Science ouverte
Université Grenoble Alpes

Les codes, des données pas comme les autres ?

Printemps de la donnée 2022

Qu’est-ce qu’un code source ?

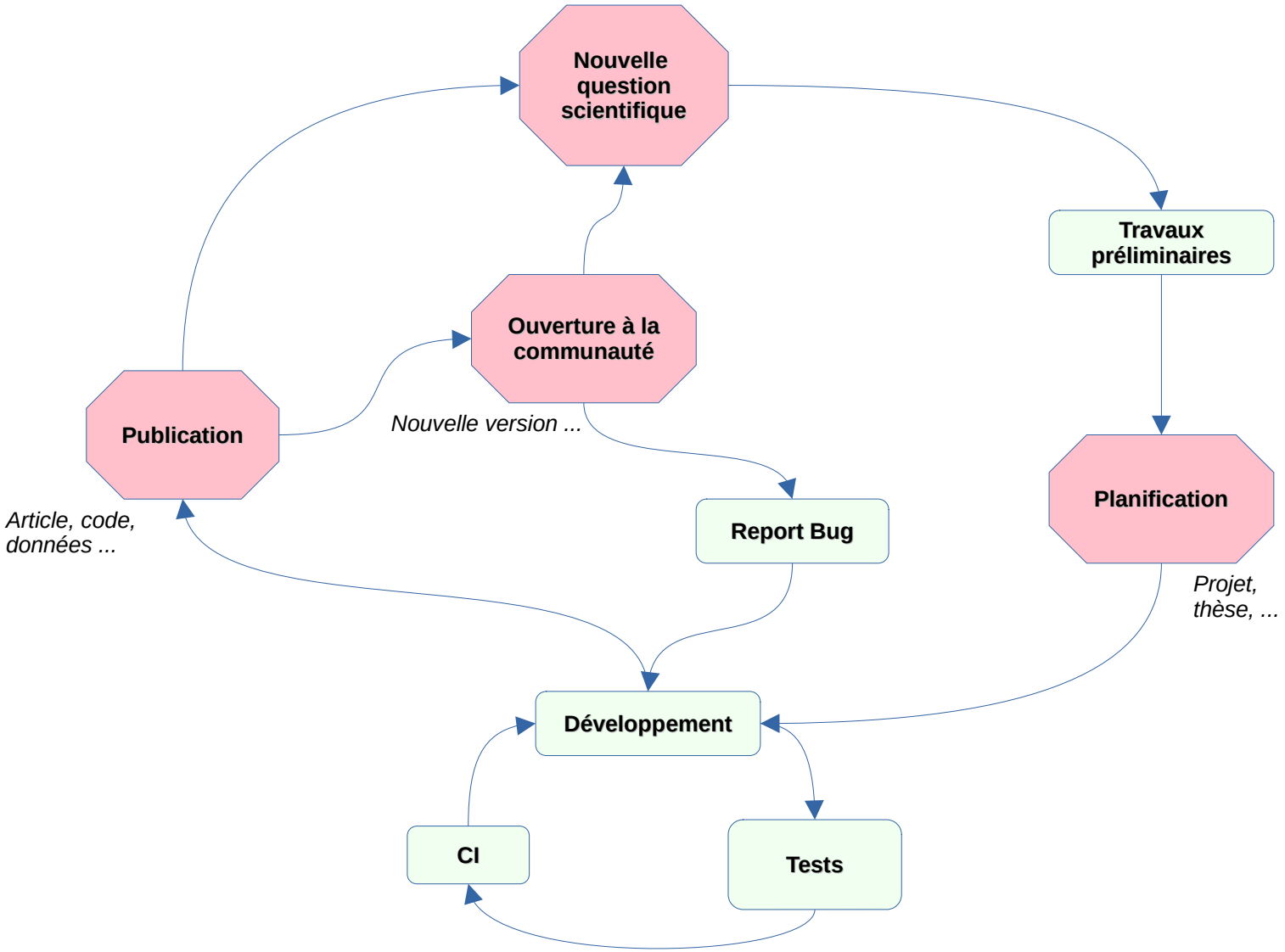
Science ouverte Grenoble Alpes

- Issu du rapport Bothorel :
  - Un code source peut être défini comme un ensemble d’instructions exécutables par un ordinateur.
  - Un algorithme n’est pas nécessairement informatisé.
  - De manière simplifiée, l’**algorithme est une recette de cuisine**, et le **code sa réalisation concrète**
- Code source vs exécutable
  - Un logiciel fonctionne grâce à du code exécutable, compréhensible uniquement par des machines.
  - **L’« âme » d’un logiciel est dans son code source**, c’est-à-dire dans les instructions telles qu’elles sont rédigées pour être lisibles par l’humain.
  - Seul le code source permet d’accéder aux informations techniques et scientifiques

Code de recherche : profils et usages

- Des **profils très variés**
  - Des codes développés par une personne (souvent un doctorant), une équipe, une communauté
  - Des complexités très différentes depuis les scripts à des programmes de millions de lignes, des niveaux de modularités et de liens à des dépendances externes très divers
  - Des durées de vie de quelques semaines à des dizaines d'années
- Des usages différents
  - Comme **outils de recherche** : collecte, traitement de données, tests de modèles … dans de nombreuses communautés disciplinaires
  - Comme **résultat ou objet de recherche** : en informatique ou en mathématique par exemple, en tant que preuve d'existence d'une solution algorithmique efficace à un problème donné
- Mais liés au processus de recherche et orientés vers un même objectif : la **production de connaissances scientifiques**
  Un des piliers de la recherche dans toutes les disciplines

Cycle de vie d'un code de recherche



Code de recherche vs données de recherche

Science ouverte Grenoble Alpes

- Les données de recherche sont plutôt passives, les codes sont **intrinsèquement vivants**
  - On ne change en général pas les données, collectées dans un contexte bien défini
  - On change éventuellement la façon dont on les traite et on les analyse (grâce à des codes)
  - Les codes sont associés à une **action** : création de connaissances, transformation d'informations, visualisation, ...
  - Le code peut être réutilisé tel que, en reproduisant son environnement et toutes ces dépendances mais on a surtout envie de le **modifier pour l'adapter** à nos besoins propres ou l'**enrichir de nouvelles fonctionnalités**

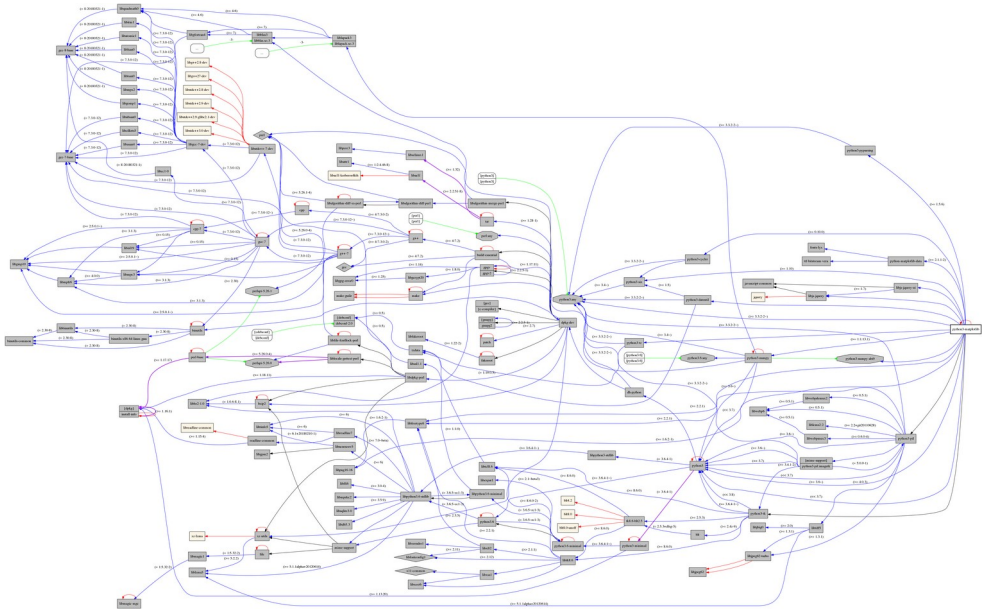
Code de recherche vs données de recherche

- Les codes représentent un **travail de création**, et correspondent à un cadre juridique différent de celui des données
  - Ils sont bien souvent développés collectivement, et la question de l'attribution des droits peut être très complexe (ce qui est également vrai pour les données)
  - D'un point de vue juridique, la création d'une **base de données** peut également être considérée comme un travail de création (dans le sens où il y a une « empreinte de la personnalité de l'auteur », un travail intellectuel …). Les BDD relèvent également du droit d'auteur comme le logiciel
- Les codes s'appuient sur des **dépendances et tout un environnement logiciel et matériel** qui évolue sans cesse
  - Cela complexifie les questions de reproductibilité

Reproductibilité

- Mes résultats ont changé ! Pourtant je n'ai touché à rien
- Mon code ne marche plus ! Pourtant il fonctionnait la semaine dernière
- Impossible de retrouver les résultats de cet article, pourtant j'ai réimplémenté l'algorithme car le code n'était pas disponible
- Impossible d'installer ce code qui est référencé dans cet article

- Il est évidement **nécessaire d'ouvrir les codes sources**
- Mais également de s'assurer de retrouver le **même environnement de travail** même longtemps après (alors qu'OS, bibliothèques, compilateurs ... ont évolué)
- Dépendances, piles logicielles : la complexité peut être rapidement très importante
  - *Un simple « import matplotlib » cache une large chaîne de dépendance*



Vers une recherche reproductible, HAL Id: hal-02144142

Forges logicielles au coeur du processus

Science ouverte Grenoble Alpes

- Système de gestion de rédaction, de partage et de maintenance collaborative de texte
  - Intégrant de nombreux outils : gestion de versions (git), rédaction en ligne, gestion de tâches, gestion de documentation, suivi de bugs, forums, ...
- **Coeur névralgique de tout développement**
  - Facilite la mise en œuvre de bonnes pratiques de développement
  - Absolument indispensable quand on développe à plus de un mais particulièrement utile aussi pour les développements individuels
  - Simplifie aussi la diffusion du logiciel, son référencement, son archivage ...
    - En particulier, c'est le seul endroit où les informations sont constamment à jour

La question des licences

- Le logiciel est protégé par le **droit d'auteur**
  - **Droits moraux** attachés à l'auteur
  - **Droits patrimoniaux** couvrent le monopole de l'exploitation, propriétés de l'administration employeur de ou des auteurs
  - Contrairement aux autres droits d'auteurs, il y a une « **dévolution automatique des droits patrimoniaux à l'employeur** »
  - Depuis le 15 décembre 2021, c'est vrai aussi pour les **stagiaires**

La question des licences

Science ouverte Grenoble Alpes

- Le droit d'auteur s'applique dès la **création du logiciel**, il est donc essentiel de pouvoir dater cette création
  - La preuve peut être faite par tout moyen, y compris des choses simples : enregistrement des versions, cahier de laboratoire, envoi de mail à soi-même avec la dernière version … tout ce qui a un **horodatage**
  - Preuve aussi par le dépôt APP
- S'il n'y a pas de droit explicitement donné à travers une licence, utiliser un logiciel relève de la **contrefaçon**.

Pas de licence == tous droits réservés

Le logiciel libre

- Deux grands types de licence :
  - **Licences libres ou Open Source**, termes plus ou moins similaires
    - Les licences dites « libres » (traduction bancale de l'anglais « royalty-free ») viennent de la Free Software Foundation
    - Open Source vient de l'Open Source Society
    - Ils n'ont pas exactement la même philosophie
  - Une licence libre **ne veut pas dire libre de droit**, bien au contraire !
  - **Licences propriétaires**, donc non libre c'est-à-dire que seul l'auteur ou l'ayant droit du logiciel peut le modifier.

Le logiciel libre

- L'**ouverture des logiciels** développés dans le cadre de la recherche publique est un point essentiel de la science ouverte

- Logiciel libre définit 4 types de libertés pour l'utilisateur :

  - Liberté **d'exécuter** le programme, pour tous les usages.

  - Liberté **d'étudier** le fonctionnement du programme, et **de l'adapter** à vos besoins. Accès au code source condition requise.

  - Liberté de **redistribuer** des copies.

  - Liberté **d'améliorer** le programme et de **publier vos améliorations**, pour en faire profiter toute la communauté. Accès au code source condition requise.

Codes de recherche et principes FAIR

- FAIR : Findable, Accessible, Interoperable, Reusable
  - **Ces principes sont-ils adaptables au logiciel ?**
- Questions ouvertes :
  - Comment définir les contributions et donc les citations ?
  - Comment intégrer l'environnement, les dépendances ?
  - Comment prendre en compte la dynamique du code dans un identifiant pérenne et unique ?
  - ...

Métadonnées

- Essentielles pour :
  - faciliter la **recherche d'un logiciel**
  - identifier son **usage**
- Outre les métadonnées spécifiques à la discipline concernée par le code, il existe des **schémas de métadonnées dédiés au logiciel** :
  - Citation File Format (CFF) : https://citation-file-format.github.io/
  - Fichier schema.org : https://schema.org/, schéma de métadonnées généraliste.
  - Fichier CodeMeta : https://codemeta.github.io/, est un format pour les métadonnées logicielles génériques, qui étend les fichiers schema.org.

Archivage, citation et identifiant unique

Science ouverte Grenoble Alpes

- **Deux piliers :**
  - Software Heritage
  - HAL

- **Software Heritage** (SWH, https://www.softwareheritage.org/?lang=fr) est une initiative lancée par INRIA en 2016, et soutenue par l'UNESCO qui vise à collecter, préserver, partager et archiver tous les logiciels disponibles publiquement sous forme de code source.

- **SWH + HAL**
  - Archivage du logiciel sur SWH
  - Identifier les objets avec un SWHID, la notice et la citation avec un HAL-ID
  - Décrire le logiciel avec les métadonnées adaptées à l'objet
  - Citer le dépôt avec une citation complète

Références

- Defining Research Software: a controversial discussion, FAIR4RS RDA WG. 10.5281/zenodo.5504015
- Software vs. data in the context of citation, https://doi.org/10.7287/peerj.preprints.2630v1
- Note d'opportunité sur la valorisation des logiciels issus de la recherche, https://dx.doi.org/10.52949/17
- Vers une recherche reproductible : Faire évoluer ses pratiques , https://hal.archives-ouvertes.fr/hal-02144142v3
- Software developpement for reproductible research, http://dx.doi.org/10.1109/MCSE.2013.91
- Assessment report on « FAIRness of software », https://doi.org/10.5281/zenodo.4095092

- **Merci aux collègues de la CDGA pour leur relecture et à Mona Roger juriste en PI à l'UGA pour son éclairage !**

Actualités

- Lancement en mars 2022 du **Collège « codes sources et logiciels »** du COSO qui fait suite au groupe thématique sur le même sujet. 5 thèmes abordés :
  - Identification et mise en avant de la production logicielle de l'ESR
  - Outils et bonnes pratiques techniques et sociales
  - Valorisation et durabilité
  - Liaison et animation nationale, européenne et internationale
  - Reconnaissance et carrières
- Évènement « **Open Science Days @ UGA** » du 13 au 15 décembre à Grenoble sur le thème des **codes et logiciels**
  - Notez la date !!