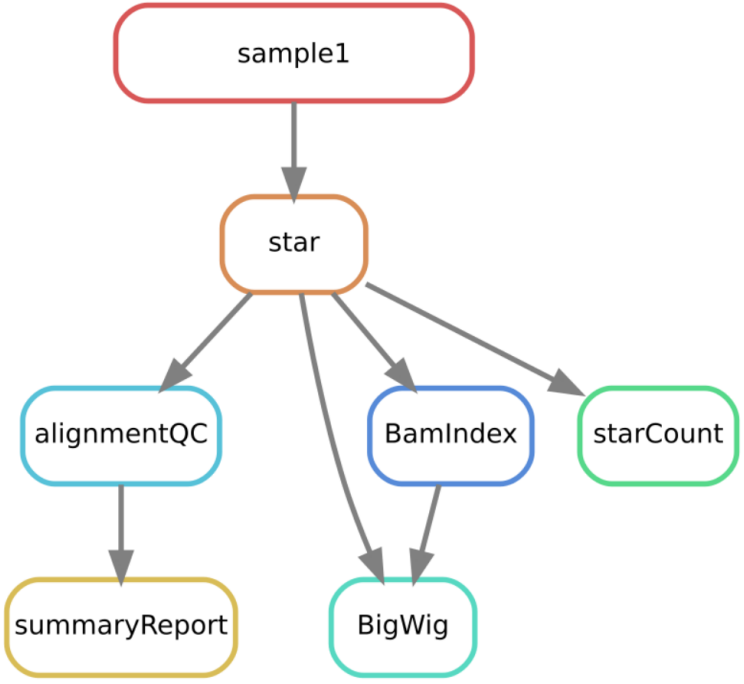
Reproductible analysis using workflow managers

Magali Hennion – october 2024

Why using a workflow manager?

One analysis : multiple steps, multiple samples

Simple example, one sample



snakemake

nextflow

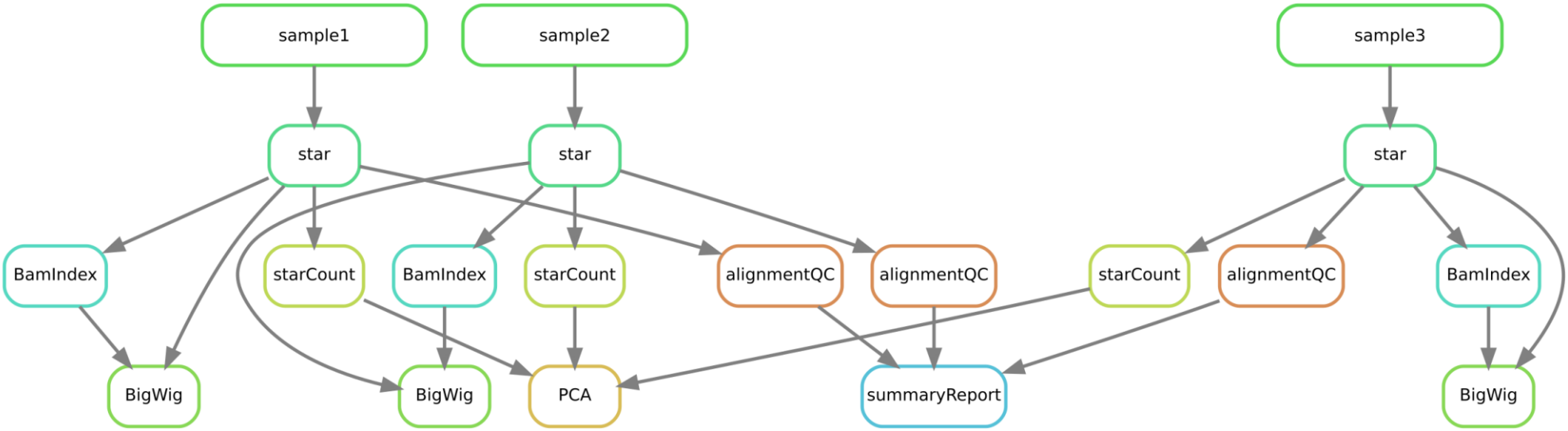
Galaxy

Some tasks can run in parallel

Some tasks have to wait for one or several others to complete

Why using a workflow manager?

Simple example, more samples (3)



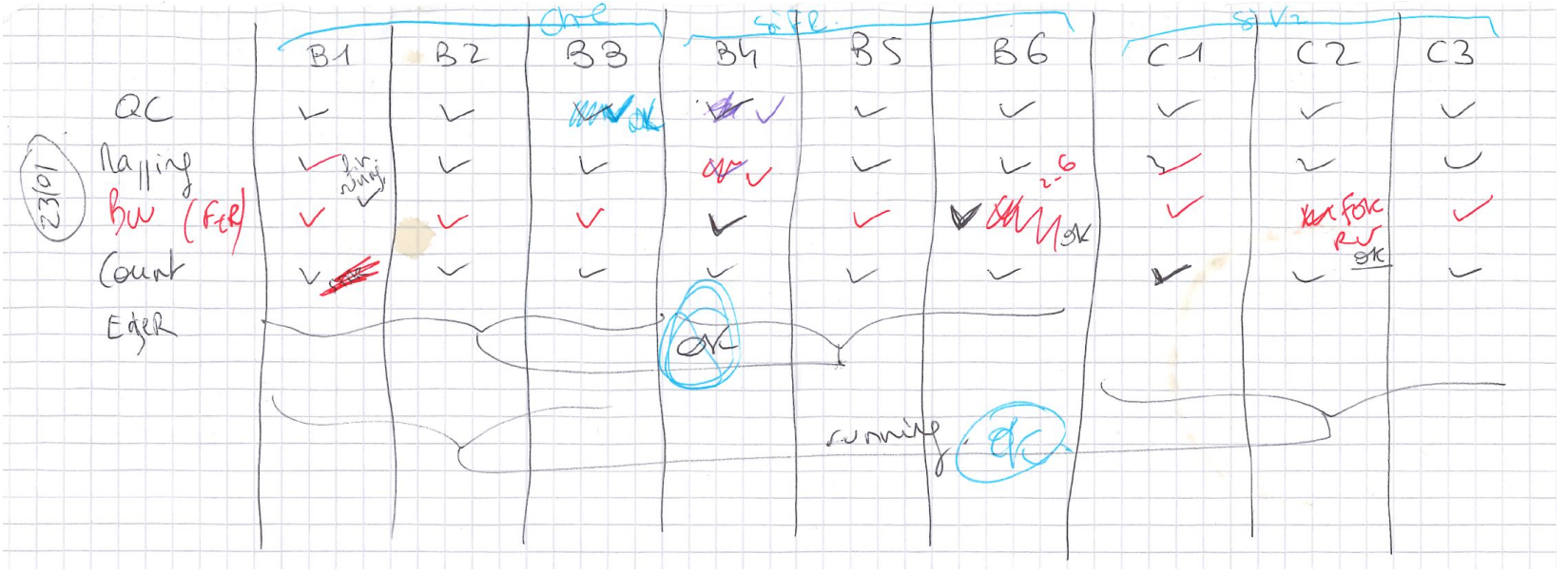
Some tasks are done for each sample

Some combine the information from several samples

→ Usually more samples, without workflow manager you may get lost...

Why using a workflow manager?

And/or end up with something like that...

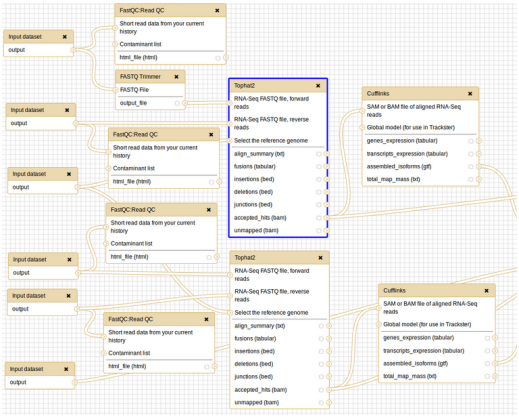


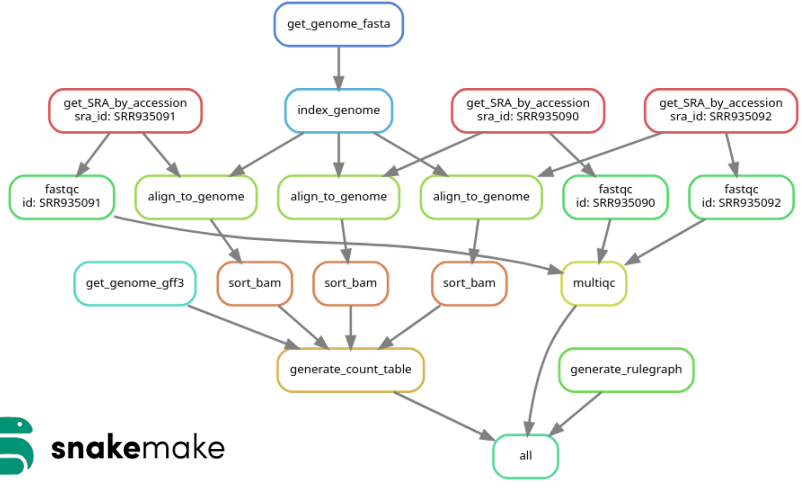
The reproducible journey

How to make my analysis reproducible in a few years / by someone else ?

Worflow managers

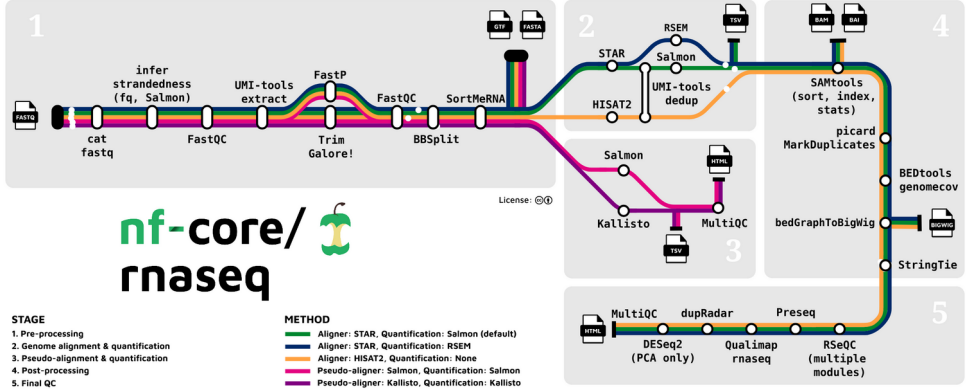
Galaxy





snakemake

nextflow



nf-core/rnaseq

STAGE

1. Pre-processing
2. Genome alignment & quantification
3. Pseudo-alignment & quantification
4. Post-processing
5. Final QC

METHOD

- Aligner: STAR, Quantification: Salmon (default)
- Aligner: STAR, Quantification: RSEM
- Aligner: HISAT2, Quantification: None
- Pseudo-aligner: Salmon, Quantification: Salmon
- Pseudo-aligner: Kallisto, Quantification: Kallisto

Pros

Reproducible and portable analysis, controled environment

Task parallelisation -> efficient and scalable (HPCs)

Resuming of failed runs or steps

Modular → library of reusable blocks (any langage), very efficient to benchmark methods and parameters

Easy configuration for bioinformaticians and biologists

Cons

Learning effort ...

Controled environments

Most of the tools in your workflow need some requirements.

Some need lot of requirements, ie Python or R libraries in specific version.

Often, those dependencies are not compatible with what you already have on our system or between them.

→ need for independant environments

- Conda environments
- Containers : Docker, Apptainer (ex Singularity)
- Modules

Conda

Automatically get packages from repositories called « channels »

Warning : Anaconda curated channels are not free !
« Utilizing Miniconda to pull package updates from the Anaconda Public Repository without a commercial licence […] is considered a violation of the Terms of Service »
Teaching is the only exception, doing public research is not !

→ Don’t use *defaults* channel which includes *main/default* and *r*.
→ Don’t use Anaconda or Miniconda softwares but Mamba or Miniforge

Use open source and free channels such as bioconda or conda-forge

Conda / Mamba

Create a new environment

```
mamba create --no-default-packages -n myPythonEnv python=3.12.7
mamba create -n jupyterEnv -c conda-forge jupyterlab
```

yaml file to define your environment
-> define channels (order matters!) and packages

```
mamba env create -f nanopore_env.yaml
```

Work in an environment

```
$ conda activate nanopore
$ samtools --help
```

nanopore_env.yaml

```
name: nanopore
channels:
  - epi2melabs
  - bioconda
  - conda-forge
  - nodefaults
dependencies:
  - samtools
  - htslib=1.14
  - modbam2bed
  - epi2melabs
```

Conda limits

- Reproducible ?

  - Environments built few months/years appart from the same recipe are differents

  - Some can’t be rebuilt (packages not on the repos anymore, changes in librairies)

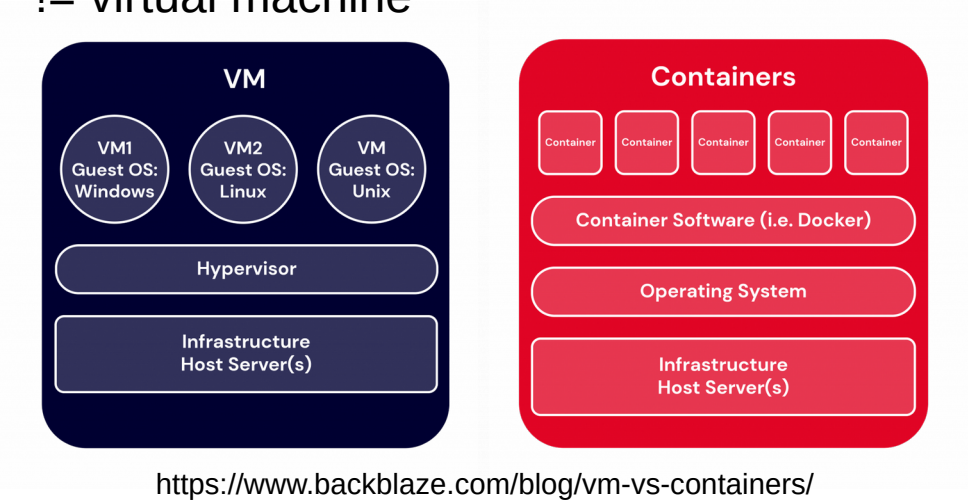
  → Use of containers solves this issue (but need to store a big file)

- Huge number of files in Conda envs (typically 10-20K...)

Containers

Container image ~ virtual computer disk with all of the necessary software, libraries and configuration to run one or more applications

!= virtual machine



https://www.backblaze.com/blog/vm-vs-containers/

A container made in one OS works only on that OS (e.g. Windows or Linux)

Containers

You can use existing containers

Docker Hub
https://hub.docker.com/

Singularity Hub (2016-2021)

https://oras.land

Sylabs
https://sylabs.io/

From Docker
$ apptainer pull tensorflow.sif docker://tensorflow/tensorflow:latest
From Shub
$ apptainer pull apptainer-images.sif shub://vsoch/apptainer-images
From supporting OCI (Open Containers Initiative) registry
$ apptainer pull image.sif oras://url/to/gitlab/container/registry:latest

Build a container

Need to be super user !

Singularity_4.0.3

Apptainer definition file

```
BootStrap: docker
From: ubuntu

%post
# update apt
apt update
# git install
apt install -y git
# clone git repo and install
git clone --recursive
https://github.com/MBoemo/DNAscent.git
cd DNAscent
git checkout 4.0.3
make

%runscript
# Define the run command
        exec echo "$@"
```

From a base image

1. Pull an existing image as a base
2. Install packages
3. Make the image

```
$ sudo apptainer build dnascent_v4.0.3.sif Singularity_4.0.3
```

Build a container

Need to be super user !

Apptainer_def

```
Bootstrap: docker
From: mambaorg/micromamba:1.5.9

%files
# Copy files for building
env.yaml /setupfile/env.yaml

%post
# Create conda env
cd /setupfile
micromamba env create -f env.yaml -n env_name
micromamba clean --yes --tarballs

# Export FAIR Files
micromamba list -n  env_name > env_installed_packages.txt
micromamba env export --no-build -n env_name > env_export.yaml

%environment
# set Conda env bin in the PATH
        export PATH=/opt/conda/envs/env_name/bin:$PATH

%runscript
# Define the run command
        echo "Container was created $NOW"
        exec echo "$@"
```

Containing a Conda environment

1. Pull an existing image as a base
2. Create Conda env from a yaml file defining the environment
3. Export path to conda bin
4. Define what does the « run » command

```
$ sudo apptainer build image.sif Apptainer_def
```

Build a container

- From a Docker image

```
$ sudo apptainer build alpine.sif docker://alpine
```

- Online on Sylabs cloud (https://cloud.sylabs.io/)

Use a container

pull : get the image

shell : open a shell inside the container

run : run the runscript contained in the container

exec : use tools contained in the container

```
$ apptainer pull docker://ghcr.io/apptainer/lolcow
```

```
$ apptainer shell lolcow_latest.sif

Apptainer> ls /bin
```

```
$ apptainer run lolcow_latest.sif
 _______________________________
< Fri Oct 11 11:40:34 CEST 2024 >
 -------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
$ apptainer exec lolcow_latest.sif cowsay moo
 _____
< moo >
 -----
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

https://apptainer.org/docs/user/main/quick_start.html

Use a container

- Direct command line

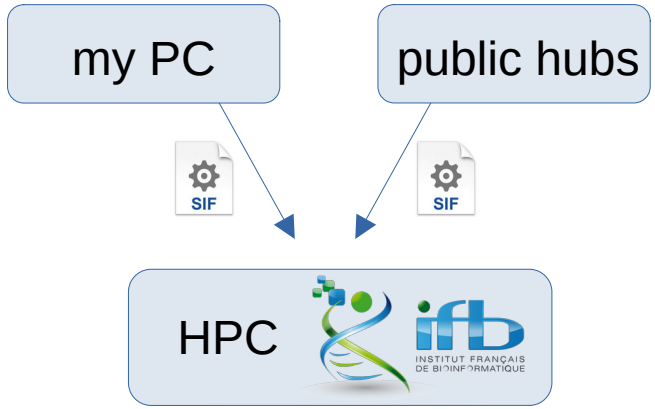
```
$ apptainer shell image.sif

$ apptainer run image.sif

$ apptainer exec image.sif hostname
```

- In sbatch script

- In workflows



nextflow

snakemake

In a workflow

One container per tool

A single container for all tools

Go to https://scrumblr.ethibox.fr/worflow and add pros and cons of both approaches

On the cluster : Modules

Tools ready to
use for everyone !

```
module load fastqc/0.11.9
```

ENVIRONMENT MODULES

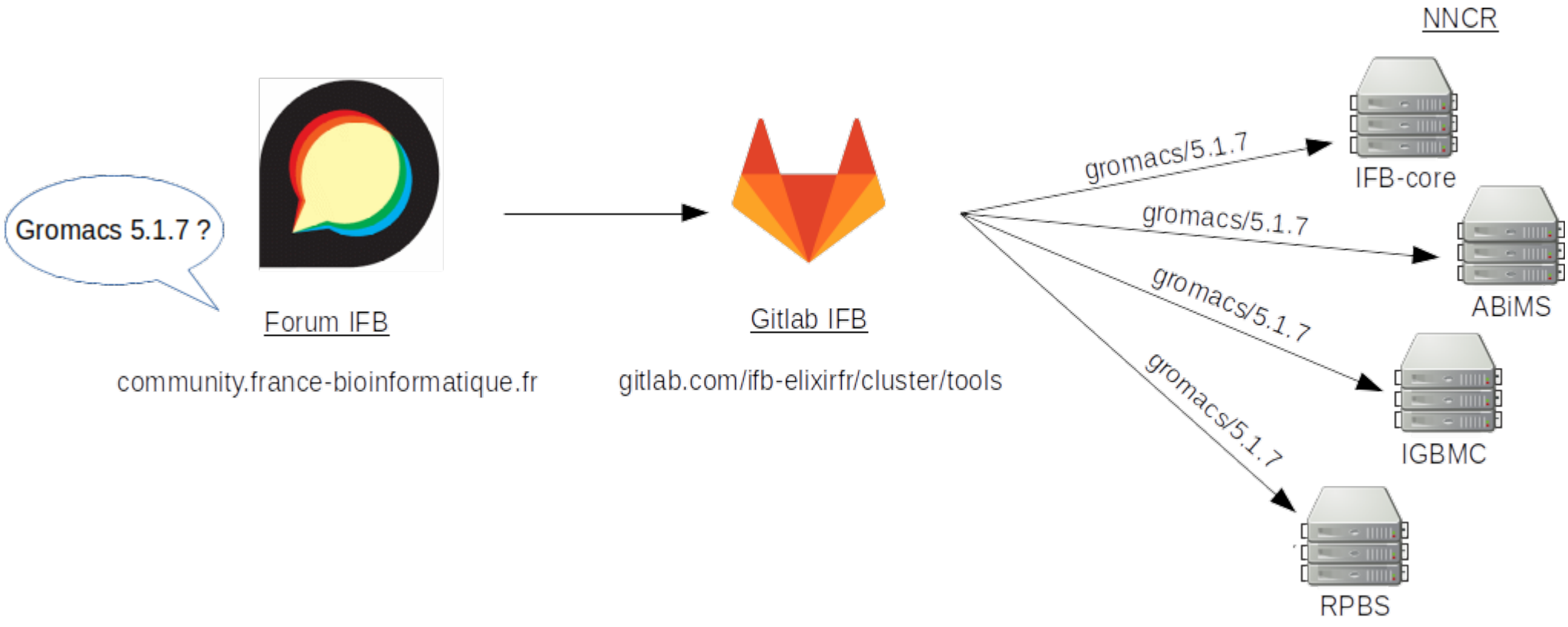
BIOCONDA®

by default

Singularity

if a licence must be accepted
if not in Bioconda and hard to integrate or urgent
if a Docker container exists

I need a new module...



Acknowledgement

Thanks to Laurent Jourdren who initiated the container presentation !

Let’s connect to IFB core cluster !