# g.citation: Scientific citation for individual GRASS GIS software modules

The authors introduce the GRASS GIS add-on module g.citation. The module extends the existing citation capabilities of GRASS GIS, which until now only provide for automated citation of the software project as a whole, authored by the GRASS Development Team, without reference to individual persons. The functionalities of the new module enable individual code citation for each of the over 500 implemented functionalities, including add-on modules. Three different classes of citation output are provided in a variety human- and machine-readable formats. The implications of this reference implementation of scientific software citation for both for the GRASS GIS project and the OSGeo foundation are outlined.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

# g.citation: Scientific citation for individual GRASS GIS software modules

**Peter Löwe[1], Vaclav Petras[2], Markus Neteler[3], and Helena Mitasova[4]**

[1]**DIW Berlin, Mohrenstraße 58, Berlin, Germany**
[2]**Center for Geospatial Analytics, Campus Box 7106, Raleigh, NC 27695, USA**
[3]**Mundialis GmbH u. Co. KG , Kölnstrasse 99, 53111 Bonn, Germany**
[4]**Center for Geospatial Analytics, Campus Box 7106, Raleigh, NC 27695, USA**

Corresponding author:
Peter Löwe[1]

Email address: ploewe@diw.de

## ABSTRACT

The authors introduce the GRASS GIS add-on module g.citation. The module extends the existing citation capabilities of GRASS GIS, which until now only provide for automated citation of the software project as a whole, authored by the GRASS Development Team, without reference to individual persons. The functionalities of the new module enable individual code citation for each of the over 500 implemented functionalities, including add-on modules. Three different classes of citation output are provided in a variety human- and machine-readable formats. The implications of this reference implementation of scientific software citation for both for the GRASS GIS project and the OSGeo foundation are outlined.

## INTRODUCTION: GRASS GIS - OVER 30 YEARS OF CONTINUOUS SOFTWARE DEVELOPMENT

GRASS GIS(GRASS GIS community, 2018) is a community driven software project already lasting over three decades with continuous community-driven development and maintenance efforts. Since 1983, the software has continuously evolved and its capabilities have been continuously extended according to the needs of the geospatial community. During this time, code management within the project also evolved: The project used manual source code management from 1983 until 1999, when the Concurrent Versions System (CVS)(The CVS Team, 2018) was introduced for revision control. Since 2006 the code management is based on Apache Subversion (SVN)(Apache Foundation, 2018) hosted by OSGeo(The GRASS GIS community, 2018a) . This is paralleled since 2015 by a subset of the development branch on GitHub(The GRASS GIS community, 2018b) .

While version control, including the tracking of code submissions by individuals, evolved over time, the capabilities of the GRASS GIS software to provide user-sided automated citation have not kept up with the current advances in software citation. A standard GRASS GIS 7.4 installation is only capable to generate a BibTex citation through the g.version module(The GRASS GIS Development Team, 2018a), which credits the GRASS Developer Team as authors of the whole GRASS GIS software system.

The term GRASS Development Team summarizes the community of individuals which have during the past and in the present developed and maintained the GRASS code base. Within the team, individuals have been and are taking on varying roles when interacting with the code base, including, but not limited to original developer and maintainer. All roles are significant to the GRASS GIS project and should receive recognition (both as team-, but also for the individual effort) by due credit when scientific results based on these efforts are published. Since GRASS GIS has been under continuous development for over three decades, for many long established GRASS GIS modules the number of persons involved in code maintance, extension and refactoring already exceed significantly the number of initial authors. A visual summary and overview of the development activities of the GRASS GIS codebase from 1999 to 2013 is given in(Markus Neteler, 2013)

It is necessary, to extend the GRASS GIS software by code-citation capabilities on the level of the

individual functionalities, which are implemented as GRASS GIS modules, to acknowledge the efforts particular members of the GRASS Development Team for the particular module.

Also, the developing best practices for software citation, especially metadata management, as currently being driven by communities like FORCE11(FORCE11 Community, 2011) or codemeta(The CodeMeta Project, 2018) remain to be acknowledged and adopted by the GRASS GIS community. This would allow to give credit to all stakeholders in the GRASS Development Team by state-of-the-art scientific citation practices.

## THE ROLE OF THE OSGEO FOUNDATION

The OSGeo foundation(The OSGeo Foundation, 2006) is an umbrella organisation which acts as a communication platform for a growing number of community driven geospatial open source projects since its founding in 2006. GRASS GIS, which preceded OSGeo by over two decades, was one of the founding projects and has ever since played an active role in shaping and advancing the OSGeo workflows and best-practices. The foundation has established common quality standards and best practices for projects, including social aspects of community governance and communication, but also technical aspects like coding standards and repository management. One central factor is the OSGeo incubation process, which is required for open source projects to become accredited within OSGeo. It is related to the Apache Foundation graduation process(The Apache Foundation, 2018) , and assesses the maturity of project processes, and their compliance with the values of OSGeo:

OSGeo embraces and fosters the paradigms of open source, open data, open standards and open education as the building blocks for open science (Wikipedia contributors, 2012a) . The foundation belongs to the signatories of the commitment statement of the Enabling FAIR Data project(Enabling FAIR Data Project, 2016) to enable FAIR data (including scientifc software) in earth, space and environmental science, committing itself to extend its support for the FAIR (Findable, Accessible, Interoperable, Reusable)(Enabling FAIR Data Project, 2018) .However, best practices for software citation remain to be included in the OSGeo incubation process.

## SOFTWARE DEVELOPMENT IN GRASS GIS

The portfolio of functionalities which are provided by the GRASS GIS software continues to grow. The adding of new functionalities, frequently triggered by science projects, results in additions to the GRASS GIS codebase. This requires a sequence of actions, which are related to code quality and license, access and repository management aspects: The code which implements the algorithm for the new functionality migrates over time from the authors personal domain (i.e. his or her local computing environment), to the community domain of the GRASS project for code review and lon term curation, paralleled by public access in the open access domain.

To initiate this process, the original code author has to consent to the basic rules of code development and adequate open source licensing(The GRASS GIS Development Team, 2018b) prior to uploading the code to the section of the GRASS software repository for experimental code (sandbox). The process is then continued by migrating the code to the add-on section of the repository, once functionality and code quality have been reviewed.

If the functionality provided by the code proves to be significant to the overall project, the code is migrated into the development branch of the GRASS codebase as a core module, to become a part of the next official GRASS release. This migration process is paralleled by iterative code quality assessment and improvement by the project community by public discussion, thorough review, refactoring and documentation according to the quality standards of the GRASS GIS project, in accordance to the best-practices of the greater OSGeo software ecotope.

Once a novel GRASS module has reached add-on module status, the GRASS add-on discovery functionality provided by the module g.extension(, original shell script) to import add-on modules makes it both discoverable and accessible to the global user community, allowing for large scale reuse.

When a functionality has become part of the main branch of the codebase, the task of code maintenance shifts from the original author to the GRASS Development Team. Participation of the authors(s) in the continuing maintenance and improvement effort is still appreciated, but not longer mandatory. The code will be continued to be maintained the GRASS Development Team after the original author(s) have left



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

the project. Over time, such well maintained and iteratively updated code can reach levels of structuring and performance beyond the programming skills of the original authors.

This is similar to paradox of the ship of Theseus(Wikipedia contributors, 2012b), which raises the question, if a wooden boat, which has had all physical parts replaced over time, is still identical to the vessel which was initially laid down. From the perspective of both the users and the GRASS Development Team, this is highly desirable and beneficial to the GRASS GIS project: In analogy to the ship of Theseus, the GRASS GIS project keeps rejuvenating its aging codebase in the face of evolving IT best practices and also extends its tonnage displacement by the growing number of included functionalities. The GRASS code repository ensures that all iterations of the GRASS GIS software (e.g. the many instances of Theseus ship) are kept available for future review and analysis.

## REWARD STRATEGIES IN SCIENCE AND SOFTWARE COMMUNITIES

Scientists, which develop research code based on GRASS GIS, which could be turned into new GRASS modules, must select at some point a strategy how to publish their code, which currently results in conflicts regarding the quality of reward, code maintenance and reuse by others.

The first strategy, by publishing the code as a new GRASS module in the GRASS GIS code repository, has already been described above. This strategy will result in re-use and potentially long term maintenance and praise by the GRASS community. However, the code author will only receive credit by citation if the prospective author of a scientific publication is determined enough to undertake the effort tp manually derive a citation from the credits on the manual page belonging to the GRASS GIS module. In this case it is also unlikely that the efforts of members of the GRASS Development Team acting in other roles than primary authors, ensuring long term usability, will receive any due credit, as most traditional citation standards do not enforce to address these software development roles.

The second strategy for the code author is to publish his or her novel GRASS-based code in an established scientific repository outside the GRASS GIS code repository, like those listed in the registry of research data repositories(The re3data Project, 2018). These repositories allow for reliable scientific citation through permanent persistent digital identifiers (PID) , like Digital Object Identifiers (DOI)(The DOI community, 2018) to reference to landing pages, instead of transient URL links to module man pages, as currently used by GRASS GIS and the other OSGeo projects.

However, from the established long term expectation for fitness for use by the GRASS community, this approach must be considered as ”dead from the start”: The task to further maintain the code in the chosen scientific repository must be shouldered entirely by the original authors, without the option of the GRASS Development Team to take over at some point. It can be anticipated that the original developers will cease to support the maintenance of their submitted code within relatively short time. The code archived in the repository will fossilize, by not being regularly updated, resulting in for compatibility with the evolving GRASS GIS code base, the need for later updates or re-implementations to make it executable in the future.

The GRASS GIS g.extension(, original shell script) module, which allows to integrate add-on modules to an existing GRASS GIS installation, provides the means to access GRASS add-on code from external code repositories, including RE3Data(The re3data Project, 2018) -listed scientific repositories like Zenodo(The Zenodo Community, 2015) . However, this requires existing prior knowledge by the prospective user where the particular module is stored and what it does. In addition, it is left to the user to assess the compatibility and trustworthiness of such unmaintained code in regard to the version of GRASS GIS currently being used. Since this applies to each user wishing to reuse the code, this can lead to repeated re-implementations over time.

## G.CITATION: SOFTWARE CITATION FOR GRASS GIS MODULES

The new g.citation module(Petras et al., 2018) complements the existing citation capabilities for GRASS GIS (g.version module) by supporting multiple scientific citation options on the granularity of particular GRASS modules1 in an automated and user-friendly way (Figure 2). This is a first step to overcome the current limitations of the GRASS GIS software regarding convenient and flexible citation capabilities, to increase the motivation for code submissions to the GRASS GIS code repository for scientists. While the development of the module is currently in its late experimental phase (GRASS sandbox code repository(Vaclav Petras, Peter Löwe, Markus Neteler and Helena Mitasova, 2018) ), it already supports three



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

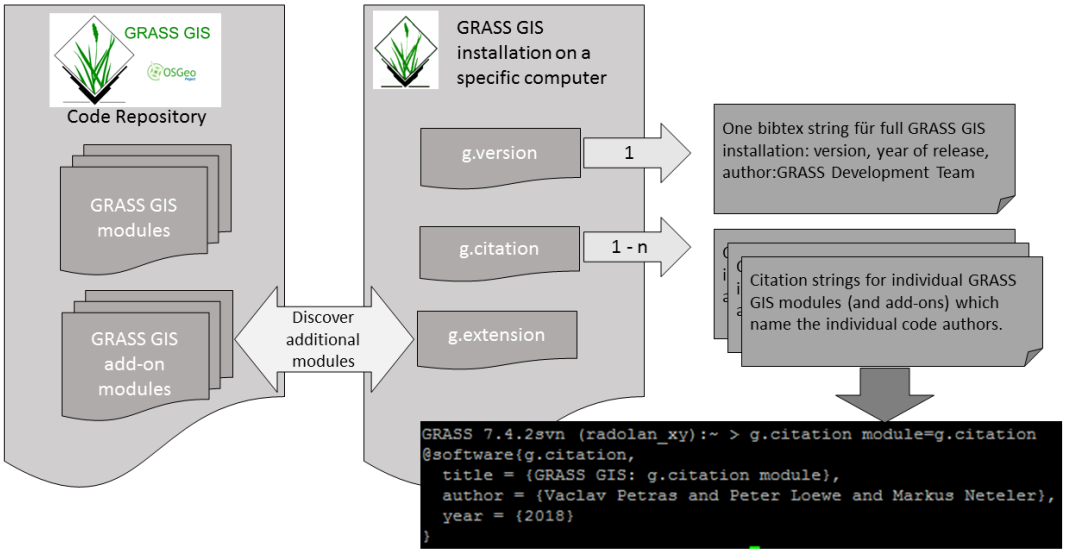


**Figure 1.** Overview of citation options for GRASS GIS: The GRASS GIS module g.version only provides a BibTeX citation string for the whole GRASS GIS installation, citing the GRASS Developer Team as author. The new g.citation module can be installed from the GRASS GIS code repository via the g.extension module. It provides citations of individual GRASS GIS modules (and add-ons) in multiple output formats. The BibTeX output of self-referential application is shown in the lower right of the figure.

distinct categories of citation options:

The first category are citation strings formatted for human use in a text processor, according to the formatting rules (e.g. Figure 3). The second category provides machine readable strings as input for reference management software used by humans for formatting lists of references (Figure 4), including a BibTex style (Figure 1). The third category provides well formatted metadata in XML-dialects like Citation File Format(Stephan Druskat, 2018) (CFF) (Figure 5) and Citation Style Language(CSL team, 2018) (upcoming), which are to be rendered by reference management tools and CSL-processors into a variety of citation styles, similar to citation rendering services already provided by scientific data repositories and citation infrastructures, as provided through the web portals of Zenodo or DataCite(Datacite team, 2018).

## NEXT STEPS

In addition to improve and extend the g.citation module regarding its functionality and code quality, several tasks related to the GRASS GIS project and the OSGeo foundation have already been identified, which can now be taken on because of the availability of g.citation.

The first task concerns the homogenization and improvement of the metadata within the GRASS GIS project: Currently, the code-related metadata, which is provided as human readayble content on the manual pages of individual GRASS GIS modules is of mixed quality regarding the listing of involved persons and their respective roles (e.g. original authors, maintainers, etc.): While best practices exist, a controlled vocabulary to describe the roles of members of the GRASS GIS Development Team is not defined yet. Either, no machine-harvestable metadata are provided on the module manual pages. This makes it computationally hard to derive well-formed citation strings. To mitigate this, the output from g.citation based on the current GRASS GIS manual pages can be used as input for a clean-up effort to homogenize and improve the structuring of the content of already existing GRASS GIS man-pages: As a follow-up step, it is intended to improve the GRASS GIS-internal code- and documentation management workflow by integrating a new layer of structured CFF-files with well-defined metadata attributes as the source for HTML-manual pages, enabling the latter to become machine-readable resulting in improved discoverability and scientific credit for content in the GRASS GIS code repository.

The second task is to establish code citation capabilities as a best practice for the OSGeo foundation:



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

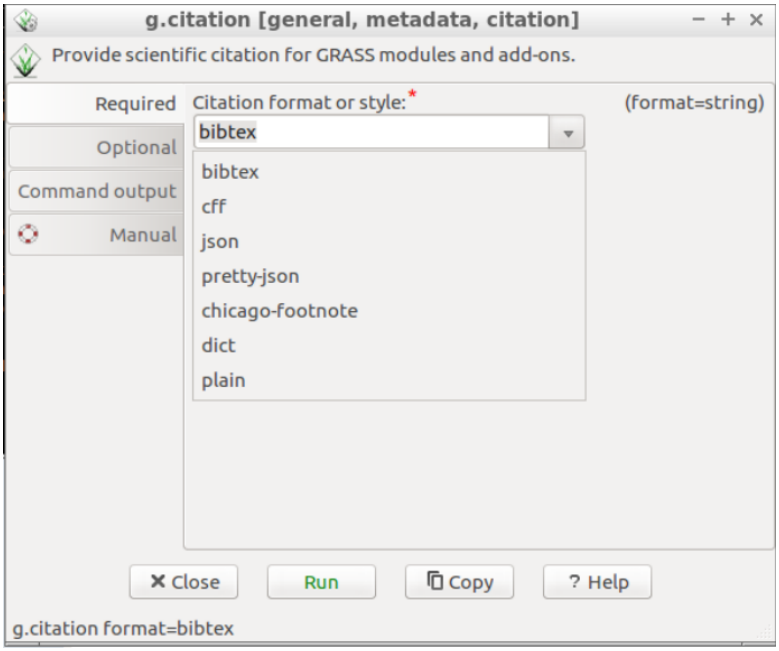


**Figure 2.** Screenshot of the GUI of the g.citation module, showing the currently included citation style options.

```
g.citation module=g.citation format=chicago-footnote

Vaclav Petras, Peter Loewe, and Markus Neteler,
GRASSS GIS module g.citation (7.4.2svn),
computer software (2018).
```

**Figure 3.** Example output for human readible output adhering the chicago footnote citation style, as to be used with word processors

Once g.citation becomes included into the main branch of the GRASS GIS codebase, GRASS GIS can become a role model within OSGeo for code citation: in a follow up step, OSGeo can elect to include the topic of code citation capabilities and best practices into incubation process check-list.

The third task is to evolve the software repositories of OSGeo software projects to meet the current requirements of scientific data repositories and to establish them as recognized scientific infrastructure. This will involve updates on the metadata schemata, inclusion of machine-harvestable metadata to improve discoverability from outside and replacement of potentially transient (i.e ensured to fail on the very long term time scale of libraries) URLs with persistent identifiers. Many aspects of the GRASS GIS project infrastructure already comply to these requirements, like the structuring of man-page content for GRASS modules, whose human readable content already meets the requirements for landing pages for DOI-referenced data sets. The GRASS GIS project could also become the driver for this within OSGeo.

## CONCLUSION

The GRASS GIS add-on module g.citation extends the existing functionality range of GRASS GIS by generating human- and machine-readable citation information for individual GRASS GIS modules. This allows to give due credit to the respective authors though a GRASS GIS functionality. This new functionality is a pragmatic step towards improvements of workflows and infrastructures within the GRASS GIS project, which can become examples for the greater OSGeo community. Based on this relatively small step, follow up efforts, which can have positive effects on larger scales, can be undertaken to homogenize the quality of metadata within the existing GRASS GIS code base, establish g.citation as a OSGeo-wide reference implementation, and make code citation capabilities a topic for the OSGeo incubation process.



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

**g.citation module=g.citation format=pretty-json**


```
{
  "authors": [
    {
      "institute": "NCSU GeoForAll Lab (ORCID: 0000-0001-5566-9236)",
      "name": "Vaclav Petras",
      "orcid": "0000-0001-5566-9236"
    },
    {
      "institute": "NCSU GeoForAll Lab (ORCID: 0000-0003-2257-0517)",
      "name": "Peter Loewe"
    },
    {
      "institute": "NCSU GeoForAll Lab (ORCID: 0000-0003-1916-1966)",
      "name": "Markus Neteler"
    }
  ],
  "code-url": "https://trac.osgeo.org/grass/browser/grass-
addons/grass7/general/g.citation",
  "grass-build-date": "2018-08-06",
  "grass-version": "7.4.2svn",
  "module": "g.citation",
  "url-code-history": "https://trac.osgeo.org/grass/log/grass-
addons/grass7/general/g.citation",
  "year": 2018
}
```


**Figure 4.** Example output both human and machine readible, in pretty printed Javascript object notation (json) to be used by humans as input for reference management systems for formatting lists of references.

## REFERENCES


Apache Foundation (2018). Concurrent versions system. [Online; accessed 13-September-2018].

CSL team (2018). Citation style language website. [Online; accessed 13-September-2018].

Datacite team (2018). Datacite website. [Online; accessed 13-September-2018].

Enabling FAIR Data Project (2016). Enabling fair data project website. [Online; accessed 13-September-2018].

Enabling FAIR Data Project (2018). Enabling fair data commitment statement in the earth, space, and environmental sciences. [Online; accessed 13-September-2018].

FORCE11 Community (2011). Force11 - future research communication and e-scholarship. [Online; accessed 13-September-2018].

GRASS GIS community (2018). Grass gis - bringing advanced geospatial technologies to the world. [Online; accessed 13-September-2018].

Markus Neteler (2013). Grass gis 6.4 development visualization from 1999 to 2013. [Online; accessed 13-September-2018].

(original shell script), M. N., Engineering, U. A. C., Landa, M., and Petras, V. (2017). Grass gis: g.extension module.

Petras, V., Loewe, P., and Neteler, M. (2018). Grass gis: g.citation module.

Stephan Druskat (2018). Citation file format (cff) github repository. [Online; accessed 13-September-2018].

The Apache Foundation (2018). Apache website. [Online; accessed 13-September-2018].

The CodeMeta Project (2018). Codemeta website. [Online; accessed 13-September-2018].

The CVS Team (2018). Concurrent versions system. [Online; accessed 13-September-2018].

The DOI community (2018). Doi website. [Online; accessed 13-September-2018].

The GRASS GIS community (2018a). Grass gis software repository. [Online; accessed 13-September-2018].

The GRASS GIS community (2018b). Grass gis software repository. [Online; accessed 13-September-




PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018

**g.citation module=g.citation format=cff**

```
cff-version: 1.0.3
message: "If you use this software, please cite it as below."
authors:
  - family-names: Petras
    given-names: Vaclav
    orcid: 0000-0001-5566-9236
  - family-names: Loewe
    given-names: Peter
  - family-names: Neteler
    given-names: Markus
title: "GRASS GIS: g.citation module"
version: 7.4.2svn
date-released: 2018-08-06
license: GPL-2.0-or-later
```

**Figure 5.** Example output both human and machine readable, in code citation format to be used for reference management software (used by humans) or as input for machine actionable citation harvesting by entities like DataCite.

2018].
The GRASS GIS Development Team (2018a). g.version. [Online; accessed 13-September-2018].
The GRASS GIS Development Team (2018b). How to contribute to grass gis (wiki). [Online; accessed 13-September-2018].
The OSGeo Foundation (2006). Osgeo website. [Online; accessed 13-September-2018].
The re3data Project (2018). re3data website. [Online; accessed 13-September-2018].
The Zenodo Community (2015). Zenodo research data repository. [Online; accessed 13-September-2018].
Vaclav Petras, Peter Löwe, Markus Neteler and Helena Mitasova (2018). g.citation preliminary code repository. [Online; accessed 13-September-2018].
Wikipedia contributors (2012a). Open science — Wikipedia, the free encyclopedia. [Online; accessed 13-September-2018].
Wikipedia contributors (2012b). Ship of theseus — Wikipedia, the free encyclopedia. [Online; accessed 13-September-2018].



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27206v1 | CC BY 4.0 Open Access | rec: 14 Sep 2018, publ: 14 Sep 2018