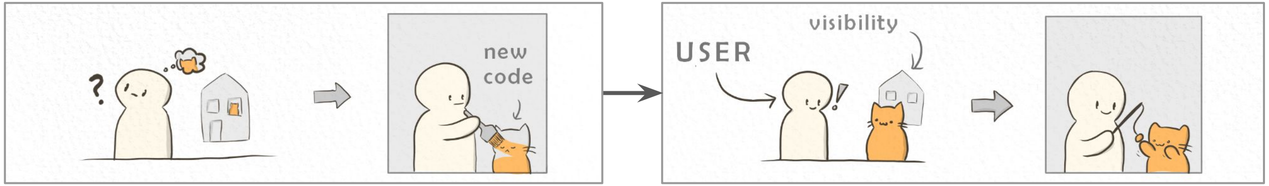
Supporting community-driven open-access simulation software with MDAnalysis toolkits

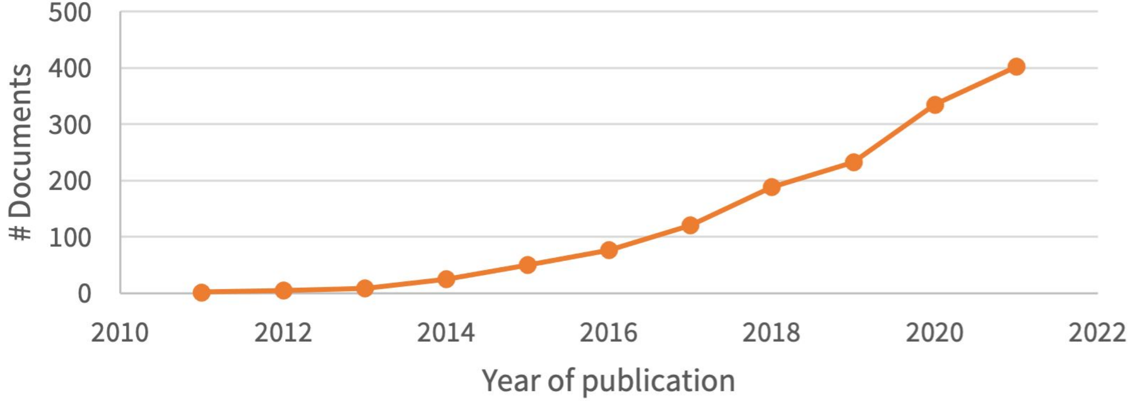
MDAKITS

Scientific code is increasingly shared

- MD community has made effort to share more and more code
- Lets you replicate methodology
- Lets you re-use code
- Reduces duplication of effort
- Reduces risk of implementation error

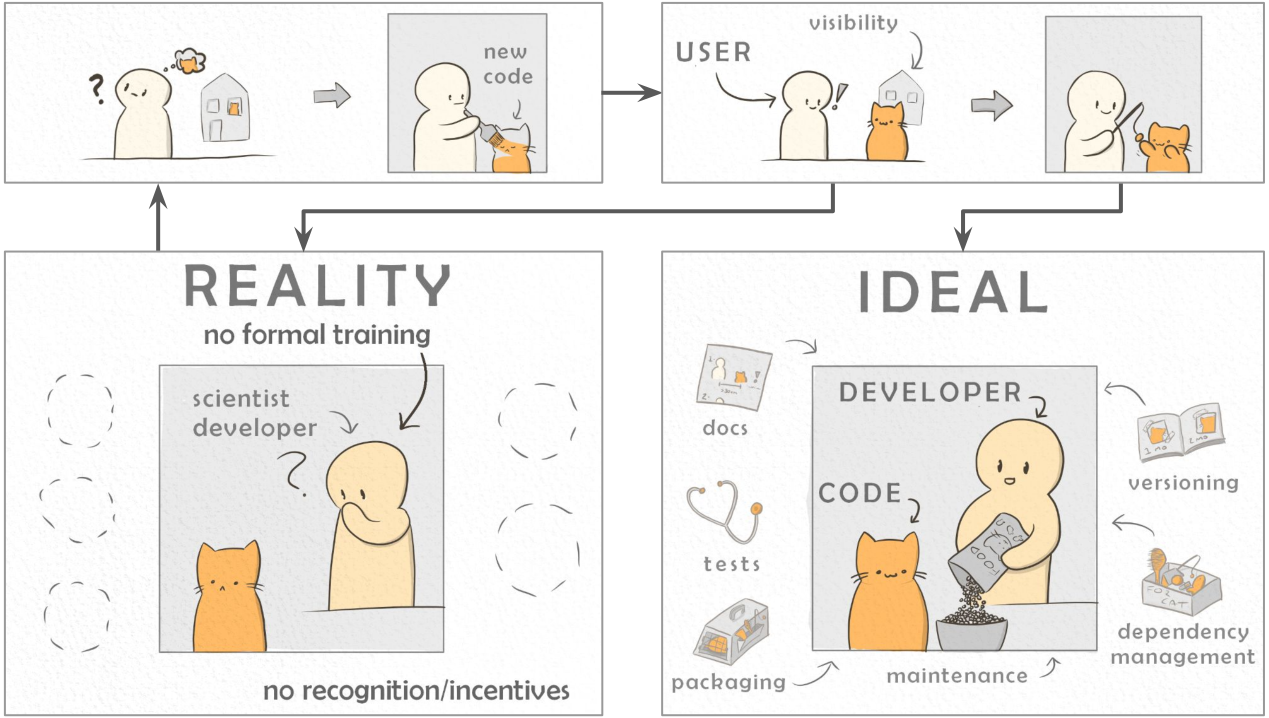


Documents in Scopus with “molecular dynamics” and github.com, gitlab.com, or bitbucket.com



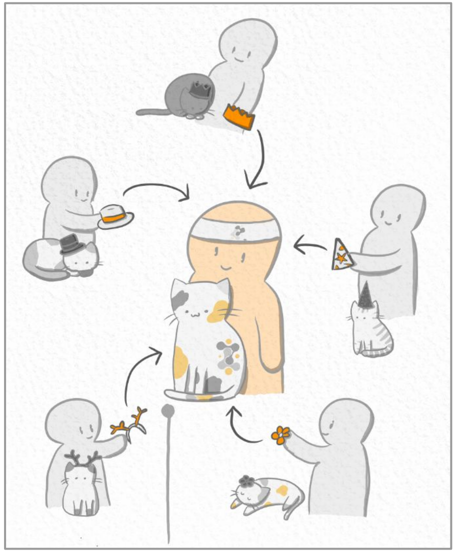
Scientific code often falls short of FAIR tenets

- **F:** findable
- **A:** accessible
- **I:** interoperable
- **R:** reusable



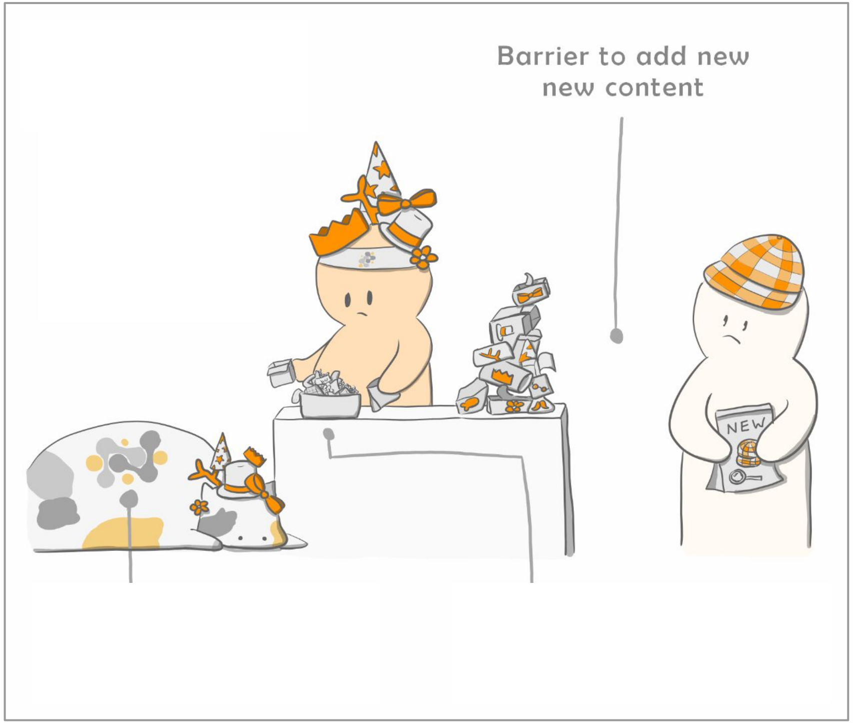
Centralising in MDAnalysis is a limited solution

- Asking for contributions to MDAnalysis has many advantages
  - Setting up documentation, packaging, automatic tests, etc already taken care of
  - Experienced developers ensure code follows best practices
  - Code automatically accessible to user base – even promoted
  - MDAnalysis developer team can maintain code, so original developers can move on

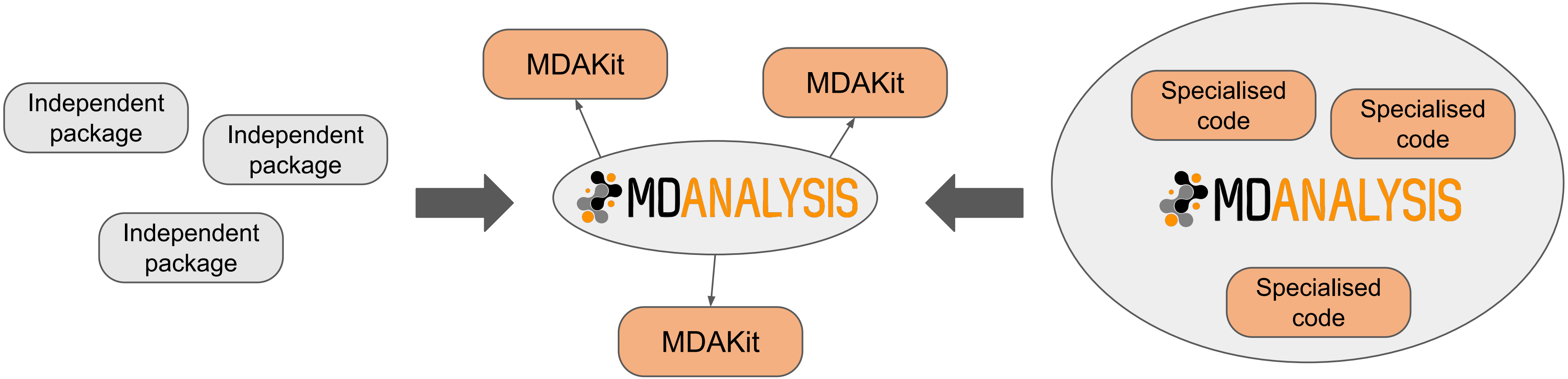


Centralising in MDAnalysis is a limited solution

- But many disadvantages make this unsustainable for us

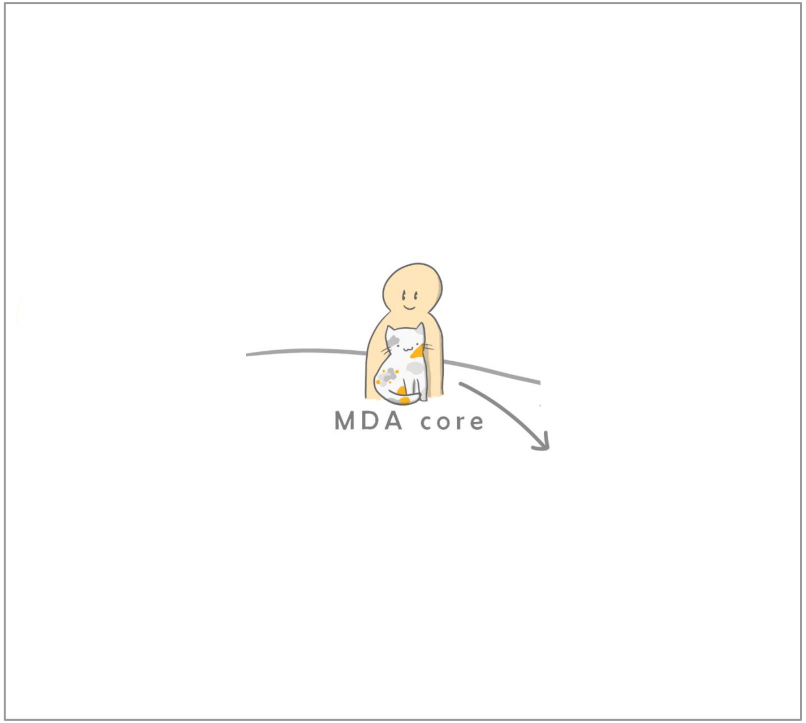


MDAKits (MDAnalysis toolkits)



MDAKits

- Researchers supported in developing software and being FAIR-compliant
- Cookiecutter template provided
- Toolkits that meet standards are promoted to the community on an MDAKit registry
- Users can check here for analysis packages
- Developers can find users and collaborate with each other



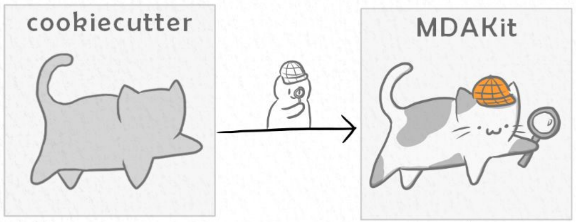
Cookiecutter template (cookiecutter-mdakit)

```
$ cookiecutter gh:MDAnalysis/cookiecutter-mdakit

project_name [ProjectName]: My Project Name
repo_name [my_project_name]: example-repository
package_name [example_repository]: package_name
description [A short description of the project.]: A package to do MD analysis
github_username [Your personal GitHub username]: my-github-username
github_host_account [GitHub account for hosting example-repository (e.g.
my-github-username or an organization account). Press Enter to use
my-github-username]:
author_name [Your name (or your organization/company/team)]: My Name
author_email [Your email (or your organization/company/team)]:
my_example_email@gmail.com
Select dependency_source:
1 - Prefer conda-forge over the default anaconda channel with pip fallback
2 - Prefer default anaconda channel with pip fallback
3 - Dependencies from pip only (no conda)
Choose from 1, 2, 3 [1]: 1

Select include_ReadTheDocs:
1 - y
2 - n
Choose from 1, 2 [1]: 1

template_analysis_class [Class name to template (e.g. MyProjectName ).
Press Enter to skip including analysis templates]: MyAnalysisClass
```

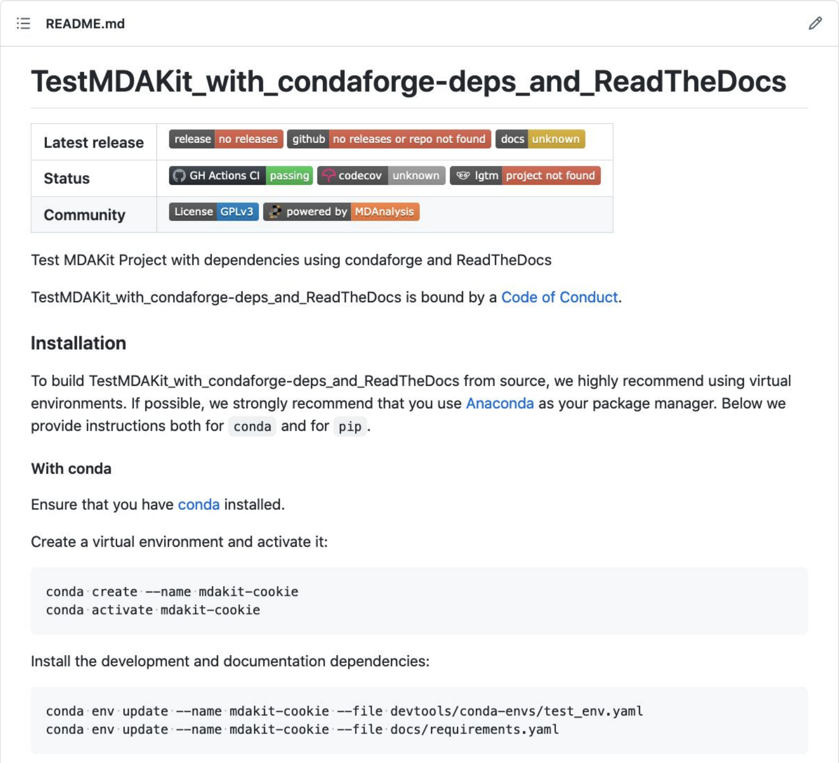


First step in MDAKit framework

*Based off MolSSI's cookiecutter-cms!

https://www.mdanalysis.org/cookiecutter-mdakit/

Cookiecutter example



README.md

TestMDAKit_with_condaforge-deps_and_ReadTheDocs

Latest release	release no releases github no releases or repo not found docs unknown
Status	GH Actions CI passing codecov unknown lgtm project not found
Community	License GPLv3 powered by MDAnalysis

Test MDAKit Project with dependencies using condaforge and ReadTheDocs

TestMDAKit_with_condaforge-deps_and_ReadTheDocs is bound by a Code of Conduct.

Installation

To build TestMDAKit_with_condaforge-deps_and_ReadTheDocs from source, we highly recommend using virtual environments. If possible, we strongly recommend that you use Anaconda as your package manager. Below we provide instructions both for `conda` and for `pip`.

With conda

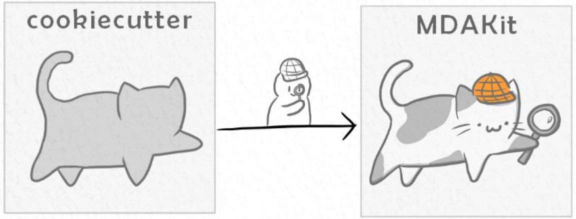
Ensure that you have conda installed.

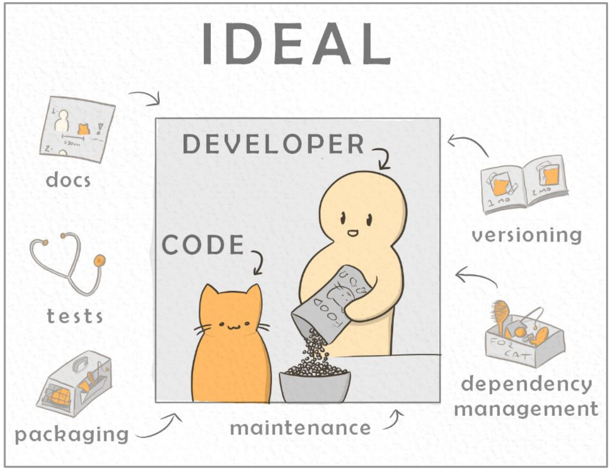
Create a virtual environment and activate it:

```
conda create --name mdakit-cookie
conda activate mdakit-cookie
```

Install the development and documentation dependencies:

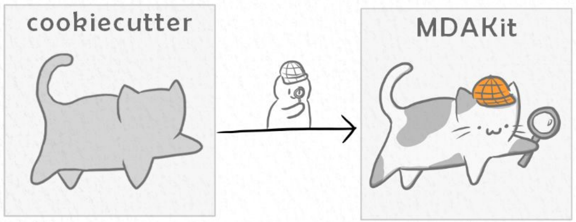
```
conda env update --name mdakit-cookie --file devtools/conda-envs/test_env.yaml
conda env update --name mdakit-cookie --file docs/requirements.yaml
```

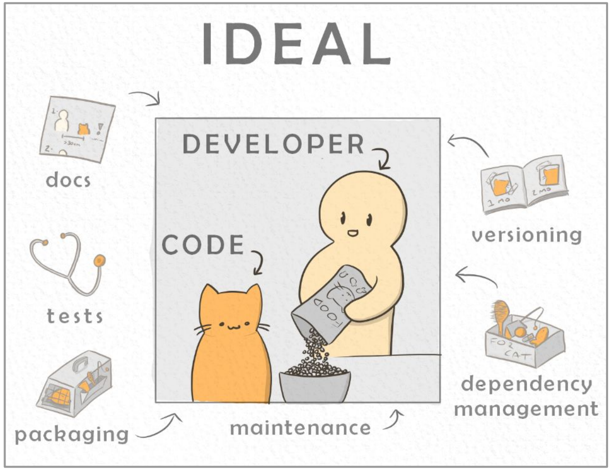




Cookiecutter template (cookiecutter-mdakit)

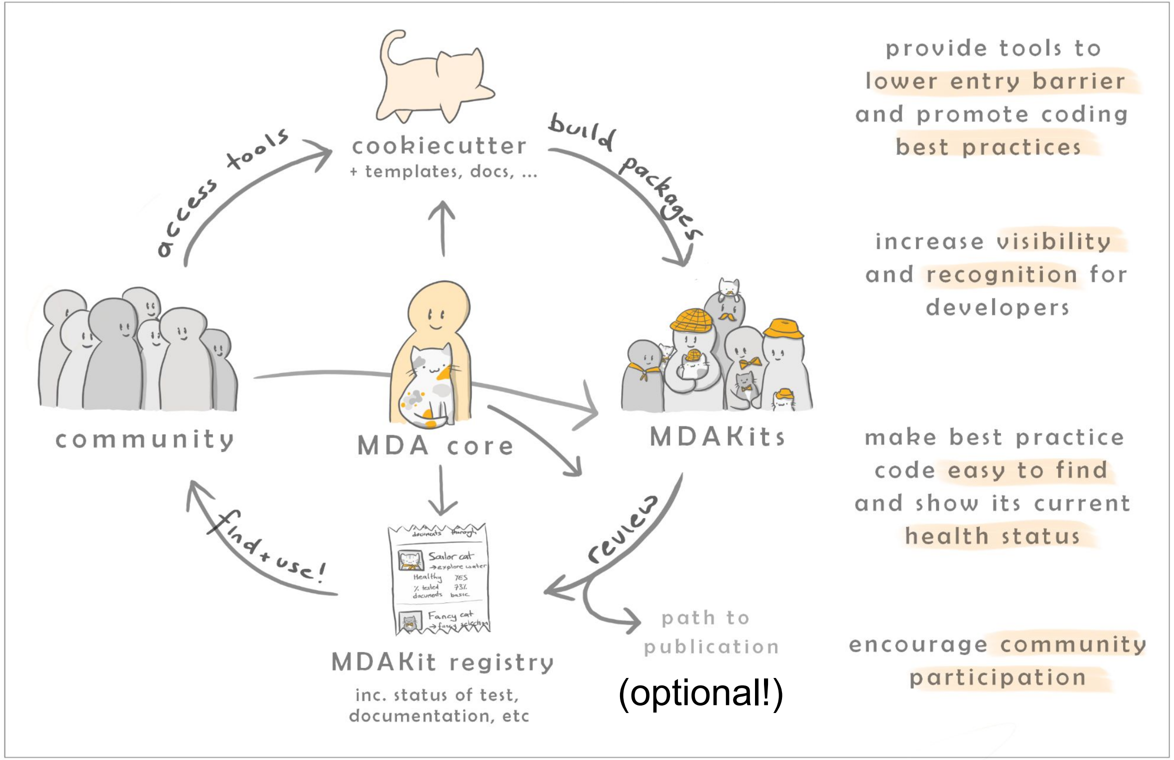
- Builds skeleton repository with:
  - **Continuous integration** for GitHub Actions with a matrix of OS, Python, MDAnalysis versions
  - Basic example **test** structure with PyTest
  - Instructions for creating development environments
  - Choice of **dependency management** from conda-forge, anaconda, or PyPI
  - Initial basic **documentation** configured
  - Modern best-practice Python **packaging**
  - Automatic **version control**
  - Configuration for helpful hooks for code coverage, Python code quality, online documentation, etc
  - Community infrastructure e.g. code of conduct, author list, change log
  - Template code for a custom Analysis class





https://www.mdanalysis.org/cookiecutter-mdakit/

MDAKits - beyond just packages



(optional!)

provide tools to lower entry barrier and promote coding best practices

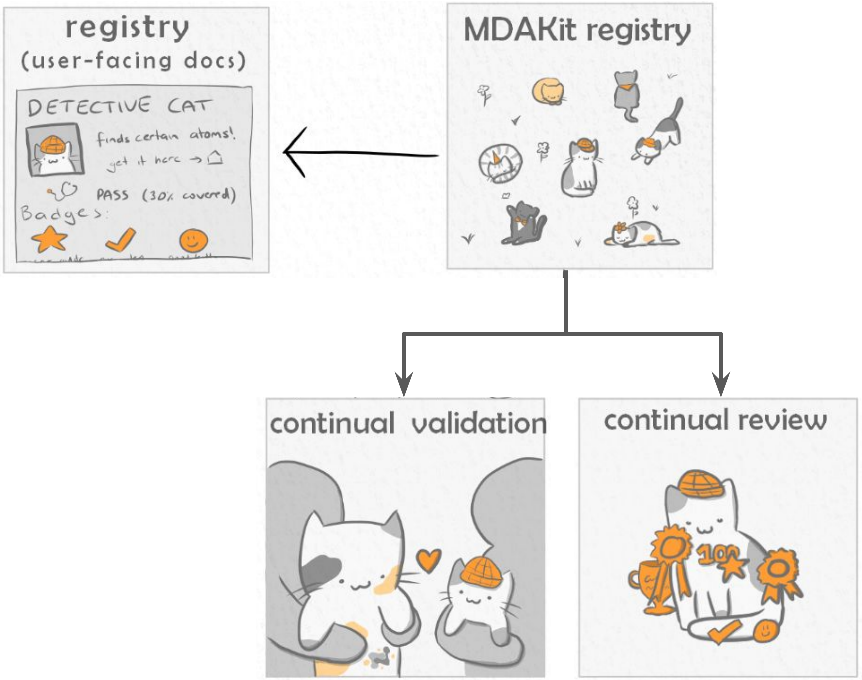
increase visibility and recognition for developers

make best practice code easy to find and show its current health status

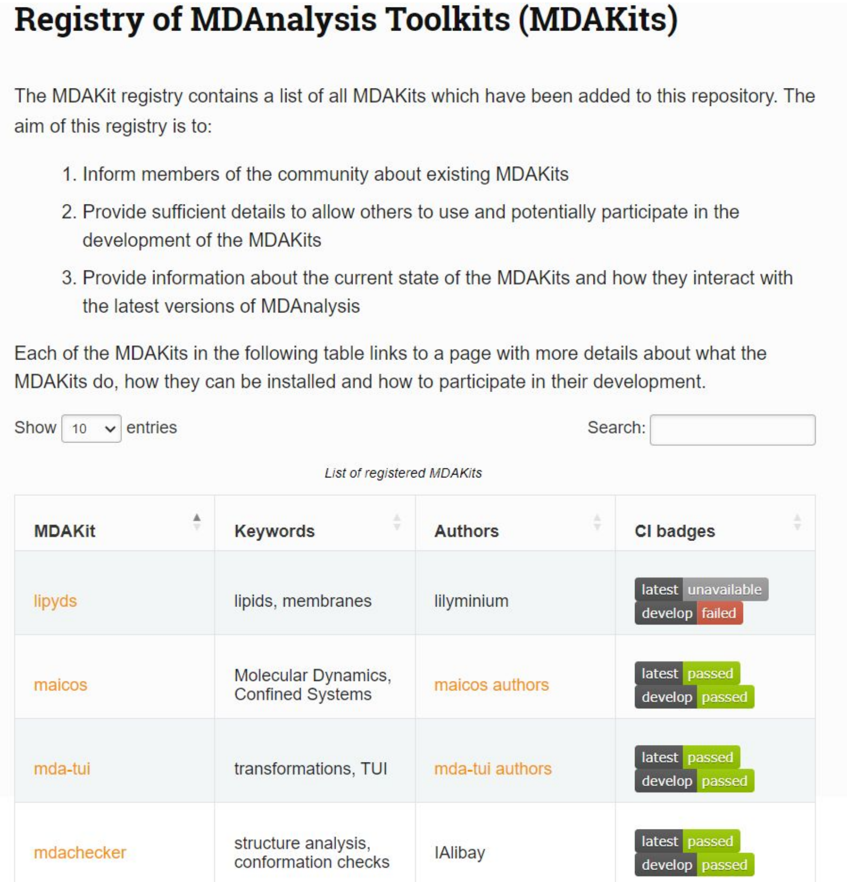
encourage community participation

MDAKit Registry

- List of MDAKits that fulfil quality standards
- Each MDAKit gets its own page with badges and recent validation data
- Badges include information about test coverage, presence of a user guide, etc
- Validation data lists versions of Python and MDAnalysis that pass tests
- The Registry runs continual checks to validate the health of each registered package



MDAKit Registry



Registry of MDAnalysis Toolkits (MDAKits)

The MDAKit registry contains a list of all MDAKits which have been added to this repository. The aim of this registry is to:

1. Inform members of the community about existing MDAKits
2. Provide sufficient details to allow others to use and potentially participate in the development of the MDAKits
3. Provide information about the current state of the MDAKits and how they interact with the latest versions of MDAnalysis

Each of the MDAKits in the following table links to a page with more details about what the MDAKits do, how they can be installed and how to participate in their development.

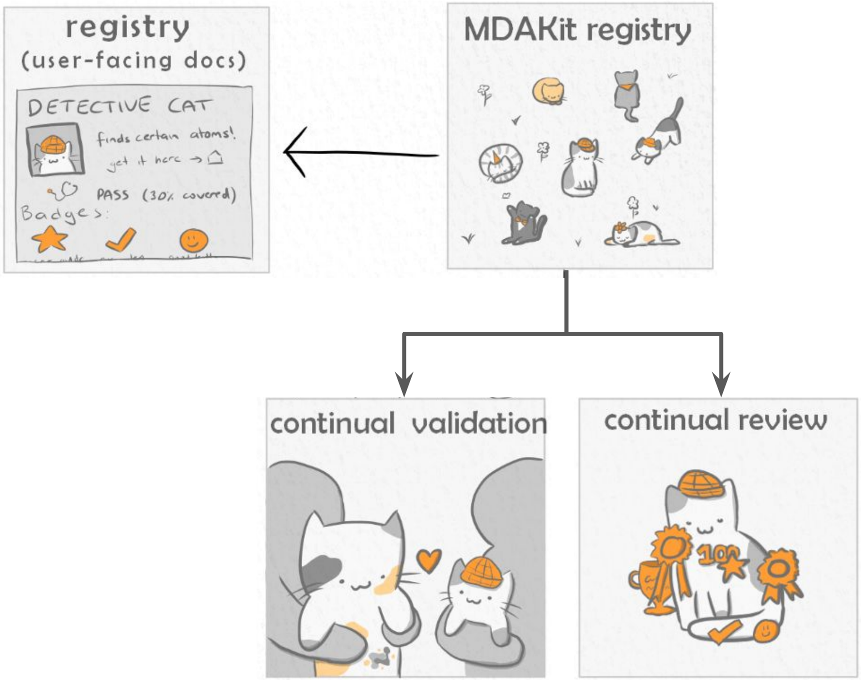
Show 10 entries Search:

List of registered MDAKits

MDAKit	Keywords	Authors	CI badges
lipyds	lipids, membranes	lilyminium	latest unavailable develop failed
maicos	Molecular Dynamics, Confined Systems	maicos authors	latest passed develop passed
mda-tui	transformations, TUI	mda-tui authors	latest passed develop passed
mdachecker	structure analysis, conformation checks	IAlibay	latest passed develop passed

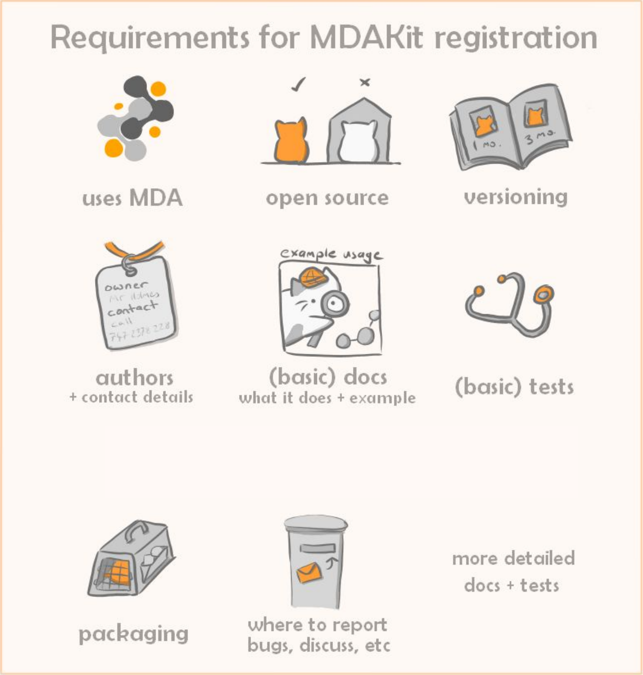
MDAKit Registry - responsibility

- Registry maintainers
  - Keeping the registry components working
  - Adding new components as needs arise
  - Helping with addition of new mdakits
  - Helping with advice on continual improvement
  - Note: packages will *not be removed* if they fail continuous validation
- MDAKits maintainers
  - Ensuring registry for their mdakits remain green
  - Ensuring the information in the registry is up to date



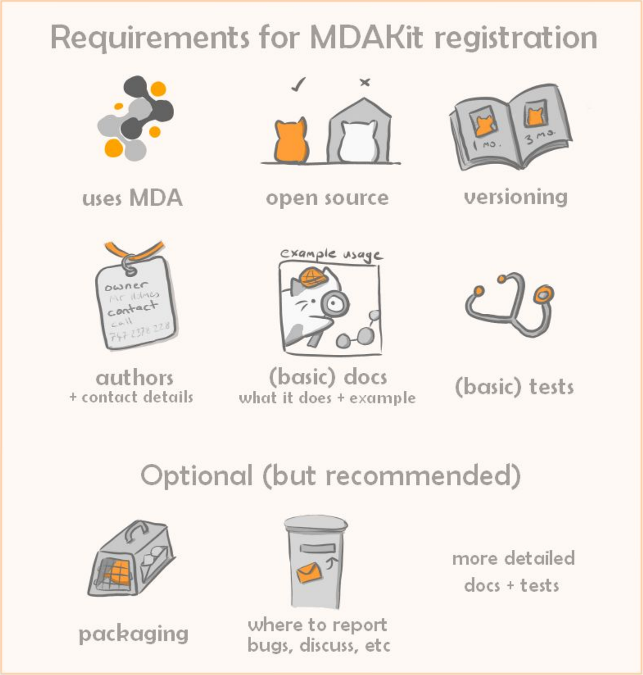
MDAKit minimal requirements

1. Must use **MDAnalysis**! Powered by MDAnalysis
2. **Open source** under an OSI approved license
3. Code is **versioned** and provided in a version-controlled repository
4. Clearly designated **authors** and maintainers
5. At least minimal **documentation**
6. Minimal regression **tests** and **CI**



MDAKit ideal standards

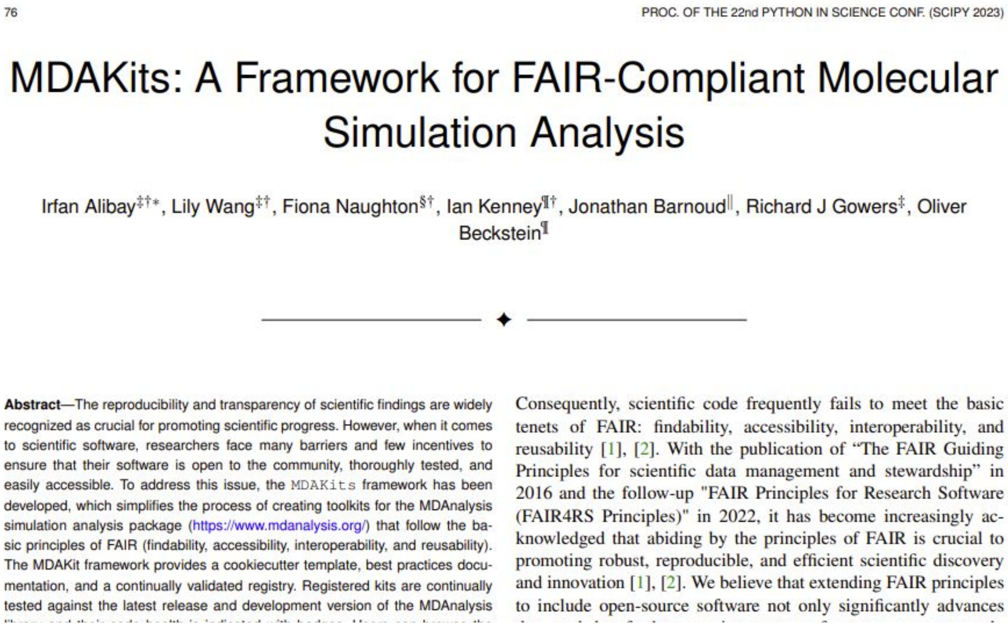
1. Must use **MDAnalysis**!
2. **Open source** under an OSI approved license
3. Code is **versioned** and provided in a version-controlled repository
4. Clearly designated **authors** and maintainers
5. API **documentation** and **user guide**
6. Unit **tests** with >80% code coverage
7. **Packaging** for easy installation
8. **Community** infrastructure (e.g. code of conduct, discussion channels)



Towards publication (future)

- Our best practices are similar to criteria in software journals (e.g. Journal of Open Source Software)
- MDAKits are encouraged to consider publication
- We are working with journal editors at JOSS to streamline the publication process

MDAKit paper



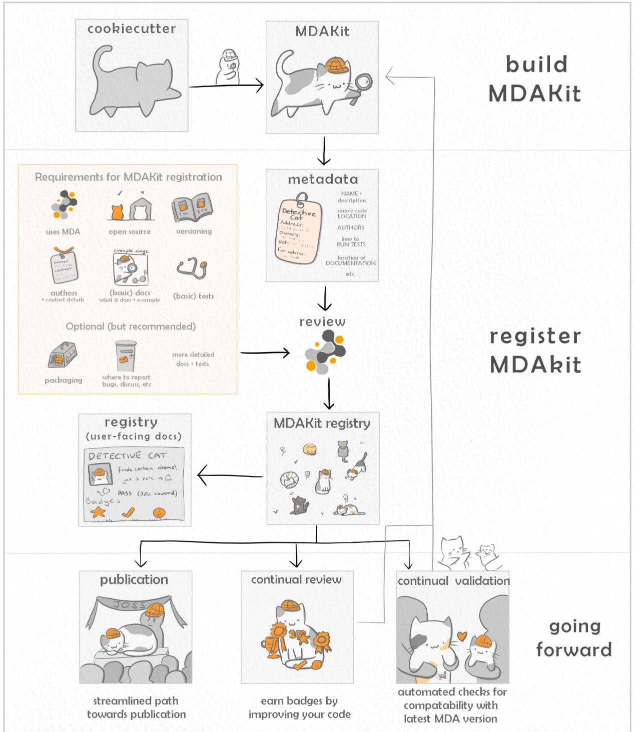
MDAKits: A Framework for FAIR-Compliant Molecular Simulation Analysis

Irfan Alibay[‡†*], Lily Wang[‡†], Fiona Naughton[§†], Ian Kenney[¶†], Jonathan Barnoud[||], Richard J Gowers[‡], Oliver Beckstein[¶]

Abstract—The reproducibility and transparency of scientific findings are widely recognized as crucial for promoting scientific progress. However, when it comes to scientific software, researchers face many barriers and few incentives to ensure that their software is open to the community, thoroughly tested, and easily accessible. To address this issue, the `MDAKits` framework has been developed, which simplifies the process of creating toolkits for the MDAnalysis simulation analysis package (https://www.mdanalysis.org/) that follow the basic principles of FAIR (findability, accessibility, interoperability, and reusability). The MDAKit framework provides a cookiecutter template, best practices documentation, and a continually validated registry. Registered kits are continually tested against the latest release and development version of the MDAnalysis

Consequently, scientific code frequently fails to meet the basic tenets of FAIR: findability, accessibility, interoperability, and reusability [1], [2]. With the publication of "The FAIR Guiding Principles for scientific data management and stewardship" in 2016 and the follow-up "FAIR Principles for Research Software (FAIR4RS Principles)" in 2022, it has become increasingly acknowledged that abiding by the principles of FAIR is crucial to promoting robust, reproducible, and efficient scientific discovery and innovation [1], [2]. We believe that extending FAIR principles to include open-source software not only significantly advances

The MDAKit framework


cookiecutter
MDAKit
build MDAKit
Requirements for MDAKit registration
uses MDA
open source
versioning
authors
+ contact details
(basic) docs
what it does + example
example usage
(basic) tests
Optional (but recommended)
packaging
where to report bugs, discuss, etc
more detailed docs + tests
metadata
Detective Cat
Address:
Owners:
Vet:
For advice:
NAME + description
source code LOCATION
AUTHORS
how to RUN TESTS
location of DOCUMENTATION
etc
review
register MDAkit
registry
(user-facing docs)
DETECTIVE CAT
finds certain atoms!
get it here →
PASS (30% covered)
Badges:
MDAKit registry
publication
JOSS
continual review
continual validation
going forward
streamlined path towards publication
earn badges by improving your code
automated checks for compatability with latest MDA version