



RR
FR



FR Reproducible
Research network

Groupe de travail logiciels et environnements computationnels

Franck Pérignon (LJK, CNRS)

Simon Tournier (Université Paris Cité, INSERM)

*Journées du réseau français de
recherche reproductible*

Bordeaux, 19-21 mai 2026



Refaire (reproduce ou replicate) un résultat ?

		audit ?		opaque		depend ?
result	←	paper	+	data	+	analysis
data	←	protocol	+	instruments	+	materials
analysis	←	script	+	data	+	environment

- **audit ?** dépend principalement de nos pratiques individuelles
- **opaque** est la partie « difficile » (en général)
- **depend ?** de nos pratiques collectives

Refaire (reproduce ou replicate) un résultat ?

		audit ?		opaque		depend ?
result	←	paper	+	data	+	analysis
data	←	protocol	+	instruments	+	materials
★ analysis	←	script	+	data	+	environment

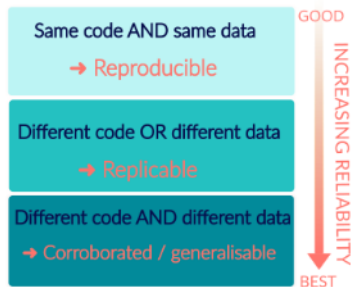
- **audit ?** dépend principalement de nos pratiques individuelles
- **opaque** est la partie « difficile » (en général)
- **depend ?** de nos pratiques collectives

★ *le sujet de notre GT*

GT "Logiciels et environnements computationnels"

Cadre : tout ce qui concerne la *reproductibilité* des procédés de calculs

👉 logiciels et leur contexte d'utilisation (à l'exclusion des « données »)



Reproductibilité : même code (et mêmes données) $\implies$ mêmes résultats

Logiciel est dual

human-readable vs machine-readable

source + **transformation** → **binaire**

Logiciel est dual

human-readable vs machine-readable

source + transformation → binaire

Programme (code source)

```
/* Hello World program */  
  
#include<stdio.h>  
  
void main()  
{  
    printf("Hello World");  
}
```

Logiciel est dual

human-readable vs machine-readable

source + transformation → binaire

Programme (code source)

```
/* Hello World program */  
  
#include<stdio.h>  
  
void main()  
{  
    printf("Hello World");  
}
```

Logiciel est dual

human-readable vs machine-readable

source + transformation → binaire

Programme (code source)

```
/* Hello World program */  
  
#include<stdio.h>  
  
void main()  
{  
    printf("Hello World");  
}
```

→

Programme (binaire)

```
4004e6: 55  
4004e7: 48 89 e5  
4004ea: bf 84 05 40 00  
4004ef: b8 00 00 00 00  
4004f4: e8 c7 fe ff ff  
4004f9: 90  
4004fa: 5d  
4004fb: c3
```


Logiciel est dual

human-readable vs machine-readable

source + transformation → binaire

Programme (code source)

```
/* Hello World program */  
  
#include<stdio.h>  
  
void main()  
{  
    printf("Hello World");  
}
```



Programme (binaire)

```
4004e6: 55  
4004e7: 48 89 e5  
4004ea: bf 84 05 40 00  
4004ef: b8 00 00 00 00  
4004f4: e8 c7 fe ff ff  
4004f9: 90  
4004fa: 5d  
4004fb: c3
```

 Software Heritage

 GitLab

reproducibilité (long terme)



Pourquoi est-ce plus « difficile » qu'il n'y paraît ?

Collaboratif, **O**uvert, **D**ocumenté, **E**xécutable
([CODE beyond FAIR](#))

- ① **code source** : gestion de versions, traçabilité, archivage, etc.
- ② **binaire / calcul** : environnements de production, d'exécution, etc.

Pourquoi est-ce plus « difficile » qu'il n'y paraît ?

Collaboratif, Ouvert, Documenté, Exécutable
(CODE beyond FAIR)

- ① **code source** : gestion de versions, traçabilité, archivage, etc.
- ② **binaire / calcul** : environnements de production, d'exécution, etc.

```
import math  
print(math.isclose(1.0,1.0000001))
```

FIGURE – code source du fichier script.py

Pourquoi est-ce plus « difficile » qu'il n'y paraît ?

Collaboratif, **O**uvert, **D**ocumenté, **E**xécutable
([CODE beyond FAIR](#))

- ① **code source** : gestion de versions, traçabilité, archivage, etc.
- ② **binaire / calcul** : environnements de production, d'exécution, etc.

```
import math  
print(math.isclose(1.0,1.0000001))
```

```
alice@laptop$ python3 script.py  
True  
blake@desktop$ python3 script.py  
False
```

Pourquoi est-ce plus « difficile » qu'il n'y paraît ?

Collaboratif, Ouvert, Documenté, Exécutable
(CODE beyond FAIR)

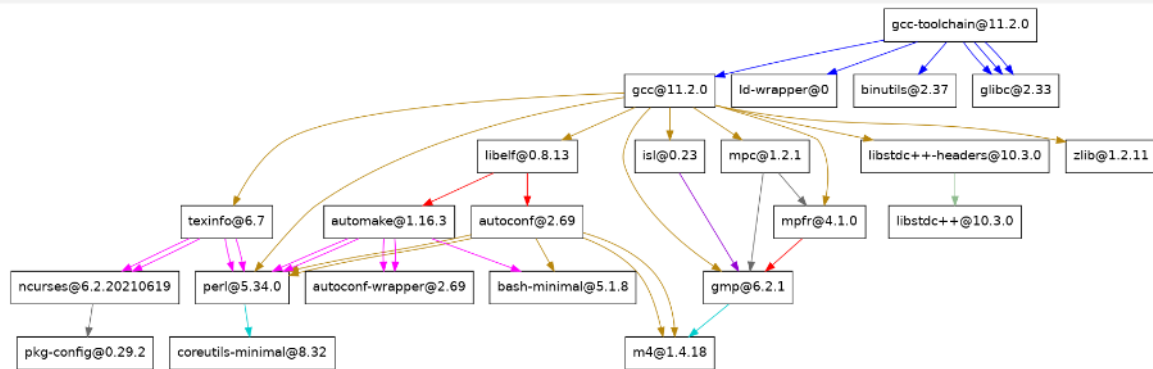
- ① **code source** : gestion de versions, traçabilité, archivage, etc.
- ② **binaire / calcul** : environnements de production, d'exécution, etc.

```
import math  
print(math.isclose(1.0,1.0000001))
```

```
alice@laptop$ python3 script.py  
True  
blake@desktop$ python3 script.py  
False
```

Alice utilise Python à la version 3.6 et Blake à la version 3.5
(les valeurs de `rel_tol` et `abs_tol` ont changé entre les deux versions)

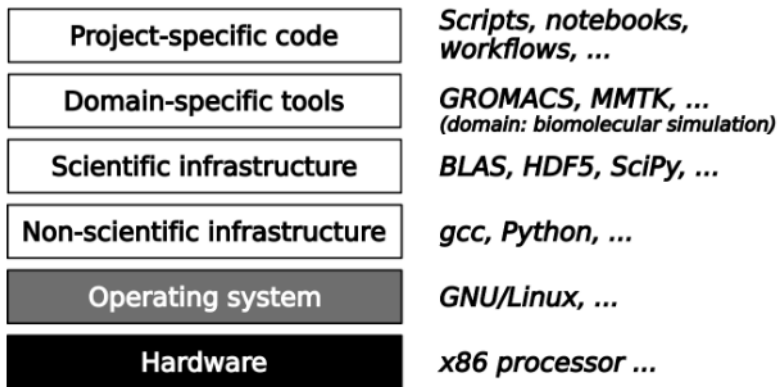
Quand Alice dit « GCC à la version 11.2.0 »



Graphe complet : 104 nœuds

Est-ce la même « version » de GCC si mpfr est remplacé par la version 4.0 ?

La « pyramide de Konrad »



Konrad Hinsén, <https://blog.khinsen.net/>

Qu'entend on par logiciel de recherche ?

(« formellement »)

Définition du COSO (COmité pour la Science Ouverte)

*[...] Les logiciels de recherche peuvent se formaliser de différentes façons (**une plateforme, un intergiciel, un workflow ou une bibliothèque, module ou greffon d'un autre logiciel**) et être ainsi en interaction dans un écosystème ou au contraire plus autonomes [...]*

⚠ On parle du code, pas des données

En d'autres termes, il s'agira de tout code source pouvant aller d'un script shell de quelques lignes à une bibliothèque parallèle de plusieurs centaines de milliers de lignes.

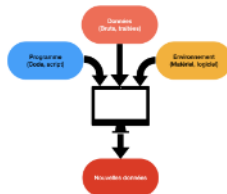
Etes vous concerné(e)s ?

Probablement !

Au final, que sont les environnements computationnels ?

Une collection de programmes, bibliothèques, utilitaires, etc. nécessaires au bon fonctionnement de votre logiciel

👉 Le **contexte précis** dans lequel votre code sera construit, exécuté, utilisé, etc.



Etes vous concerné(e)s ?

Oui !

Comment faire fonctionner votre logiciel ?

Reproduire les résultats numériques d'un papier ?

Reproductibilité et science ouverte pour les codes et logiciels de recherche

Quelques questions pour commencer :

- Qu'entend on par « codes et logiciels » de recherche ?
- Environnements computationnels ?
- Quelles démarches adopter ? Quelles pratiques mettre en place ?
- Quels outils utiliser ?
- Pour quels bénéfices ?
- Êtes vous concerné·e·s ?
- Avez-vous déjà été capables de reproduire un calcul ou une analyse et datant de 6 mois, 1 an, 2 ans, plus ?
 - Si oui, partagez votre pratiques, démarches, outils, etc.
 - Si non, partagez vos difficultés et collaborons

Logiciels, environnements : quels choix, attentes, exigences ... ?

Un outil pour résoudre un problème implique des arbitrages et des choix :

- très précis et intensif ou moins précis et modéré
- dépendance à une bibliothèque en développement (instable ?)
ou à une bibliothèque versionnée (stable ?)
- robuste, portable, performant ?
- être reproductible (avec quelles ressources et moyens humains ?)

Une piste (?) : **penser aux collaborateurices et à soi du futur**
leurs/se simplifier au maximum la vie

- ce qui est simple n'est pas toujours facile
- complexe ne signifie pas compliqué

Quels bénéfices à aller vers la reproductibilité computationnelle ?

- Augmenter la confiance dans les produits de la recherche, pour soi, pour les collègues et pour le grand public en général
- Encourager la réutilisation : codes et environnements peuvent être repris par d'autres équipes, dans un autre cadre
- Optimiser l'usage des ressources : temps, financements et infrastructures mieux utilisés.
- Garantir la fiabilité : cela permet de vérifier que les conclusions ne reposent pas sur d'éventuelles erreurs.

Objectifs du groupe de travail

Dans ce cadre, reproductibilité du procédé de calcul*

- Définir les concepts et les méthodes
- Identifier et présenter les difficultés, orienter vers des solutions
- Recenser les ressources documentaires disponibles
- Fournir des ressources pour aider à la sensibilisation et à la formation

👉 Encourager et faciliter la mise en place de pratiques menant à la reproductibilité

*en excluant les données

Les axes de travail du groupe

Reproductibilité du procédé de calcul

4 axes de travail, séries de questions ouvertes

- **Identification**

- Comment citer ? que citer ? quoi décrire ?
- Arbre de dépendances ? Quelles dépendances (construction vs exécution) ?
- Construction ? On lit le code source mais on exécute un binaire

- **Utilisation**

- Reproductibilité bit-à-bit ou pas ? parfois oui, parfois non, pourquoi ?
- Métrologie du calcul ? Et les nombres flottants ?
- L'environnement logiciel ?
- Le matériel ?

Reproductibilité du procédé de calcul

4 axes de travail, séries de questions ouvertes

- **Développement**

- Bonnes pratiques (gestion de version, documentation,...)
- Quelle finalité (script glue vs bibliothèque support ?)
- Quelle limite entre simple "Utilisation" vs "Développement" ? Parle-t-on alors de la même "reproductibilité logicielle" ?
- Maintenance ?

- **Publication**

- Que publier ?
- Comment ?
- Comment évaluer ?

Organisation et fonctionnement du groupe

- Une dizaine de personnes dans l'équipe de pilotage
- Une liste de diffusion : `gt-env-logiciel [at] groupes.renater.fr` (une trentaine de personnes actuellement)
- Un groupe Gitlab [hébergé chez Gricad-gitlab](#)
- Une réunion mensuelle (visio)
- Travail ciblé (2-3 personnes) sur des questions ponctuelles

Rédaction de fiches (*boussole*)

Comment produire un logiciel reproductible ? Où, quoi, comment accéder à un code source ? Quelle est la différence entre développer et déployer ? Conteneur, en ai-je besoin ? Mais qu'est-ce déjà ?

- Un site web reproducibility.gricad-pages.univ-grenoble-alpes.fr/guidelines-4-computing
- Fiches
 - Définitions
 - Parcours "bonnes pratiques"
 - Fiches techniques (outils ...)

Difficultés...

- Maintenir une dynamique (visio, travail non prioritaire, etc.)
- Ne pas être trop centré·e·s "ingé calcul et informatique"
- Ne pas dupliquer les ressources déjà existantes
- Ne pas être trop technique (mais suffisamment ...)
- ...

Appel à participation !

N'hésitez pas à nous contacter : [gt-env-logiciel \[at\] groupes.renater.fr](mailto:gt-env-logiciel@groupes.renater.fr)
Voir notre page "[How to contribute ?](#)"

Quelles contributions ?

Quelque soit votre profil et votre communauté scientifique, vous pouvez contribuer utilement !

- Relire, commenter
- Proposer de nouvelles idées
- Pointer des ressources venant de vos champs
- Rédiger des fiches
- ...

reproducibility.gricad-pages.univ-grenoble-alpes.fr/guidelines-4-computing

Un grand merci à toutes les personnes ayant participé au GT !