Nextflow and nf-core

First steps

Alexis Praga[1]

FOSDEM 2025

[1]M.D, Ph.D, Besançon Hospital, France - alexis@praga.dev

Introduction

Do you want to make pipelines

without caring about parallelization, dependencies, intermediate file names, data structures, handling exceptions, resuming executions, etc. ?

In short

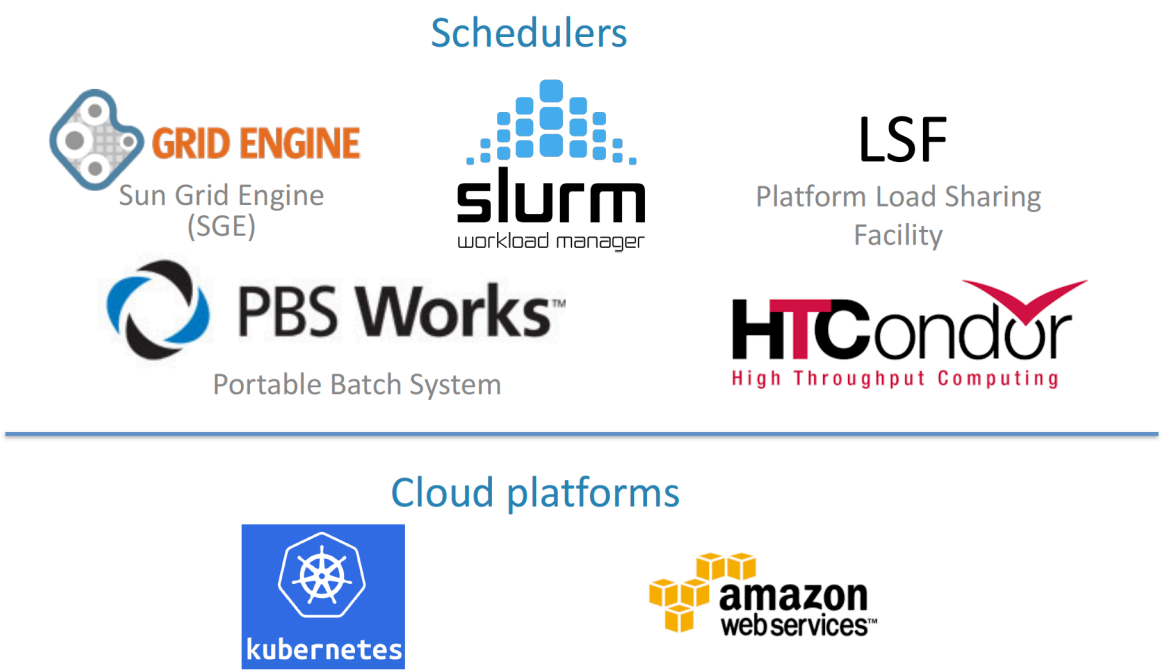
Nextflow : a DSL to write workflow

- portable (run “everywhere”)
- reproductible
- easy and scalable parallelization
- easy and scalable deployment

Useful for chaining CLI apps/scripts

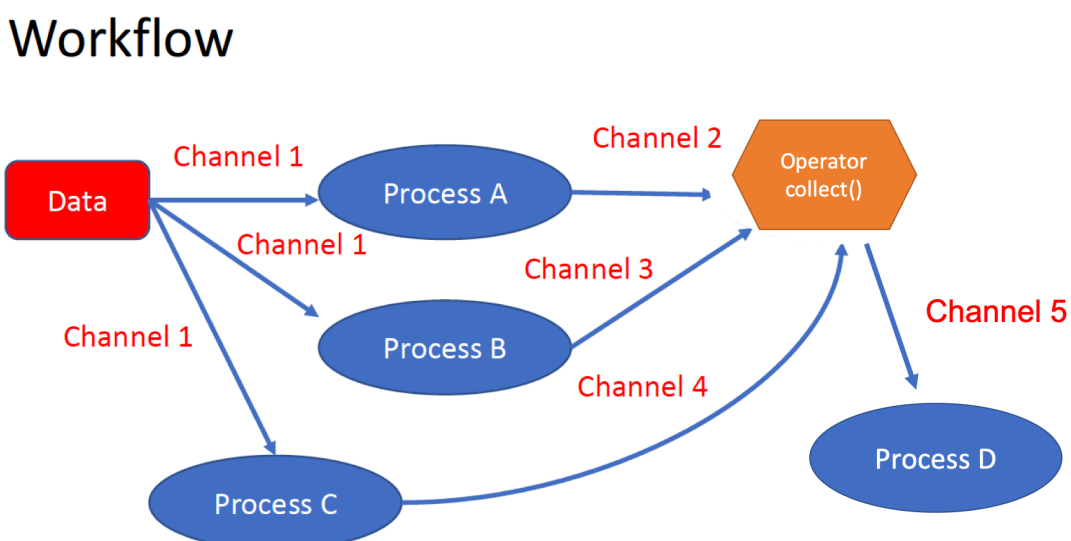
Nextflow: portable

- Run “everywhere” without changing the code



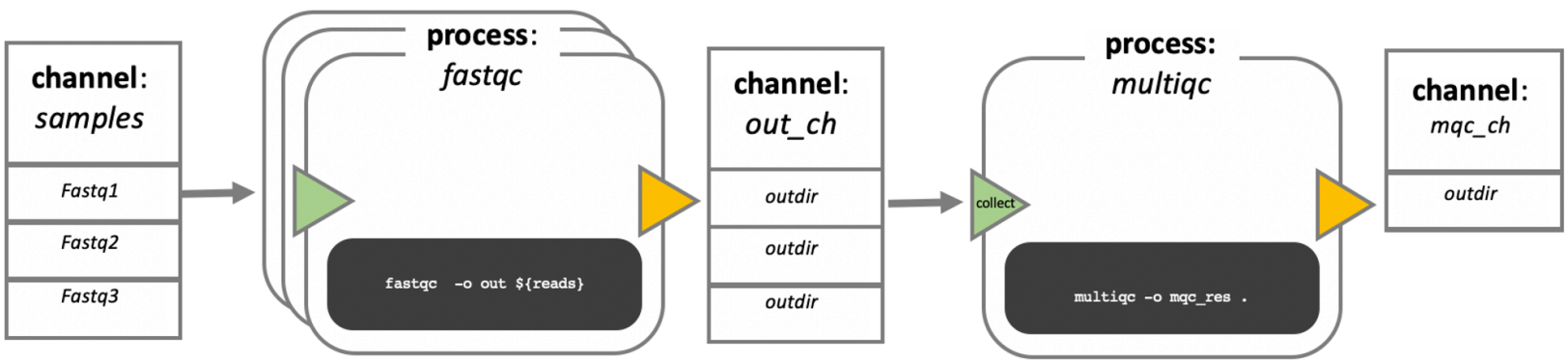
Nextflow: scalable

- Parallelize process when possible (Directed Acyclic Graph)



Nextflow: scalable

- Parallelize process when possible (Directed Acyclic Graph)



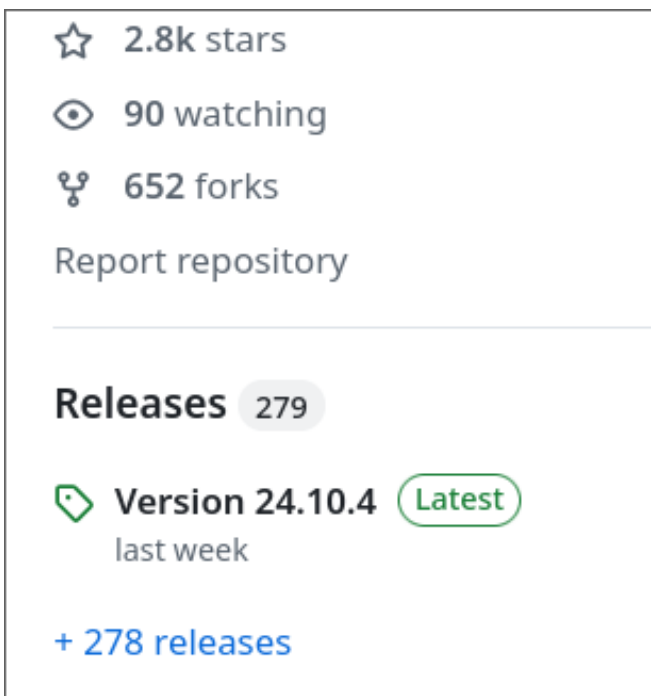
Nextflow: others pros

- **resumable** from last failed process
- fast prototyping
- “pre-made” pipelines (`nf-core`)

Nextflow: is it good ?

Yes: more than 3 000 citations

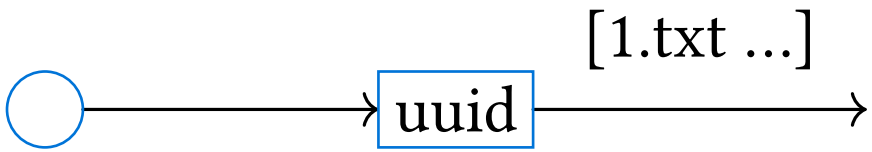
Nextflow enables reproducible computational workflows
P Di Tommaso, M Chatzou, EW Floden, PP Barja… - Nature …, 2017 - nature.com
… We present a qualitative comparison between **Nextflow** and other similar tools in Table 1 (ref. …
Another key specification of **Nextflow** is its integration with software repositories (including …
☆ Save Cite Cited by 3074 Related articles All 7 versions



Tutorial: Generating random files

```
 process randomFiles {
    publishDir 'results', mode: 'copy'
    output:
    path("*.txt")

    script:
    """
    for i in {1..5}; do uuidgen > \$i.txt; done
    """
}
workflow {
  randomFiles()
}
```

```
$ nextflow run test1.nf

 N E X T F L O W   ~  version 24.08.0-edge

Launching test1.nf [amazing_crick] DSL2 - revision: f3d07993e5

executor >  local (1)
[e6/b16bcd] randomFiles | 1 of 1 ✔
```


[1.txt ...]
uuid

Temporary files

```
$ ls work/e6/b16bcdae36e772dc2b2211af21ecc9/
.command.sh    # your command
.command.err   # stderr
.command.out   # stdout
.command.log   # log
.exitcode      # exit code for the command
.command.begin # task metadata
.command.run   # nextflow shenaningan to run your command
1.txt    # our output files !
2.txt    # our output files !
...
```

With `publishDir`

```
$ ls results/
1.txt
2.txt
3.txt
4.txt
5.txt
```

Tutorial: Sorting a list of files

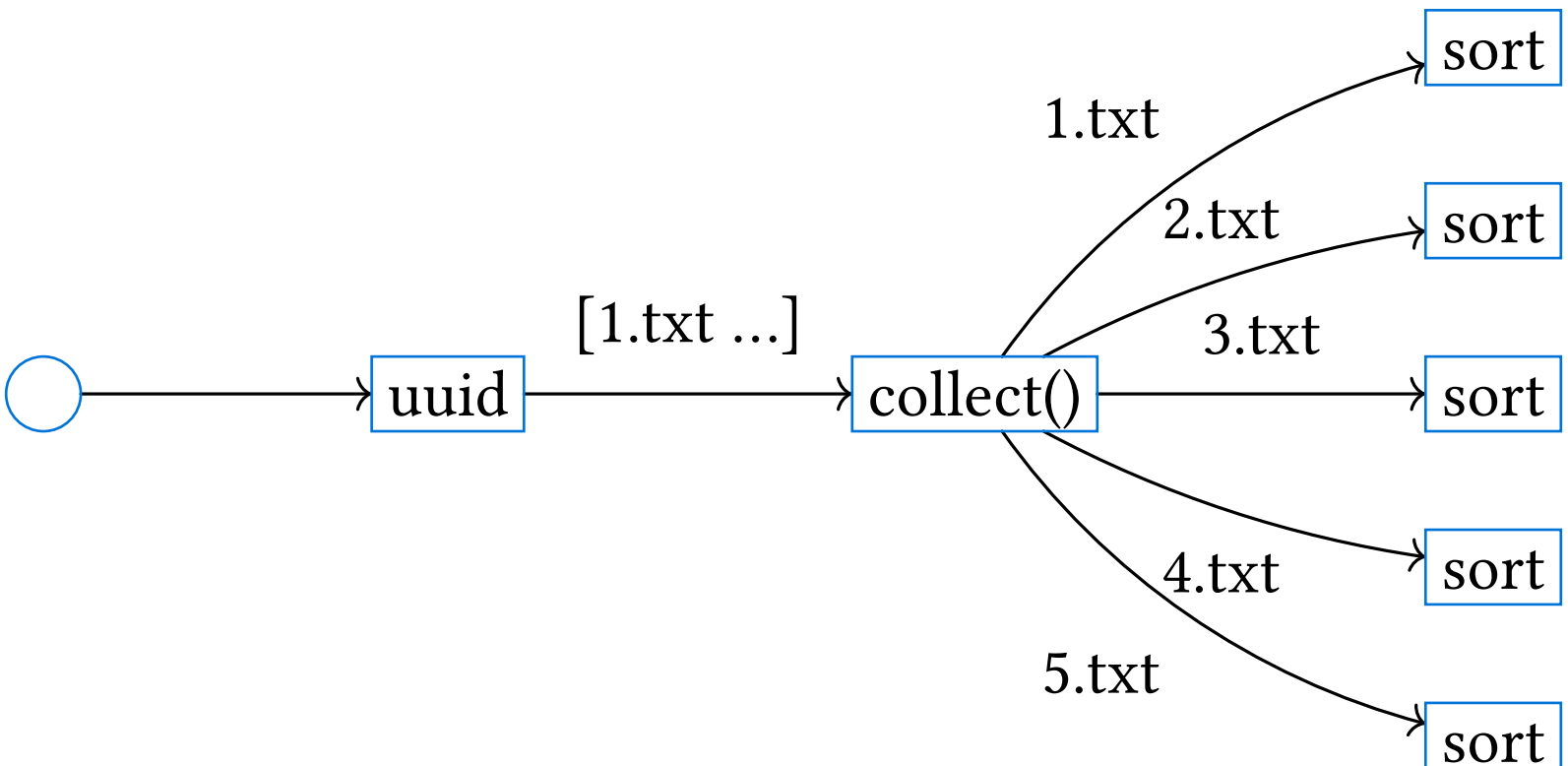
```
process sortFiles {
    input:
    path f

    output:
    path("${f}.sorted")

    script:
    """
    sort $f  > ${f}.sorted
    """
}
```

```
workflow {
  randomFiles().flatten() |
sortFiles
}
```

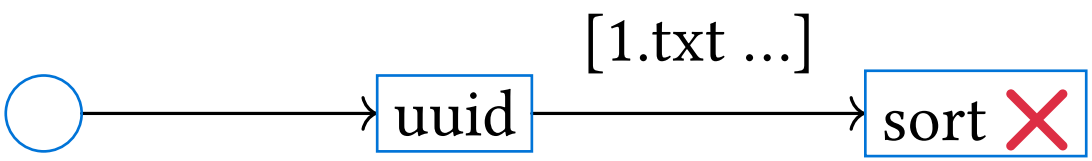
```
$ nextflow run test2.nf

executor >  local (6)
[71/17ff8b] randomFiles   | 1 of 1 ✔
[f3/002f95] sortFiles (5) | 5 of 5 ✔
```

```
$ nextflow run test2.nf -ansi-log false

[f7/0b0a49] Submitted process > randomFiles
[4a/3c6242] Submitted process > sortFiles (5)
[a0/fc20ef] Submitted process > sortFiles (1)
[e7/c3da13] Submitted process > sortFiles (4)
[c1/51fa80] Submitted process > sortFiles (2)
[18/c1afef] Submitted process > sortFiles (3)
```

`randomFiles().flatten() | sortFiles` ✓



`randomFiles() | sortFiles` ❌



"Thinking in nextflow" needs a bit of time ...

Run it on a cluster ?

Just add `nextflow.config` in project directory

```
process {
    executor = 'slurm'
    queue = 'smp'
}
```

Run it on a cluster ?

Runtime requirement

```
process {
    executor = 'slurm'
    queue = 'smp'

    withLabel:process_high { // if the process has label: 'high'
        cpus = 64
    }

    withName: sortFiles {
        memory = '50GB'
    }
 }
```

Nf-core

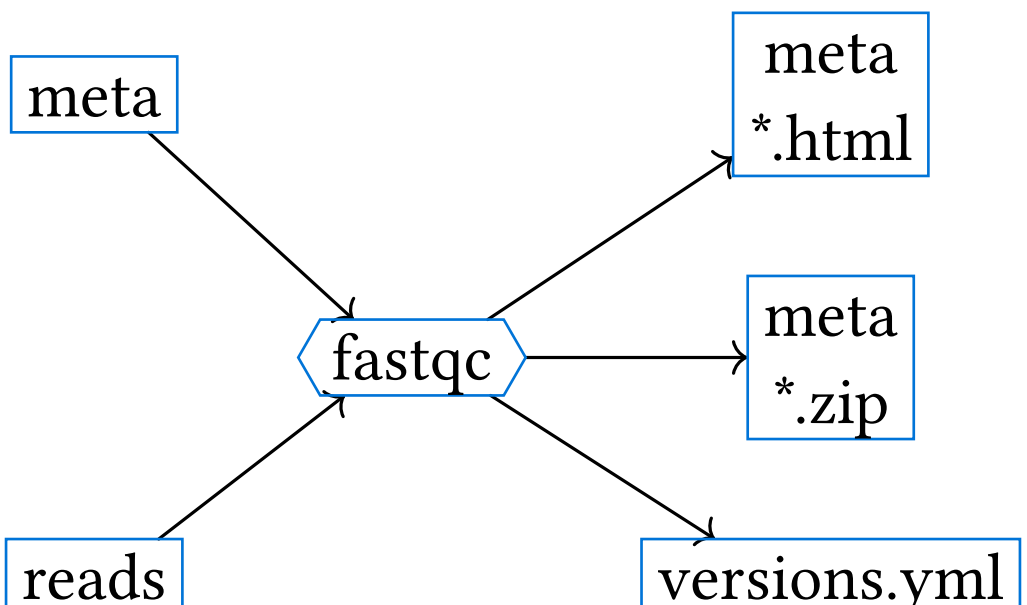
Don't reinvent the wheel

74 ready-to-use pipelines

- RNA-seq: `rnaseq` (971 ⭐)
- WGS germline/somatic: `sarek` (419 ⭐)
- single-cell RNAseq: `scrnaseq` (243 ⭐)
- methylation analysis: methylseq (156 ⭐)
- ancient DNA analysis: eager (156 ⭐)
- detection of gene-fusions (RNA-seq): rnafusion (150 ⭐)

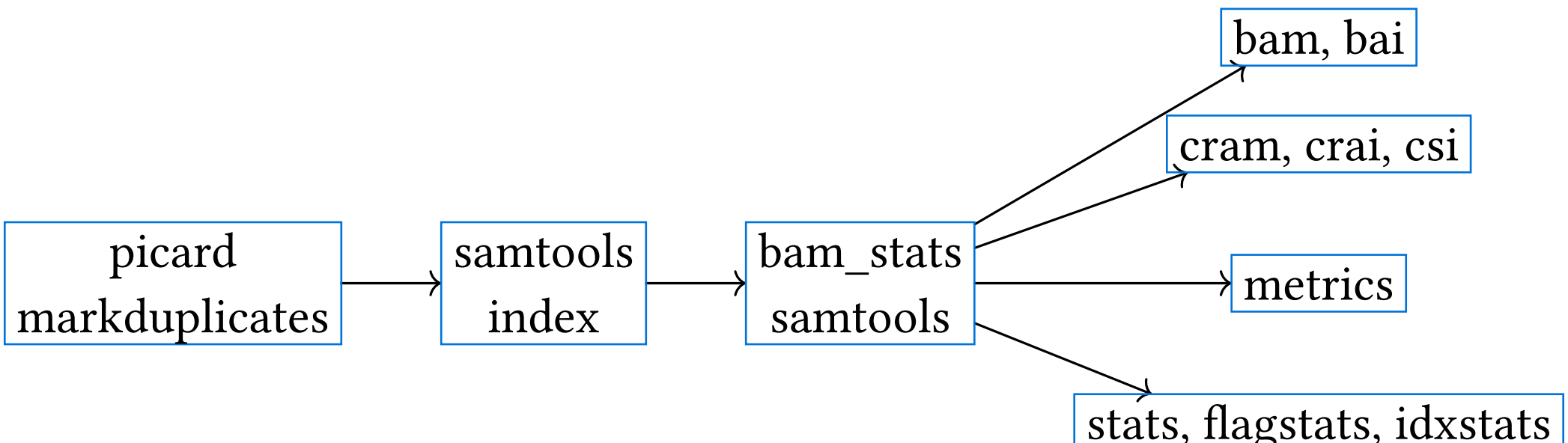
Ready-to-use components

- existing wrapper (`modules`) : `fastqc`



Ready-to-use components

- subworkflows: `bam_markduplicates_picard`



Dependencies

- Docker
- Singularity
- Apptainer
- Podman, Charliecloud and Shifter
- Conda (last resort !)
- Mamba

Automatically installed when running the pipeline

In practice

Create `samplesheet.csv`

```
patient,sample,lane,fastq_1,fastq_2
ID1,S1,L002,test_R1_001.fastq.gz,test_R2_001.fastq.gz
```

Run it like a 🥷

```
nextflow run nf-core/sarek \    # fetch pipeline
   -profile singularity \       # install deps
   --input samplesheet.csv \
   --outdir out
```

Take home message

Give it a try ! Community ❤️

https://training.nextflow.io - https://nf-co.re/pipelines

Join us on Matrix

https://matrix.to/#/#fosdembioinformatics:matrix.orgatrix