Mooc recherche reproductible

Alexandre Changenet

Module 1: Cahier de notes, cahier de laboratoire

1. Nous utilisons tous des cahiers de notes

2. Un aperçu historique de la prise de notes

- Index, chapitre etcs ... aspects organisationel
  - Passage du volumen (rouleau de papyrus) au codex => éléments organisationnels du livres.
  - Codex => reliure des papyrus + alternative au papyrus = le parchemin
  - Rubrication => Permets de séparer les paragraphes (couleurs, ou blancs)
  - Notes => Linée
  - Idem en Chine (volumen au codex), impression sur papier
  - Eusebe et les canons eusébien => Table de références croisées
- Navigation dans les notes
  - Fiches => Permette de naviguer, plus certains découpent directement
  - Numérotation
  - Mots clefs
  - Méthode de John locke => Indexations 1ere lettre du mot et première voyalle

3. Fichier texte au balisage léger

Editeur de texte =! éditeur de texte

- Pourquoi utiliser un éditeur de texte ? Car UTF-8 permettant de les lire des années plus tard
- Problème pas possible mettre en gras ou d'utilier hyper lien. Pas facile de collaborer (track changes)

Langage de balisage léger permets de:

- Confort de lecture
- légereté d'un fichier texte
- Et organisation
- Convertir en pdf, word etc ...

On peut facilmement introduire du `chasse fixe` pour du code par exemple

Ici j'introduit une image de mon sosi 

- Une stratégie qui est souvent employée et qui fonctionne bien en pratique consiste à faire le gros du travail de rédaction d'un article ou d'un mémoire en `Markdown`. La rédaction terminée, le fichier est exporté au format docx (ou `LaTeX`) et des ajustements de mise en page sont alors effectués avec un logiciel de traitement de texte (ou un éditeur `LaTeX`).
- TinyTex => Editeur latex léger pour R fait par le type de Bookdown
- TEI => Text encoding Initiative

4. Pérénité et évolubilité avec la gestion de versions

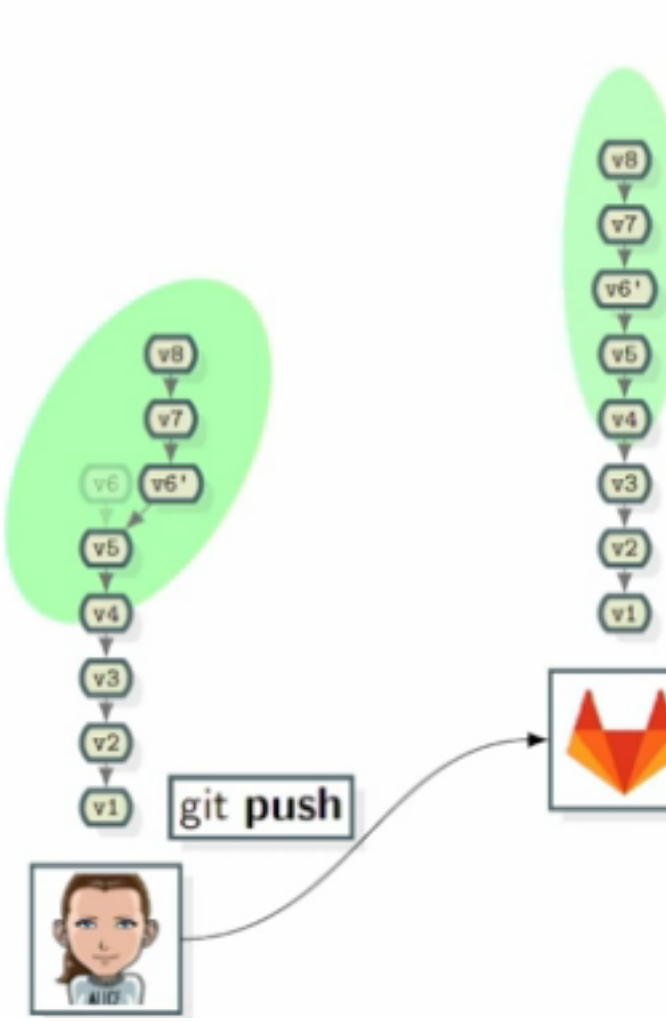
- Problème de libre office par exemple:
  - Sauvegarde indépendante de la getion des versions
  - Pas du fichier texte
- Docuwiki est une solution qui a des avantages mais aussi des inconvénients
- Git meilleur outil actuel car permets de corriger plusieurs fichier en même temps (ce qui est crucial quand on travail sur des logiciel par exemple)

Démystifions Git, Github, Gitlab

- Historique des fichiers sans mes démultiplier

- Fusions facile Petite blague rigolote
- Git est explicite
- git add pour indiquer quelle modif on veut ajouter (equivalent cocher des trucs sur Rstudio)
- Ce qui est important c'est la branche principale
- SHA1 est l'ecnodage des modif qui permets de suivre les modifs.
- git diff permets de comparer plusieurs versions (v4 et v8 par exmple avec le sha1)
- git checkout permets de revenir a une version précédente
- process:
  - Cloner un repertoire vite
  - Travailler en local
  - Commit et push

– Un push ne permets que de pusher la branche principale ! A moins de spécifier explicitement que l'on veut pusher cette branche aussi. Exemple:



2. Travailler à plusieurs

- On ne peut pas pusher si quelqu'un a push avant nous
- Obliger de faire un pull d'abords
  - Si travail sans conflit (parties du code différentes et/ou fichiers différents) pas de soucis
  - Si conflit => inspection et merge
  - push
  - et pull par quelqu'un d'autres
- Ce système est très résilient !
- Attention a ne pas faire des commit pour rien !! Genre réindentation de code
- ou bien dissocier les commit de fond et de forme
- git statuts, git diff, git add avec des petits commits logiques !
- Eviter git commit -a !!
- Favoriser le format texte !
- Git hist => Ensemble des modifs !

3. L'écosystème Git

- Permetttent de faire des commit en ligne ! Pratiques pour les petites choses

- Issues cool pour les devloppeur
- Faire de la revue de code
- Fork et pull request
  - Pour proposer des modifs sur un trucs sécurisé exemple li,ux =)> git fork => faire une copie du projet sur son espace perso
  - Puis pull request pour demander à ce qu'il y est revue de code
  - intégration continue => tests
  - connexion a des archives
  - github leaders mais logiciel propriétaire
  - gitlab logiciel libre permets de déployer des instances

4. Etiquettage et indexation: se retrouver dans ses notes

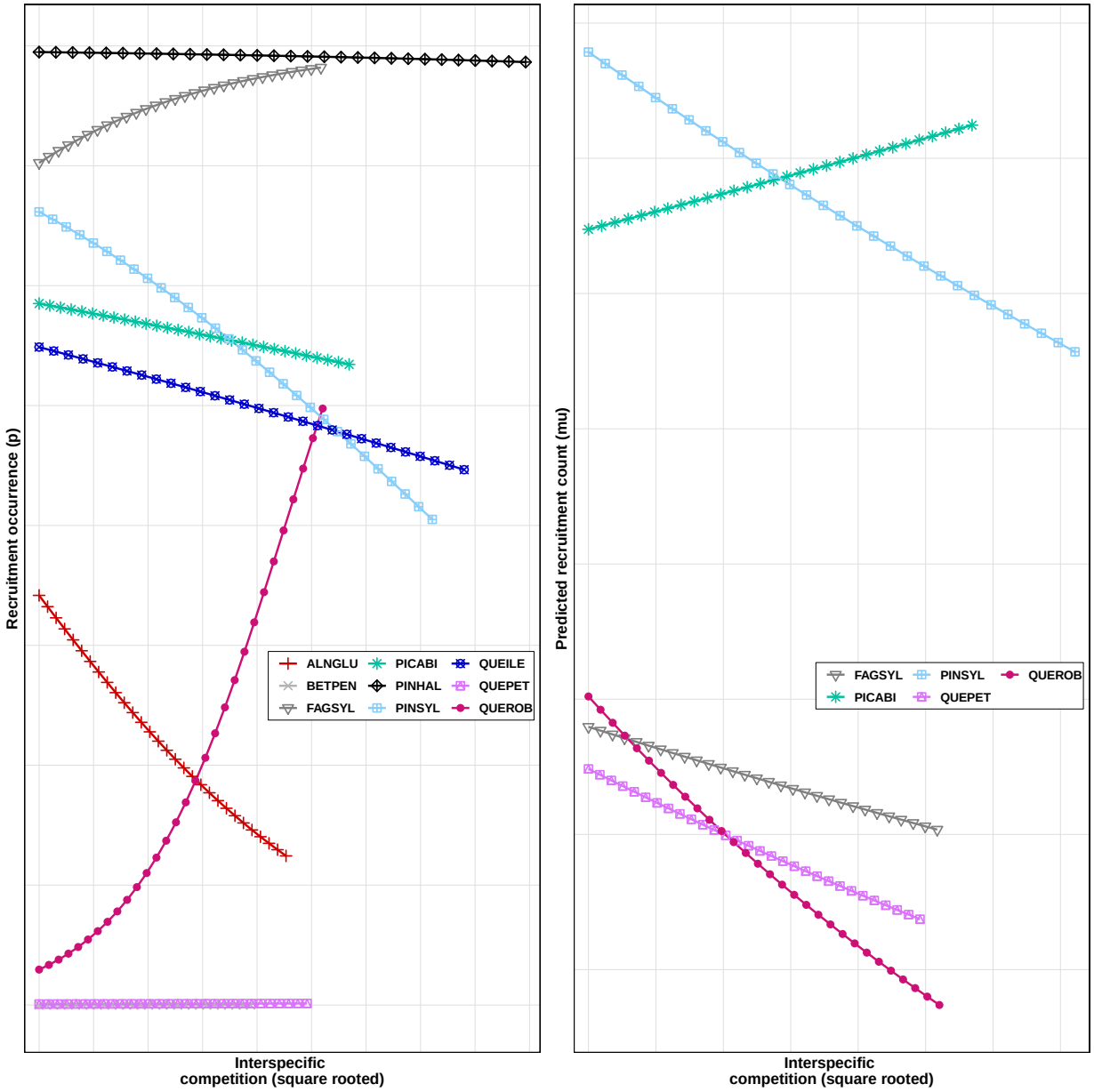
- Recherche dans un fichier texte
- Index dans des fiches => locke ou Leibniz
- Pratiques modernes:
  - Indexation au sens large (même fichiers non numériques)
  - Moteur de recherche de bureau: **DocFetcher**
    - Localisation d'occurrence dans beaucoup de repertoire du disque dur.
    - Problèmes quand beaucoup de fichiers: Submergé par le nombre d'infos
    - Ajout des étiquettes ou **mot clés Markdown avec des étiquettes**
    - Recherche ensuite avec les étiquetttes
- Métadonnées ajoutées sur des images aussi avec * exiftool *

Exo

Here I will try to import a rds figure file, print it to a pdf file, and finally annotate it using the pdf_write_metadata function of the R package threadr

Here we load the file and we save it

```
save_plot(filename = paste0("~/Google Drive/Cours/MOOC_Recherche_Reproductible/",MyVAR,"Test2.pdf"),
          plot = pall,base_width = 8, base_height = 8, dpi = 300 ,units = "in",nrow=1,ncol=2)
```



And finally we use the homemade function to tag the pdf output

```
pdf_keywords(
  paste0("~/Google Drive/Cours/MOOC_Recherche_Reproductible/",MyVAR,"Test.pdf"),
  Keywords=":MOOC::TEST:"
)
```

```
## [1] "    1 image files updated"
```