
Report from 2015 OHBM Hackathon (HI)

# DueCredit: automated collection of citations for software, methods, and data

**Project URL:** https://github.com/duecredit/duecredit

Yaroslav O. Halchenko* and Matteo Visconti di Oleggio Castello

## 1 Introduction

Data analysis software and canonical datasets are the driving force behind many fields of empirical sciences. Despite being of paramount importance, those resources are most often not adequately cited. Although some can consider this a "social" problem, its roots are technical: Users of those resources often are simply not aware of the underlying computational libraries and methods they have been using in their research projects. This in-turn fosters inefficient practices that encourage the development of new projects, instead of contributing to existing established ones. Some projects (e.g. FSL [1]) facilitate citation of the utilized methods, but such efforts are not uniform, and the output is rarely in commonly used citation formats (e.g. BibTeX). DueCredit is a simple framework to embed information about publications or other references within the original code or dataset descriptors. References are automatically reported to the user whenever a given functionality or dataset is being used.

## 2 Approach

DueCredit is currently available for Python, but we envision extending support to other frameworks (e.g., Matlab, R). Until DueCredit gets adopted natively by the projects, it provides the functionality to "inject" references for 3rd party modules.

For the developer, DueCredit implements a decorator `@due.dcite` that allows to link a method or class to a set of references that can be specified through a doi or BibTeX entry. For example (from PyMVPA):

```
@due.dcite(
    Doi('10.1016/j.neuron.2011.08.026'),
    description="Hyperalignment of data to a ...",
    tags=["implementation"])
def train(self, datasets):
    """Derive a common feature space ...
    ....
```

The end-user simply needs to run the script loading the `duecredit` module. For example, with the following minimal script

```
# A tiny analysis script to demonstrate duecredit
#
# Import of duecredit is not necessary if you just run
# this script with
#    python -m duecredit
# import duecredit  # Just to enable duecredit
from scipy.cluster.hierarchy import linkage
from scipy.spatial.distance import pdist
from sklearn.datasets import make_blobs

print("I: Simulating 4 blobs")
data, true_label = make_blobs(centers=4)

dist = pdist(data, metric='euclidean')

Z = linkage(dist, method='single')
print("I: Done clustering 4 blobs")
```

by running it with `python -m duecredit`, we get a summary of the packages and methods used

```
$python -m duecredit example_scipy.py
I: Simulating 4 blobs
I: Done clustering 4 blobs
DueCredit Report:
- Scientific tools library / numpy (v 1.10.4) [1]
- Scientific tools library / scipy (v 0.17) [2]
  - Single linkage hierarchical clustering ... [3]
```

*Correspondence: yoh@onerussian.com
Department of Psychological & Brain Sciences, Dartmouth College, Hanover, 6207 Moore Hall, 3755, New Hampshire, USA
Full list of author information is available at the end of the article

```
- Machine Learning library / sklearn ... [4]

3 packages cited
0 modules cited
1 functions cited

References
----------

[1] Van Der Walt, S., Colbert, S.C. & ...
structure for efficient numerical comp...
Engineering, 13(2), pp.22{30.
[2] Jones, E. et al., 2001. SciPy: Open ...
[3] Sibson, R., 1973. SLINK: an optimall...
single-link cluster method. The Computer...
[4] Pedregosa, F. et al., 2011. Scikit-l...
Journal of Machine Learning Research, 12...
```

The references can then be easily converted into BibTeX format by using the `duecredit summary --format bibtex` command.

## 3 Results

The initial release of DueCredit (0.1.0) was implemented during the OHBM 2015 hackathon and uploaded to pypi and is freely available. DueCredit provides a concise API to associate a publication reference with any given module or function. For example: To provide a reference for an entire module the cite function can be used, while functions and methods can be conveniently decorated using dcite. DueCredit comes with a simple demo code which demonstrates its utility. Running a sample analysis produces a summary of references. At each run, the information is stored in a pickled file, and incremental runs update that file. Thus, DueCredit summary can be used to show that information again or export it as a BibTeX file ready for reuse.

## 4 Conclusions

DueCredit is in its early stages of development, but two days of team development at the OHBM hackathon were sufficient to establish a usable prototype implementation. Since then, the code-base was further improved and multiple beta-releases followed, expanding the coverage of citable resources (e.g., within scipy, sklearn modules via injections and PyMVPA natively).

**Availability of Supporting Data**
More information about this project can be found at: https://github.com/duecredit/duecredit. Further data and files supporting this project are hosted in the *GigaScience* repository REFXXX.

**Competing interests**
None

**Author's contributions**
YOH and MVdOC performed the project and wrote the report.

**Acknowledgements**
The authors would like to thank the organizers and attendees of the 2015 OHBM Hackathon. This project is supported in part by a grant from the NSF (award 1429999).

**References**

1. Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E., Johansen-Berg, H., Bannister, P.R., De Luca, M., Drobnjak, I., Flitney, D.E., *et al.*: Advances in functional and structural mr image analysis and implementation as fsl. Neuroimage **23**, 208–219 (2004)