

OCTOBRE 2024
OCÉANE VALENCIA

Archiver les codes et logiciels



Document confidentiel –
ne peut être reproduit ni diffusé
sans l'accord préalable
de Sorbonne Université.

L'archivage comme un outils au service des chercheurs

Les archives courantes et intermédiaires relèvent du régime de droit commun du RGPD et de la loi Informatique et Libertés, mais:

- ⇒ Les archives historiques bénéficient d'un statut spécifique, dérogatoire au régime de droit commun, statut justifié par la finalité de la conservation des archives (apporter des preuves, documenter l'histoire) qui implique la conservation d'archives intègres.
- ⇒ **Le SARA fournit à ceux qui en font la demande, un stockage sécurisé, pour permettre de conserver les données et archives encore non librement communicables avec un identifiant et une gestion des droits d'accès, pour le papier comme pour le numérique.**

Accompagner tout au long du cycle de vie des données – le Lab de ressources



Accueil

Politiques institutionnelles



Votre projet par étapes

Cadre juridique

Besoin d'accompagnement?



<https://labrador.sorbonne-universite.fr/>

LabRADoR, *Lab de Ressources et d'Accompagnement aux Données de la Recherche*, est un projet pilote pour répondre à vos questions concernant les données de la recherche, qu'il s'agisse de les ouvrir ou de les protéger.

Porté par l'unité de service SACADO et la bibliothèque de Sorbonne Université, ce portail a pour but de traiter vos questions dans toute leur complexité et leurs différentes dimensions (technique, juridique, documentaire) en faisant appel aux compétences de plusieurs structures de l'université, direction de la recherche et de l'innovation, directions de la recherche et de la valorisation, direction des services informatiques, direction des affaires juridiques et institutionnelles, déléguée à la protection des données.

Vous pouvez adresser toutes vos questions sur notre boîte mail : labrador@sorbonne-universite.fr



L'équipe LabRADoR

LabRADoR est un projet porté par la bibliothèque de Sorbonne Université et l'unité de service SACADO.

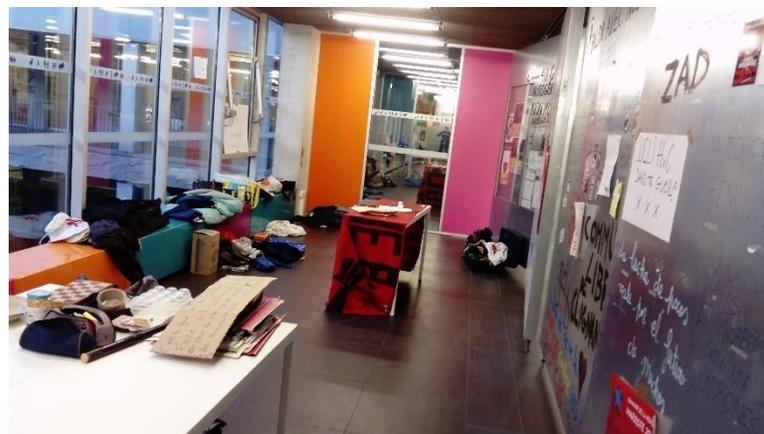
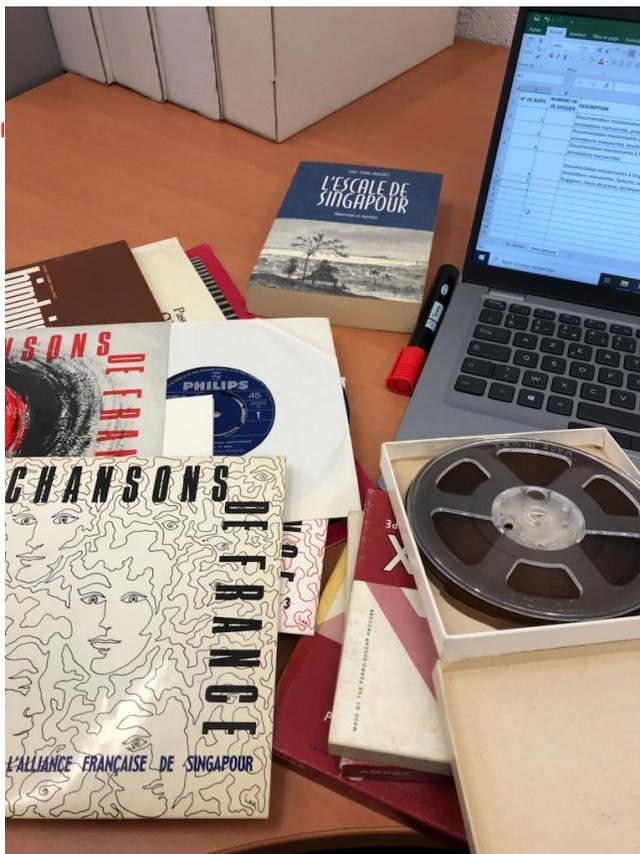
L'équipe travaillant sur la phase pilote de ce projet est composée de :

- Benjamin Abensour, DRV, faculté de sciences et ingénierie
- Meriç Akdogan, BSU
- Charles Allaire, DRV, faculté des sciences et ingénierie
- Cécile Arènes, BSU
- Nicolas Benoit, SACADO
- Ana Bobier, DRV, faculté de santé
- Koumba Koné, Bureau de la PI, DR&I
- Aurélien Moisan, BSU
- Barbara Van Doosselaere, DRV, faculté des lettres

1

ARCHIVER DES CODES SOURCES ET DES
LOGICIELS ?

LES ARCHIVES, C'EST QUOI ?



Un défi pour la recherche : la préservation des données

« For papers where the authors gave the status of their data, the odds of a data set being extant fell by 17% per year. In addition, the odds that we could find a working e-mail address for the first, last, or corresponding author fell by 7% per year. »

Vines TH, Albert AYK, Andrew RL, Débarre F, Bock DG, Franklin MT, et al.
The Availability of Research Data Declines Rapidly with Article Age.
Current Biology. 6 janv 2014;24(1):94-7.



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

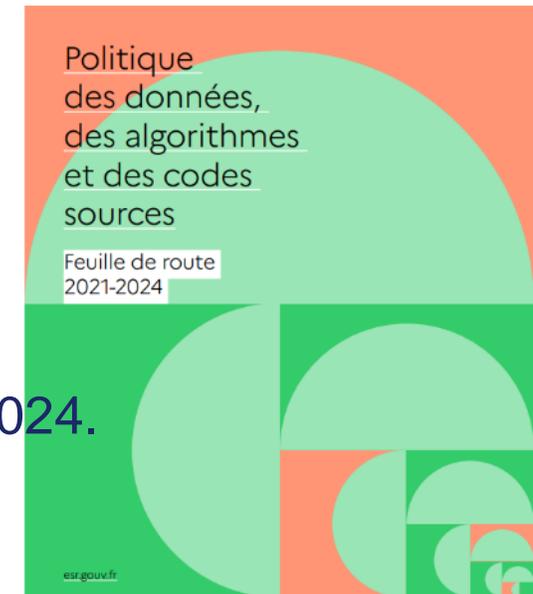
- Valoriser et soutenir la diffusion sous licence libre des codes sources issus de recherches financées sur fonds publics
- Mettre en valeur la production des codes sources de l'enseignement supérieur, de la recherche et de l'innovation
- Définir et promouvoir une politique en matière de logiciels libres



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

LES CODES ET LOGICIELS DE RECHERCHE, DES
ARCHIVES PUBLIQUES COMME LES AUTRES



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

<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

Les licences pouvant être utilisées par les administrations



La loi pour une République numérique a prévu la création d'une liste, fixée par décret, de licences qui peuvent être utilisées par les administrations pour la réutilisation à titre gratuit de leurs informations publiques, qu'il s'agisse de données ou de code source d'un logiciel (article D.323-2-1 du code des relations entre le public et l'administration (CRPA)).

Les licences applicables spécifiquement aux codes source de logiciels :

Licences permissives	identifiant SPDX
Apache License 2.0	Apache-2.0
BSD 2-Clause "Simplified" License	BSD-2-Clause
BSD 3-Clause "New" or "Revised" License	BSD-3-Clause
CeCILL-B Free Software License Agreement	CECILL-B
MIT License	MIT

Licences avec obligation de réciprocité	identifiant SPDX
CeCILL Free Software License Agreement v2.1	CECILL-2.1
CeCILL-C Free Software License Agreement	CECILL-C
GNU General Public License v3.0 or later	GPL-3.0-or-later
GNU Lesser General Public License v3.0 or later	LGPL-3.0-or-later
GNU Affero General Public License v3.0 or later	AGPL-3.0-or-later
Mozilla Public License 2.0	MPL-2.0

Il est précisé que l'obligation faite aux administrations de choisir parmi ces licences ne vaut que lorsque les administrations **peuvent** établir une licence de réutilisation. En particulier, les administrations participant à un projet déjà initié sous une licence déjà imposée ont bien entendu la faculté de contribuer au projet sous cette licence ([plus de détails](#)).

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**, concentrateur 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.**

Outils, résultats ou objets de recherche, panorama des plateformes au service des logiciels

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



Le logiciel pour documenter les projets de recherche et assurer leur reproductibilité

Documenter les projets de recherche :

- Jupyter notebooks
- RStudio
- OrgMode
- MkDocs



The screenshot shows a GitLab repository page for 'Notebook Api Nakala'. The left sidebar contains a navigation menu with options like 'Gestion', 'Programmation', 'Code', 'Déploiement', 'Opération', 'Surveillance', and 'Analyse'. The main content area displays the repository's file structure and a table of recent commits. The table has columns for 'Nom', 'Dernière validation', and 'Dernière mise à jour'. Below the table, there is a section for 'README.md' with the title 'Notebooks API NAKALA' and a brief description of the project.

Nom	Dernière validation	Dernière mise à jour
dataset-anf	Import dataset ANF-2021	il y a 3 ans
illustrations	add logo	il y a 3 ans
simple-dataset	Import dataset ANF-2021	il y a 3 ans
README.md	update README.md	il y a 3 ans
output.csv	add https://apitest.nakala.fr as exemple	il y a 2 ans
presentation-api.ipynb	Update how to create a group with roles	il y a 2 ans
requirements.txt	change requirement for MyBinder	il y a 3 ans
tp-depot-par-lot.ipynb	add https://apitest.nakala.fr as exemple	il y a 2 ans

Notebooks API NAKALA

Ces notebooks Jupyter ont été mis en place à l'occasion de l'ANF 2021: [Gérer ses données en SHS avec les services et outils proposés par la TGR Huma-Num](#). Ils ont été réalisés à l'aide de [Callisto](#), preuve de concept du [HN Lab](#).

Assurer la reproductibilité de la recherche :
-> Voir le [MOOC d'INRIA](#) : Recherche reproductible : principes méthodologiques pour une science transparente.

Différents niveaux d'ouverture



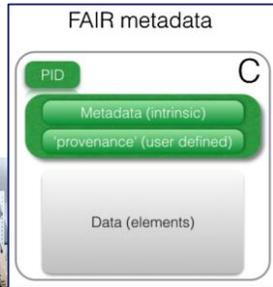
Archives papier

A. Paléographie antique et médiévale

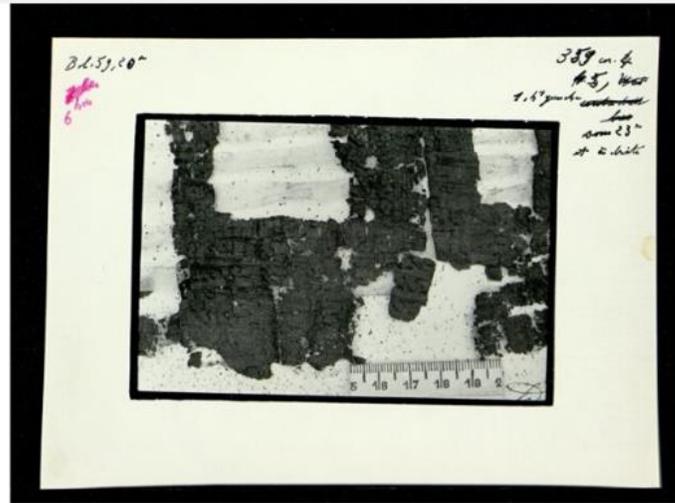
1. Évolution des écritures
 - Écritures grecques et latines du 1er au 5e siècle
 - Alphabet latin
 - Nomenclature des écritures livresques
 - B "baroque"
 - "Onciale" épigraphique
 - III ème - Xème siècles
 - XI ème - XII ème siècles
 - Pecia XII ème - XIV ème siècles
 - XV ème siècle, Humanistique
 - XVI ème siècle
2. Études et transcriptions, par lieux:
 - 2.1. Écritures romaines
 - 2.1.1. Ostie
 - 2.1.2. Rome
 - 2.1.3. Palatin
 - 2.1.4. Catacombes
 - 2.1.5. Herculanum
 - 2.1.6. Pompéi
 - 2.2. Afrique du Nord
3. Études et transcription, par support d'écriture
 - 3.1. Graffiti et dipinti
 - 3.2. Tablettes
 - 3.3. Amphores
 - Autres inscriptions
4. Vocabulaire militaire
 - Généralités
 - Vocabulaire par ordre alphabétique.
 - Écritures:
 - Suisse
 - Kaiseraugst
 - Vindonissa



Source: Scappaticcio, *Per i testi latini*.



Numérisation et ses métadonnées



FAIR data-
Open Access/Functionally Linked

Rustic Capital Writing: typed drafts and photos

Titre	Rustic Capital Writing: typed drafts and photos
Auteur	Robert Marichal
Date de création	Inconnue
Licence	Creative Commons Attribution 4.0 International License
Type	Autre
Descriptions	vue 17
Http://purl.org/dc/terms/publisher	VP
Http://purl.org/dc/terms/type	text ## image
Http://purl.org/dc/terms/identifier	PLATINUM_4MAR_10_2_5-19.6(532)

PID F

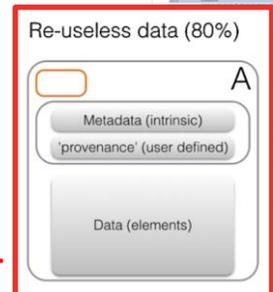
Metadata (intrinsic)

'provenance' (user defined)

Data (elements)

source: Marichal, Robert (2015) «Rustic Capital Writing: typed drafts and photos» [text ## image] NAKALA.
<https://doi.org/10.12801/a32894f>

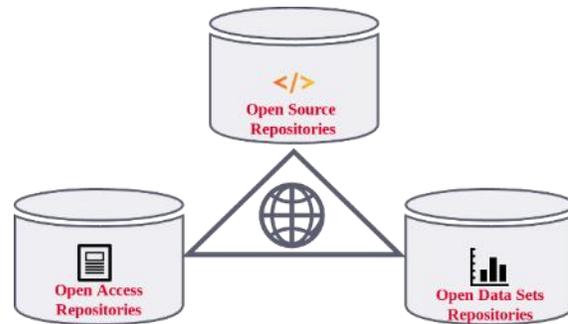
Données produites pendant le projet



3

LES ENJEUX SPÉCIFIQUES À L'OUVERTURE DES
CODES ET LOGICIELS

Pourquoi ouvrir les codes sources ?



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

“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 pour la science ouverte :

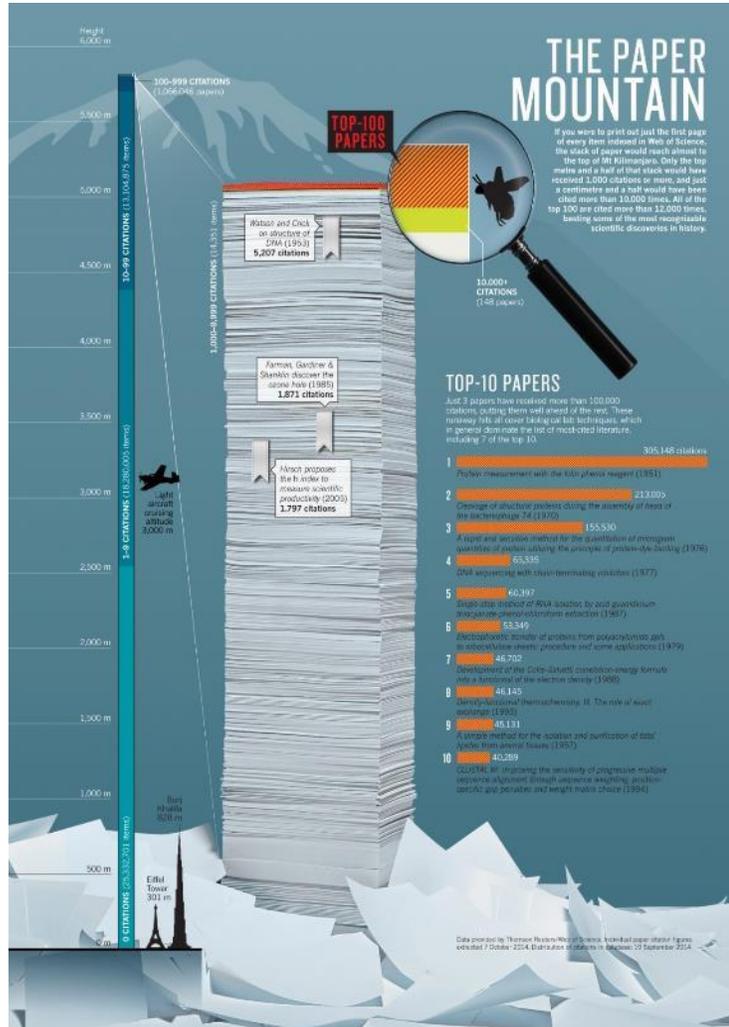
- open access
- open data
- open source repositories

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



Les principes FAIR pour les codes et logiciels, extrait : R pour 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.	Software metadata and documentation meet domain-relevant community standards.	Rephrased

4

LES OPTIONS : FOCUS SUR LA SAUVEGARDE DES
CODES SUR SOFTWARE HERITAGE



Software Heritage

une archive pour les codes et logiciels



- 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>

Depuis 2020 : accompagner avec Software Heritage



Deux profils d'ambassadrices :

- **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

Des actions communes :

- Offre de formation à SWH au catalogue
- Sur-mesure : audit, sensibilisation, aide au dépôt, formation, collecte retrospective
- Depuis 2023, avec le département Publications et Open Acces, modération des dépôts SWH dans HAL.

Partager logiciels et codes source avec



Software Heritage

Sauvegarder son code :

1. Préparer

2. Déposer (deux possibilités) :

- Dépôt direct dans Software Heritage
- Dépôt via HAL

Préparation du logiciel dans une forge

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

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 commit. The commit history shows several files added: bin, glpk_files, src, AUTHORS, LICENSE, README.md, and codemeta.json. The README.md file is open, showing the title 'projet en Recherche Opérationnelle' and a DOI '10.5281/zenodo.438684'. The README content describes a distributed system for solving a generalized assignment problem. The right sidebar shows project details, releases, and language usage (Java 83.7%, AMPL 16.0%, Batchfile 0.3%).

Préparer le logiciel pour un dépôt source

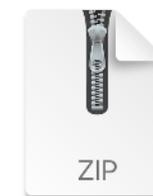
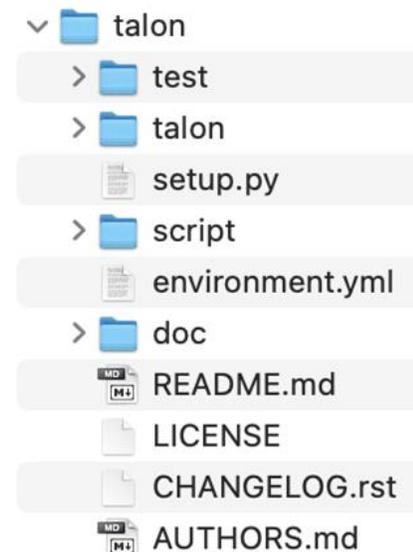
Ces *éléments*
sont *vérifiés*
par les
modérateurs
HAL ou les
archivistes

L'auteur ajoute les fichiers suivants :

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



talon
Cossier - 472 Ko



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

Dépôt source: déposer .zip /.tar.gz

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.

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/](https://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 HAL et les archivistes**
 - **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.
```

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](#)

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

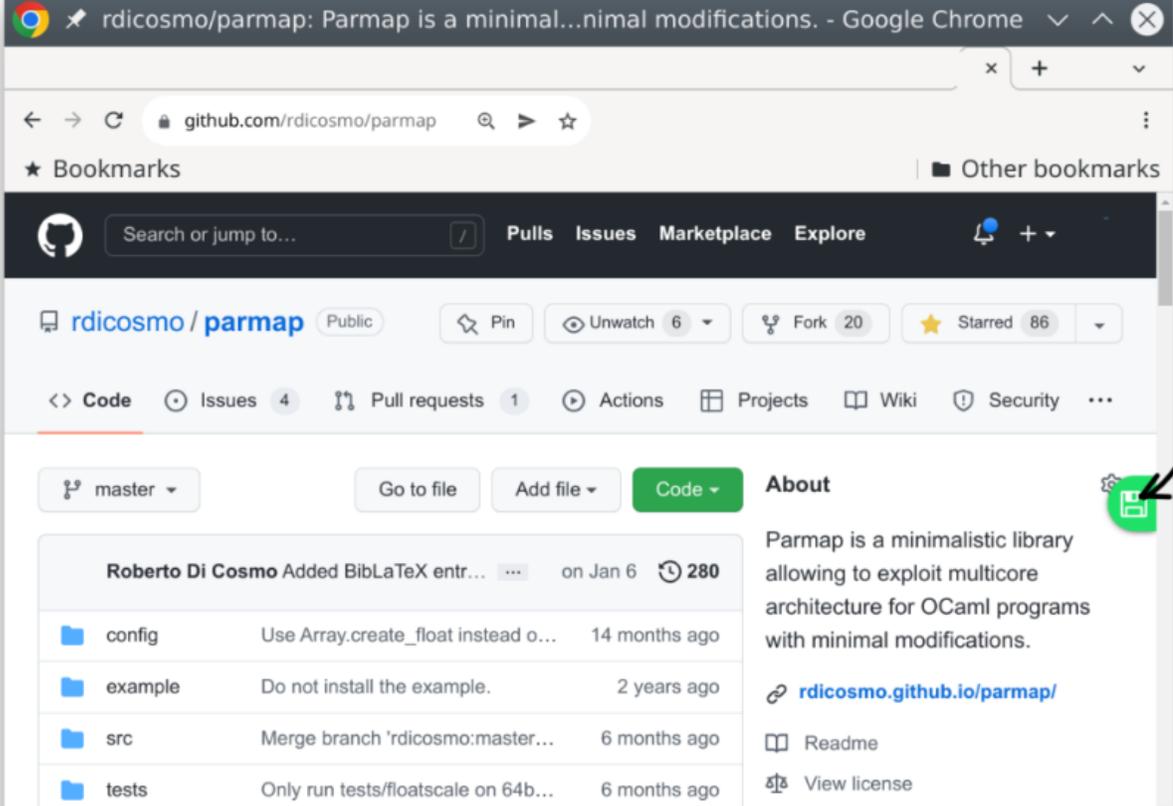
YYYY-MM-DD

Fichier CodeMeta: un exemple

```
{
  "@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"
      }
    }
  ]
}
```

Un plugin pour sauvegarder du code

UpdateSWH: check and update archival of a repository



The screenshot shows the GitHub repository page for 'rdicosmo/parmap'. The repository is public and has 20 forks and 86 stars. The 'Code' tab is selected, showing a list of files and folders. A green icon with a document and a checkmark is visible in the bottom right corner of the repository page, indicating the archival status.

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.

Un exemple d'article citant des logiciels déposés dans SWH :

- 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-02487579v1⟩](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/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/hal-02487579).
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/hal-02487579).
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/hal-02487579).
10. [SW EXC.] R. Di Cosmo, "Driver command," from **The OCamlP3l library** 1998. LIC: GPL. SWHID: [⟨swh:1:cnt:c428f4deb1cdff8500fff5c449b99454a816c163;origin=https://gitorious.org/ocamlp3l/ocamlp3l_cvs.git;anchor=swh:1:rev:2db189928c94d62a3b4757b3eec68f0a4d4113f0;path=/Tools/ocamlp3lrun⟩](https://sw.hal.archives-ouvertes.fr/hal-02487579).
11. R. Di Cosmo. "How to use Software Heritage for archiving and referencing your source code: guidelines and walkthrough." HAL preprint hal-02263344 . Apr. 2019.

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*

Et pour ce qui est protégé ?

- Dépôt au service d'archives le temps de sa durée d'utilité administrative et de la protection !

Dépôt source: déposer .zip /.tar.gz

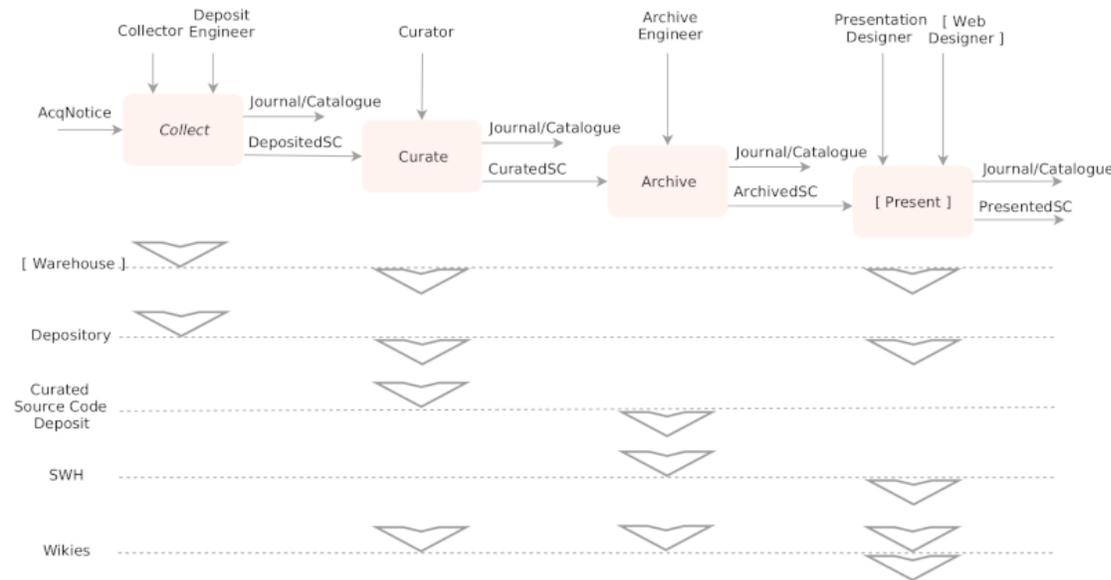
- Ajout des métadonnées dans le SIP (si un codemeta existe, possible d'en récupérer une partie pour alimenter), en particulier celles des habilitations d'accès et des délais de communicabilité, en accord avec le producteur.
- Archivage en fonction des options mises en œuvre pour l'archivage électronique (actuellement sauvegarde sur serveur avec accès restreint et sauvegarde hors ligne)

Pour les dépôts de codes et logiciels historiques :

SWHAP: le processus pour l'acquisition du logiciel historique

Une guide pour vous accompagner de bout en bout

The Software Heritage Acquisition Process





MERCI DE VOTRE ATTENTION

sara-archives@sorbonne-universite.fr

Oceane.valencia@sorbonne-universite.fr

<https://labrador.sorbonne-universite.fr/>

<https://www.sorbonne-universite.fr/archives>