SouthGreen bioinformatics platform

MAKING REPRODUCIBLE WORKFLOWS WITH

Jacques Dainat Ph.D.

Why Workflows Management System (WMS)?

- To automate series of bioinformatic processing steps.
- To standardize analysis for large projects or core facilities (repetitive task)
- Optimise computation (parallelization, times, …)
- Simplify deployment of complex pipelines
- Improve reproducibility

What Workflows Management System

There are over 150 workflow managers currently in use and under development
https://github.com/pditommaso/awesome-pipeline
https://github.com/common-workflow-language/common-workflow-language/wiki/Existing-Workflow-systems

Overview of workflow managers for bioinformatics

Tool	Class	Ease of use[a]	Expressiveness[b]	Portability[c]	Scalability[d]	Learning resources[e]	Pipeline initiatives[f]
Galaxy	Graphical	●●●	●○○	●●●	●●●	●●●	●●○
KNIME	Graphical	●●●	●○○	○○○	●●◐	●●●	●●○
Nextflow	DSL	●●○	●●●	●●●	●●●	●●●	●●●
Snakemake	DSL	●●○	●●●	●●◐	●●●	●●○	●●●
GenPipes	DSL	●●○	●●●	●●○	●●○	●●○	●●○
bPipe	DSL	●●○	●●●	●●○	●●◐	●●○	●○○
Pachyderm	DSL	●●○	●●●	●○○	●●○	●●●	○○○
SciPipe	Library	●●○	●●●	○○○	○○○	●●○	○○○
Luigi	Library	●●○	●●●	●○○	●●◐	●●○	○○○
Cromwell + WDL	Execution + workflow specification	●○○	●●○	●●●	●●◐	●●○	●●○
cwltool + CWL	Execution + workflow specification	●○○	●●○	●●◐	○○○	●●●	●●○
Toil + CWL/WDL/Python	Execution + workflow specification	●○○	●●●	●◐○	●●●	●●○	●●○

https://doi-org.insb.bib.cnrs.fr/10.1038/s41592-021-01254-9

Differences between Snakemake and Nextflow

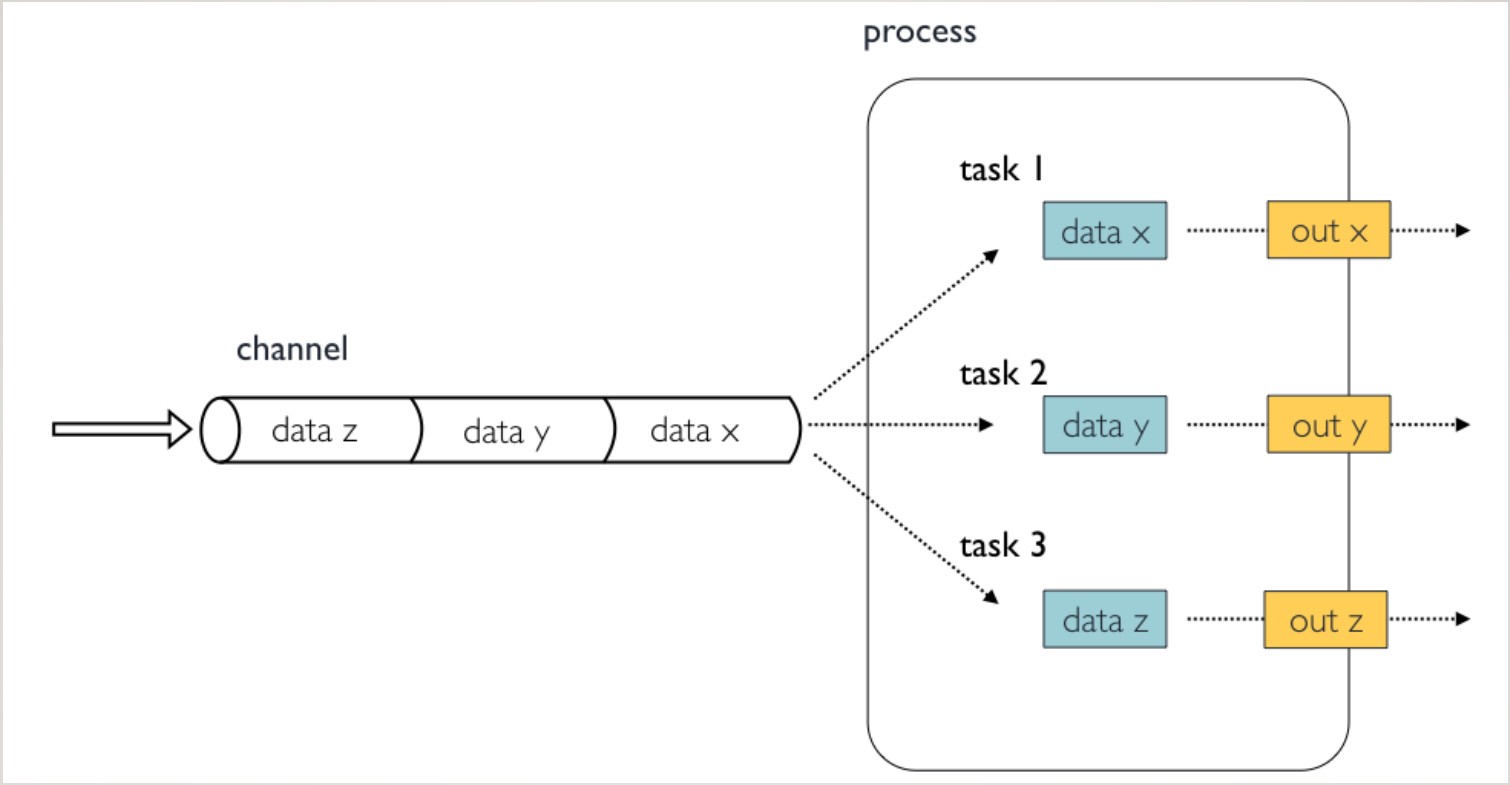
	Snakemake	Nextflow
Language	Python	Groovy
Data	Everything is a file	Can use both files and values
Execution	Working directory	Each job in its own directory
Philosophy	« Pull »	« Push »
Dry-runs	Yes	No
Track code changes	No	Yes

Question: But, which one is the best?
Answer: Both– it's mostly up to personal preference

Nextflow features

- Generalisable
- Portable
- Scalable
- Platform-agnostic
- Based on Groovy and Java
- Large active community in e.g. nf-core

Concepts and nomenclature



- Channels contain data, e.g. input files
- Processes run some kind of code, e.g. a script or a command-line program
- Tasks are instances of a process, one per process input

Anatomy of a process

```
process GET_SRA_BY_ACCESSION {

    input:
        val(sample)

    output:
        path("${sample}.fastq.gz")

    script:
        """
        fastq-dump ${sample} > ${sample}.fastq.gz
        """
}
```

Anatomy of a process

```
process GET_SRA_BY_ACCESSION {

    input:
        val(sample)

    output:
        tuple val(sample), path("${sample}.fastq.gz")

    script:
        """
        fastq-dump ${sample} > ${sample}.fastq.gz
        """
}
```

Anatomy of a process

```
process GET_SRA_BY_ACCESSION {

    cpus 2
    memory '8 GB'

    input:
        val(sample)

    output:
        tuple val(sample), path("${sample}.fastq.gz")

    script:
        """
        fastq-dump ${sample} > ${sample}.fastq.gz
        """
}
```

Anatomy of a process

```
process GET_SRA_BY_ACCESSION {

    cpus 2
    memory '8 GB'

    conda 'sra-tools=2.11.0'
    container 'ncbi/sra-tools:2.11.0'

    input:
        val(sample)

    output:
        tuple val(sample), path("${sample}.fastq.gz")

    script:
        """
        fastq-dump ${sample} > ${sample}.fastq.gz
        """
}
```

Anatomy of a process

```
process GET_SRA_BY_ACCESSION {

    cpus 2
    memory '8 GB'

    conda 'sra-tools=2.11.0'
    container 'ncbi/sra-tools:2.11.0'

    input:
        val(sample)

    output:
        tuple val(sample), path("${sample}.fastq.gz")

    script:
        """
        fastq-dump ${sample} -X {params.depth} > ${sample}.fastq.gz
        """
}
```

Anatomy of a workflow

```
workflow {

    // Define SRA input data channel
    ch_sra_ids = Channel.fromList ( ["SRR935090", "SRR935091"] )

    // Define the workflow
    GET_SRA_BY_ACCESSION ( ch_sra_ids )
}
```

Anatomy of a workflow

```
workflow {

    // Define SRA input data channel
    ch_sra_ids = Channel.fromList ( ["SRR935090", "SRR935091"] )

    // Define the workflow
    GET_SRA_BY_ACCESSION ( ch_sra_ids )

    RUN_FASTQC ( GET_SRA_BY_ACCESSION.out )

}
```

Anatomy of a workflow

```
workflow {

    // Define SRA input data channel
    ch_sra_ids = Channel.fromList ( ["SRR935090", "SRR935091"] )

    // Define the workflow
    GET_SRA_BY_ACCESSION ( ch_sra_ids )

    RUN_FASTQC ( GET_SRA_BY_ACCESSION.out )

    RUN_MULTIQC ( RUN_FASTQC.out.collect() )
}
```

Executing Nextflow

```
# Execute a workflow
$ nextflow run main.nf
```

```
# Re-run using cached results
$ nextflow run main.nf -resume
```

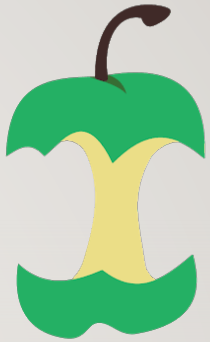
```
# Executing with a specific configuration file
$ nextflow run main.nf -c nextflow.config
```

```
# Supply a custom parameter
$ nextflow run main.nf --my_param "my value"
```

```
# Use Docker or Singularity
$ nextflow run main.nf -with-docker
$ nextflow run main.nf -with-singularity
```

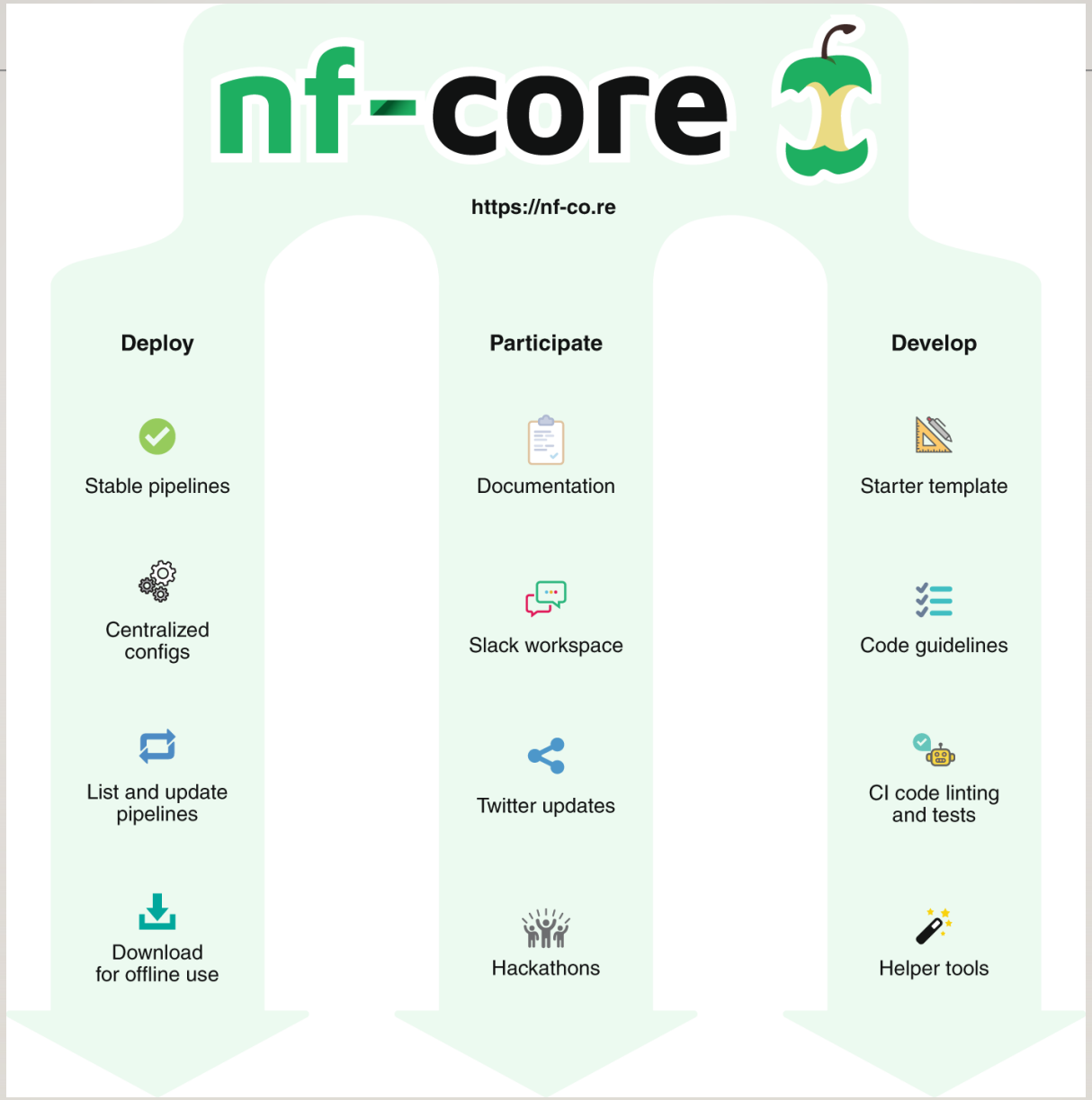
```
# Use a pre-defined configuration profile
$ nextflow run main.nf -profile my_cluster_profile
```

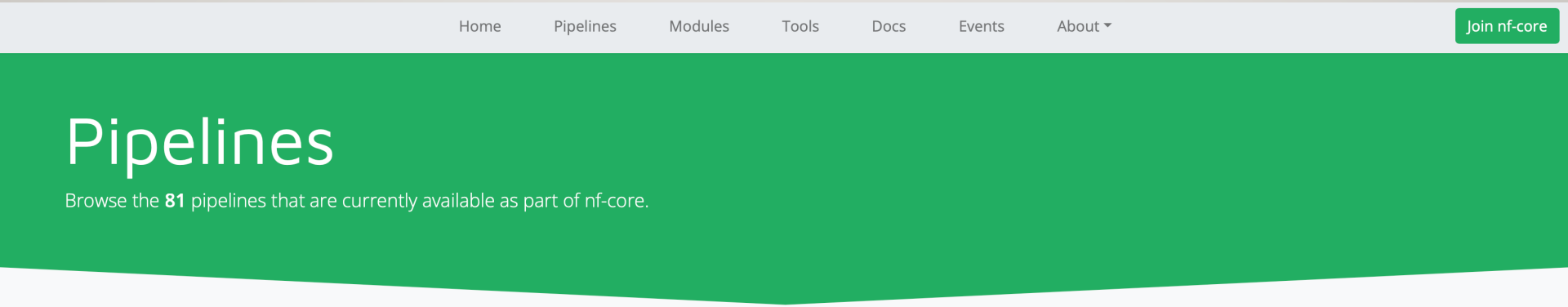
nf-core



Start of 2018 / NGI Stockholm

A community effort to collect a curated set of analysis pipelines built using Nextflow.


nf-core
https://nf-co.re
Deploy
Stable pipelines
Centralized configs
List and update pipelines
Download for offline use
Participate
Documentation
Slack workspace
Twitter updates
Hackathons
Develop
Starter template
Code guidelines
CI code linting and tests
Helper tools



Available Pipelines

Can you think of another pipeline that would fit in well? Let us know!

Search keywords | Filter: Released 49 | Under development 20 | Archived 12 | Sort: Last Release | Alphabetical | Stars | Display:

nf-core/hic

☆ 55

chromosome-conformation-capture hi-c

Analysis of Chromosome Conformation Capture data (Hi-C)

Version 2.1.0 Published 2 hours ago

nf-core/demultiplex

☆ 26

bases2fastq bcl2fastq demultiplexing elementbiosciences illumina

Demultiplexing pipeline for sequencing data

Version 1.3.0 Published 22 hours ago

nf-core/bamtofastq

☆ 9

bamtofastq conversion cramtofastq

Converts bam or cram files to fastq format and does quality control.

Version 2.0.0 Published 1 day ago

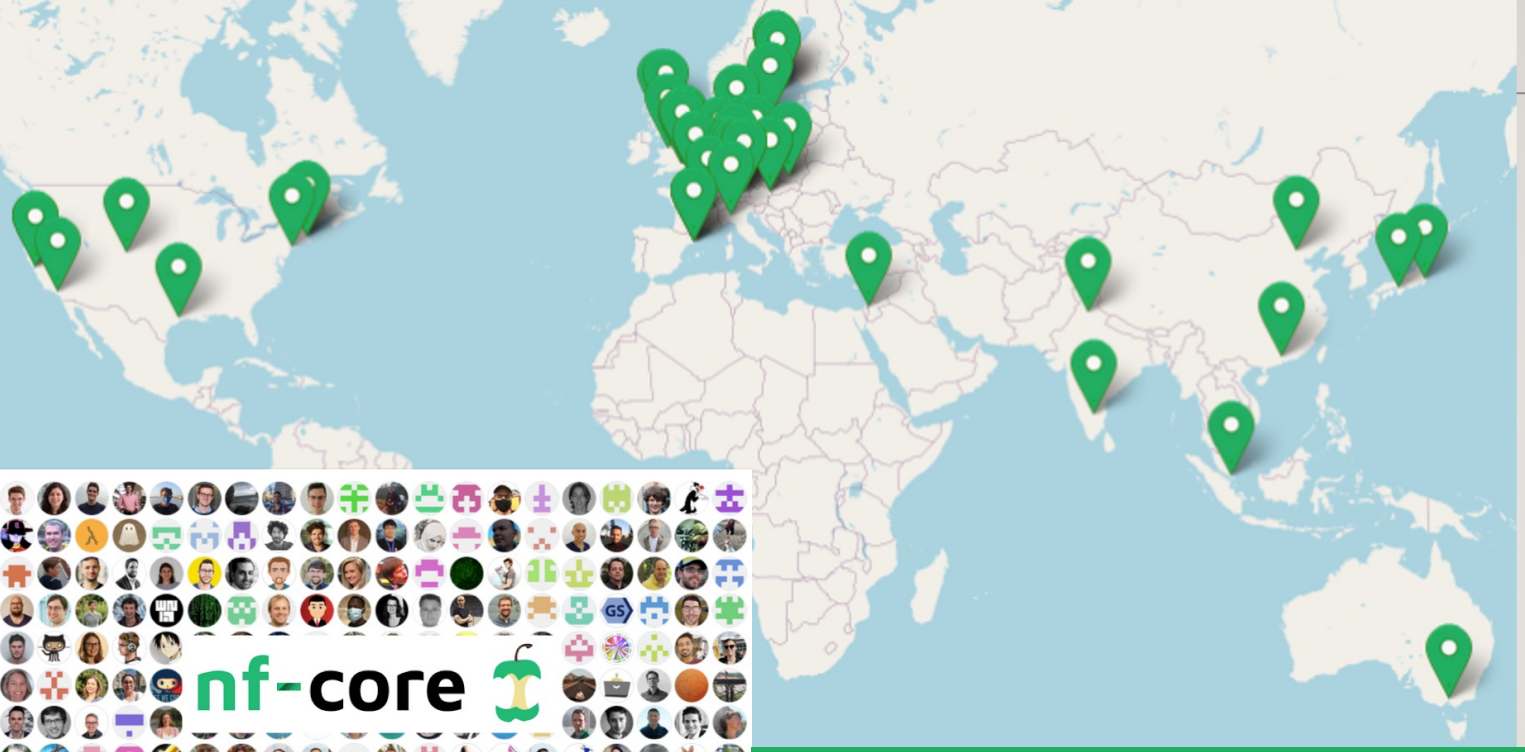
nf-core/funcscan

☆ 33

amp amr antibiotic-resistance antimicrobial-peptides antimicrobial-resistance-genes arg assembly bgc biosynthetic-gene-clusters contigs function metagenomics natural-products screening secondary-metabolites

(Meta-)genome screening for functional and natural product gene sequences

Version 1.1.1 Published 1 week ago



Join nf-core

Read about the different ways you can get involved with nf-core

We use a few different tools to organise the nf-core community - you are welcome to join us at any or all!

Slack | GitHub | twitter | mastodon | YouTube

⚠ All nf-core community members are expected to adhere to the nf-core code of conduct

If your question is about Nextflow and not directly related to nf-core, please use the Slack community chat or the discussion forum on GitHub.