

# annotate.Snakefile

Un outil **répétable**, **réutilisable** et **réplicable**  
pour l'annotation de séquences synthétiques

*Cette présentation est sponsorisée par la*

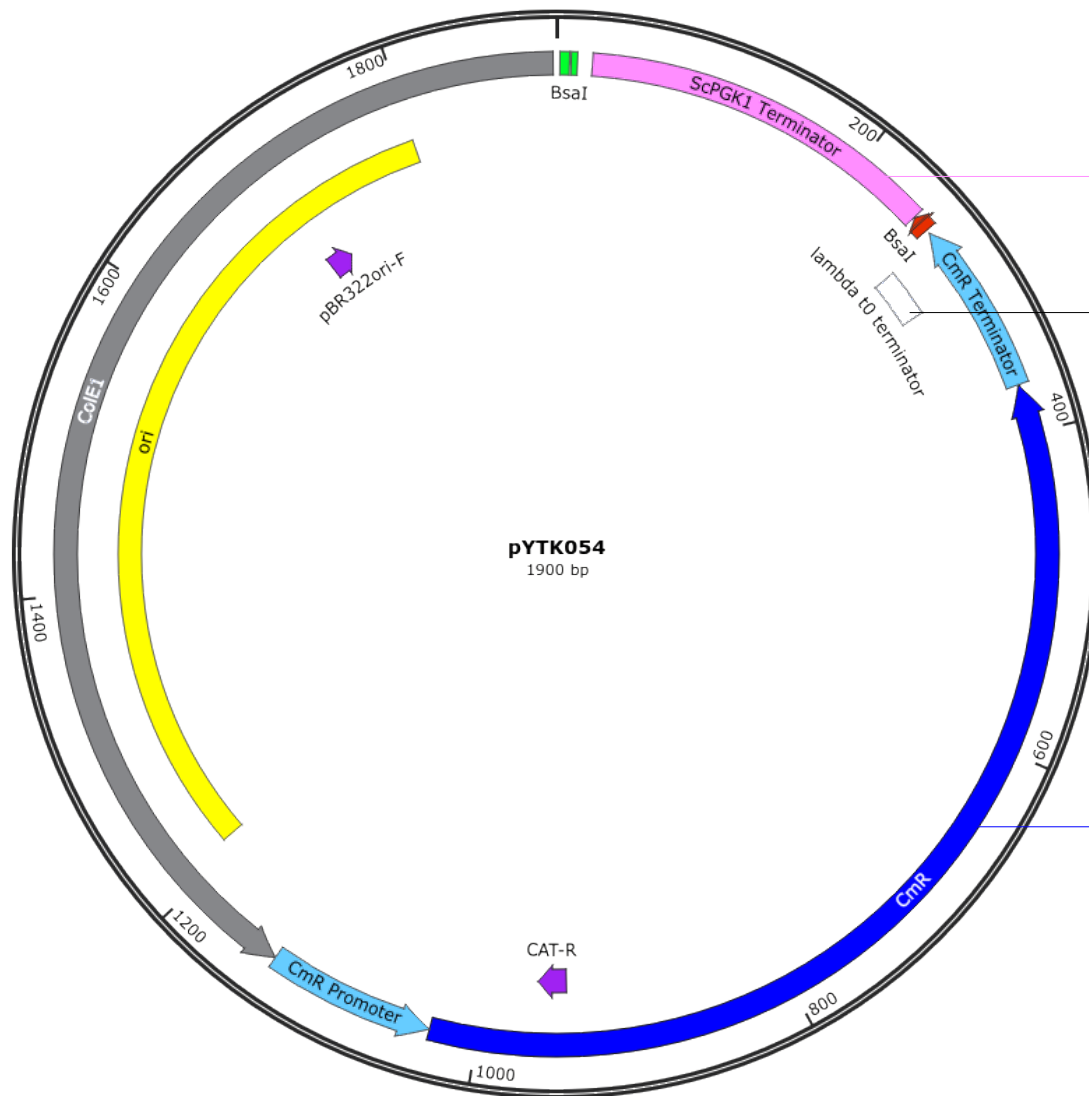
***Commission générale  
de terminologie et de néologie***

*de l'Académie Française*



Académie française

# Contexte – Séquences synthétiques



/label = ScPGK1 Terminator

/label = lambda t0 terminator

/note = minimal transcription terminator from phage lambda (Scholtissek and Grosse, 1987)

/label = CmR

/note = confers resistance to chloramphenicol

/db\_xref = UniProtKB/Swiss-Prot: P62577

/db\_xref = GO: 0008811

/db\_xref = GO: 0016740

/db\_xref = GO: 0016746

/db\_xref = GO: 0046677

/db\_xref = PFAM: PF00302

## Contexte – Distribution

[Kits](#) / [MoClo Guide](#) / [Densmore CIDAR MoClo Parts Kit](#)

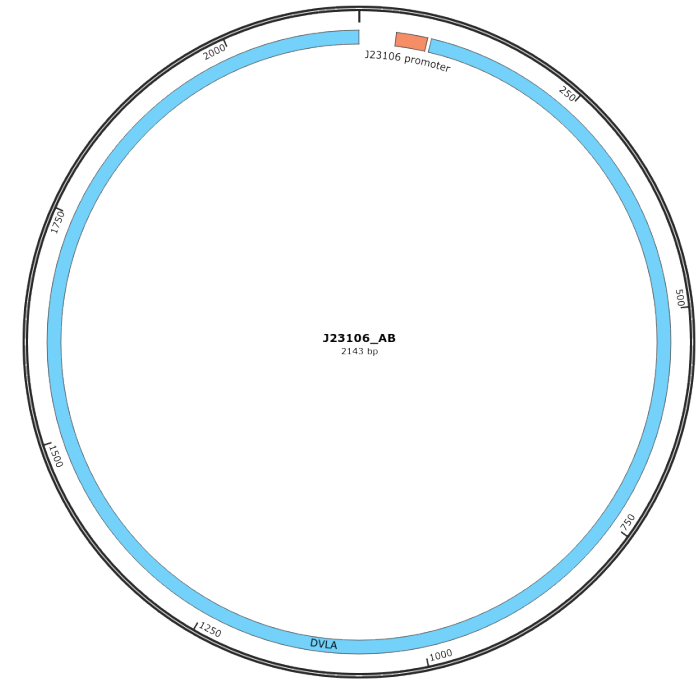
## CIDAR MoClo Parts Kit

(Kit #1000000059 )

Depositing Labs: Douglas Densmore

The Cross-disciplinary Integration of Design Automation Research lab (CIDAR) MoClo Library is collection of modular DNA parts and enhanced MoClo protocols to enable rapid one-pot, multipart assembly, combinatorial design and expression tuning in *E. coli*.

This kit will be sent as bacterial glycerol stocks in 96-well plate format.



### Detailed Information

## Protocols & Resources

## Kit Documentation

Download a ZIP file containing the plasmid sequences in GenBank format:



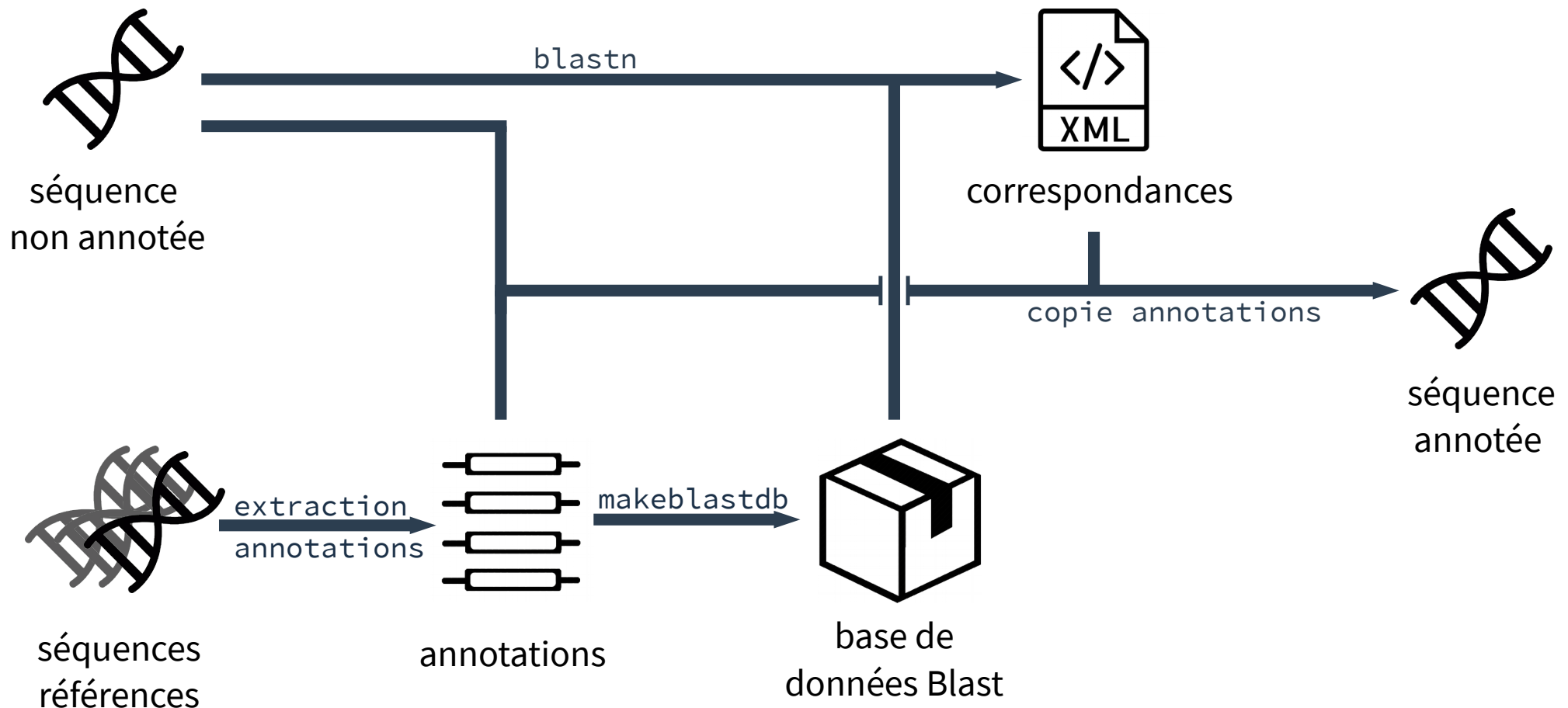
Genbank Files (228.3 KB)

Documents describing this kit:

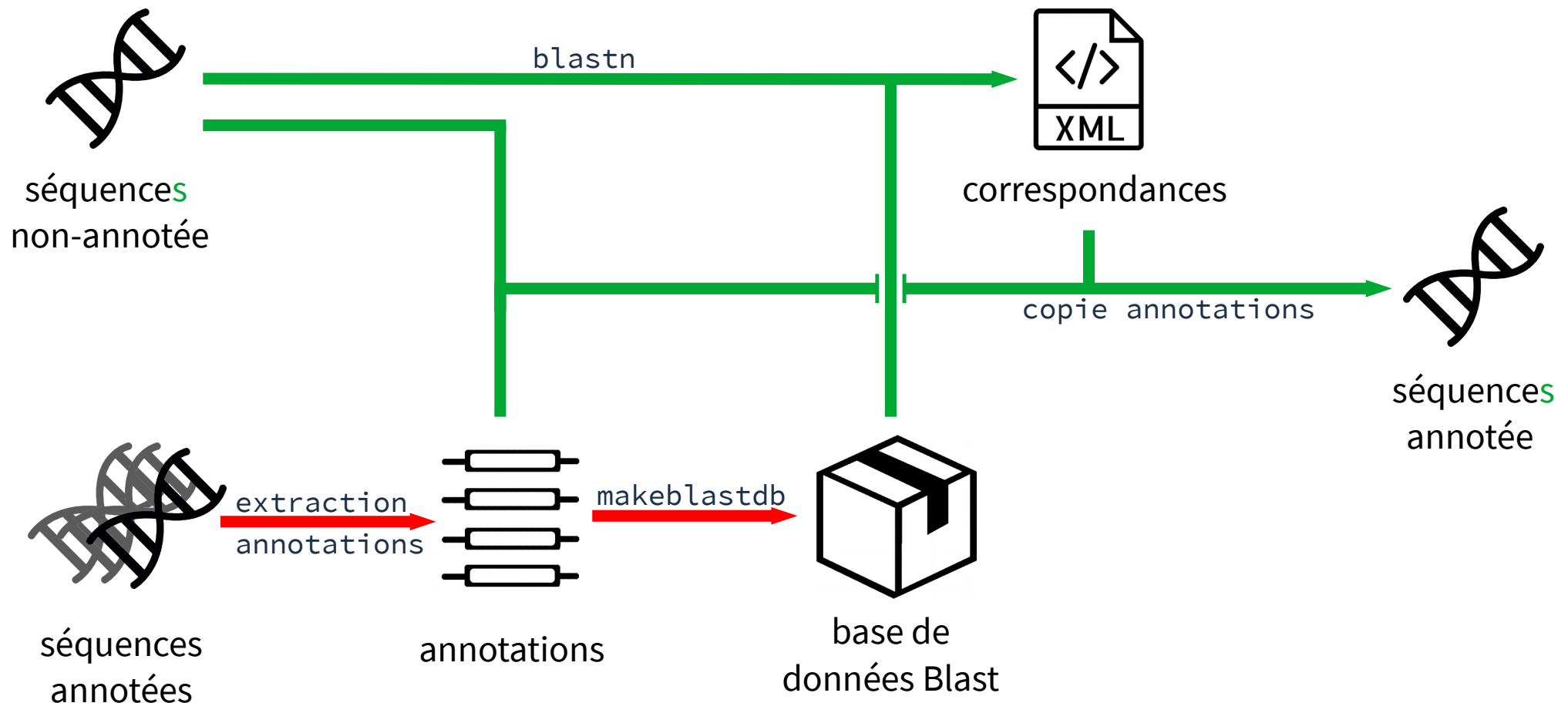


[CIDAR MoClo summary \(129.1 KB\)](#)

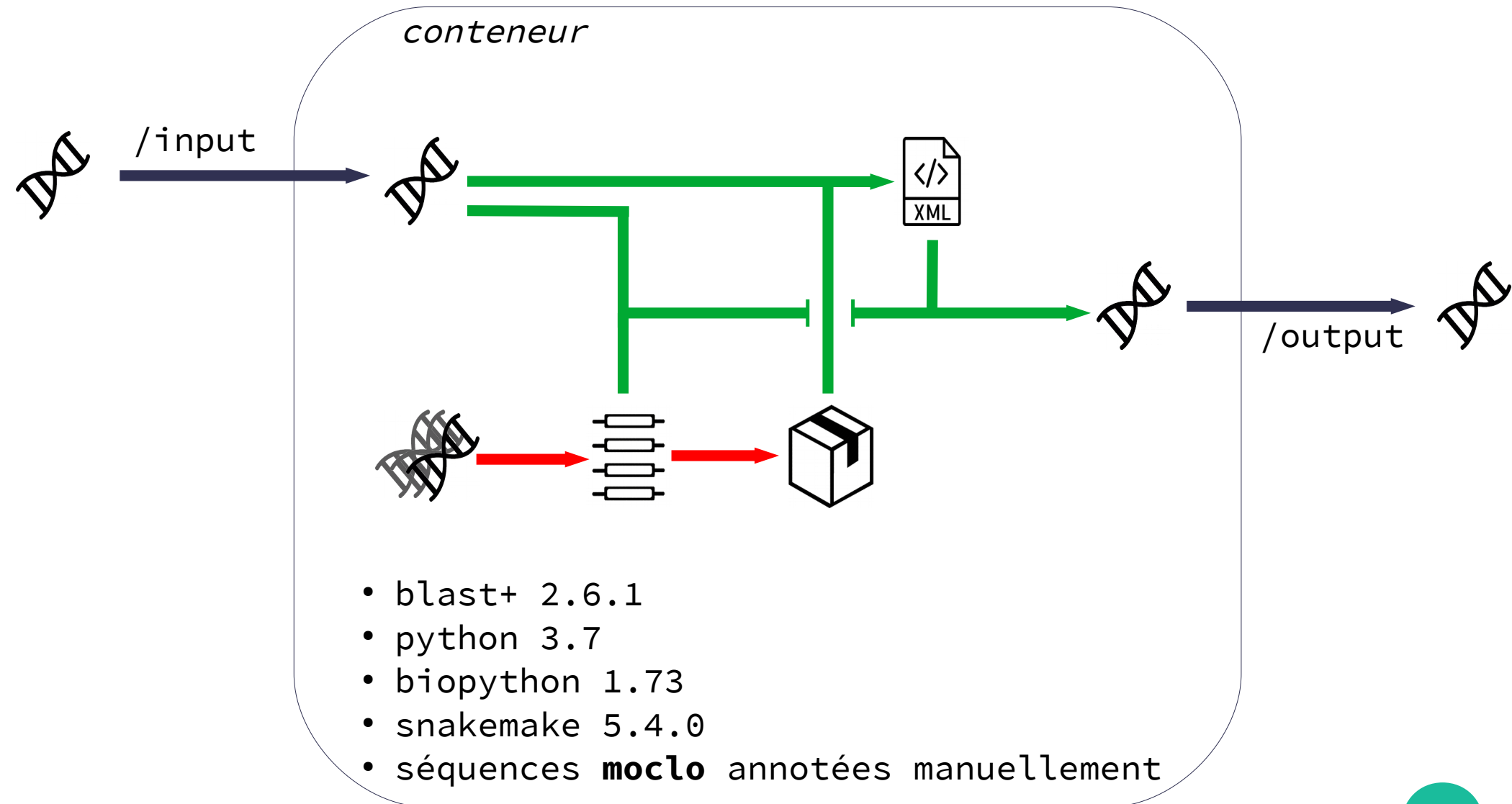
# Flux de travaux\* - Organisation



# Flux de travaux - Parallélisation



# Conteneurisation avec Docker



**Démonstration !**



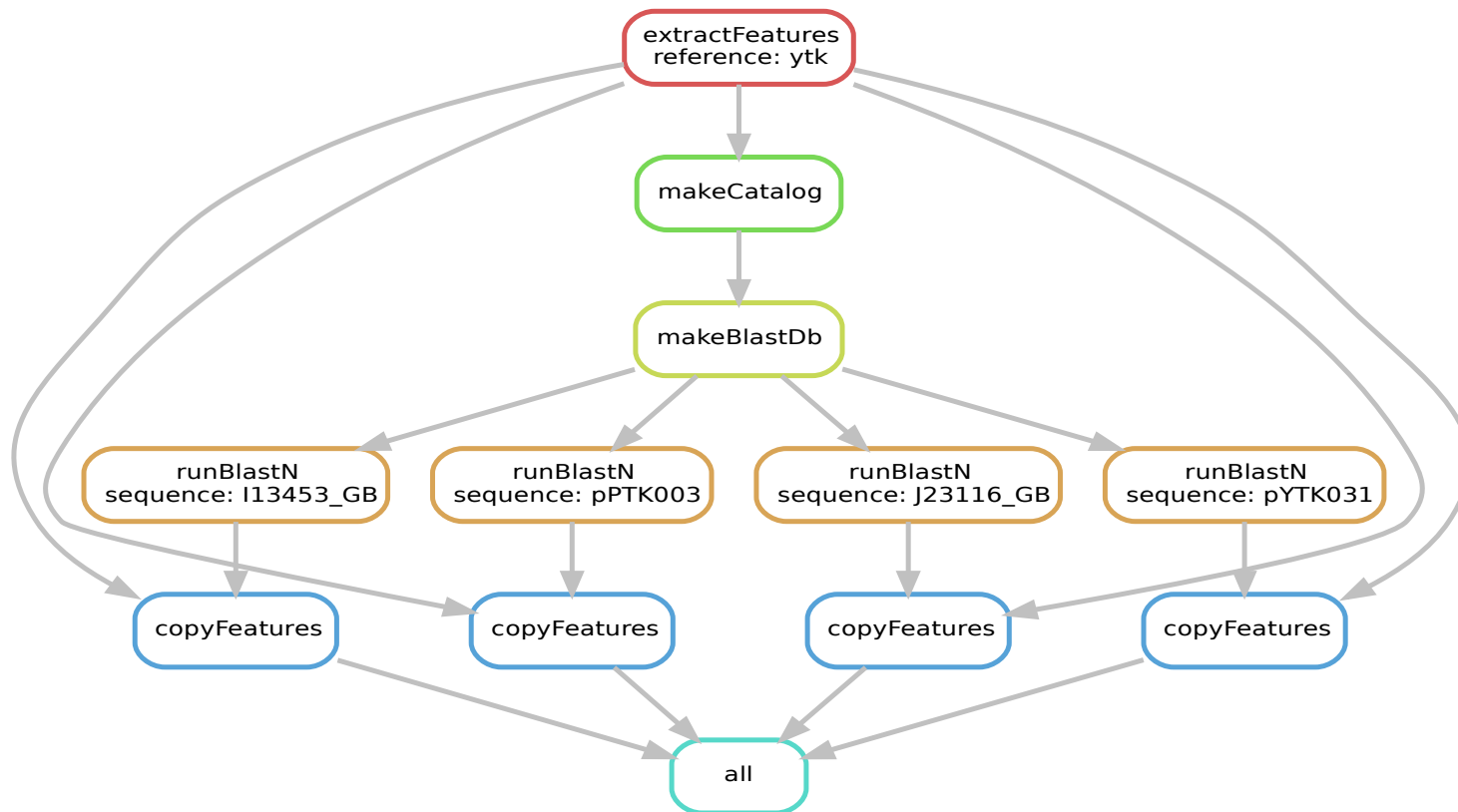
# Snakemake - 1

- Gestionnaire de flux opérationnels à base de règles à la façon d'un *Makefile*

```
rule makeCatalog:
    """Build a FASTA catalog of sequences from a directory of GenBank files.
    """
    input:
        "features/{reference}"
    output:
        "catalog/{reference}.fa"
    run:
        files = glob.iglob(os.path.join(input[0], "*.gb"))
        features = map(lambda f: Bio.SeqIO.read(f, 'gb'), files)
        with open(output[0], 'w') as f:
            Bio.SeqIO.write(features, f, 'fasta')
```

# Snakemake - 2

- Génération du DAG de manière à minimiser le nombre de nœuds (étapes du flux)



# Snakemake - 3

- **Avantages :**

- syntaxe simple et proche du Python
- utilisation de Python (et de ses bibliothèques) et/ou de commandes de la coquille\* du système

- **Limitations :**

- travailler avec un nombre variable de fichiers (!)
- établissement des règles (partant de la fin) qui peut être contre-intuitif

# Docker - 1

- **Moteur de conteneurisation :**
  - génère une image contenant l'intégralité des fichiers nécessaires à l'exécution d'un programme
  - lance cette image sur n'importe quelle machine
  - la compilation des images se fait à l'aide d'un *Dockerfile* qui permet de faire tourner des commandes d'installation à l'intérieur du conteneur

# Docker - 2

- **Avantages :**

- supprime le problème de l'installation des dépendances
- fige l'état d'un système, permettant la répétabilité/reproductibilité/répliquabilité
- plus léger qu'une machine virtuelle

- **Limitations:**

- les images s'alourdissent très rapidement
- l'isolement rend plus difficile le partage de ressources entre les conteneurs
- gestion plus complexe que certaines alternatives

**Cours accéléré:  
générer une image avec  
gestion des versions**

# Docker + GitHub = ♥

- **Objectif :**

- développer du code pour l'intégrer dans une image
- (optionnel) tester ce code pour éviter l'introduction de bogues
- créer une image contenant ce code et la publier

- **Intérêt :**

- permettre l'utilisation de la dernière version du code sans avoir à compiler l'image soi-même
- distribuer votre projet simplement (pour l'utilisateur)

# Docker + GitHub = ♥

- Mettre en place un dépôt public sur GitHub

The screenshot shows the GitHub interface for a repository named 'althonos / annotate.Snakefile'. At the top, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The repository description is 'A Snakemake pipeline to copy annotations between GenBank files', with an 'Edit' button. Below the description is a bar showing repository statistics: '12 commits', '1 branch', '0 releases', '1 contributor', and 'MIT' license. A secondary bar contains buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table lists the following files and their commit details:

File	Commit Message	Time Ago
<a href="#">.gitignore</a>	Add `Snakefile` with first steps of the pipeline	24 days ago
<a href="#">COPYING</a>	Initial commit	24 days ago
<a href="#">Dockerfile</a>	Build docker image on `debian` instead of `alpine`	4 days ago
<a href="#">README.md</a>	Add `README.md` file with usage information	2 minutes ago
<a href="#">Snakefile</a>	Build docker image on `debian` instead of `alpine`	4 days ago

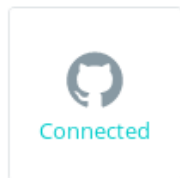


# Docker + GitHub = ♥

## • Configurer un nouveau dépôt sur le Docker Hub

Build Settings *(optional)*

Autobuild triggers a new build with every git push to your source code repository [Learn More](#)



← sélectionner la forge



althonos



annotate.Snakefile



← sélectionner le dépôt

▼ Click here to customize the build settings

BUILD RULES +

The build rules below specify how to build your source into Docker images.

Source Type	Source	Docker Tag	Dockerfile location	Build Caching
Branch ▼	master	latest	Dockerfile	<input checked="" type="checkbox"/>

► View example build rules

← ajouter des règles  
une image *latest* pour  
la dernière soumission\*  
de la branche *master*

# Docker + GitHub = ♥

- L'image est recompilée à chaque fois que la branche *master* est mise à jour dans le dépôt

Recent Builds			
✓ <a href="#">Build in 'master' (0cd89d1f)</a>	🔧 latest	🕒 22 minutes ago	
✓ <a href="#">Build in 'master' (a271f31f)</a>	🔧 latest	🕒 4 days ago	
✗ <a href="#">Build in 'master' (f794d3d2)</a>	🔧 latest	🕒 6 days ago	🔄
✗ <a href="#">Build in 'master' (af850087)</a>	🔧 latest	🕒 6 days ago	🔄
✓ <a href="#">Github Ping</a>		🕒 6 days ago	

# Docker + GitHub = ♥

- On a accès au fichier journal\* créé par la commande *docker build* en cas d'erreur

Build logs Dockerfile Readme

```
Building in Docker Cloud's infrastructure...
Cloning into '.'...
Warning: Permanently added the RSA host key for IP address '192.30.253.113' to the list of known hosts.
Reset branch 'master'
Your branch is up-to-date with 'origin/master'.
KernelVersion: 4.4.0-1060-aws
Components: [{u'Version': u'18.03.1-ee-3', u'Name': u'Engine', u'Details': {u'KernelVersion': u'4.4.0-1060-aws', u'Os': u'linux', u'BuildTime': u'2018-08-30T18:42:30.000000000+00:00', u'Platform': {u'Name': u''}, u'GoVersion': u'go1.10.2', u'GitCommit': u'b9a5c95', u'Os': u'linux', u'Architecture': u'amd64'}}]
Arch: amd64
BuildTime: 2018-08-30T18:42:30.000000000+00:00
ApiVersion: 1.37
Platform: {u'Name': u''}
Version: 18.03.1-ee-3
MinAPIVersion: 1.12
GitCommit: b9a5c95
Os: linux
GoVersion: go1.10.2
Starting build of index.docker.io/althonos/annotate:latest...
Step 1/6 : FROM python:alpine
---> 1a8edcb29ce4
Step 2/6 : ADD Snakefile Snakefile
```

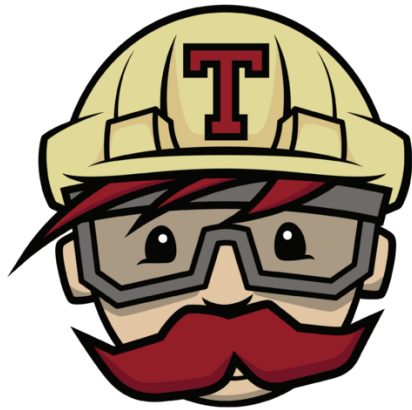
■■■

```
error: command 'gcc' failed with exit status 1
```

```
-----
```

```
Command "/usr/local/bin/python -u -c "import setuptools, tokenize;__file__='/tmp/pip-install-7x2k28ym/datrie/setup.py';f=getattr(tokenize, 'open', open).open(__file__);code=f.read().replace(r'\n', r'\n\n');exec(code)" failed with error code 1
Removing intermediate container e10736f5ee89
```

**Docker + CI + GitHub = ♥**



Travis CI



circleci



GitLab CI

**FAIRE UN WORKFLOW DANS  
UN SCRIPT SHELL**



**FAIRE UN WORKFLOW AVEC  
SNAKEMAKE**



**FAIRE UN WORKFLOW AVEC  
SNAKEMAKE DANS UNE  
IMAGE DOCKER**



**FAIRE UN WORKFLOW AVEC  
SNAKEMAKE DANS UNE  
IMAGE DOCKER GÉNÉRÉE  
AUTOMATIQUEMENT**



**FAIRE UN WORKFLOW AVEC  
SNAKEMAKE DANS UNE IMAGE  
DOCKER GÉNÉRÉE  
AUTOMATIQUEMENT PAR UN  
SERVICE D'INTÉGRATION  
CONTINUE**



**FAIRE UN SCRIPT SHELL POUR  
LANCER UN WORKFLOW AVEC  
SNAKEMAKE DANS UNE IMAGE  
DOCKER GÉNÉRÉE  
AUTOMATIQUEMENT PAR UN  
SERVICE D'INTÉGRATION  
CONTINUE**



**Merci pour votre  
attention !**