



HAL
open science

Software Heritage: archive ouverte des codes et logiciels

Océane Valencia, Cécile Arènes

► To cite this version:

Océane Valencia, Cécile Arènes. Software Heritage: archive ouverte des codes et logiciels. Printemps de la Donnée 2024, INRAE; Université Haute-Alsace; Université de Strasbourg; INSA; PNDB; AgroParisTech; Université de Lille; Sorbonne Université; Data Terra, May 2024, Paris, France. hal-04684387

HAL Id: hal-04684387

<https://hal.inrae.fr/hal-04684387v1>

Submitted on 2 Sep 2024

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 4.0 International License



Le Printemps
de la Donnée

BIBLIOTHÈQUE DE SORBONNE UNIVERSITÉ

OCÉANE VALENCIA
CÉCILE ARÈNES

16 MAI 2024

SOFTWARE HERITAGE
ARCHIVE OUVERTE
DES CODES ET LOGICIELS



BIBLIOTHÈQUE
UNIVERSITAIRE



Qui sommes-nous ?

- **Cécile Arènes**, chargée de mission Données de la recherche et Humanités numériques, bibliothèque de Sorbonne Université
- **Océane Valencia**, responsable du Service des Archives et du Recueil des Actes, Sorbonne Université



1

LE MOUVEMENT DE LA SCIENCE OUVERTE

Le mouvement du logiciel libre

- **Copyleft, 1970**
- **Manifeste GNU / Richard Stallman, 1985**
- **Free Software Foundation, 1985**
- **Licence GNU GPL : 1989**
- **Linux, 1991**
- **Parution de La Cathédrale et le bazar / Eric Raymond, 1997**
- **Open Source Initiative, 1998**

4 libertés :

- **modification**
- **diffusion**
- **étude du code source**
- **utilisation**



open source
initiative

Le plan national pour la science ouverte

[Plan national pour la science ouverte 2](#), 2021-2024.

Axe 3 : ouvrir et promouvoir les codes sources produits par la recherche

- Valoriser et soutenir la diffusion sous licence libre des codes sources issus de recherches financées sur fonds publics
- Mettre en valeur la production des codes sources de l'enseignement supérieur, de la recherche et de l'innovation
- Définir et promouvoir une politique en matière de logiciels libres



La feuille de route du MESRI pour les données et codes

Politique des données, des algorithmes et des codes sources, 2021-2024.

- **Suivre l'ouverture des données et des codes de la recherche**
Baromètre de la science ouverte – Action 7
- **Collecter, préserver et partager les codes sources – Software Heritage**
– Action 11
- **Mettre en place des identifiants uniques et pérennes** – Actions 22 à 27
- **Accompagner l'adoption de licences libres** – Action 45



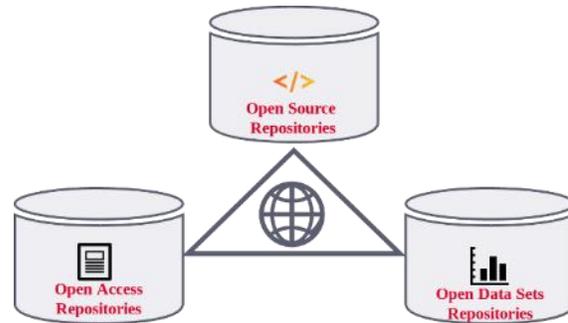
2

OUVRIR LES CODES ET LOGICIELS

Pourquoi ouvrir les codes sources ?

“Sometimes, if you don't have the software, you don't have the data.”

Christine L. Borgman, *Journées Nationales pour la Science Ouverte*, 2018



Trois piliers de la Science Ouverte, Software Heritage CC-BY 4.0 2019

Trois piliers pour la science ouverte :

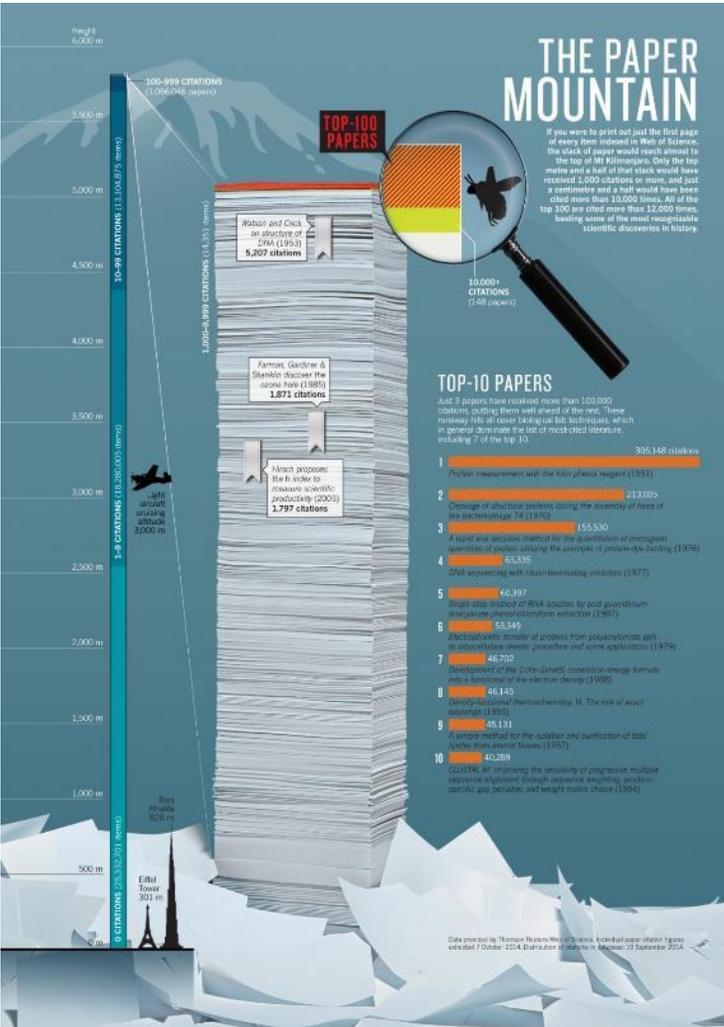
- open access
- open data
- open source repositories

Des données sans logiciel ne sont pas toujours reproductibles !

Voir : Stodden, Victoria, et Sheila Miguez. « Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research ». SSRN Scholarly Paper. Rochester, NY: Social Science Research Network, 6 septembre 2013.

<https://papers.ssrn.com/abstract=2322276> .

Pourquoi ouvrir les codes sources ?



« [...] software [...] essential in their fields »

Top 100 papers (Nature, 2014)

« Without the data and computer codes that underlie scientific discoveries, published findings are all but impossible to verify »

V. Stodden, S. Miguez, *Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2322276

➤ Sur 531 000 articles dans le domaine de la biosanté et montre que lorsqu'un article est associé à des données en annexe, il est cité 25% de fois en plus par rapport à un article sans données associées.

Plos One, 2020,

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230416>

=> avoir des données associées est un facteur de publicité et de crédibilité accrue des résultats

Plateformes au service des logiciels : panorama

Plateformes de développement collaboratif, “forges logicielles” :

- [GitHub](#), [GitLab.com](#), [BitBucket](#), instances de GitLab institutionnelles, etc.
 - Finalité : développer un logiciel de façon collaborative
 - Risques : aucune garantie de pérennité (nombreux exemples de fermetures brutales : Gitorious.org, Google Code, Bitbucket)



Plateformes de distribution:

- Debian, Ubuntu, CTAN, CRAN, CPAN, Maven, etc.
 - finalité: faciliter la diffusion de logiciels, souvent sous forme de paquetage

Plateformes d'archivage :

- [Software Heritage](#), [CINES](#), [BnF](#), Archives nationales, etc.
 - Finalité : garantir une conservation et un accès pérennes
 - Enjeux : garantir l'accès et la possibilité d'utiliser des codes et des logiciels de manière pérenne
 - Constituer un patrimoine informatique



Spécificités du logiciel

Le logiciel est à la fois un **outil**, un *résultat* et un *objet de recherche* :

- un *outil* : pour effectuer des traitements
 - R, utilisé largement pour des analyses statistiques,
 - Scilab, pour le traitement numérique, etc.
 - Jupyter, pour des notebooks exécutables
- un *résultat* : issu de travaux de recherche
 - SAGE, centre de nouveaux algorithmes en calcul formel
 - CGAL, bibliothèque d'algorithmes en géométrie computationnelle
- un *objet de recherche* : le code source des logiciels, avec son historique de développement, est notamment l'objet d'étude privilégié dans le Génie Logiciel (conférences MSR et ICSE)

Comme tout résultat de recherche, un logiciel doit pouvoir être accessible à la communauté de recherche à des fins de reproductibilité.



Software Heritage

**une archive pour les
codes et logiciels**



BIBLIOTHÈQUE
UNIVERSITAIRE

- **Une initiative d'INRIA**
- **Portée par Roberto Di Cosmo**

- **Trois principaux objectifs pour les codes sources :**
 - Collecter,
 - Préserver à long terme
 - Partager

Abramatic, J.-F., Di Cosmo, R., & Zacchioli, S. (2018). *Building the universal archive of source code*. <https://cacm.acm.org/magazines/2018/10/231366-building-the-universal-archive-of-source-code/fulltext>



Software Heritage

un peu d'histoire



BIBLIOTHÈQUE
UNIVERSITAIRE

- Juin 2018 :
Ouverture de l'archive Software Heritage
- Novembre 2018 :
Appel de Paris : le code source des logiciels comme patrimoine pour le développement durable
Mission et collaboration continue avec l'UNESCO.
- Juillet 2018 :
Software Heritage for Open Science
- Novembre 2019 :
ENEA, miroir italien de Software Heritage
- Octobre 2019
Curating and archiving landmark legacy source code

Principes FAIR pour les codes et logiciels

Lamprecht, Anna-Lena, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, et al. « **Towards FAIR Principles for Research Software** ». *Data Science* 3, n° 1 (1 janvier 2020): 37-59.

<https://doi.org/10.3233/DS-190026>.

Travail d'adaptation de chaque principe FAIR pour les codes et logiciels



Détail des principes FAIR :

<https://www.force11.org/group/fairgroup/fairprinciples>

Principes FAIR pour les codes et logiciels

F : Findable ou Facile à trouver

| | FAIR for data | FAIR for software | Operation |
|----|--|---|------------------------|
| F1 | (Meta)data are assigned a globally unique and persistent identifier. | Software and its associated metadata have a global, unique and persistent identifier for each released version. | Rephrased |
| F2 | Data are described with rich metadata. | Software is described with rich metadata. | Rephrased |
| F3 | Metadata clearly and explicitly include the identifier of the data it describes. | Metadata clearly and explicitly include identifiers for all the versions of the software it describes. | Rephrased and extended |
| F4 | (Meta)data are registered or indexed in a searchable resource. | Software and its associated metadata are included in a searchable software registry. | Rephrased |

Principes FAIR pour les codes et logiciels

A : Accessible

| | FAIR for data | FAIR for software | Operation |
|------|--|---|-----------------|
| A1 | (Meta)data are retrievable by their identifier using a standardized communications protocol. | Software and its associated metadata are accessible by their identifier using a standardized communications protocol. | Rephrased |
| A1.1 | The protocol is open, free, and universally implementable. | The protocol is open, free, and universally implementable. | Remain the same |
| A1.2 | The protocol allows for an authentication and authorization procedure, where necessary. | The protocol allows for an authentication and authorization procedure, where necessary. | Remain the same |
| A2 | Metadata are accessible, even when the data are no longer available. | Software metadata are accessible, even when the software is no longer available. | Rephrased |

Principes FAIR pour les codes et logiciels

I : Interopérable

| | FAIR for data | FAIR for software | Operation |
|-------|--|--|-----------------------------------|
| I1 | (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation. | Software and its associated metadata use a formal, accessible, shared and broadly applicable language to facilitate machine readability and data exchange. | Rephrased and extended |
| I2 | (Meta)data use vocabularies that follow FAIR principles. | – | Reinterpreted, extended and split |
| I2S.1 | – | Software and its associated metadata are formally described using controlled vocabularies that follow the FAIR principles. | Reinterpreted, extended and split |
| I2S.2 | – | Software use and produce data in types and formats that are formally described using controlled vocabularies that follow the FAIR principles. | Reinterpreted, extended and split |
| I3 | (Meta)data include qualified references to other (meta)data. | – | Discarded |
| I4S | – | Software dependencies are documented and mechanisms to access them exist. | Newly proposed |

Principes FAIR pour les codes et logiciels

R : Réutilisable

| | FAIR for data | FAIR for software | Operation |
|-------------|---|---|------------------------|
| R1 | (Meta)data are richly described with a plurality of accurate and relevant attributes. | Software and its associated metadata are richly described with a plurality of accurate and relevant attributes. | Rephrased |
| R1.1 | (Meta)data are released with a clear and accessible data usage license. | Software and its associated metadata have independent, clear and accessible usage licenses compatible with the software dependencies. | Rephrased and extended |
| R1.2 | (Meta)data are associated with detailed provenance. | Software metadata include detailed provenance, detail level should be community agreed. | Rephrased |
| R1.3 | (Meta)data meet domain-relevant community standards, where necessary. | Software metadata and documentation meet domain-relevant community standards. | Rephrased |

Documenter les projets et assurer leur reproductibilité

Documenter les projets de recherche :

- Jupyter notebooks
- RStudio
- OrgMode
- MkDocs

Assurer la reproductibilité de la recherche :

- Voir le **MOOC d'INRIA** : Recherche reproductible : principes méthodologiques pour une science transparente

<https://www.fun-mooc.fr/fr/cours/recherche-reproductible-principes-methodologiques-pour-une-science-transparente/>

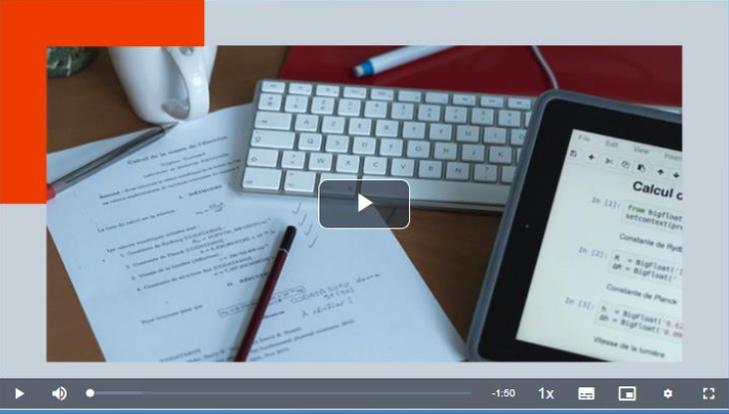
Outils pour la recherche Numérique et technologie

Recherche reproductible : principes méthodologiques pour une science transparente

Réf. 41016

Ce Mooc propose des principes méthodologiques pour une science ouverte et transparente. Il aborde de manière pratique la prise de notes, le document computationnel, la répliquabilité des analyses.

🕒 Effort : 24 heures 🔄 Rythme: Auto-rythmé
🌐 Langues: Français



Inscription
Du 10 janv. 2020 au 11 déc. 2024

Cours
Du 20 mars 2020 au 31 déc. 2024

Langues
français

Connectez-vous pour vous inscrire





Pour le déposant

Améliorer la visibilité de son projet

- Facile à trouver dans les moteurs de recherche
- Accessible à tous
- Réutilisable
- Reproductible
- Identifié

Garantie contre le plagiat :

- identifiant pérenne du dépôt
- horodatage du dépôt

3

SÉCURISER LES CONDITIONS JURIDIQUES DE LA
RÉUTILISATION



Ouvrir le code

Propriété intellectuelle

- Le code est protégé par le droit de la propriété intellectuelle (loi 85-660 du 3 juillet 1985)
 - Définition : « Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données »
 - Titularité des droits : « Les droits patrimoniaux sur les logiciels et leur documentation créés par un ou plusieurs employés dans l'exercice de leurs fonctions ou d'après les instructions de leur employeur sont dévolus à l'employeur qui est seul habilité à les exercer. » (article L113-9 du CPI)

Ouvrir le codes et les logiciels

Les licences



L'Open Source Initiative maintient la liste des licences Open Source, disponible ici:

<https://opensource.org/licenses>



La Free Software Foundation maintient les licences GPL et LGPL, ainsi qu'une liste de licences de logiciels libres, voir: <https://www.fsf.org/licensing/>



SPDX (Linux foundation), maintient une liste de noms de licences canoniques, voir <https://spdx.org/licenses/>



[Licences GNU](#)

Un outil pour trouver une licence publique :
<https://ufal.github.io/public-license-selector/>

Licences ouvertes



BIBLIOTHÈQUE
UNIVERSITAIRE

<https://spdx.org/licenses/>



SPDX License List

The SPDX License List is an integral part of the SPDX Specification. The SPDX License List itself is a list of commonly found licenses and exceptions used in free and open or collaborative software, data, hardware, or documentation. The SPDX License List includes a standardized short identifier, the full name, the license text, and a canonical permanent URL for each license and exception.

The purpose of the SPDX License List is to enable efficient and reliable identification of such licenses and exceptions in an SPDX document, in source files or elsewhere.

- [License Exceptions](#) are commonly found exceptions to free and open source licenses, used with the [License Expression](#) operator, "WITH" to create a license with an exception.
- The [matching guidelines](#) define what constitutes a license or exception match. The license text on the HTML pages here will display omissible text in blue and replaceable text in red (see Guideline #2 for more information).
- [Explanation of fields](#) used on the SPDX License List
- [License inclusion principles](#) for adding new licenses or exceptions to the SPDX License List
- [Contribute](#) to the project or request a new license
- Use [short identifiers in your source code](#)
- [Github repo](#)
- Machine readable [data files](#) for the SPDX License List

Version: 3.12 2021-03-07

Note: You can sort by each column by clicking on the column header. By default, the table sorts by the Identifier column.

| Full name | Identifier | FSF Free/Libre? | OSI Approved? |
|--|-------------|-----------------|---------------|
| BSD Zero Clause License | 0BSD | | Y |
| Attribution Assurance License | AAL | | Y |
| Abstyles License | Abstyles | | |
| Adobe Systems Incorporated Source Code License Agreement | Adobe-2006 | | |
| Adobe Glyph List License | Adobe-Glyph | | |
| Amazon Digital Services License | ADSL | | |
| Academic Free License v1.1 | AFL-1.1 | Y | Y |
| Academic Free License v1.2 | AFL-1.2 | Y | Y |
| Academic Free License v2.0 | AFL-2.0 | Y | Y |
| Academic Free License v2.1 | AFL-2.1 | Y | Y |
| Academic Free License v2.2 | AFL-2.2 | Y | Y |

4

DÉPOSER SUR SOFTWARE HERITAGE

Exemples de projets de recherche avec dépôt de code



BIBLIOTHÈQUE
UNIVERSITAIRE

Un exemple d'article citant des logiciels déposés :

- Roberto Di Cosmo, Marco Danelutto. [Rp] Reproducing and replicating the OCamlP3l experiment. *The ReScience journal*, GitHub, 2020, [⟨10.5281/zenodo.3763416⟩](https://doi.org/10.5281/zenodo.3763416). [⟨hal-02885664⟩](https://hal.archives-ouvertes.fr/hal-02885664)
- Les citations 8 à 10 font référence à des éléments précis de codes ou logiciels déposés.

References

1. M. Danelutto, R. Di Cosmo, X. Leroy, and S. Pelagatti. "Parallel Functional Programming with Skeletons: the OCamlP3L experiment." In: **ACM Workshop on ML and its applications**. ACM, Baltimore, United States, Sept. 1998.
2. [SW] R. Di Cosmo, M. Danelutto, X. Leroy, and S. Pelagatti, **The OCamlP3l library**, 1998. LIC: GPL. URL: https://archive.softwareheritage.org/gitorious.org/ocamlp3l/ocamlp3l_cvs.git.
3. X. Leroy, D. Doligez, A. Frisch, J. Garrigue, D. Rémy, and J. Vouillon. **The OCaml system release 4.09: Documentation and user's manual**. Intern report. Inria, Sept. 2019, pp. 1–789.
4. M. Cole. **Algorithmic skeletons: structured management of parallel computation**. MIT Press, 1989.
5. J.-F. Abramatic, R. Di Cosmo, and S. Zacchiroli. "Building the Universal Archive of Source Code." In: **Commun. ACM** 61.10 (Sept. 2018), pp. 29–31.
6. [SW] R. Di Cosmo, P. Weis, F. Clement, and Z. Li, **OcamlP3l release 2.03**, 2007. LIC: LGPL-2. HAL: [⟨hal-02487579⟩](https://hal.archives-ouvertes.fr/hal-02487579), SWHID: [⟨swh:1:dir:85642a2e0333bbd6340c0a84ae6bad48cba11940;origin=https://hal.archives-ouvertes.fr/hal-02487579⟩](https://sw.hal.archives-ouvertes.fr/sw/1:dir:85642a2e0333bbd6340c0a84ae6bad48cba11940;origin=https://hal.archives-ouvertes.fr/hal-02487579).
7. [SW REL.] R. Di Cosmo, M. Danelutto, X. Leroy, and S. Pelagatti, **The OCamlP3l library** version 1.0, 1998. LIC: GPL. SWHID: [⟨swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git/⟩](https://sw.hal.archives-ouvertes.fr/sw/1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git/).
8. [SW EXC.] M. Danelutto, "Simple Farm test program," from **The OCamlP3l library** 1998. LIC: GPL. SWHID: [⟨swh:1:cnt:4d99d2d18326621ccdd70f5ea66c2e2ac236ad8b;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Examples/SimpleFarm/simplefarm.ml⟩](https://sw.hal.archives-ouvertes.fr/sw/1:cnt:4d99d2d18326621ccdd70f5ea66c2e2ac236ad8b;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Examples/SimpleFarm/simplefarm.ml).
9. [SW EXC.] M. Danelutto, "Pipeline test program," from **The OCamlP3l library** 1998. LIC: GPL. SWHID: [⟨swh:1:cnt:8415f9451cf1ecaef70daab45c0ea2e5200f7d38;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Examples/PerfTuning/pipeline.ml⟩](https://sw.hal.archives-ouvertes.fr/sw/1:cnt:8415f9451cf1ecaef70daab45c0ea2e5200f7d38;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Examples/PerfTuning/pipeline.ml).
10. [SW EXC.] R. Di Cosmo, "Driver command," from **The OCamlP3l library** 1998. LIC: GPL. SWHID: [⟨swh:1:cnt:c428f4deb1cdf8500fff5c449b99454a816c163;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Tools/ocamlp3lrun⟩](https://sw.hal.archives-ouvertes.fr/sw/1:cnt:c428f4deb1cdf8500fff5c449b99454a816c163;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;/anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Tools/ocamlp3lrun).
11. R. Di Cosmo. "How to use Software Heritage for archiving and referencing your source code: guidelines and walkthrough." HAL preprint [hal-02263344](https://hal.archives-ouvertes.fr/hal-02263344). Apr. 2019.

5

SAUVEGARDER SON CODE SUR SOFTWARE
HERITAGE

Partager logiciels et codes source



Software Heritage

Sauvegarder son code:

1. Préparer
2. Déposer

Deux possibilités :

- Dépôt direct dans Software Heritage
- Dépôt via HAL

Préparation du logiciel

Avant le dépôt sur HAL préparer le code source du logiciel sur la plateforme de développement (GitHub, Gitlab, etc.).

Ces éléments sont vérifiés après par les modérateurs.

2.1.1 ajouter les fichiers suivants sur le `code repository` :

- ❑ **README**
- ❑ **AUTHORS**
- ❑ **LICENSE (à choisir avec les titulaires des droits patrimoniaux du dit logiciel - liste de référence)**

- ❑ **codemeta.json** - facultatif mais pratique car les métadonnées sont ajoutées dans le formulaire HAL automatiquement en utilisant le SWHID approprié

The screenshot shows a GitHub repository page for 'moranegg / AffectationRO'. The repository is public and has 1 pull request, 0 stars, and 0 forks. The file list shows several files, with 'AUTHORS', 'LICENSE', 'README.md', and 'codemeta.json' highlighted in a yellow box. The 'README.md' file is open, showing the title 'projet en Recherche Opérationnelle', a DOI '10.5281/zenodo.438684', and the subtitle 'Problème d'affectation généralisée'. The README content describes a distributed system with processors and tasks. The right sidebar shows project information, including 'Projet en recherche opérationnelle', 'operational-research' tags, 'GPL-3.0 License', and a release 'Project in OR: The assi...' from Mar 28, 2017. The 'Languages' section shows a bar chart with Java at 83.7%, AMPL at 16.0%, and Batchfile at 0.3%.

Le fichier Readme

De nombreux templates existent.

- Un exemple, [Make a Readme](#)

Contenu du fichier décrivant le logiciel :

- Nom,
- Objectif,
- Liens vers un site,
- Documentation
- Plateforme de développement
- Contacts et informations sur le support



Make a README

Because no one can read your mind (yet)



Need an interactive developer hub, complete with documentation, reference guides, tutorials, realtime API logs and community? [Try ReadMe.com!](#)

Make a README is proudly sponsored by ReadMe.com

README 101

What is it?

A **README** is a text file that introduces and explains a project. It contains information that is commonly required to understand what the project is about.

Why should I make it?

It's an easy way to answer questions that your audience will likely have regarding how to install and use your project and also how to collaborate with you.

Who should make it?

Anyone who is working on a programming project, especially if you want others to use it or contribute.

Pourquoi CodeMeta ?

- Un vocabulaire qui étend le vocabulaire schema.org
 - <https://codemeta.github.io/terms/>
- Une **communauté académique**
- Une **table** de correspondances qui permet de traduire plusieurs ontologies/vocabulaires vers CodeMeta

Pour créer facilement un fichier codemeta.json utiliser l'[outil en ligne](#)

➤ [Pour contribuer](#)

Outil pour créer un codemeta.json

CodeMeta generator

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

The software itself

Name

My Software

the software title

Description

My Software computes ephemerides and orbit propagation. It has been developed from early '80.

Creation date

YYYY-MM-DD

First release date

YYYY-MM-DD



Auteurs : bonnes pratiques

Voir :

Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, et al..
Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria.
2019. [hal-02135891v1](https://hal.archives-ouvertes.fr/hal-02135891v1)

Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria

Pierre Alliez¹, Roberto Di Cosmo^{2,3,4}, Benjamin Guedj^{5,6,7,8}, Alain Girault⁹, Mohand-Said Hacid¹⁰, Arnaud Legrand^{11,12}, Nicolas P. Rougier¹³ [Détails](#)

- 1 TITANE - Geometric Modeling of 3D Environments
CRISAM - Inria Sophia Antipolis - Méditerranée
- 2 Software Heritage
- 3 UPD7 - Université Paris Diderot - Paris 7
- 4 Inria Siège
- 5 UCL-CS - Computer science department [University College London]
- 6 MODAL - MOdel for Data Analysis and Learning
LPP - Laboratoire Paul Painlevé - UMR 8524, Université de Lille, Sciences et Technologies, Inria Lille - Nord Europe, METRICS - Evaluation des technologies de santé et des pratiques médicales - ULR 2694, Polytech Lille - École polytechnique universitaire de Lille
- 7 UCL - University College of London [London]
- 8 Inria-CWI - Inria-CWI
- 9 SPADES - Sound Programming of Adaptive Dependable Embedded Systems
Inria Grenoble - Rhône-Alpes, LIG - Laboratoire d'Informatique de Grenoble
- 10 BD - Base de Données
LIRIS - Laboratoire d'Informatique en Image et Systèmes d'information
- 11 LIG - Laboratoire d'Informatique de Grenoble
- 12 POLARIS - Performance analysis and optimization of LARge Infrastructures and Systems
Inria Grenoble - Rhône-Alpes, LIG - Laboratoire d'Informatique de Grenoble
- 13 Mnemosyne - Mnemonic Synergy
LaBRI - Laboratoire Bordelais de Recherche en Informatique, Inria Bordeaux - Sud-Ouest, IMN - Institut des Maladies Neurodégénératives [Bordeaux]

Abstract : Software is a fundamental pillar of modern scientific research, not only in computer science, but actually across all fields and disciplines. However, there is a lack of adequate means to cite and reference software, for many reasons. An obvious first reason is software authorship, which can range from a single developer to a whole team, and can even vary in time. The panorama is even more complex than that, because many roles can be involved in software development: software architect, coder, debugger, tester, team manager, and so on. Arguably, the researchers who have invented the key algorithms underlying the software can also claim a part of the authorship. And there are many other reasons that make this issue complex. We provide in this paper a contribution to the ongoing efforts to develop proper guidelines and recommendations for software citation, building upon the internal experience of Inria, the French research institute for digital sciences. As a central contribution, we make three key recommendations. (1) We propose a richer taxonomy for software contributions with a qualitative scale. (2) We claim that it is essential to put the human at the heart of the evaluation. And (3) we propose to distinguish citation from reference.

FICHIERS

main.pdf
Fichiers produits par l'(les) auteur(s)

IDENTIFIANTS

- HAL Id : hal-02135891, version 1
- ARXIV : 1905.11123

COLLECTIONS

UNIV-PARIS7 | USPC

CITATION

Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, et al.. Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria. 2019. (hal-02135891v1)

EXPORTER

BibTeX TEI DC DCTerms EndNote

PARTAGER



MÉTRIQUES

Consultations de la notice Téléchargements de fichiers

Partager logiciels et codes source



Sauvegarder son code

- Dépôt direct dans Software Heritage

Guide pas à pas : Dépôt dans Software Heritage

Possibilité d'import direct

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 an origin save request. To do so, fill the required info in the form below:

| | | |
|-------------|---|--------|
| Origin type | Origin url | |
| git | https://github.com/carenes/notes-brutes | Submit |

The "save code now" request has been accepted and will be processed as soon as possible.

Help Browse save requests

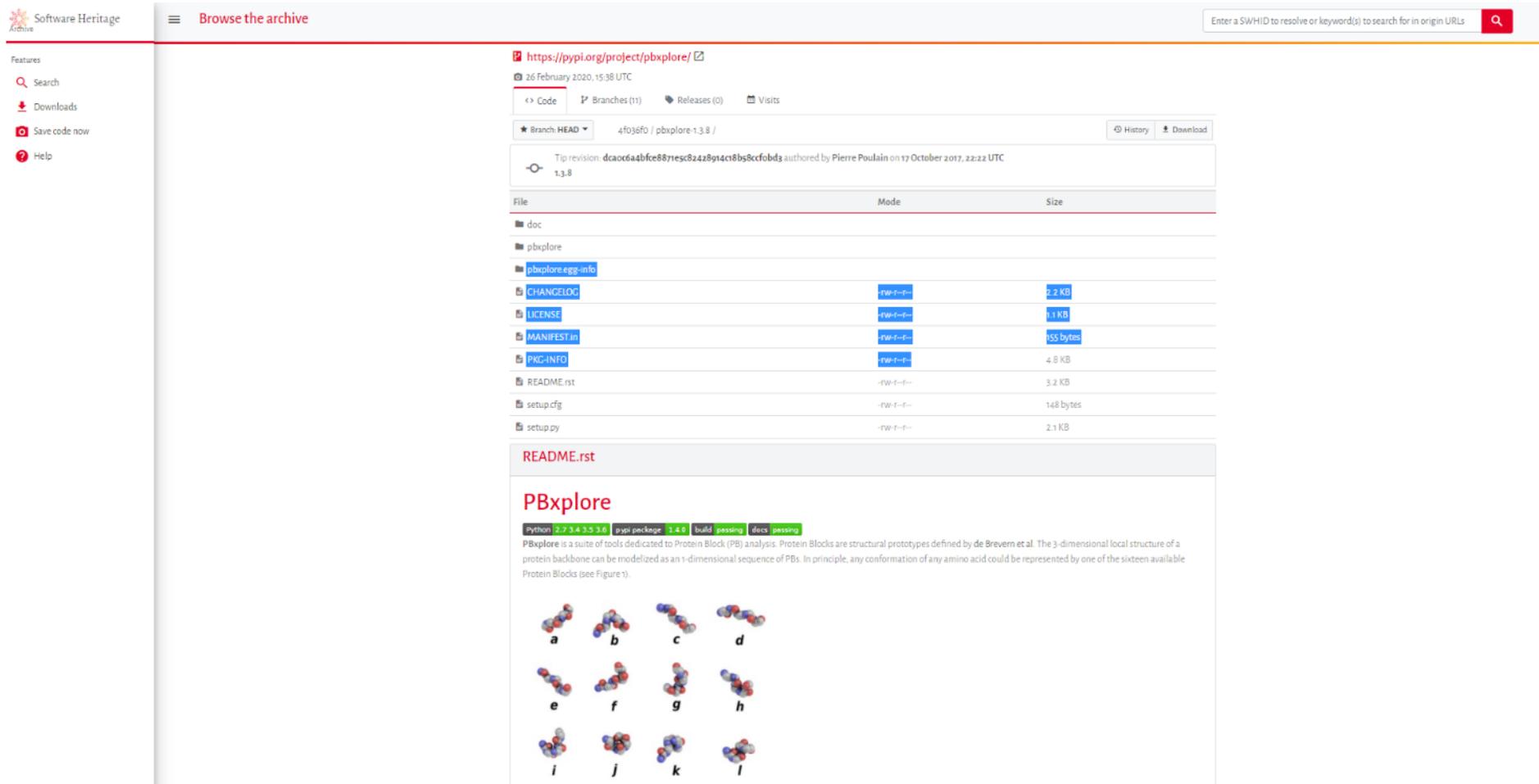
Show 10 entries

Search:

| Date | Type | Url | Request | Status | Info |
|-----------------------|------|---|----------|-------------------|------------|
| 08/07/2020 à 18:13:13 | git | https://github.com/carenes/notes-brutes | accepted | not yet scheduled | Save again |

Guide pas à pas : Dépôt dans Software Heritage

Exemple : https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://pypi.org/project/pbxplore/&path=pbxplore-1.3.8



The screenshot shows the Software Heritage archive page for the `pbxplore` project. The page is titled "Browse the archive" and displays the following information:

- Project URL: <https://pypi.org/project/pbxplore/>
- Timestamp: 26 February 2020, 15:38 UTC
- Navigation: Code, Branches (11), Releases (0), Visits
- Current Branch: HEAD (4f036f0 / pbxplore-1.3.8)
- Tip revision: `dca0c6a4bfc887f5c82428914c18b58ccf0bd5` authored by Pierre Poulain on 17 October 2017, 22:22 UTC
- Version: 1.3.8
- File listing table:

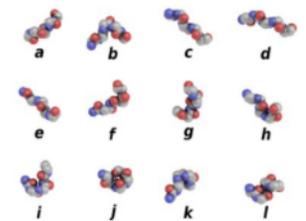
| File | Mode | Size |
|-------------------|------------|-----------|
| doc | | |
| pbxplore | | |
| pbxplore.egg-info | | |
| CHANGELOG | -rw-r--r-- | 2.2 KB |
| LICENSE | -rw-r--r-- | 1.1 KB |
| MANIFEST.in | -rw-r--r-- | 155 bytes |
| PKG-INFO | -rw-r--r-- | 4.8 KB |
| README.rst | -rw-r--r-- | 3.2 KB |
| setup.cfg | -rw-r--r-- | 148 bytes |
| setup.py | -rw-r--r-- | 2.1 KB |

The `README.rst` file content is visible below the table:

PBxplore

Python 2.7 3.4 3.5 3.6 | pypi package 1.4.0 | build | passing | docs | passing

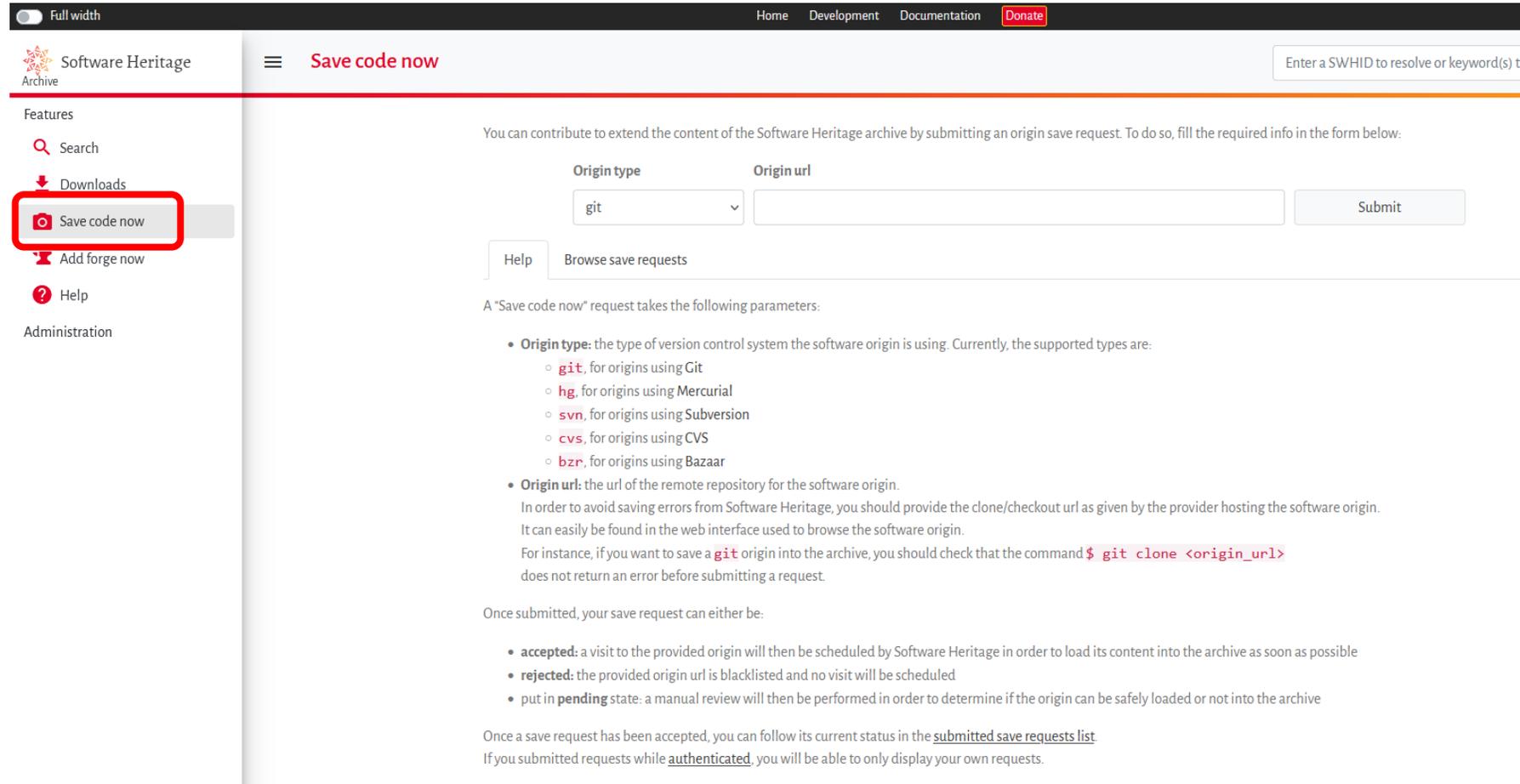
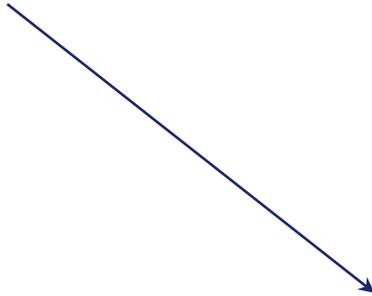
PBxplore is a suite of tools dedicated to Protein Block (PB) analysis. Protein Blocks are structural prototypes defined by de Brevern et al. The 3-dimensional local structure of a protein backbone can be modeled as an n-dimensional sequence of PBs. In principle, any conformation of any amino acid could be represented by one of the sixteen available Protein Blocks (see Figure 1).



Sauvegarder code sur Software Heritage

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

Vérifier si l'archive est sur SWH



Full width Home Development Documentation **Donate**

Software Heritage Archive

Save code now

Enter a SWHID to resolve or keyword(s) to search

Features

- Search
- Downloads
- Save code now**
- Add forge now
- Help

Administration

You can contribute to extend the content of the Software Heritage archive by submitting an origin save request. To do so, fill the required info in the form below:

Origin type: git

Origin url:

Submit

Help Browse save requests

A "Save code now" request takes the following parameters:

- Origin type:** the type of version control system the software origin is using. Currently, the supported types are:
 - git, for origins using Git
 - hg, for origins using Mercurial
 - svn, for origins using Subversion
 - cvs, for origins using CVS
 - bzr, for origins using Bazaar
- Origin url:** the url of the remote repository for the software origin.
In order to avoid saving errors from Software Heritage, you should provide the clone/checkout url as given by the provider hosting the software origin. It can easily be found in the web interface used to browse the software origin.
For instance, if you want to save a git origin into the archive, you should check that the command `$ git clone <origin_url>` does not return an error before submitting a request.

Once submitted, your save request can either be:

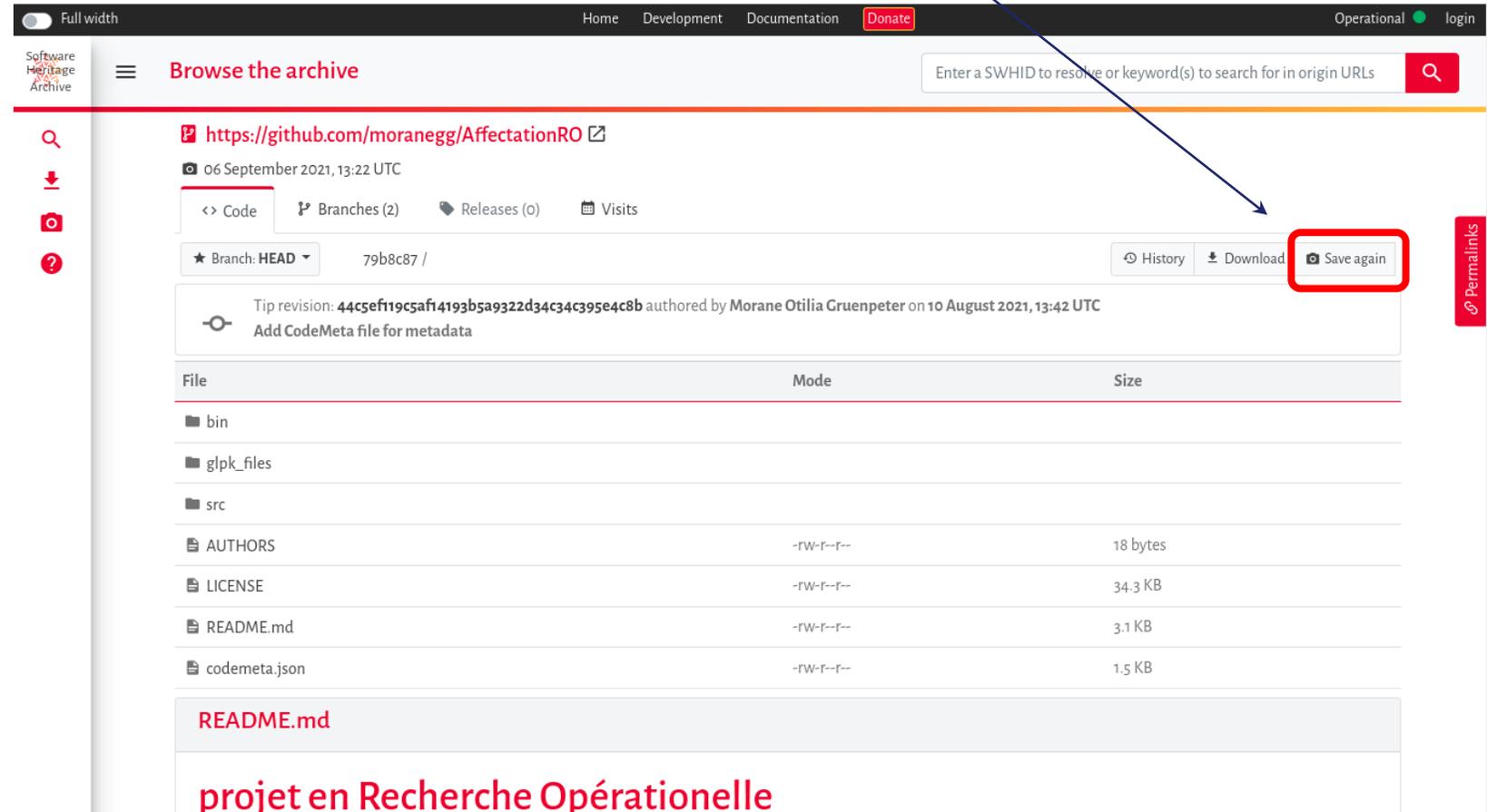
- accepted:** a visit to the provided origin will then be scheduled by Software Heritage in order to load its content into the archive as soon as possible
- rejected:** the provided origin url is blacklisted and no visit will be scheduled
- put in **pending** state: a manual review will then be performed in order to determine if the origin can be safely loaded or not into the archive

Once a save request has been accepted, you can follow its current status in the [submitted save requests list](#).
If you submitted requests while [authenticated](#), you will be able to only display your own requests.

Sauvegarder code sur Software Heritage

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

Si l'archive est sur SWH -> la
mettre à jour



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: 44c5ef119c5af14193b5a9322d34c34c395e4c8b authored by Morane Otilia 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

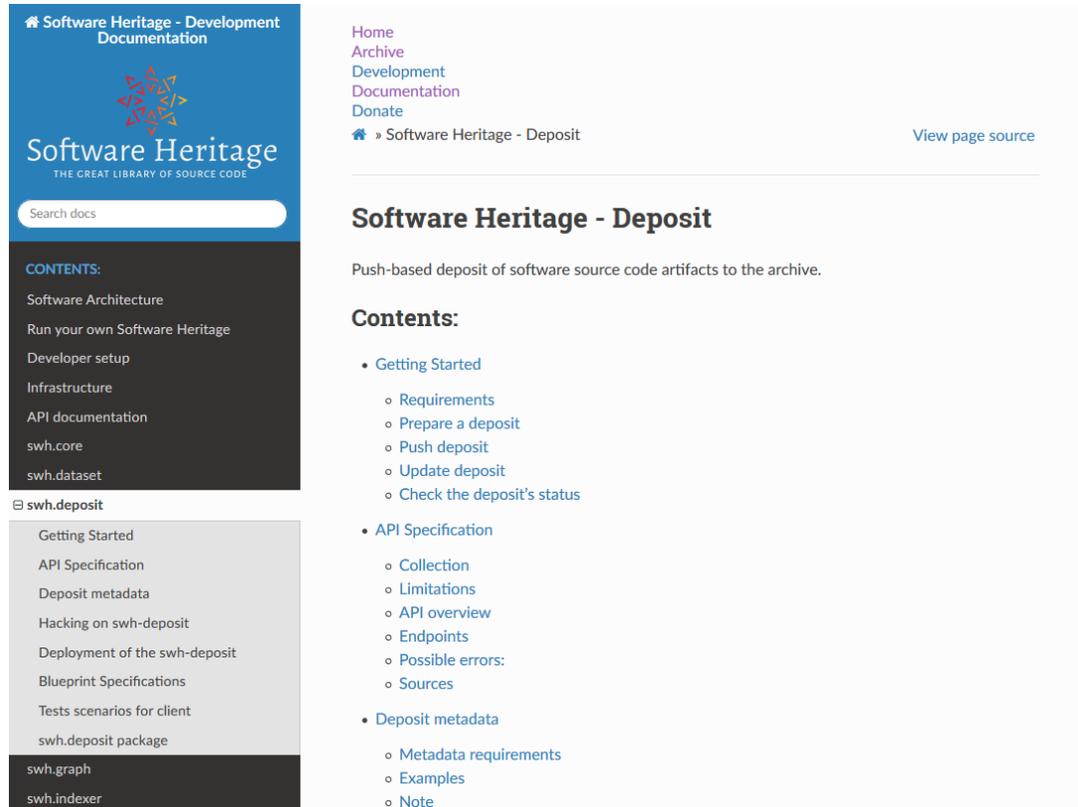
projet en Recherche Opérationnelle

Permalinks

Guide pas à pas : Dépôt dans Software Heritage

Documentation

<https://docs.softwareheritage.org/devel/swh-deposit/index.html>



The screenshot shows the documentation page for Software Heritage - Deposit. The page has a blue header with the Software Heritage logo and the tagline 'THE GREAT LIBRARY OF SOURCE CODE'. Below the header is a search bar and a navigation menu with links for Home, Archive, Development, Documentation, and Donate. The main content area is titled 'Software Heritage - Deposit' and includes a brief description: 'Push-based deposit of software source code artifacts to the archive.' Below this is a 'Contents' section with a list of links: Getting Started (Requirements, Prepare a deposit, Push deposit, Update deposit, Check the deposit's status), API Specification (Collection, Limitations, API overview, Endpoints, Possible errors, Sources), and Deposit metadata (Metadata requirements, Examples, Note). A sidebar on the left lists various documentation topics, with 'swh.deposit' selected and expanded to show its sub-topics.

Software Heritage - Development Documentation

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Search docs

CONTENTS:

- Software Architecture
- Run your own Software Heritage
- Developer setup
- Infrastructure
- API documentation
- swh.core
- swh.dataset

swh.deposit

- Getting Started
 - API Specification
 - Deposit metadata
 - Hacking on swh-deposit
 - Deployment of the swh-deposit
 - Blueprint Specifications
 - Tests scenarios for client
 - swh.deposit package
- swh.graph
- swh.indexer

Home
Archive
Development
Documentation
Donate

» Software Heritage - Deposit [View page source](#)

Software Heritage - Deposit

Push-based deposit of software source code artifacts to the archive.

Contents:

- Getting Started
 - Requirements
 - Prepare a deposit
 - Push deposit
 - Update deposit
 - Check the deposit's status
- API Specification
 - Collection
 - Limitations
 - API overview
 - Endpoints
 - Possible errors:
 - Sources
- Deposit metadata
 - Metadata requirements
 - Examples
 - Note

Partager logiciels et codes source



Software Heritage

Sauvegarder son code

- Dépôt via HAL

HAL et Software Heritage

Une collaboration étroite entre Hal-Inria (l'archive ouverte d'Inria), le CCSD et Software Heritage a abouti à la mise en place d'un processus permettant de **déposer des codes sources de logiciels via HAL, et leur archivage pérenne dans Software Heritage.**

Objectifs :

- dépôt dans HAL pour la **citabilité**,
- **archivage** par Software Heritage.

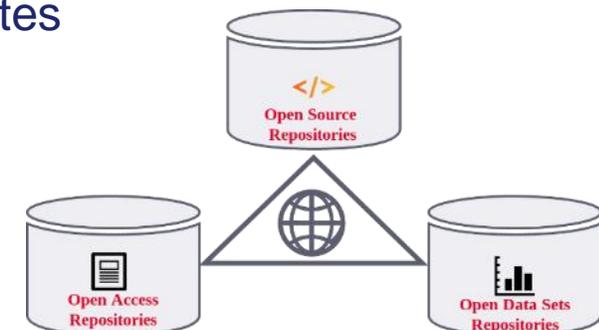
Conditions pour être transféré à Software Heritage :

- **licence libre**
- **pas d'embargo**

Voir : <https://doc.archives-ouvertes.fr/deposer/deposer-le-code-source/>

Pourquoi déposer sur HAL ?

- **Archiver** le logiciel sur HAL et sur SWH
 - Grande **visibilité** aux logiciels dans une démarche de science ouverte
 - **Archivage pérenne**, en transférant votre code vers Software Heritage, l'archive universelle du code source
- **Identifier**
 - les objets avec un **SWHID** (SoftWare Heritage Identifier)
 - la notice et la citation avec un **HAL-ID**
- **Décrire** le logiciel avec des métadonnées qui sont modérées
 - Modération des **métadonnées** par l'équipe des documentalistes
- **Citer** le dépôt avec une citation complète
 - Différents **formats** d'export pour faciliter la citation



Quel objet logiciel déposer sur HAL?

- ★ Le code source d'un logiciel (ne pas déposer les exécutables)
- ★ Le logiciel qui est développé dans un milieu académique
- ★ Seuls les créateurs/auteurs du logiciel ou leurs représentants peuvent déposer un logiciel sur HAL

Cas d'utilisations

Je **développe** mes logiciels **localement** et je les partage sur mon **site web** personnel ou institutionnel

Dépôt source: déposer .zip / .tar.gz

Je **développe** mes logiciels sur une **plateforme collaborative** (type GitHub, GitLab, Bitbucket...)

Dépôt SWHID: déposer métadonnées et SWHID

Checklist par cas d'utilisation

Dépôt source : déposer .zip /.tar.gz

1. **Préparer le logiciel (localement)**
 - Fichiers AUTHORS, LICENSE & README
 - Compresser les documents en .zip / .tar.gz
2. **Déposer l'archive compressée**
3. **Compléter les métadonnées**
 - Choisir le type de dépôt
 - Ajouter des métadonnées générales
 - Ajouter des métadonnées spécifiques aux logiciels
 - Renseigner les auteurs
 - Valider le dépôt

Dépôt SWHID : déposer métadonnées et SWHID

1. **Préparer le logiciel (sur un code repository)**
 - Fichier AUTHORS, LICENSE & README
 - Fichier codemeta.json
 - Save code now sur Software Heritage
 - Choisir un SWHID
2. **Déposer le lien (SWHID) vers le contenu**
3. **Compléter les métadonnées**
 - Ajouter le domaine
 - Contrôler les entrées codemeta
 - Compléter des métadonnées
 - Renseigner les auteurs
 - Valider le dépôt

Soumettre le code repository sur SWH avec le Save Code Now pour un dépôt SWHID :

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

Dépôt SWHID : déposer métadonnées et SWHID

The screenshot shows the 'Save code now' page on the Software Heritage website. The page has a header with the Software Heritage logo and a search bar. A left sidebar contains navigation links: Search, Downloads, Save code now (highlighted), and Help. The main content area is titled 'Save code now' and contains a form for submitting code. The form has two main fields: 'Origin type' and 'Origin url'. The 'Origin type' field is a dropdown menu with 'git' selected. 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. A purple circle with the number '1' points to the 'Origin type' dropdown menu. 2. A purple circle with the number '2' points to the 'Origin url' text input box. 3. A purple circle with the number '3' points to the 'Submit' button. Three grey callout boxes with blue borders contain the following text: '1. Choisir le type de système de contrôle de version', '2. Ajouter l'url', and '3. Soumettre'.

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. To do so, fill the required info in the form below:

Origin type: git (1) | Origin url: (2) | Submit (3)

1. Choisir le type de système de contrôle de version

2. Ajouter l'url

3. Soumettre

Choisir un SWHID sur l'archive Software Heritage

The screenshot shows the Software Heritage interface for a repository. The main content area displays a directory listing with a 'directory' button highlighted in red. Below the listing, there is a section for 'Add contextual information' with a checked checkbox, and a 'Copy identifier' button. The SWHID is displayed as `swh:1:dir:79b8c8755dbed34f01a6a7184ffc196f3c58cb5d`. The repository name is 'AffectationRO' by 'moranegg' on GitHub.

1. Cliquer sur `Permalinks`

2. Choisir le type d'objet - `directory`

3. Ajouter le contexte du SWHID

4. Copier l'identifiant sur le formulaire HAL

projet en Recherche Opérationnelle

Déposer le lien vers le contenu

→ **Mettre le SWHID sur le formulaire HAL**

◆ préférer un SWHID avec contexte (pour garder le lien vers l'historique de développement)

◆ Si un codemeta.json existe dans le code source - la plateforme HAL pourra récupérer les métadonnées automatiquement

→ Vérifier et compléter les métadonnées

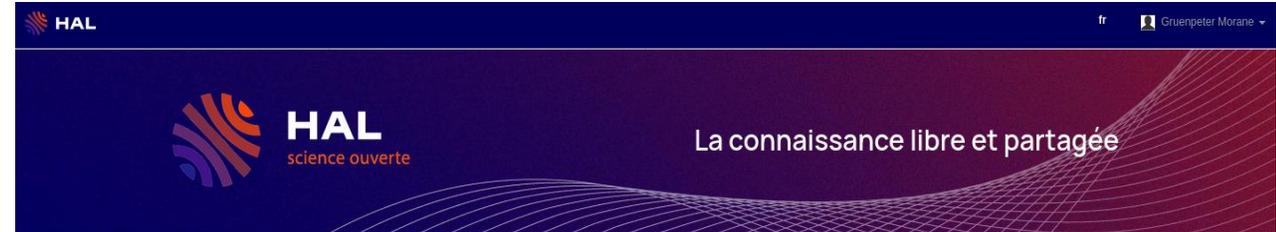
◆ vérifier les métadonnées insérées

◆ compléter les métadonnées manquantes

◆ choisir le domaine

◆ vérifier les auteurs et affiliations

→ Valider le dépôt



Déposer le(s) fichier(s)

Afficher la vue détaillée

Vous pouvez déposer un fichier et/ou compléter votre dépôt à partir d'un identifiant externe.

Glissez-déposez ou cliquez pour choisir vos fichiers
Taille maximale du fichier : 200M

Chargez les métadonnées à partir d'un identifiant

Les informations associées à cet identifiant permettront de remplir automatiquement votre dépôt. Par exemple, le DOI est un code qui apparaît généralement sur la page de votre article sur le site de l'éditeur.

SWHID

Récupérer les métadonnées

Si vous souhaitez ajouter un embargo, récupérer des fichiers de votre espace FTP ou toute autre action avancée, veuillez afficher la vue détaillée.

SWHID

Dépôt des fichiers sources



BIBLIOTHÈQUE
UNIVERSITAIRE

Dépôt source: déposer .zip /.tar.gz

Préparer le logiciel

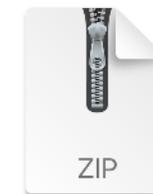
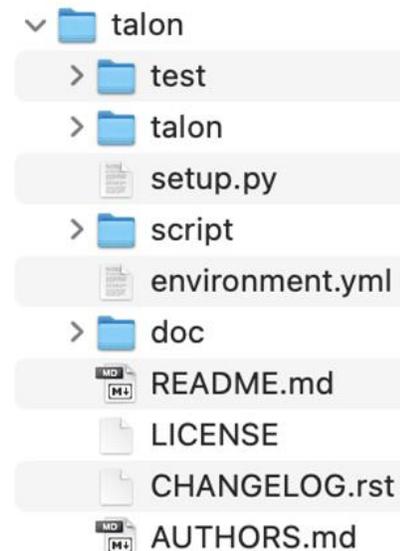
Dépôt source: déposer .zip / .tar.gz

L'auteur ajoute les fichiers suivants :

- README
- AUTHORS
- LICENCE (à choisir avec les titulaires des droits patrimoniaux du dit logiciel - liste de référence)



talon
Fichier - 472 Ko



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

Créer une archive compressée .zip
ou .tar.gz

Nommer de préférence avec le nom du
logiciel et sa version

Fichier README

- ★ Obligatoire:
- Nom du logiciel / projet
- Description du projet
- ★ Recommandé:
- Site du projet
- lien vers la documentation
- Contact & support
- Liste des fonctionnalités
- Environnement de développement
- build, installation, requirements
- comment exécuter le code
- ★ Possible:
- Usage - comment utiliser le code
- Nouvelles récentes du projet
- Visuels du logiciel

Références: [Software Release Practice HOWTO](#) by Eric Steven Raymond and [Make a README](#)

```
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
```

```
-----
```

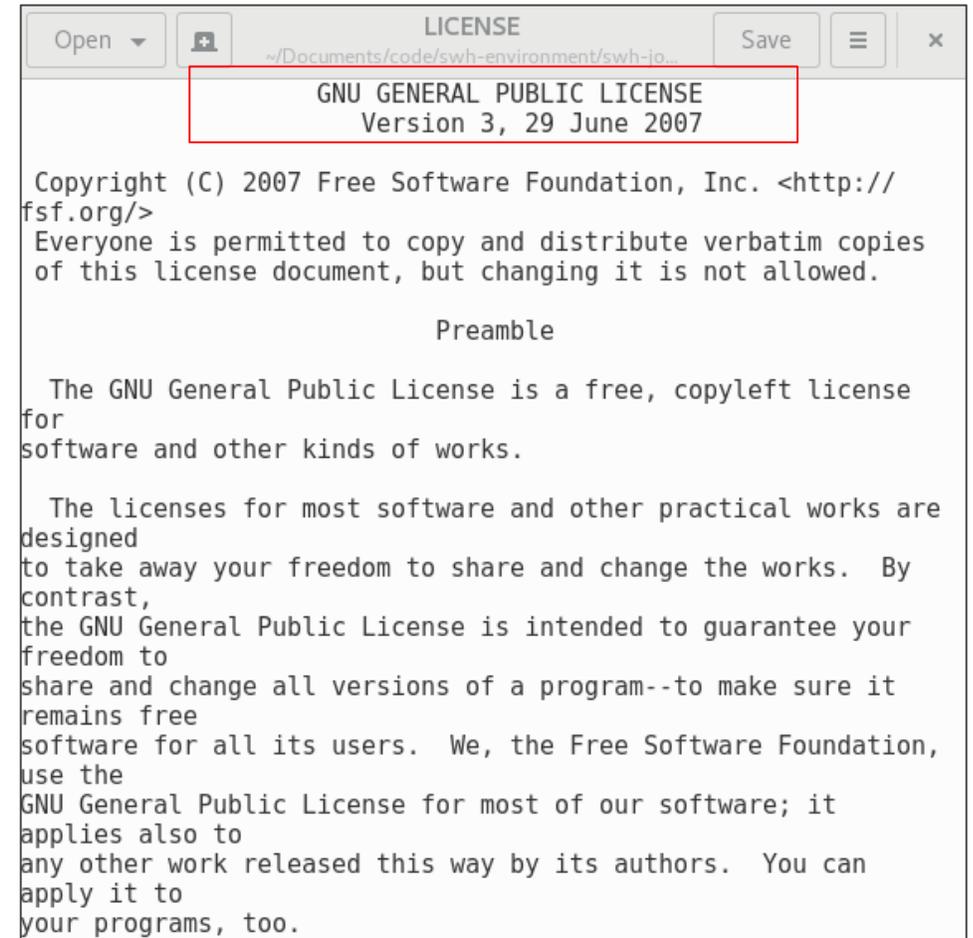
```
Installable Python kits, and information about using Python, are available at  
'python.org'_.
```

Readme du projet CPython archivé sur SWH :

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

Fichier LICENCE

- ★ Avant tout dépôt de code source, vous devez réfléchir à la licence que vous souhaitez pour votre code.
- ★ Consulter les personnes en charge dans votre établissement
- ★ Pour vous aider à choisir:
 - <https://choosealicense.com/>
 - <https://reuse.software/>
- ★ Noter la/les licence/s utilisé/s dans le code source dans les métadonnées
 - **La licence est vérifiée par les modérateurs**
 - **Le déposant est responsable de la compatibilité entre licences (entre le code déposé et ses dépendances)**



```
Open [icon] LICENSE Save [icon] x
~/Documents/code/sw-h-environment/sw-h-io...
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://
fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

                                Preamble

The GNU General Public License is a free, copyleft license
for
software and other kinds of works.

The licenses for most software and other practical works are
designed
to take away your freedom to share and change the works. By
contrast,
the GNU General Public License is intended to guarantee your
freedom to
share and change all versions of a program--to make sure it
remains free
software for all its users. We, the Free Software Foundation,
use the
GNU General Public License for most of our software; it
applies also to
any other work released this way by its authors. You can
apply it to
your programs, too.
```

Déposer les fichiers

- Depuis l'onglet "Déposer", sélectionnez l'archive compressée ou Glissez/Déposez

Il n'est pas possible d'ajouter plusieurs archives ou plusieurs fichiers distincts.

Le .zip doit contenir tous les fichiers et ne doit pas être décompressé avant la validation.

Le .zip ne doit pas contenir un autre .zip



The screenshot shows the HAL website interface. At the top, there is a dark blue header with the HAL logo and the text "science ouverte". Below the header, there is a navigation menu with links for "Accueil", "Dépôt", "Consultation", "Recherche", "Documentation", and "Mon espace". The main content area is titled "Déposer le(s) fichier(s)" and contains a large dashed box for file upload. Inside this box, there is a cloud icon with a downward arrow, a document icon, and the text ".zip / .tar.gz". Below the icons, it says "Glissez-déposez ou cliquez pour choisir vos fichiers" and "Taille maximale du fichier : 200M". To the right of the upload area, there is a section titled "Chargez les métadonnées à partir d'un identifiant" with a text input field containing "swh:1:dir:79b8c8755dbed34f01a6a7184fc196f3c58cb5d:origi" and a button labeled "Récupérer les métadonnées".

Compléter les métadonnées

→ Choisir le type de document* :
Logiciel

Pour afficher la liste complète des métadonnées cocher la case



BIBLIOTHÈQUE
UNIVERSITAIRE

*Les métadonnées obligatoires sont indiqués par un **

→ Ajouter les **métadonnées générales** :

- Nom*
- Domaine*
- Description (*Il est recommandé d'avoir une description cohérente avec le contenu du fichier README*)
- Mots-clés
- Identifiants
- Données associées
- Publications associées
- Date de production/écriture
- Editeur
- Institution

[Afficher la liste complète des métadonnées](#)

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.

Type de document * * Champs requis
Logiciel

Nom * Vous pouvez renseigner le titre en plusieurs langues : choisir la langue et cliquer sur + pour ajouter une nouvelle langue
TALON: Tractograms As Linear Operators in Neuroimaging
anglais

français

Domaine * Le domaine que vous mettez en premier, sera considéré comme la discipline principale du dépôt (article, logiciel ou autre). L'ordre des domaines peut être changé par glissé/déposé.
 Imagerie médicale
 Traitement du signal et de l'image [eess.SP]
Ajouter un domaine

Description Cliquer sur + pour ajouter une nouvelle langue
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.
anglais

Ajouter les métadonnées spécifiques

- Licences*
- Langage de programmation
- Code Repository
- Platform/OS
- Version
- Etat du développement
- Outils de développement

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

Compléter les données auteur(s)

- Ajouter un **auteur**
- Ajouter une **affiliation** (au moins un auteur doit être affilié)

- Il est possible d'ajouter plusieurs auteurs et de leur associer une **fonction** (développement, maintenance, design, architecture, débogage, documentation, test, support, management).

- Il est nécessaire de renseigner le/les auteurs qui sont mentionnés dans le fichier AUTHORS.

- Vous pouvez ajouter un fichier CONTRIBUTORS pour les personnes qui ne sont pas auteurs dans la notice.

Compléter les données auteur(s) ✓

Afficher toutes les options ✓

Matteo Frigo **IdHAL : matteofrigo** Auteur

- ATHENA - Computational Imaging of the Central Nervous System
- UCA - Université Côte d'Azur
- Ajouter une affiliation

Mauro Zucchelli **IdHAL : mauro-zucchelli** Auteur

- ATHENA - Computational Imaging of the Central Nervous System
- UCA - Université Côte d'Azur
- Ajouter une affiliation

Rachid Deriche **Auteur**

- ATHENA - Computational Imaging of the Central Nervous System
- UCA - Université Côte d'Azur
- Ajouter une affiliation

Samuel Deslauriers-Gauthier **Auteur**

- ATHENA - Computational Imaging of the Central Nervous System
- UCA - Université Côte d'Azur

Matteo Frigo

Fonction

- ✓ Développement
- Maintenance
- Design
- Architecture
- Débogage
- Documentation
- Test
- Support
- Management

Pour cet auteur ▼

Modifier

Supprimer

Choisir la fonction

Pour cette structure ▼

Pour cette structure ▼

Pour cet auteur ▼

Pour cette structure ▼

Pour cette structure ▼

Pour cet auteur ▼

Pour cette structure ▼

Pour cette structure ▼

Valider le dépôt dans HAL

- Valider le transfert et contribuer au patrimoine logiciel constitué par Software Heritage
- Accepter les conditions
- Cliquer sur Déposer

☰ 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**

Partager logiciels et codes source

Documentation :

- Guide de dépôt de logiciel sur HAL (version FR et version EN accessible sur la page)

<https://hal.archives-ouvertes.fr/hal-01872189>

- Guide d'utilisation de l'archive SWH pour sauvegarder et référencer des logiciels de recherche sur le site: <https://www.softwareheritage.org/save-and-reference-research-software/>

- Blog post sur l'utilisation de l'archive

<https://www.softwareheritage.org/2018/09/22/browsing-the-software-heritage-archive-a-guided-tour/>

6

RÉFÉRENCER SON TRAVAIL

Partager logiciels et codes source



Software Heritage

Référencer son travail

Une fois le code archivé, vous pouvez citer votre code :

- **Citation du code** : ajout du lien vers le dépôt complet sur Software Heritage
- **Citation d'une version précise** : ajout du lien de la version sur Software Heritage
- **Citation d'une ou des lignes spécifiques** dans une version précise : ajout du lien vers la ou les lignes de la version sur Software Heritage

Citer le code et ses versions

Features

- 🔍 Search
- ⬇️ Downloads
- 📄 Save code now
- ❓ Help

<https://pypi.org/project/pbxplore/>

🕒 26 February 2020, 15:38 UTC

<> Code Branches (11) Releases (0) Visits

★ Branch: HEAD 4f036f0 / pbxplore-1.3.8 /

🕒 History ⬇️ Download

🔗 Tip revision: `dca0c6a4bfce8871e5c82428914c18b58ccf0bd3` authored by Pierre Poulain on 17 October 2017, 22:22 UTC
1.3.8

| File | Mode | Size |
|-------------------|------------|-----------|
| doc | | |
| pbxplore | | |
| pbxplore.egg-info | | |
| CHANGELOG | -rW-r--r-- | 2.2 KB |
| LICENSE | -rW-r--r-- | 1.1 KB |
| MANIFEST.in | -rW-r--r-- | 155 bytes |
| PKG-INFO | -rW-r--r-- | 4.8 KB |
| README.rst | -rW-r--r-- | 3.2 KB |
| setup.cfg | -rW-r--r-- | 148 bytes |
| setup.py | -rW-r--r-- | 2.1 KB |

README.rst

PBxplore

Python 2.7 3.4 3.5 3.6 pypi package 1.4.0 build passing docs passing

PBxplore is a suite of tools dedicated to Protein Block (PB) analysis. Protein Blocks are structural prototypes defined by de Brevern et al. The 3-dimensional local structure of a protein backbone can be modeled as an 1-dimensional sequence of PBs. In principle, any conformation of any amino acid could be represented by one of the sixteen available

Citer les logiciels

Extension pour BibLaTeX pour citer du code et des logiciels :

<http://ctan.mines-albi.fr/macros/latex/contrib/biblatex-contrib/biblatex-software/software-biblatex.pdf>

Exemples :

R. Di Cosmo, M. Danelutto, X. Leroy, and S. Pelagatti, The OCamlP3I library version 1.0, 1998. LIC: GPL. SWHID:

[sw:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git/](https://sw.hiclipper.com/1/rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git/).

[Software] R. Di Cosmo and M. Danelutto, The Parmap library version 1.1.1, 2020. Inria, University of Paris, and University of Pisa. vcs: <https://github.com/rdicosmo/parmap>

Plus d'informations :

<https://www.softwareheritage.org/2020/05/26/citing-software-with-style/?lang=fr>

D'autres possibilités de partage

- Imports de dépôts et attribution automatique de DOI à chaque nouvelle version.

Zenodo et Github :

<https://guides.github.com/activities/citable-code/>

Zenodo et Binder :

<https://blog.jupyter.org/binder-with-zenodo-af68ed6648a6>



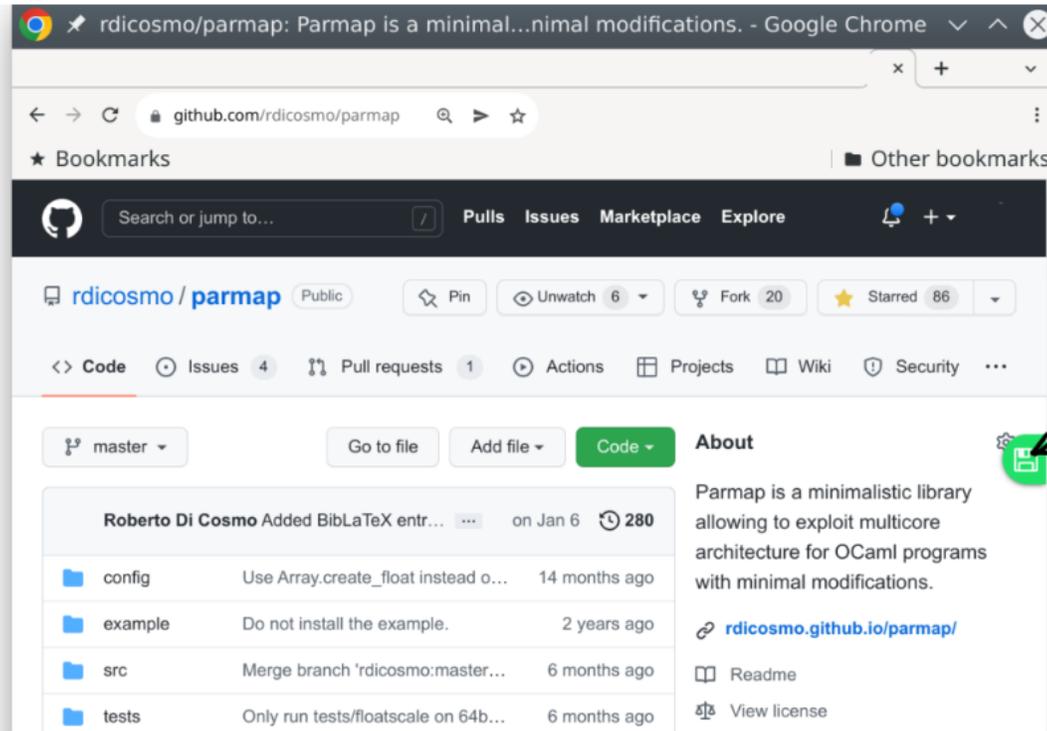
- *Attention, ces solutions ne remplacent pas un archivage pérenne sur Software Heritage*

7

OUTILS

Un plugin pour sauvegarder du code

UpdateSWH: check and update archival of a repository



This tab shows the archival status of the repository

Green up to date
Yellow not up to date
Grey: not archived yet
Red: not archivable (private)

Sauvegarder une forge

Features

 Search

 Downloads

 Save code now

 Add forge now

 Help

Administration

Request the addition of a forge into the archive

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



“Add forge now” provides a service for Software Heritage users to save a complete forge in the Software Heritage archive by requesting the addition of the forge URL into the list of regularly visited forges.

Submit a Request

Browse Requests

Help

Forge type *

bitbucket

Supported forge types in software archive.

Forge URL *

Remote URL of the forge.

Forge contact name *

Name of the forge administrator.

Forge contact email *

Email of the forge administrator. The given email address will not be used for any purpose outside the “add forge now” process.

I consent to add my username in the communication with the forge.

Comment

Optionally, leave a comment to the moderator regarding your request.

8

S'INVESTIR

Rejoindre la communauté Software Heritage



Rejoindre la communauté



BIBLIOTHÈQUE
UNIVERSITAIRE



Twitter/X: [@SWHeritage](https://twitter.com/SWHeritage)



Mastodon :
<https://mstdn.social/@swheritage>

LinkedIn:

<http://www.linkedin.com/company/swheritage>

Youtube:

<https://www.youtube.com/channel/UCe26PIE3YKTXyYvEY9HWdIIQ>

Contact:

info@softwareheritage.org

Ressources

Forge

Notre travail de développement logiciel concerne les référentiels Git, les logiciels de suivi des problèmes et des *bugs*, les révisions du code, etc.

[Plongez dans l'univers du code](#)

Canal IRC

Notre équipe de développement se trouve sur le canal IRC #swh-devel sur FreeNode.

[Discuter avec nous](#)

Liste de diffusion

Les discussions techniques sur Software Heritage ont lieu sur la liste de diffusion publique swh-devel.

[S'abonner, lire les archives](#)

Wiki

C'est le lieu où nous collectons des informations publiques, organisées par la communauté sur Software Heritage et son développement.

[Lire, contribuer](#)

Software Source Code IG RDA



BIBLIOTHÈQUE
UNIVERSITAIRE

<https://www.rd-alliance.org/groups/software-source-code-ig>



O&A Members

61

Active Organisational & Affiliate members

MEMBERSHIP

Members: 11670

Becoming a member of RDA is simple and open to both individuals and organizations

[Register now](#)

RDA Groups

WG & IGs: 95

Discover what RDA Working and Interest Groups and all other Groups are up to and find out how to join them. [Explore Groups](#)

[ABOUT RDA](#) [GET INVOLVED](#) [GROUPS](#) [RECOMMENDATIONS & OUTPUTS](#) [RDA FOR DISCIPLINES](#) [PLENARIES & EVENTS](#) [NEWS & MEDIA](#) [SEARCH](#)

[Home](#) » [Working and Interest Groups](#) » [Interest Group](#) » [Software Source Code IG](#)

IG

Software Source Code IG

Taxonomy:



Posts



Wiki



Events



Repository



Outputs



Charter



Plenaries



Members

[create new content](#)

Group Status: IG Established

[Join Group](#)

Status: Recognised & Endorsed

Chair (s): Roberto Di Cosmo, Neil Chue Hong, Julia Collins

Secretariat Liaison: enquiries[at]rd-alliance.org

TAB Liaison: Gretchen Greene

Software source code plays a critical role in all fields of modern research, where source code is written and developed to address a variety of needs, like cleaning, processing and visualising data.

Software source code is a necessary component for research reproducibility and reusability.

Thus software source code should be properly curated in the same way as other research inputs and outputs such as research data and paper publication. Software source code developers and organisations that sponsor software development should also be properly credited and attributed.

This interest group will provide a forum to discuss issues on management, sharing, discovery, archival and provenance of software source code. It will pay special attention to source code that generates research data and plays an important role in scientific publications.

Merci !

Un grand merci à Morane Gruenpeter,
de Software Heritage !

Sources des slides utilisées :

- Morane Gruenpeter, Jozefina
Sadowska. **Create software deposit:
User guide and best practices.**
[Technical Report] Inria; CCSD; Software
Heritage. 2018. [hal-01872189](https://hal.archives-ouvertes.fr/hal-01872189)



BIBLIOTHÈQUE DE SORBONNE UNIVERSITÉ

Cellule Données de la recherche & Humanités numériques : data-bsu@sorbonne-universite.fr

Service des Archives et du Recueil des Actes : sara-archives@sorbonne-universite.fr

SOFTWARE HERITAGE

Contact : info@softwareheritage.org



Sauf logos et mention contraire, cette présentation est mise à disposition selon les termes de la Licence Creative Commons Attribution 4.0 International.