Un système de workflow au service du docking

A Barnabé, H Ben Yedder, M Davy, N Poulain

université PARIS-SACLAY

MASTER BIOINFORMATIQUE PARIS SACLAY

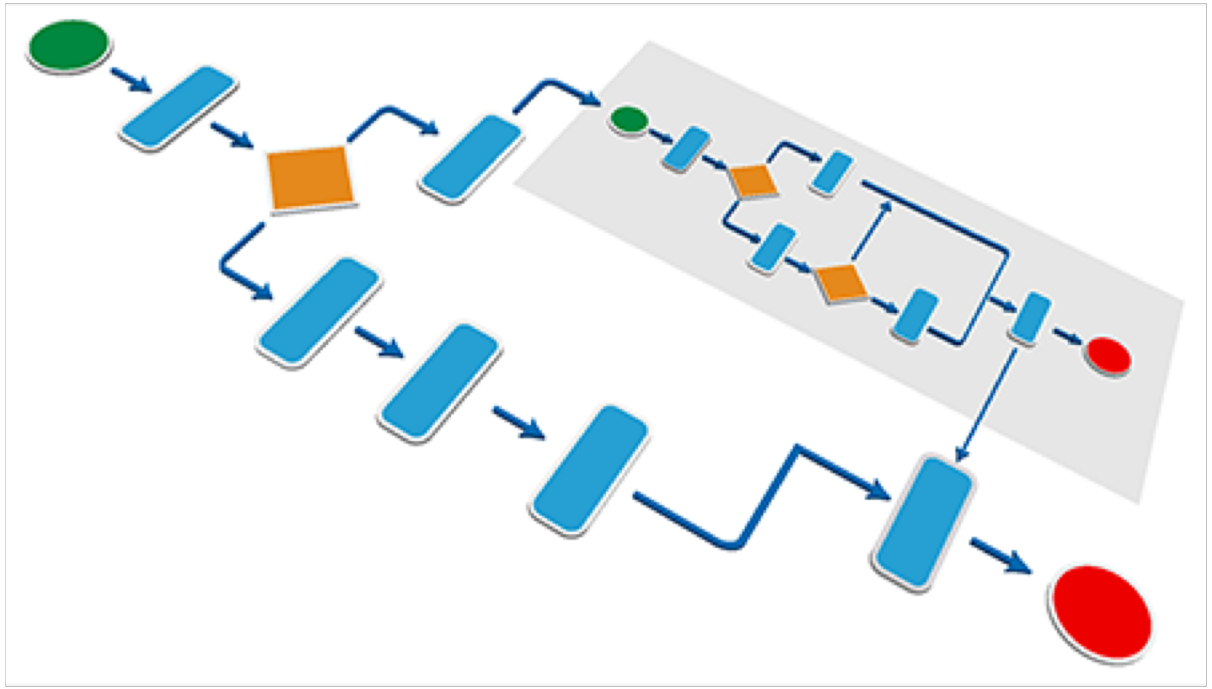
Le système de workflow

Pourquoi ?

- Besoin de reproduire des résultats
- Besoin d'un environnement « stable » *(librairie, os, logiciels)*
- Pouvoir utiliser le workflow avec d'autres entrées
- Avoir un workflow robuste

Définition

- Décrire le circuit de traitements de données
- Système de workflow : logiciel executant le workflow
  - Snakemake
  - Knime
  - Galaxy
  - …

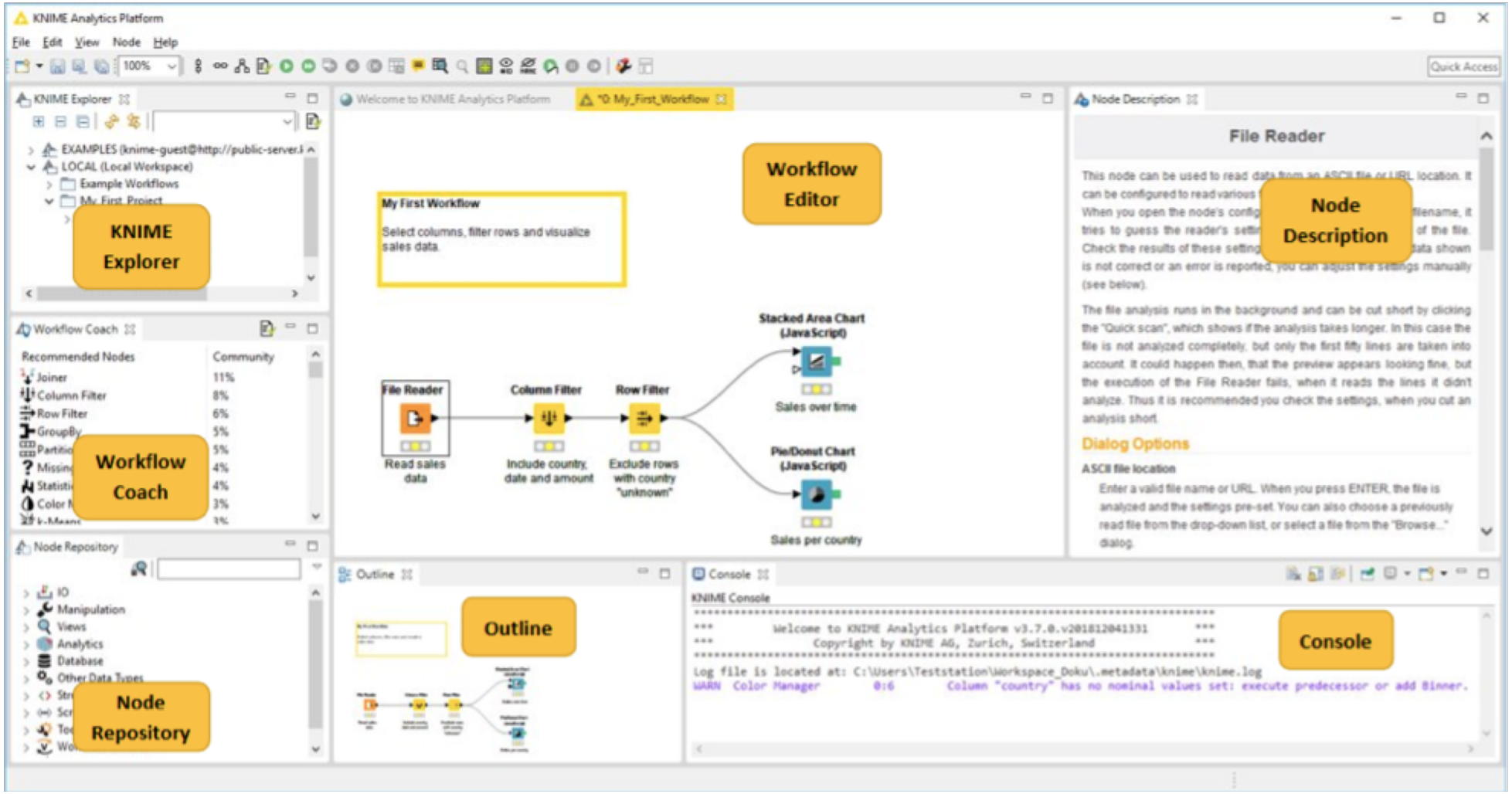


Des exemples

- Logiciel intuitif pour l’analyse de données
- Interface graphique pour la construction du workflow
- Plateforme open-source avec des extensions robustes
- Utilisation dans de nombreux domaines (finance, industrie, sciences…)

Open for Innovation®

KNIME



Source : https://www.knime.com/getting-started

Snakemake

- Workflow se basant sur le langage python
- Fonctionne avec des règles
- Création de fichiers

```
rule targets:
    input:
        "plots/dataset1.pdf",
        "plots/dataset2.pdf"

rule plot:
    input:
        "raw/{dataset}.csv"
    output:
        "plots/{dataset}.pdf"
    shell:
        "somecommand {input} {output}"
```


Source : https://snakemake.readthedocs.io/en/stable/

Snakemake

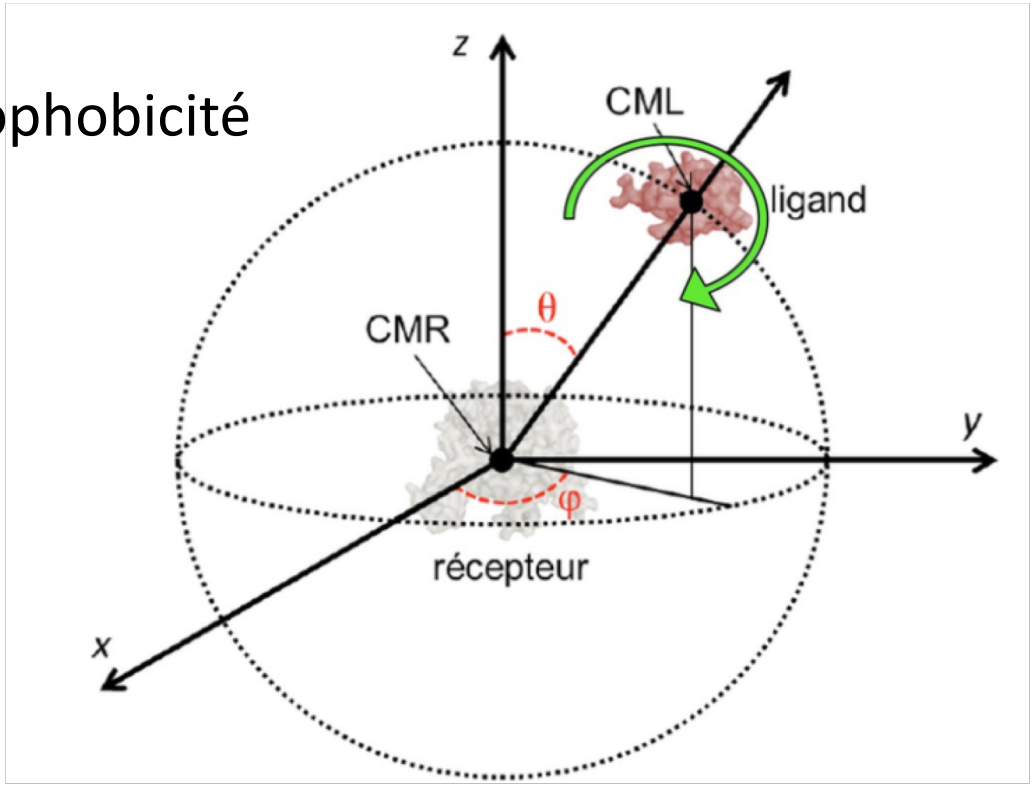
- Possibilité de stocker un ensemble de **paramètres** dans un fichier (yaml, bam)
- Permet de faire appel à différent scripts *(dans différents langages)* en interagissant avec le terminal
- Permet la **parallélisation**
- Pas d’interface graphique
- **Simple** d’utilisation

Cas pratique

Rendre reproductible notre projet de docking

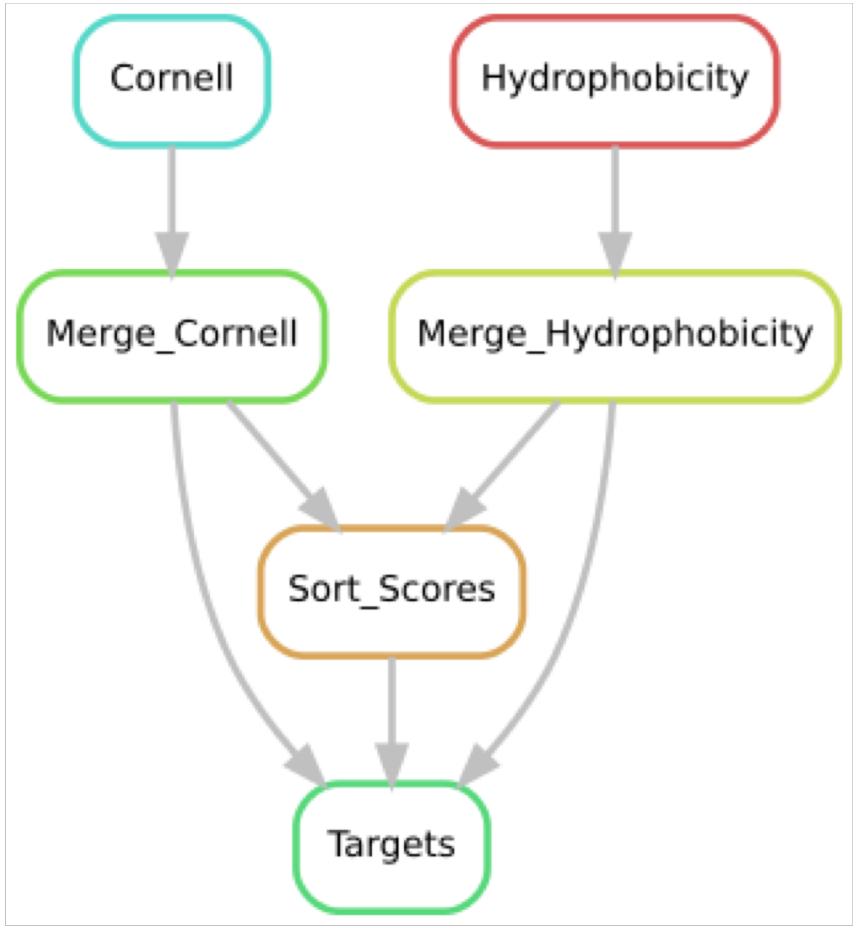
Docking

- Objectif : trouver la position optimale du ligand par rapport au récepteur
- 948 positions et orientations d'un ligand
- Calcul de deux scores : Cornell et hydrophobicité
- Un récepteur fixe



- Problèmes rencontrés:
  - Peu de documentation pour l'intégration de Python
  - Système de gestion de données rigides (format, structure)
  - Intégration de python sous forme de scripts indépendants (pas d'include local)
  - Requière une modification profonde du code
- Intérêts :
  - Adapté à de l'analyse statistique sur de grands jeux de données homogènes
  - Conservation du protocole statistique et de paramétrage

Utilisation de Snakemake



Calcul des scores

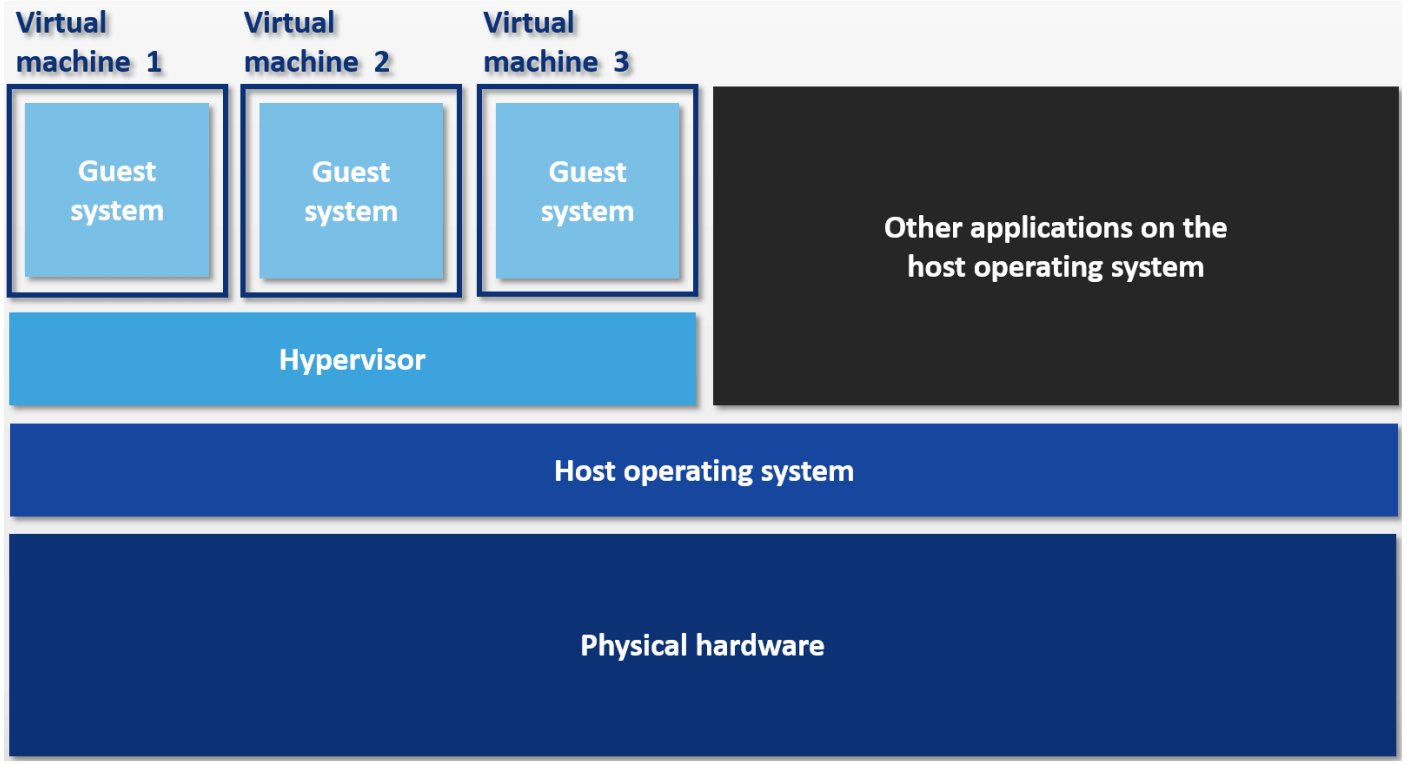
Tri des scores et écriture du meilleur hit

Snakemake

>snakemake --cores X

Machine Virtuelle

- 3ème ingredient de la reproductibilité
- Environnement isolé
- Robustesse du projet
- Portabilité de la machine virtuelle



VirtualEnv

- Spécifique à Python

- Machine virtuelle dédiée à une version Python

- Integration de Snakemake

- Commandes pratiques :
  - pip_pypy install virtualenv
  - pypy –m virtualenv name

python™
virtualenv

Conclusion

- **Knime** n’était pas adapté pour ce projet
- Le workflow conçu est fonctionnel avec **Snakemake**
  - Reproductible
  - Parallélisable
- L’association avec une **machine virtuelle** rend le programme portable et robuste