

Research References Tracking Tool (R2T2)

Dr Diego Alonso Álvarez, Research Computing Service, ICT, Imperial College London

<https://imperialcollegelondon.github.io/R2T2/>

Problem: Incomplete citation loop for reproducible research



Research citing software

- ✗ Software itself lacks full bibliographic info
- ✗ Documentation references are not indexed

Software citing research?

Aims of R2T2

1. Facilitate **giving credit** to all the works software is based on.
2. Promote those works' **visibility and impact**.
3. Promote the **transparency of the code**: bi-directional link between theory and the specific code that implements it.
4. Facilitate code maintenance and improve its sustainability.

Adding references

1. Decorator for functions, classes or methods where:
 - algorithms described in a paper are implemented
 - data stored in a repository is used

```
from r2t2 import add_reference
@add_reference(short_purpose="some context",
               reference="Reference text")
def my_great_function():
    pass
```

2. Multiple references can be stacked:

```
@add_reference(short_purpose="A", reference="Ref1")
@add_reference(short_purpose="B", reference="Ref2")
def my_great_function():
    pass
```

Getting involved



← Fill in this form to get involved

- What is your use case?
- What citation tools do you use?
- What options would you find useful?
- Would you test newly implemented features?

Retrieving references

1. **Runtime tracker**: records the execution of annotated functions when running a particular script:

```
$ python -m r2t2 my_script.py arg1 arg2
```

2. **Static parser**: list of all references that a given folder or file contains:

```
$ python -m r2t2 some/folder
$ python -m r2t2 -s my_script.py
```

3. Currently available **output formats** are "terminal", "markdown", or explore the BIBLIOGRAPHY dictionary if used interactively:

```
$ python -m r2t2 -f markdown -o my_refs.md target
```

```
from r2t2 import BIBLIOGRAPHY
```

Next steps

1. Resolve references provided as a DOI online:

```
@add_reference(doi="10.1007/s10825-018-1171-3")
def my_fancy_algorithm()
    pass
```

2. Resolve references provided in docstrings as **Sphinx** or **Doxygen cite** keywords described in an external Bibtex file:

```
@add_reference()
def my_fancy_algorithm()
    """Implements Eq. 7 in \cite Smith2019."""
    pass
```

3. Ultimately, add standard, **human and machine readable formats**, eg. `.zenodo.json`, `CFF` (Citation File Format) or Bibtex.