



Create software deposit in HAL

Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault, Alain Monteil

► To cite this version:

Morane Gruenpeter, Jozefina Sadowska, Estelle Nivault, Alain Monteil. Create software deposit in HAL: User guide and best practices. [Technical Report] Inria; CCSD; Software Heritage. 2022. hal-01872189v2

HAL Id: hal-01872189

<https://inria.hal.science/hal-01872189v2>

Submitted on 13 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike| 4.0 International License

Create software deposit in HAL

User guide and best practices

Version 2.0 (March 2022)

Authors :

- Morane Gruenpeter
morane@softwareheritage.org
- Jozefina Sadowska
jozefina.sadowska@inria.fr
- Estelle Nivault
estelle.nivault@inria.fr
- Alain Monteil
alain.monteil@inria.fr



HAL
open science

CCSD . . .

Inria



Software Heritage

Table of contents: Create software deposit in HAL

[Introduction](#)



[Checklist](#)



[Chapter 1: Prepare your source code](#)



Local method: .zip /.tar.gz



SWHID method: with the SoftWare Heritage ID

[Chapter 2: Deposit local source code archive to HAL](#)

[Chapter 3: Deposit SoftWare Heritage ID \(SWHID\) with metadata](#)



[Chapter 4: behind the scenes - the moderation process](#)



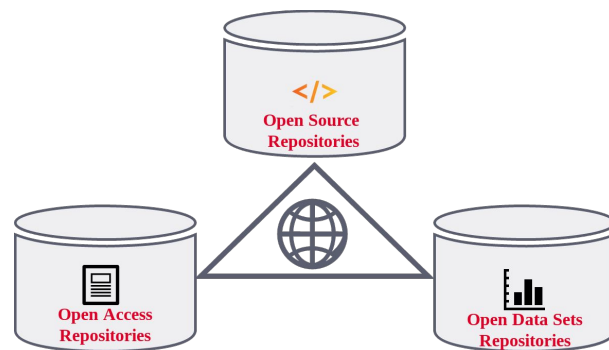
[Chapter 5: The deposit publication and exports](#)



[References](#)

Introduction: why deposit on HAL?

- ★ **Archive** software in HAL and in SWH
 - Better **visibility** for software in open science context
 - **Long term preservation** HAL and SWH have a long term preservation service
- ★ **Identify**
 - the software artifacts with a **SWHID** (SoftWare Heritage Identifier)
 - the metadata record and citation with the **HAL-ID**
- ★ **Describe** source code with verified metadata
 - **Moderation** and control of the metadata by librarians and/or curators
- ★ **Cite** the software deposit with a complete citation
 - Several **exports format** make it easier for citation



The three pillars of Open Science, Software Heritage CC-BY 4.0 2019

Introduction: What software object to deposit in HAL?

- ★ The **source code** of the software (not compatible for executables)
- ★ Software that was developed in **academia** for research purposes
- ★ Only the software **creators/authors** of the software or their representatives can deposit software in HAL

Use cases

I **develop** my software **locally**
And I share my code on my personnel website or my institution's website

Local method: deposit .zip /.tar.gz

I **develop** my software on a **collaborative plateforme** using a **version control system** (on GitHub, GitLab, Bitbucket...)

SWHID method: deposit SWHID with metadata

Introduction: The deposit steps

- Deposit on HAL
 - ◆ Local method or the SWHID method
- Validation of the form by the contributor
- Deposit in progress waiting for **verification**
 - ◆ **Dialogue** between contributor and moderator
- **Validation** of the deposit by the moderator
- Deposit **published on HAL** and **transfer to SWH**
- Export formats
 - ◆ Citation
 - ◆ BibTeX
 - ◆ codemeta.json
 - ◆ TEI

Checklist depending on use case

Local method: deposit .zip / .tar.gz

1. **Prepare your source code (locally)**
 - ☐ AUTHORS, LICENSE & README files
 - ☐ Compress documents into .zip / .tar.gz
2. **Deposit compressed archive**
3. **Complete metadata**
 - ☐ Choose deposit type
 - ☐ Add generic metadata
 - ☐ Add software specific metadata
 - ☐ Add authors
 - ☐ Validate deposit

SWHID method: deposit SWHID with metadata

1. **Prepare your source code (in code repository)**
 - ☐ AUTHORS, LICENSE & README files
 - ☐ Codemeta.json file
 - ☐ [Save code now](#) on Software Heritage
 - ☐ Choose SoftWare Heritage IDentifier (SWHID)
2. **Deposit SWHID in HAL interface**
3. **Complete metadata**
 - ☐ Add domain
 - ☐ Control entries from codemeta
 - ☐ Add authors
 - ☐ Validate deposit

Chapter 1:

Prepare the source code for archival

1.1 Prepare your code

1.1.1 Add the following files :

these *elements* are *verified* by the *moderators*

- ☐ README
- ☐ AUTHORS
- ☐ LICENSE (Choose with the right holders of the software - [Here you can find the SPDX reference list of licenses](#))



talon
Dossier - 472 Ko

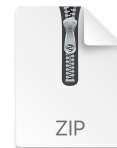
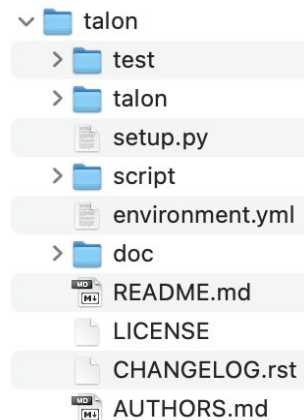
1.1.2 Local method: create a compressed archive

.zip ou .tar.gz

It is preferable to name the compressed file with the software name and version number

1.1.3 SWHID method: use your version control system

local method: deposit .zip / .tar.gz



talon-source.zip
Archive dans un fichier ZIP - 247 Ko

Updating content on the version control system



SWHID method: deposit SWHID with metadata

1.1.1 README file



Mandatory:

- Software / project name
- Description of the software



Recommended:

- Website
- Link to the documentation
- Contact & support
- List of functionalities
- Development environment
 - build, installation, requirements
 - How to execute the code



Possible:

- Usage - How to use the code
- News about the project
- Visuals

This is Python version 3.8.0 alpha 0

```
.. image:: https://travis-ci.org/python/cpython.svg?branch=master
:alt: CPython build status on Travis CI
:target: https://travis-ci.org/python/cpython

.. image:: https://ci.appveyor.com/api/projects/status/4mew1a93xdkbf5ua/branch/master?svg=true
:alt: CPython build status on Appveyor
:target: https://ci.appveyor.com/project/python/cpython/branch/master

.. image:: https://dev.azure.com/python/cpython/_apis/build/status/Azure%20Pipelines%20CI?branchName=master
:alt: CPython build status on Azure DevOps
:target: https://dev.azure.com/python/cpython/_build/latest?definitionId=4&branchName=master

.. image:: https://codecov.io/gh/python/cpython/branch/master/graph/badge.svg
:alt: CPython code coverage on Codecov
:target: https://codecov.io/gh/python/cpython

.. image:: https://img.shields.io/badge/zulip-join_chat-brightgreen.svg
:alt: Python Zulip chat
:target: https://python.zulipchat.com
```

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018 Python Software Foundation. All rights reserved.

See the end of this file for further copyright and license information.

.. contents::

General Information

```
- Website: https://www.python.org
- Source code: https://github.com/python/cpython
- Issue tracker: https://bugs.python.org
- Documentation: https://docs.python.org
- Developer's Guide: https://devguide.python.org/
```

Contributing to CPython

For more complete instructions on contributing to CPython development, see the "Developer Guide".

.. _Developer Guide: https://devguide.python.org/

Using Python

Installable Python kits, and information about using Python, are available at `python.org`.

Readme from the CPython project archived on SWH :

swh:1:cnt:970f62e6938908a9caaaa0a07fe425bd3976c101:origin=https://github.com/python/cpython/

1.1.2 AUTHORS files

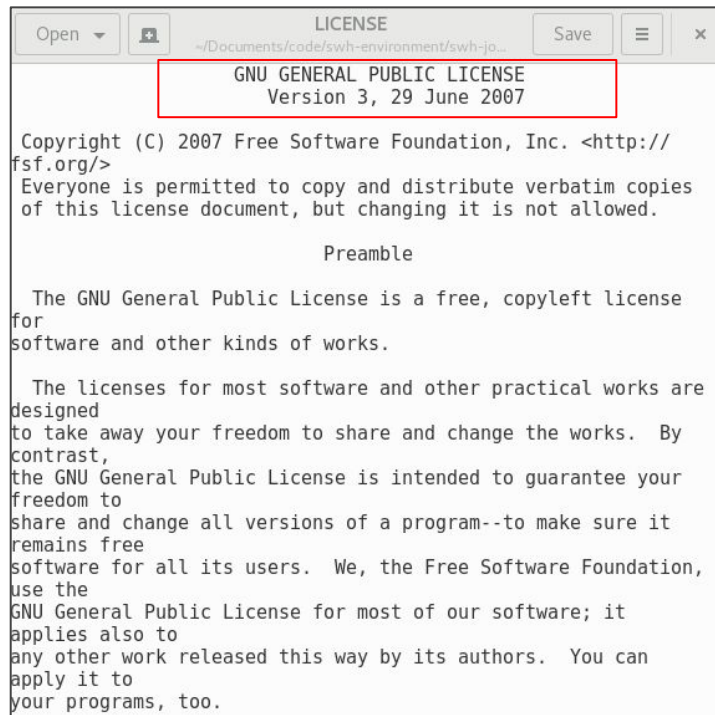
- ★ Identify the people involved in the creation of the software who should be credited for the creative work
 - Others contributors can be identified in a separate file or list, called “contributors”
- ★ Note all the authors in the file and on the form
- ★ The people that are noted on the HAL form and on the AUTHORS file, have authors right on the source code
- ★ You can use the following file names for this file: AUTHORS, AUTHORS.md, AUTHOR.rst, CONTRIBUTORS, CREDITS, CITATION, CITATION.cff, etc.

Open  AUTHORS
~/

```
The wonderland team (1953-1960)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Captain Hook <c.hook@wonderland.org>
The Crocodile/Tic Toc <croco@wonderland.org>
George Darling <g.darling@wonderland.org>
John Darling <g.darling@wonderland.org>
Mary Darling <ma.darling@wonderland.org>
Michael Darling <mi.darling@wonderland.org>
Nana <nana@wonderland.org>
Peter Pan <peter.pan@wonderland.org>
Mr. Smee <m.smee@wonderland.org>
Tiger Lily <tiger.lily@wonderland.org>
Tinker Bell <tinker.belg@wonderland.org>
Wendy Darling <w.darling@wonderland.org>
Jane <jane@wonderland.org>
Danny <danny@wonderland.org>
Edward <edi@wonderland.org>
```

1.1.3 Fichier LICENSE

- ★ Before depositing the source code, you must choose a license with the appropriate rights owner
 - at INRIA, the CPPI is the best service to discuss the license question
 - Consult the appropriate service in your institution
- ★ If you have more than one license create a `LICENSES/` directory with all licenses
- ★ Note the same license/s in the HAL form's metadata
 - The compatibility between the license in the form and in the code is verified by the moderators
 - The **contributor is responsible** of the compatibility between licenses (between code and dependencies)



Resources to help review different licenses & practices:

- <https://choosealicense.com/>
- <https://reuse.software/>

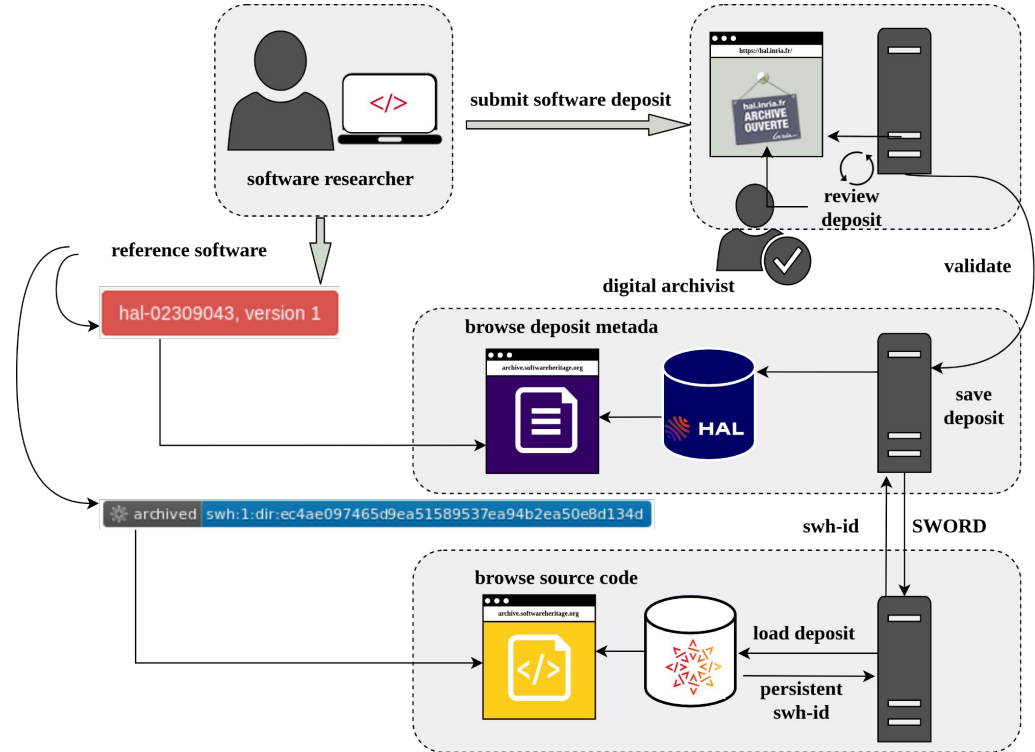
Chapter 2:

Deposit local source code

local method: `deposit .zip /.tar.gz`

The local method deposit

- One compressed **archive** containing source code (mostly text files)
- A collection of **metadata**
 - ◆ Generic metadata
 - ◆ Software specific metadata



2.1: Deposit the files

After logging in into HAL, use the “submit” tab , Drag and drop or click to upload compressed file.

Note that it is not possible for the software deposit to submit more than one compressed file.

The archive (.zip, .tar.gz) MUST contain all files and shouldn't be decompressed before validation.

The archive (.zip, .tar.gz) shouldn't contain another archive (.zip, .tar.gz).

The screenshot displays the HAL science ouverte website. The top navigation bar includes links for Accueil, Dépôt, Consultation, Recherche, Documentation, and Mon espace. The main header features the HAL logo and the tagline "La connaissance libre et partagée". Below the header, the "Déposer le(s) fichier(s)" section is visible. It contains a large dashed box for file upload with a cloud icon and a document icon, labeled ".zip / .tar.gz". Text inside the box instructs users to "Glissez-déposez ou cliquez pour choisir vos fichiers" and notes the "Taille maximale du fichier : 200M". To the right, there is a section titled "Chargez les métadonnées à partir d'un identifiant" with a text input field containing a SWHID and a "Récupérer les métadonnées" button. A small note at the bottom of the interface states: "Si vous souhaitez ajouter un embargo, récupérer des fichiers de votre espace FTP ou toute autre action avancée, veuillez afficher la vue détaillée."

2.2: Complete the metadata

→ 2.2.1 Choose the document type* :

Software

To see the full list of properties check the checkbox on the top right

Mandatory properties are identified with a star ()*

→ 2.2.2 Add the general properties :

- Title*
- Domain*
- Description (Il est recommandé d'avoir une description cohérente avec le contenu du fichier README)
- Keywords
- Identifiers
- Related data
- Associated publications
- Production date
- Publisher
- Institution

The screenshot shows a web form for submitting software metadata. At the top right, there is a checkbox labeled "Afficher la liste complète des métadonnées" which is checked. Below this, an orange banner contains the text: "Avant tout dépôt de code source, vous devez réfléchir à la licence que vous souhaitez pour votre code. Chez INRIA, votre CPPI est votre interlocuteur pour discuter de ces questions." The form has several sections: "Type de document *" with a dropdown menu set to "Logiciel"; "Nom *" with a text input field containing "TALON: Tractograms As Linear Operators in Neuroimaging" and language selection buttons for "anglais" and "français"; "Domaine *" with a list of domains including "Imagerie médicale" and "Traitement du signal et de l'image [eess.SP]", and an "Ajouter un domaine" button; and "Description" with a text input field containing a detailed description of the TALON software and its functions, also with language selection buttons for "anglais" and "français".

→ 2.2.3 Add **software specific metadata** :

- Licenses* (based on the SPDX reference list, it is also possible to enter a license that is not in this list)
- Programming language
- Code repository
- Platform/OS - environment
- Version
- Development status
- Runtime Platform

Licenses *	La ou les licences sous lesquelles est publié ce logiciel (vous pouvez utiliser l'autocomplétion)
	MIT License 
	
Langage de programmation	Python 
	
Code Repository	Lien où se trouve le développement du code (SVN, github, gitlab, CodePlex).
	https://gitlab.inria.fr/cobcom/talon
Platform/OS	Le système d'exploitation compatible avec le logiciel
	
Version	la version du logiciel (peut être différente de la version publiée sur HAL)
	0.3.0
Etat du développement	L'état du développement du logiciel au moment du dépôt (Concept, WIP, Suspendu, Actif, Inactif)
	active
Outils de développement	Les outils de développement associés au logiciel (Framework, middleware, plateforme logicielle)
	

2.3 Complete information about author(s)

- Add the author(s)
- Add affiliation for each author (*at least one author must be affiliated*)

- It is possible to add different authors and add a role (development, maintenance, design, architecture, debugging, documentation, test, support, management).

- It is necessary to add all authors from the AUTHORS file.

- You may add a CONTRIBUTORS file for people who participated in the creation of the software but are not considered authors.

The screenshot shows a web interface titled "Compléter les données auteur(s)" with a green checkmark. It lists four authors: Matteo Frigo, Mauro Zucchelli, Rachid Deriche, and Samuel Deslauriers-Gauthier. Each author has a list of affiliations (ATHENA, UCA) and a button to "Ajouter une affiliation". A dropdown menu for "Afficher toutes les options" is open, showing "Modifier", "Supprimer", and "Choisir la fonction". A modal window for "Matteo Frigo" is also open, showing a list of roles: Développement (checked), Maintenance, Design, Architecture, Débogage, Documentation, Test, Support, and Management.

Auteur	Rôle(s)	Affiliations
Matteo Frigo	Développement	ATHENA - Computational Imaging of the Central Nervous System, UCA - Université Côte d'Azur
Mauro Zucchelli		ATHENA - Computational Imaging of the Central Nervous System, UCA - Université Côte d'Azur
Rachid Deriche		ATHENA - Computational Imaging of the Central Nervous System, UCA - Université Côte d'Azur
Samuel Deslauriers-Gauthier		ATHENA - Computational Imaging of the Central Nervous System, UCA - Université Côte d'Azur

2.4 Submit the deposit

- Accept transfer to Software Heritage and contribute to the largest software source code archive in the world.
- Accept HAL's conditions
- Click on “Upload” to submit

 Valider le dépôt 

Logiciel

Matteo Frigo, Mauro Zucchelli, Rachid Deriche, Samuel Deslauriers-Gauthier. TALON: Tractograms As Linear Operators in Neuroimaging. 2021

 Software Heritage

Voir les conditions pour le transfert

Software Heritage a pour objectif de collecter, préserver, et rendre accessible, à tous, le code source de tous les logiciels disponibles.

Pour pouvoir transférer votre logiciel dans la plus grande archive de code source, votre dépôt doit satisfaire les conditions suivantes :

- Les fichiers déposés doivent être sous une licence libre.
- Les fichiers ne peuvent pas être sous embargo.

L'accès à votre dépôt sur Software Heritage sera disponible dans un délais de 7-30 jours (le temps de traitement de votre dépôt)

Conditions

En déposant ce document, le contributeur (je) accorde la licence suivante à HAL :

- J'autorise HAL à mettre en ligne et à distribuer cet article ;
- Je reconnais avoir pris connaissance que les dépôts ne peuvent pas être supprimés, une fois acceptés ;
- Je comprends que HAL se réserve le droit de reclasser ou de rejeter tout dépôt.

☒ J'accepte ces conditions

Vider

Annuler

 Déposer

Chapter 3:

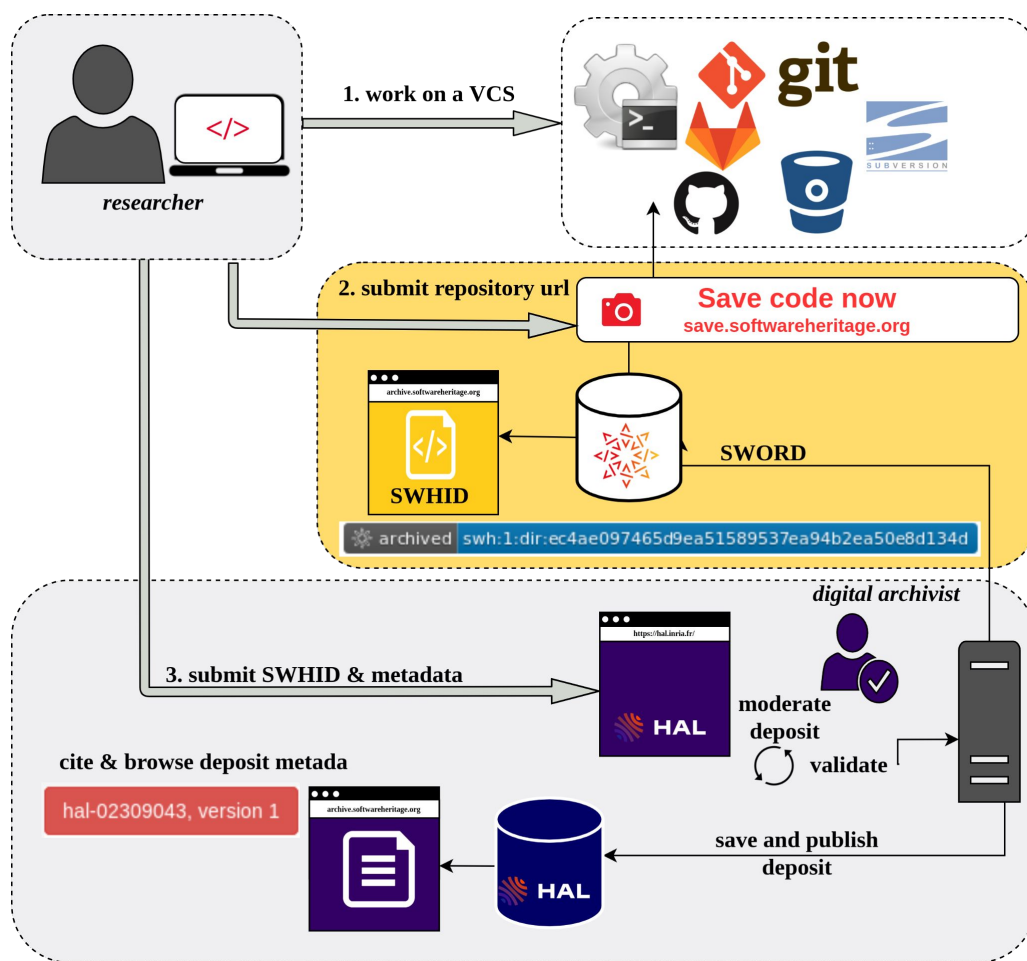
Deposit SoftWare Heritage ID (SWHID) with metadata

SWHID method: deposit SWHID with metadata

SWHID method: deposit SWHID with metadata

The SWHID deposit

- The SWHID deposit is composed by:
- A SoftWare Heritage identifier (**SWHID**) that can be found on the Software Heritage archive or calculated locally. The SWHID references the source code.
- A metadata collection
 - ◆ the metadata can be **pulled** into the form using the **SWHID**. The properties available on **codemeta.json** file in the root directory of your code can be extracted automatically on HAL's form.
 - ◆ **filling** the form with metadata properties to complete the form



*VCS = Version Control System

3.1 Prepare the source code

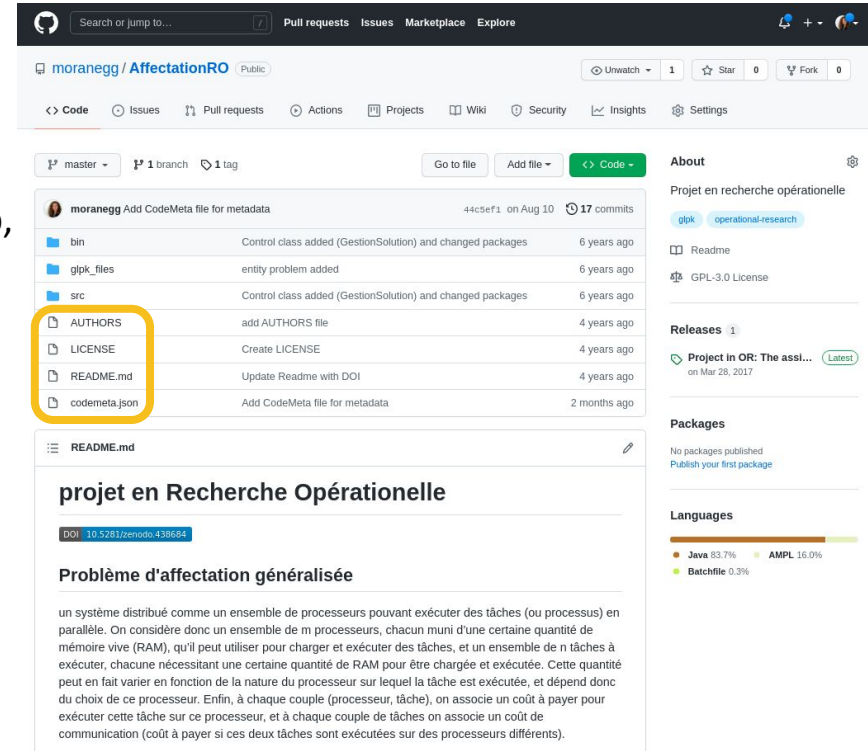
Before depositing on HAL you must prepare your source code on the collaborative development platform (GitHub, Gitlab, etc.).

The following elements are verified by a moderator

3.1.1 Add the following files to your source code and push to your `code repository`:

- ❑ README (see section [1.1.1 README file](#))
- ❑ AUTHORS (see section [1.1.2 AUTHORS files](#))
- ❑ LICENSE (see section [1.1.3 Fichier LICENSE](#))

- ❑ **codemeta.json** - isn't mandatory but useful to complete the form especially when using the SWHID method.



3.1.2 Why CodeMeta ?

- A vocabulary extending schema.org
 - <https://codemeta.github.io/terms/>
- An [academic community](#)
- A [crosswalk table](#) enabling translations between different ontologies/vocabularies to CodeMeta

Tool to create a codemeta.json file

CodeMeta generator

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name

the software title

Description

Creation date

First release date

To create easily a codemeta.json file use the [online tool](#)

➤ [You can contribute here](#)

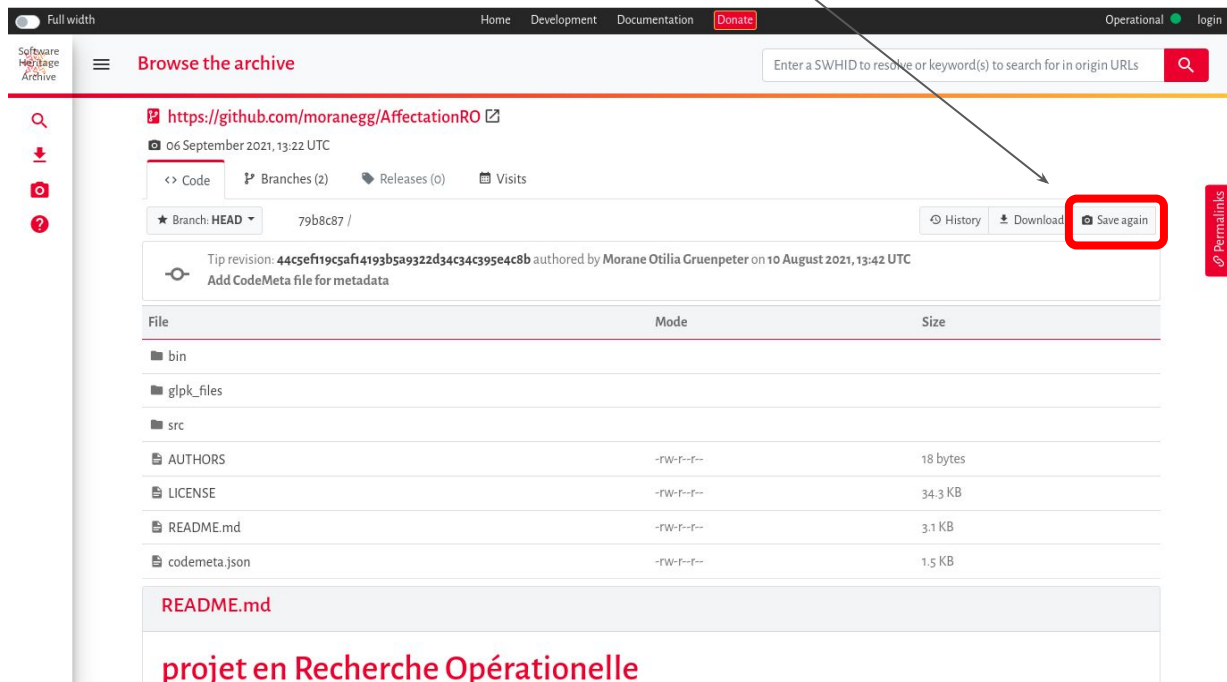
3.1.3 CodeMeta file: an example

```
{
  "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
  "@type": "SoftwareSourceCode",
  "license": "https://spdx.org/licenses/AGPL-3.0",
  "codeRepository": "https://github.com/moranegg/AffectationRO",
  "dateCreated": "2014-01-01",
  "dateModified": "2019-06-26",
  "name": "AffectationRO- The assignment problem",
  "version": "1.0.0",
  "description": "A java implementation for The Assignment Problem a distributed system as a set of processors that can perform tasks (or processes) in parallel. We therefore consider a set of m processors, each equipped with a certain amount of random access memory (RAM).\\r\\n We associate a cost to pay to perform this task on this processor, and each pair of tasks is associated with a communication cost. The Assignment problem works on minimizing the cost and maximizing the tasks performed.",
  "applicationCategory": "info",
  "releaseNotes": "First release with GLPK, in Beta testing",
  "developmentStatus": "concept",
  "keywords": [
    "distributed systems",
    "glpk",
    "optimisation",
    "OR"
  ],
  "programmingLanguage": [
    "Java"
  ],
  "author": [
    {
      "@type": "Person",
      "@id": "https://orcid.org/0000-0002-9777-5560",
      "givenName": "Morane",
      "familyName": "Gruenpeter",
      "email": "morane.gg@gmail.com",
      "affiliation": {
        "@type": "Organization",
        "name": "Software Heritage"
      }
    }
  ]
}
```


3.2 Save your code on Software Heritage

<https://archive.softwareheritage.org/>

3.2.1 Verify if your **code repository** and the **specific version** you want to submit are already in **Software Heritage**



Full width Home Development Documentation Donate Operational login

Software Heritage Archive

Browse the archive

Enter a SWHID to resolve or keyword(s) to search for in origin URLs

https://github.com/moranegg/AffectationRO

06 September 2021, 13:22 UTC

<> Code Branches (2) Releases (0) Visits

★ Branch: HEAD 79b8c87 / History Download Save again

Tip revision: 44c5ef19c5af14193b5a9322d34c34c395e4c8b authored by Morane Otília Gruenpeter on 10 August 2021, 13:42 UTC
Add CodeMeta file for metadata

File	Mode	Size
bin		
glpk_files		
src		
AUTHORS	-rw-r--r--	18 bytes
LICENSE	-rw-r--r--	34.3 KB
README.md	-rw-r--r--	3.1 KB
codemeta.json	-rw-r--r--	1.5 KB

README.md

projet en Recherche Opérationnelle

3.2.2 If your code isn't in Software Heritage- **Submit** the code repository url on the **Save Code Now**:

<https://save.softwareheritage.org/>

The screenshot shows the 'Save code now' page on the Software Heritage website. The page has a sidebar on the left with the 'Software Heritage Archive' logo and a 'Features' section containing links for 'Search', 'Downloads', 'Save code now' (which is highlighted), and 'Help'. The main content area is titled 'Save code now' and includes a search bar for SWHIDs or keywords. Below this, a form is provided for submitting code repository information. The form consists of two main fields: 'Origin type' and 'Origin url'. The 'Origin type' field is a dropdown menu currently showing 'git'. The 'Origin url' field is a text input box. A 'Submit' button is located to the right of the 'Origin url' field. Three numbered callouts are present: '1. Choose the VCS type' points to the 'git' dropdown; '2. Add url' points to the 'Origin url' text box; and '3. Submit' points to the 'Submit' button. The form is outlined with a yellow border.

Software Heritage Archive

Save code now

Enter a SWHID to resolve or keyword(s) to search for in origin URLs

You can contribute to extend the content of the Software Heritage archive by submitting code repository information. To do so, fill the required info in the form below:

Origin type Origin url

git

Submit

1. Choose the VCS type

2. Add url

3. Submit

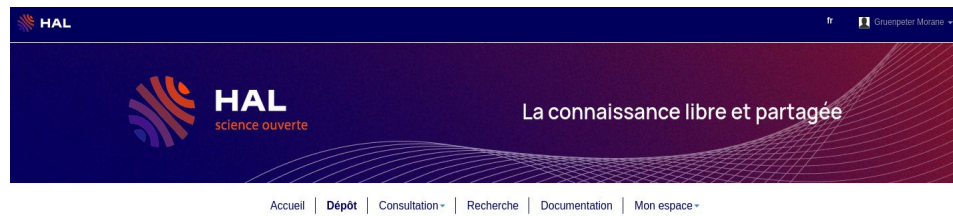
3.3 Choose a SWHID on the Software Heritage archive

The screenshot shows the Software Heritage archive interface. The top navigation bar includes 'Home', 'Development', 'Documentation', and a 'Donate' button. A search bar at the top right prompts the user to 'Enter a SWHID to resolve or keyword(s) to search for in origin URLs'. The main content area displays a repository page for 'https://github.com/moranegg/AffectionationR0'. The 'Permalinks' tab is selected, showing a list of object types: 'file', 'directory', 'revision', and 'snapshot'. The 'directory' type is highlighted with a red box. Below this, a SWHID is displayed: 'swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d'. A blue box highlights the 'Add contextual information' checkbox, which is checked. To the right, a blue box highlights the 'Copy identifier' button. The bottom of the page shows a 'README.md' file and a footer with the text 'projet en Recherche Opérationnelle'.

1. Click on `Permalinks` tab
2. Choose the object type - `directory`
3. Add the contextual information
4. Copy identifier on the HAL form

3.4 Deposit SWHID - the reference to the content

- Put the **SWHID** on the HAL form
 - ◆ Prefer a SWHID with contextual information (to keep the link with the contextual information)
 - ◆ If a codemeta.json is present in the root directory of the deposited SWHID - the HAL platform will pull the metadata automatically
- Verify and complete metadata
 - ◆ Verify inserted metadata
 - ◆ Complete missing metadata
 - ◆ Choose domain
 - ◆ Verify authors and add affiliations
- Validate deposit

The image shows a screenshot of the HAL deposit form. On the left, there is a dashed box with a cloud icon and the text 'Glissez-déposez ou cliquez pour choisir vos fichiers' and 'Taille maximale du fichier : 200M'. On the right, there is a section titled 'Chargez les métadonnées à partir d'un identifiant'. Below this title, there is a text input field containing the SWHID 'swf:1.doi:79b8c8755dbed3401a6a7184ffc1963c58cb5d.origi'. This field is highlighted with a red rectangle. Below the input field is a blue button labeled 'Récupérer les métadonnées'. At the bottom of the form, there is a small note: 'Si vous souhaitez ajouter un embargo, récupérer des fichiers de votre espace FTP ou toute autre action avancée, veuillez afficher la vue détaillée'.

SWHID

Chapitre 4:

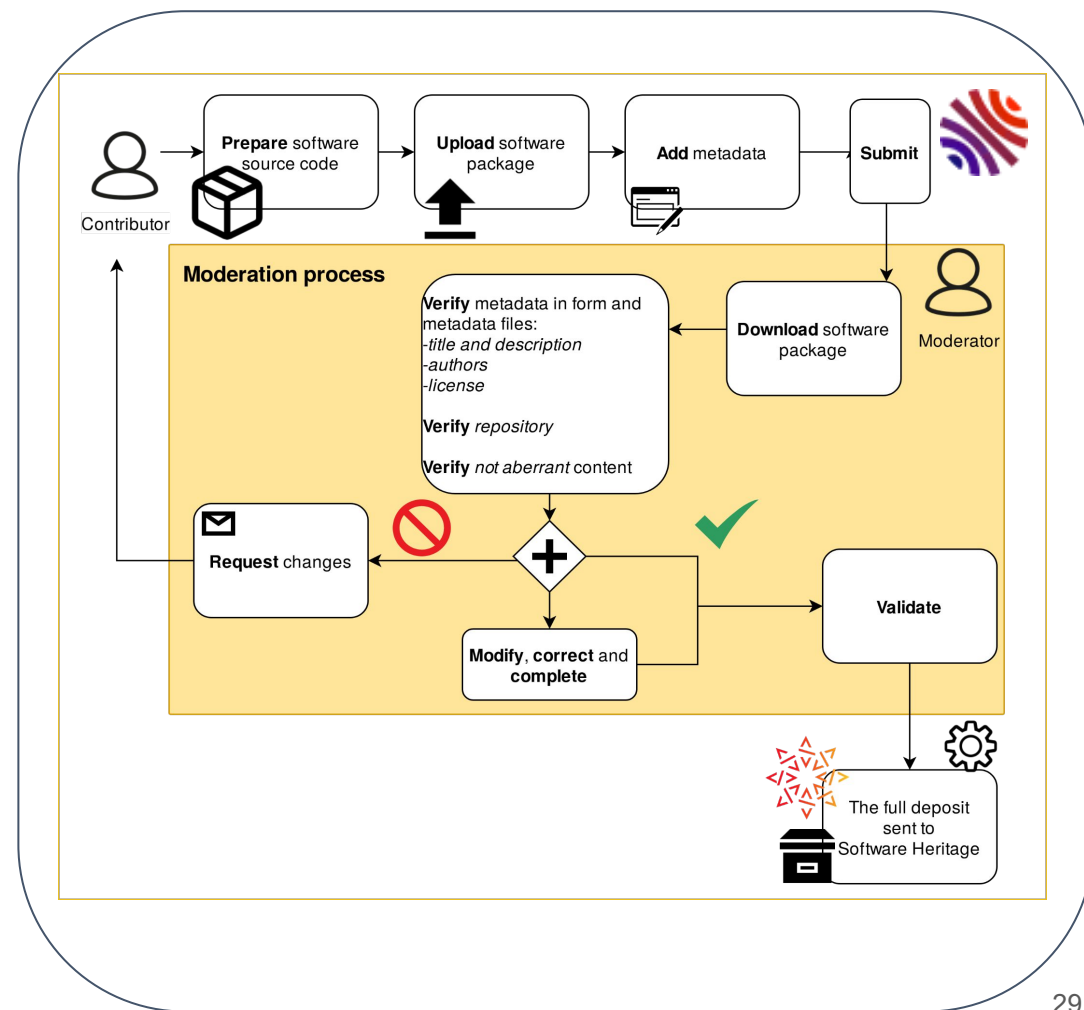
Behind the scenes- the moderation process

The moderation process

- Deposit in **moderation**
- **Dialogue** between contributor and moderator
- Deposit **validation** in moderation

Moderation guide :

Morane Gruenpeter, Jozefina Sadowska. La modération d'un dépôt logiciel : Bonnes pratiques et guide pour le modérateur. [Rapport Technique] Inria; CCSD; Software Heritage. 2018. [hal-01876705](https://hal.archives-ouvertes.fr/hal-01876705)



Chapitre 5:

The deposit publication and exports

5.1 The deposit is transferred to SWH

→ The local method deposit is transferred to SWH with:

- ◆ Content
- ◆ Metadata

→ The SWHID method deposit is transferred to SWH with:

- ◆ Metadata

When the local method deposit is transferred to SWH it will be visible on HAL without the link to SWH. When it is ingested by SWH, the link will appear on the deposit

hal-02522751, version 1

TALON: Tractograms As Linear Operators in Neuroimaging

Matteo Frigo^{1,2}, Mauro Zucchelli^{1,2}, Rachid Deriche^{1,2}, Samuel Deslaurliers-Gauthier^{1,2} [Détails](#)

1 ATHENA - Computational Imaging of the Central Nervous System
CRISAM - Inria Sophia Antipolis - Méditerranée

2 UCA - Université Côte d'Azur

Abstract : TALON is a pure Python package that implements Tractograms As Linear Operators in Neuroimaging. The software provides the TALON Python module, which includes all the functions and tools that are necessary for filtering a tractogram. In particular, specific functions are devoted to: - Transforming a tractogram into a linear operator. - Solving the inverse problem associated to the filtering of a tractogram.

Type de document : [Logiciel](#)

Domaine : [Informatique \[cs\]](#) / [Imagerie médicale](#)
[Informatique \[cs\]](#) / [Traitement du signal et de l'image \[eess.SP\]](#)

Liste complète des métadonnées [Voir](#)

TÉLÉCHARGER

Fichier : talon-source.zip [Télécharger](#)

<https://hal.halpreprod.archives-ouvertes.fr/hal-02522751>
Contributeur : [Emilie Nivault](#) [Contacter le contributeur](#)
Soumis le : vendredi 28 mai 2021 - 16:02:01
Dernière modification le : lundi 31 mai 2021 - 18:08:40

MÉTADONNÉES

Keywords : [Diffusion MRI](#) [dMRI](#) [tractography](#) [python](#)
[optimization](#)

version [0.3.0](#)

Licences [MIT License](#)

Langage de programmation [Python](#)

Code Repository <https://gitlab.inria.fr/cobcom/talon>

Outils de développement [Active](#)

CITATION

Matteo Frigo, Mauro Zucchelli, Rachid Deriche, Samuel Deslaurliers-Gauthier. TALON: Tractograms As Linear Operators in Neuroimaging. 2021. (hal-02522751)

EXPORTER

[CodeMeta](#) [BibTeX](#) [TEI](#) [DC](#) [DCterms](#)
[EndNote](#)

PARTAGER

[f](#) [t](#) [e](#) [+](#)

5.2 The final deposit

hal-02522751, version 1

TALON: Tractograms As Linear Operators in Neuroimaging

Matteo Frigo^{1,2}, Mauro Zucchelli^{1,2}, Rachid Deriche^{1,2}, Samuel Deslauriers-Gauthier^{1,2} [Détails](#)

1 ATHENA - Computational Imaging of the Central Nervous System
CRISAM - Inria Sophia Antipolis - Méditerranée

2 UCA - Université Côte d'Azur

Abstract : TALON is a pure Python package that implements Tractograms As Linear Operators in Neuroimaging. The software provides the `TALON` Python module, which includes all the functions and tools that are necessary for filtering a tractogram. In particular, specific functions are devoted to: - Transforming a tractogram into a linear operator. - Solving the inverse problem associated to the filtering of a tractogram.

Type de document : **Logiciel**

Domaine : **Informatique [cs] / Imagerie médicale**
Informatique [cs] / Traitement du signal et de l'image [eess.SP]

Liste complète des métadonnées [Voir](#)

CONSULTER

 Software Heritage

swlh:1:dir:f25157ad1b13cb20ac3457d4f6756b49ac63d079;origin=https://inria.halpreprod.archives-ouvertes.fr/hal-02522751;visit=swlh:1:snp:8a2cb6ecd1478c63550e524a5723e06597259a07;anchor=swlh:1:rev:5c9642143d7f1c22e1bb641561e210eb52a94;path=/
[Consulter](#)

<https://hal.halpreprod.archives-ouvertes.fr/hal-02522751>
Contributeur : Estelle Nivault Connectez-vous pour contacter le contributeur
Soumis le : vendredi 28 mai 2021 - 16:02:01
Dernière modification le : mercredi 2 juin 2021 - 13:59:59

MÉTADONNÉES

Keywords : **Diffusion MRI** **dmRI** **tractography** **python**
optimization

version
0.3.0

Licences
MIT License

Langage de programmation
Python

Code Repository
<https://gitlab.inria.fr/cobcom/talon>

Outils de développement
Active

CITATION

Matteo Frigo, Mauro Zucchelli, Rachid Deriche, Samuel Deslauriers-Gauthier. TALON: Tractograms As Linear Operators in Neuroimaging. 2021, (swlh:1:dir:f25157ad1b13cb20ac3457d4f6756b49ac63d079;origin=https://inria.halpreprod.archives-ouvertes.fr/hal-02522751;visit=swlh:1:snp:8a2cb6ecd1478c63550e524a5723e06597259a07;anchor=swlh:1:rev:5c9642143d7f1c22e1bb641561e210eb52a94;path=). (hal-02522751)

EXPORTER

CodeMeta **BibTeX** **TEI** **DC** **DCterms**
EndNote

To consult the content on SWH

Citation and exports

5.3 The identifiers of the software deposit

5.3.1 Reference with the SWHID

(SoftWare Heritage Identifiers)

 archived `swh:1:dir:ec4ae097465d9ea51589537ea94b2ea50e8d134d`

- ★ Identification of the software source code artifact
- ★ A digital fingerprint specific source code content

Needed to :

- **Identify - reproduce**
- **Archive**

5.3.2 Cite with the HAL-ID

`hal-02309043, version 1`

- ★ Identification of the software record
- ★ Metadata of the deposit
- ★ Authors and contributors are verified in the moderation process

Needed to :

- **Give credit to the authors**
- **Index**

5.4 The citation and the BibTeX export

- The citation is accessible on the HAL record
- Export BibTeX using the format [BibLaTeX](#) @software or @softwareversion (if a version property was submitted)
- Export used in activity reports for scientific outputs at Inria since 2020.

Softwares

[38] [SW] M. Frigo, M. Zucchelli, R. Deriche and S. Deslauriers-Gauthier, *TALON: Tractograms As Linear Operators in Neuroimaging* version 0.3.0, 19th Jan. 2021. HAL: [hal-03116143](#), URL: <https://hal.archives-ouvertes.fr/hal-03116143>, VCS: <https://gitlab.inria.fr/cobcom/talon>, SWHID: [swh:1:dir:f25157ad1b13cb20ac3457d4f6756b49ac63d079;origin=https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:snp:465d89956196578717f4cb515e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70ccac;path=/](https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:snp:465d89956196578717f4cb515e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70ccac;path=/).

HAL's citation format

Matteo Frigo, Mauro Zucchelli, Rachid Deriche, Samuel Deslauriers-Gauthier. TALON: Tractograms As Linear Operators in Neuroimaging. 2021.

[swh:1:dir:f25157ad1b13cb20ac3457d4f6756b49ac63d079;origin=https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:snp:465d89956196578717f4cb515e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70ccac;path=/](https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1:snp:465d89956196578717f4cb515e456c279aa6a22;anchor=swh:1:rev:10247a14640a280b9140a27ce003d382d70ccac;path=/).
([hal-03116143](#))

```
@softwareversion{frigo:hal-03116143v1,
  TITLE = {{TALON: Tractograms As Linear Operators in
Neuroimaging}},
  AUTHOR = {Frigo, Matteo and Zucchelli, Mauro and
Deriche, Rachid and Deslauriers-Gauthier, Samuel},
  URL = {https://hal.archives-ouvertes.fr/hal-03116143},
  NOTE = {},
  YEAR = {2021},
  MONTH = Jan,
  SWHID =
{swh:1:dir:f25157ad1b13cb20ac3457d4f6756b49ac63d079;origin
=https://hal.archives-ouvertes.fr/hal-03116143;visit=swh:1
:snp:465d89956196578717f4cb515e456c279aa6a22;anchor=swh:1
:rev:10247a14640a280b9140a27ce003d382d70ccac;path=/},
  VERSION = {0.3.0},
  REPOSITORY = {https://gitlab.inria.fr/cobcom/talon},
  LICENSE = {MIT License},
  KEYWORDS = {diffusion MRI ; dMRI ; tractography ; python
; optimization},
  FILE =
{https://hal.archives-ouvertes.fr/hal-03116143/file/talon-
source.zip},
  HAL_ID = {hal-03116143},
  HAL_VERSION = {v1},
}
```

References

- ❖ Y. Barborini, R. Di Cosmo, Antoine R. Dumont, M. Gruenpeter, B. Marmol, A. Monteil, J. Sadowska.. La création du nouveau type de dépôt scientifique - Le logiciel. *JSO 2018 - 7es journées Science Ouverte Couperin : 100 % open access : initiatives pour une transition réussie*, Jan 2018, Paris, France. 2018. ([hal-01688726](#))
- ❖ R. Di Cosmo, M. Gruenpeter, B. Marmol, A. Monteil, L. Romary, J. Sadowska. *Curated Archiving of Research Software Artifacts: lessons learned from the French open archive*. IJDC. 2020 ([10.2218/ijdc.v15i1.698](#)). ([hal-02475835](#))
- ❖ R. Di Cosmo, M. Gruenpeter, S. Zacchiroli *Referencing Source Code Artifacts: a Separate Concern in Software Citation*, CiSE, IEEE, pp.1-9. 2020. ([10.1109/MCSE.2019.2963148](#)) ([hal-02446202](#))
- ❖ P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M.-S. Hacid, et al.. *Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria*. Computing in Science and Engineering, Institute of Electrical and Electronics Engineers, 2019, pp.1-14. ([10.1109/MCSE.2019.2949413](#)). ([hal-02135891](#))
- ❖ A. Monteil, M. Gruenpeter, J. Sadowska, E. Nivault. *Garantir la cohérence des données constitue le cœur de notre activité: entretien autour des enjeux descriptifs du code source*. *Bulletin des bibliothèques de France*, Ecole Nationale Supérieure des Sciences de l'Information et des Bibliothèques (ENSSIB), 2021, Dossier BBF 2021-1 • Code source : libérer le patrimoine !. ([hal-03239502](#))