Welcome to this Training Session with Theiagen Genomics

We will soon be getting started

Software Development Practices for Public Health Bioinformatics

Week 04: Workflow Managers and Nextflow

A Mid-Atlantic Workforce Development Offering Provided by the Division of Consolidated Laboratory Services in Collaboration with Theiagen Genomics

Course Introduction

Training Workshop Instructors

Andrew Hale, BSc

- Bioinformatics Developer at Theiagen Genomics since 2025
- BSc in Computational Bioinformatics

Course Agenda

Course Agenda

Use of Workflow Managers

Week 4 - February 25/27, 2025

- Basics of workflow managers, with a focus on Nextflow
    - Package managers
    - Workflow managers
    - Nextflow

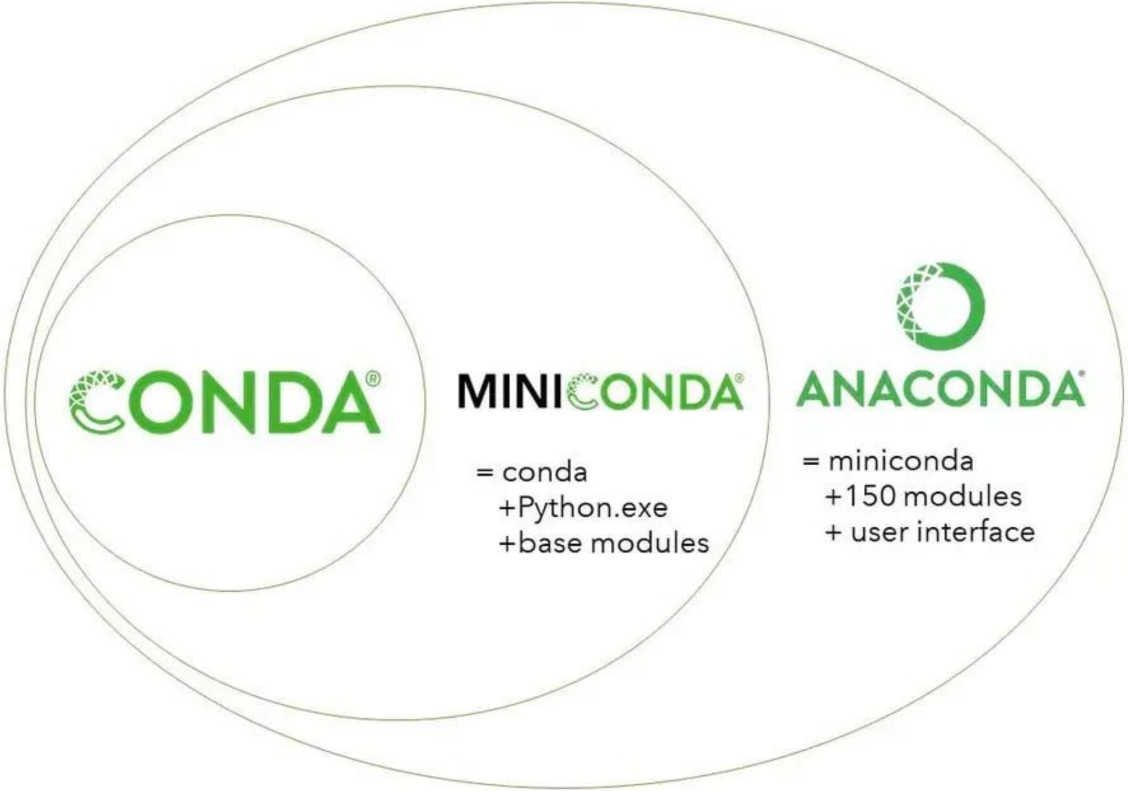
Package Managers

CONDA

Conda is a powerful command line tool for **package and environment management** that runs on Windows, macOS, and Linux.

Anaconda - Comes with a lot of pre-installed packages

Miniconda - Comes with the bare necessities



CONDA®
MINICONDA®
ANACONDA®

= conda
+Python.exe
+base modules

= miniconda
+150 modules
+ user interface

CONDA

Why Conda?

- Non-Root access → Use non-root user
- Dependency conflicts → Installs in separate environments
- Shareable → Installs are modular and easy to share
- Easy to execute → Easy to load/unload environments

```
# Download Miniconda installer
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# Run installer
bash Miniconda3-latest-Linux-x86_64.sh
```

Conda Rules of Thumb

There is an exception to this rule of thumb though!

Rule Number 1 - Keep base clean

Keep your base environment clean! Try to avoid installing anything in your base environment. If your base environment breaks, you have to reinstall Miniconda.

Rule Number 2 - Create environments

`conda create` is your friend, use it for everything! Treat environments as consumables, create them, install in them, delete them.

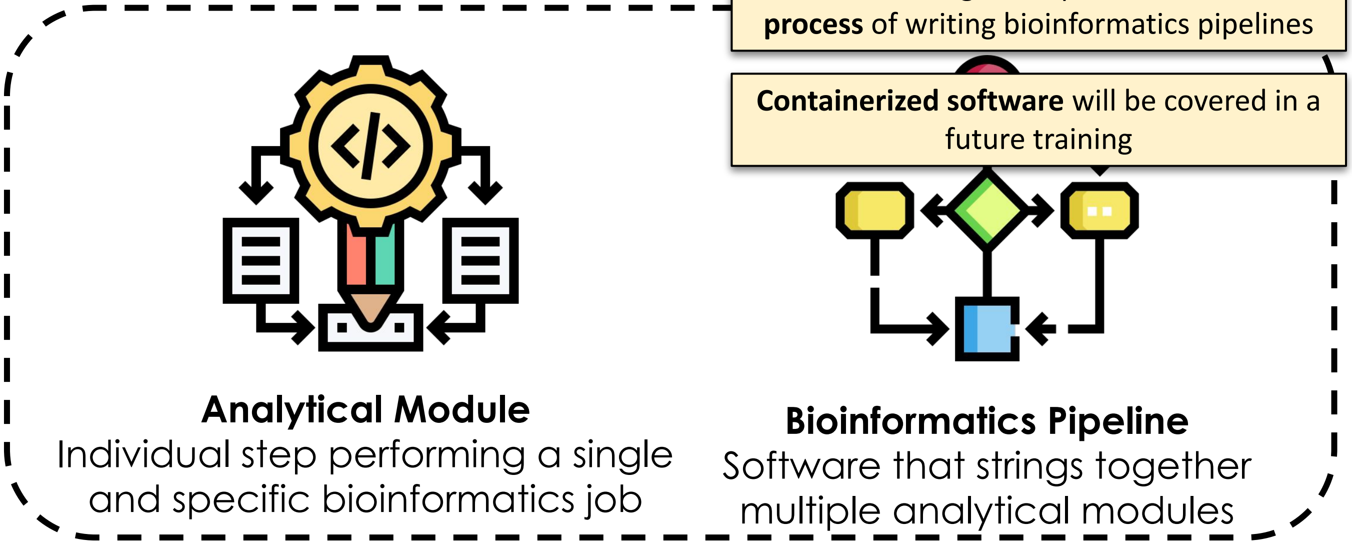
Rule Number 3 - Use containers for critical tasks

Conda is great, but for critical things its best to use containers (Docker or Singularity).

Workflow Managers

Introduction to Workflow Managers

Workflow managers provide a **standardized framework** for creating reproducible and interoperable **bioinformatics pipelines**, especially when **containerized software are utilized**



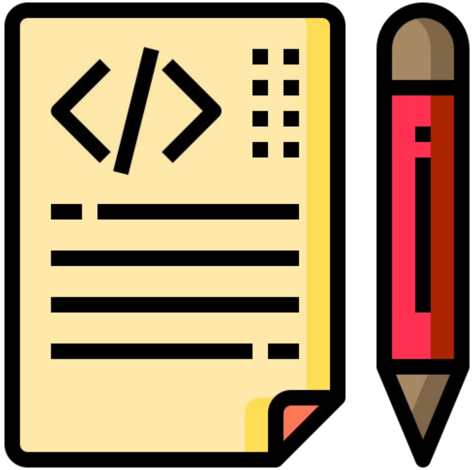
Workflow managers help to **standardize the process** of writing bioinformatics pipelines

Containerized software will be covered in a future training

Analytical Module
Individual step performing a single and specific bioinformatics job

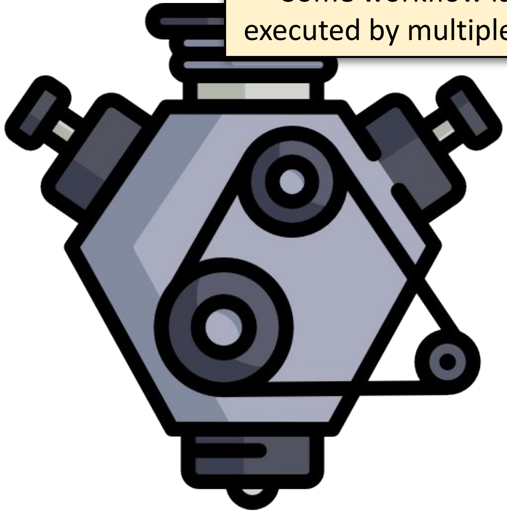
Bioinformatics Pipeline
Software that strings together multiple analytical modules

Introduction to Workflow Managers



Workflow Language

Programming language used to describe the bioinformatics pipeline



Some workflow languages can be executed by multiple workflow engines

Workflow Engine

Software to interpret and execute the workflow itself

Theiagen®

Nextflow

Introduction to Nextflow

Workflows

Using computers to collect, store, analyze and disseminate data and information

Large files

> 10 GB for a metagenomic sample

Many languages

Bash, Python, PERL…

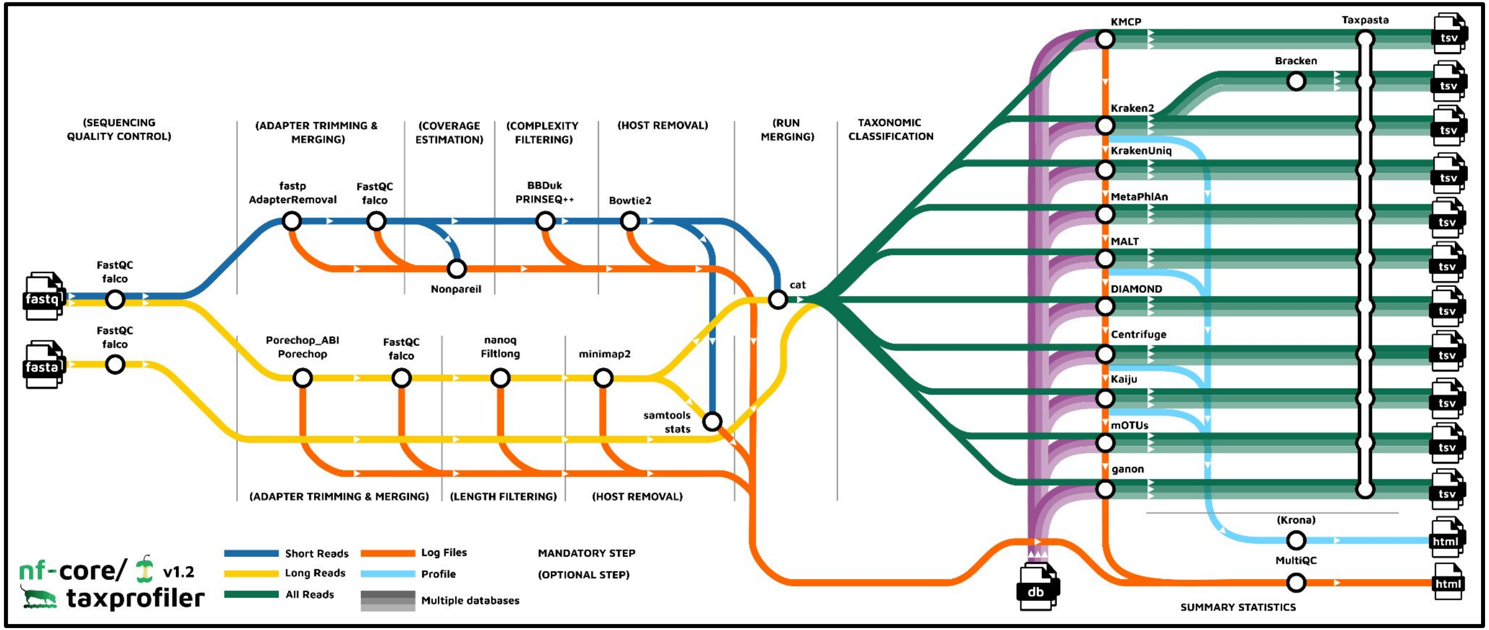
Complex interactions

Network of software and their dependencies

Introduction to Nextflow

Nextflow

Managing modern workflows is complicated


(SEQUENCING QUALITY CONTROL)
(ADAPTER TRIMMING & MERGING)
(COVERAGE ESTIMATION)
(COMPLEXITY FILTERING)
(HOST REMOVAL)
(RUN MERGING)
TAXONOMIC CLASSIFICATION
fastq
fasta
FastQC falco
FastQC falco
fastp AdapterRemoval
FastQC falco
Nonpareil
BBDuk PRINSEQ++
Bowtie2
cat
Porechop_ABI Porechop
FastQC falco
nanoq Filtlong
minimap2
samtools stats
(ADAPTER TRIMMING & MERGING)
(LENGTH FILTERING)
(HOST REMOVAL)
KMCP
Kraken2
KrakenUniq
MetaPhlAn
MALT
DIAMOND
Centrifuge
Kaiju
mOTUs
ganon
Bracken
Taxpasta
tsv
html
(Krona)
MultiQC
db
SUMMARY STATISTICS
nf-core/ taxprofiler v1.2
Short Reads
Long Reads
All Reads
Log Files
Profile
Multiple databases
MANDATORY STEP
(OPTIONAL STEP)

https://nf-co.re/taxprofiler/1.2.2/

Introduction to Nextflow

Reactive workflow framework

Create pipelines with asynchronous data streams

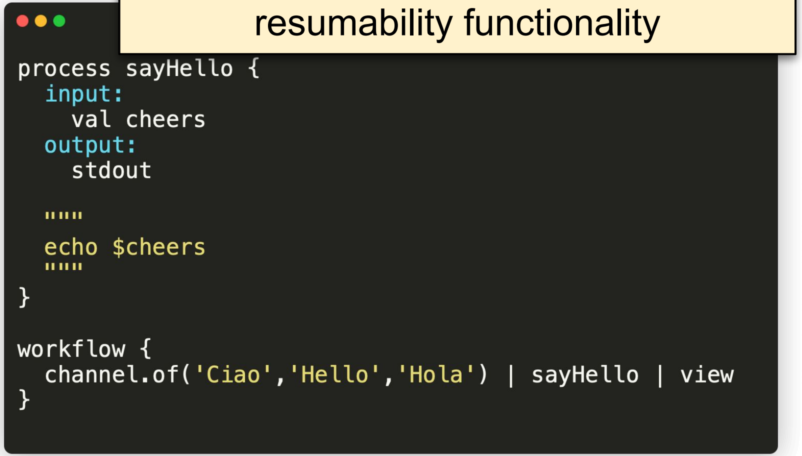
Programing DSL

Has its own language for building a pipeline

Containerized

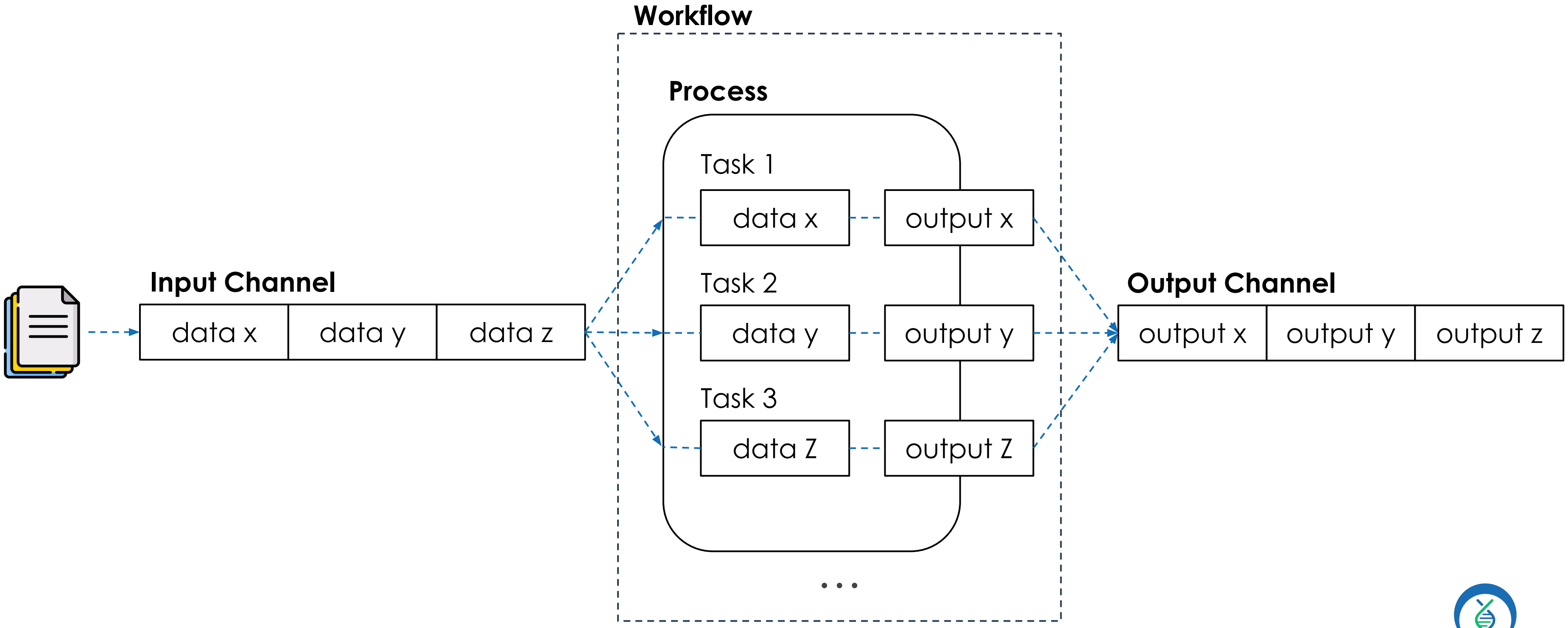
Out of the box integrating with container engines

Simplifies the deployment of complex parallel and reactive workflows with resumability functionality



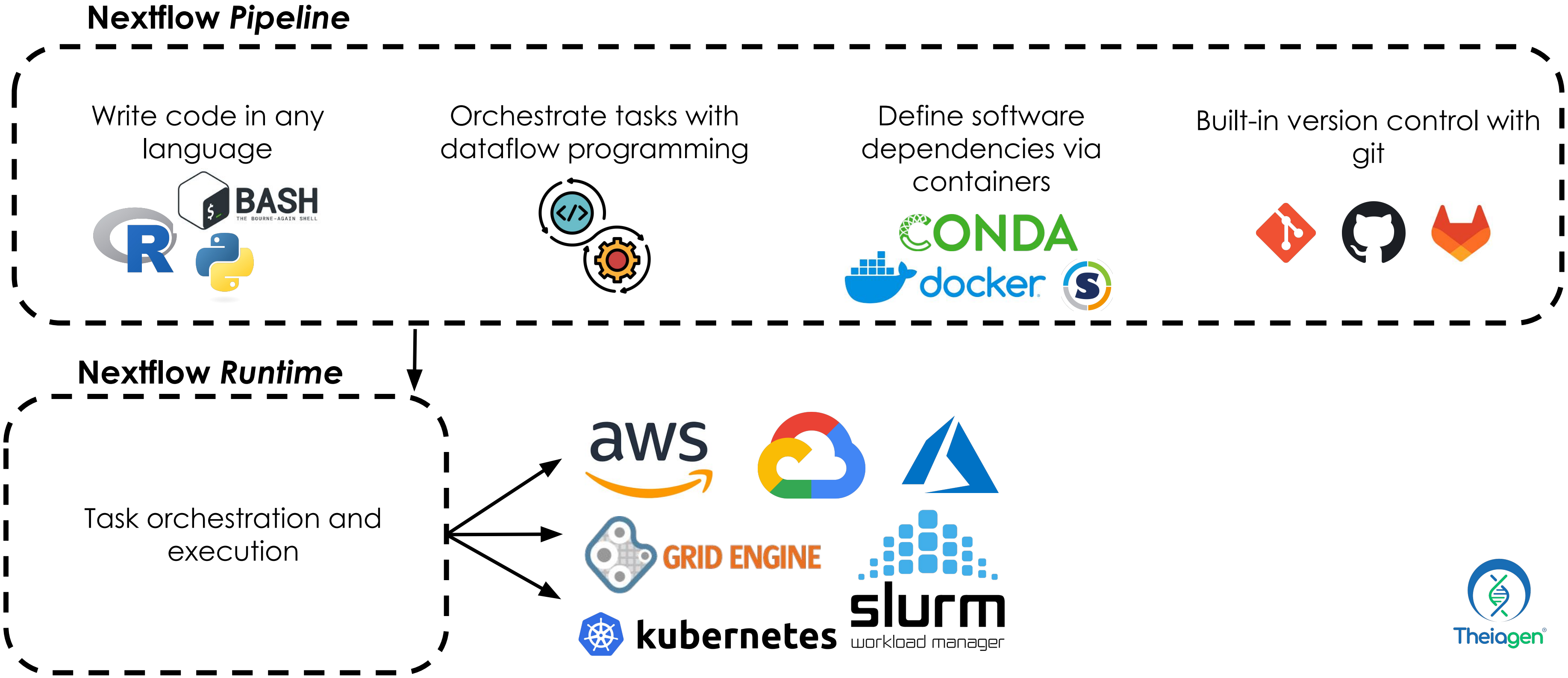
```
process sayHello {
  input:
    val cheers
  output:
    stdout

  """
  echo $cheers
  """
}

workflow {
  channel.of('Ciao','Hello','Hola') | sayHello | view
}
```

Introduction to Nextflow



Workflow
Process
Input Channel
data x
data y
data z
Task 1
data x
output x
Task 2
data y
output y
Task 3
data Z
output Z
. . .
Output Channel
output x
output y
output z

Introduction to Nextflow


BASH
THE BOURNE-AGAIN SHELL
R
CONDA
docker
aws
GRID ENGINE
kubernetes
slurm
workload manager

Nextflow *Pipeline*

Write code in any language

Orchestrate tasks with dataflow programming

Define software dependencies via containers

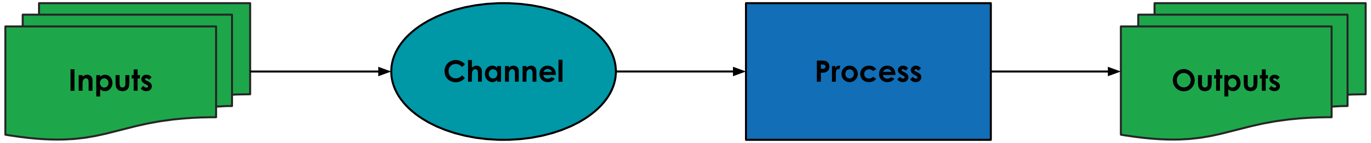
Built-in version control with git

Nextflow *Runtime*

Task orchestration and execution

1. Nextflow Basics

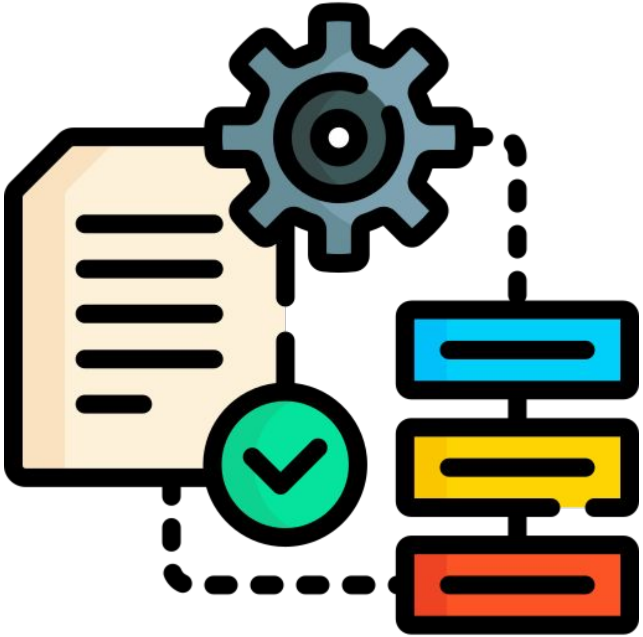
Nextflow Basic Structure


Inputs
Channel
Process
Outputs

- Channel
    - Queue Channel
        - "First-in, First-out" (FIFO) connecting processes and operators
    - Value Channel
        - Stores a single value (e.g., genome size)
        - Can be a named list (e.g., ["id": "my_sample", "genome_size": 360000])

- Channels can be used by **Operators** and **Processes**

Nextflow Basic Structure



- Methods to **connect** channels, or **transform** values of channels
    - Filtering: filter, randomSample, take, unique
    - Transforming: collect, groupTuple, reduce
    - Text Processing: splitCsv, splitJson, splitText
    - Combining: combine, concat, join, mix
    - Forking: branch, multiMap
    - Math: count, max, min, sum
    - Other: ifEmpty, map, set, view
- More than [50 Operators]() available to use

Nextflow Basic Structure

Process example:

```
bwa mem reference.fa sample.fq \
    | samtools sort -o sample.bam
```

Nextflow Basic Structure

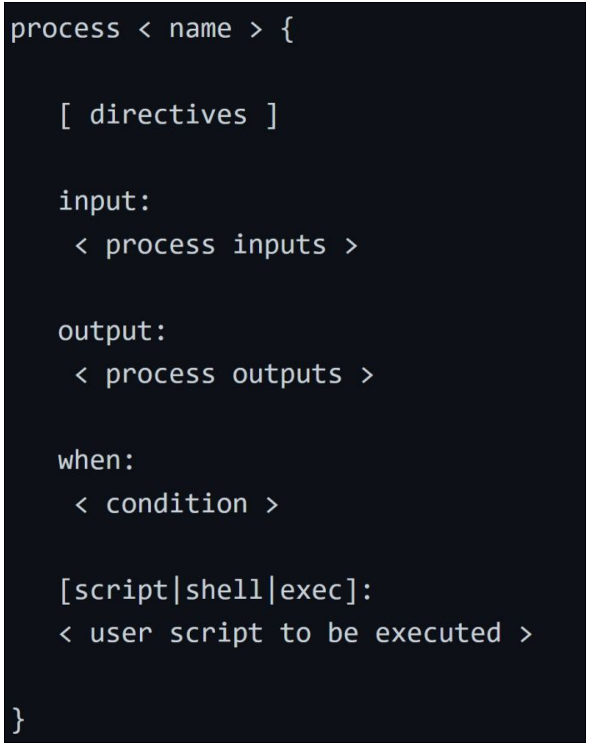
Process example:

```
process align_sample {

    input:
    file ‘reference.fasta’
    file ‘sample.fq’

    output:
    file ‘sample.bam’

    script:
    “““
    bwa mem reference.fa sample.fq \
            | samtools sort -o sample.bam
    ”””
}
```

Nextflow Basic Structure

Processes:

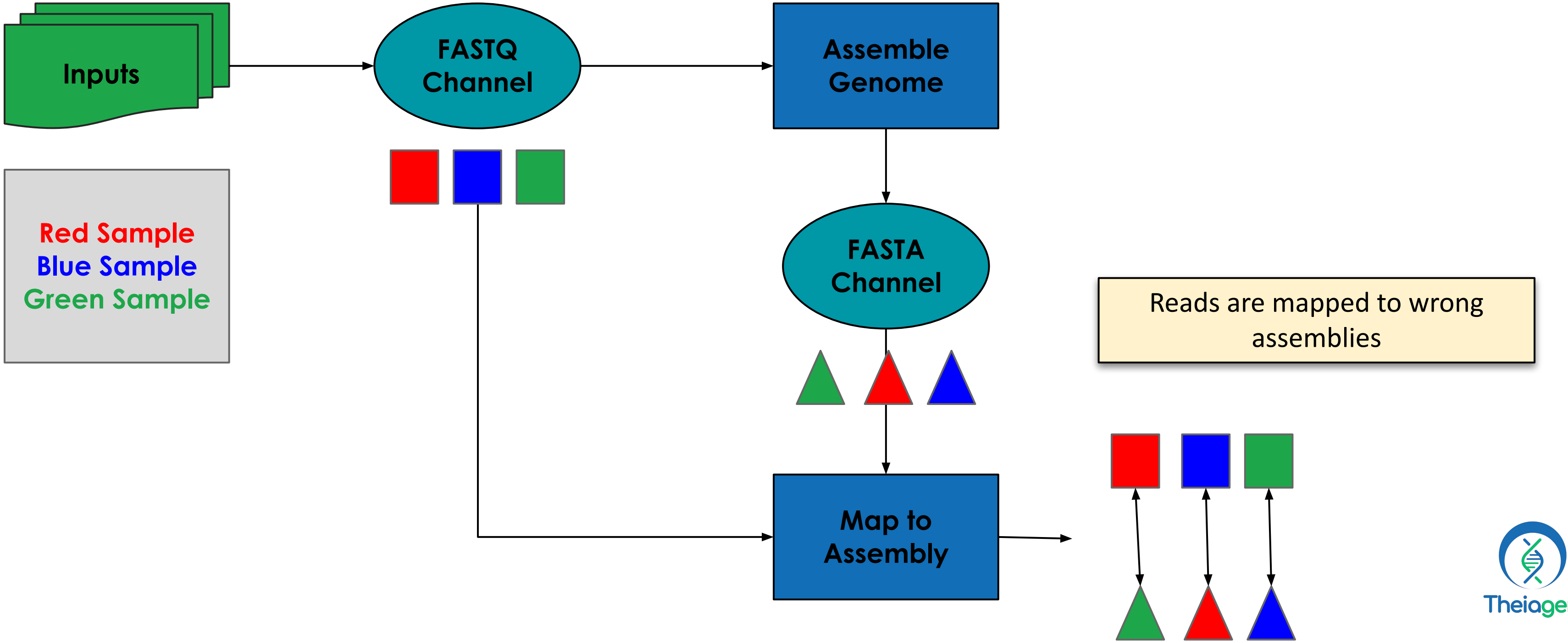
- The basic unit for executing user scripts
- [30+ directives] adjust optional settings
- Can be dynamic
- Inputs and outputs are channels
- Can be optional
- Conditional executions using ‘when’
- The script block executes user code



```
process < name > {

   [ directives ]

   input:
    < process inputs >

   output:
    < process outputs >

   when:
    < condition >

   [script|shell|exec]:
   < user script to be executed >

}
```

Nextflow Basic Structure

FIFO Can Cause "Unexpected" Results:

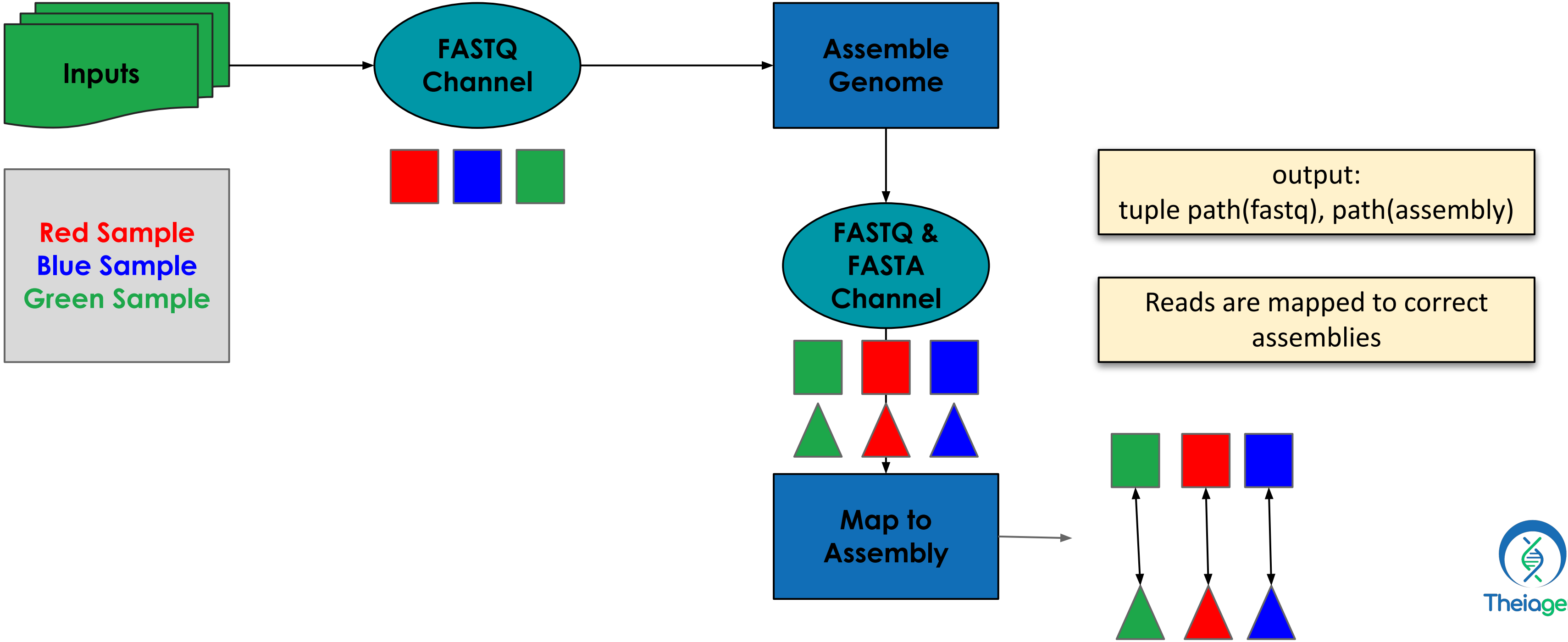


Inputs
FASTQ Channel
Assemble Genome
FASTA Channel
Map to Assembly
Red Sample
Blue Sample
Green Sample

Reads are mapped to wrong assemblies

Nextflow Basic Structure

FIFO Can Cause "Unexpected" Results:



Inputs
FASTQ Channel
Assemble Genome
Red Sample
Blue Sample
Green Sample
FASTQ & FASTA Channel
Map to Assembly

output:
tuple path(fastq), path(assembly)

Reads are mapped to correct assemblies

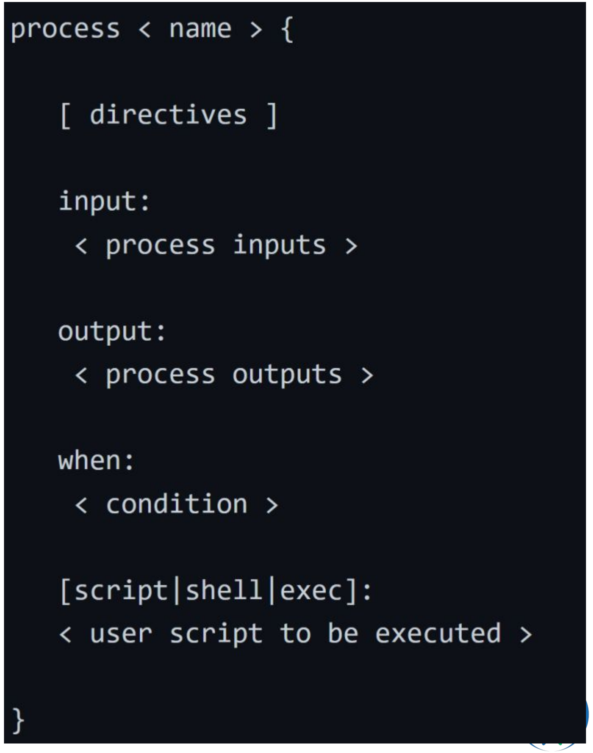
2. Nextflow Files

Nextflow Files

The Nextflow module file

- Modules – A **reusable** Nextflow script with a process definition
- Subworkflows – Multiple modules linked together

Modules are portable and easily shared between workflows



```
process < name > {

   [ directives ]

   input:
    < process inputs >

   output:
    < process outputs >

   when:
    < condition >

   [script|shell|exec]:
   < user script to be executed >

}
```

Nextflow Files

The Nextflow script file
The `main.nf` file is the central script in a Nextflow workflow.

It defines the structure and execution logic of the pipeline by **orchestrating the processes and connecting them with channels**. This file acts as the "blueprint" for how data flows through the pipeline and how tasks are executed.

```
#!/usr/bin/env nextflow
nextflow.enable.dsl=2

// Correctly include the process
// definition from the module
include { newModule } from
'./modules/new_module'

workflow {
    // Define the input channel from the
    // user-specified path
    IN_FilePath =
Channel.fromPath(params.input).ifEmpty {
        exit 1, "No file provided with
pattern: ${params.input}"
    }

    // Execute the 'newModule' process
    newModule(IN_FilePath) | view
}
```

Nextflow Files

The Nextflow config file

When a pipeline script is launched, Nextflow looks for configuration files. By default it searches for `nextflow.config` but one can be provided directly via the `-c <config-file>` optional parameter.

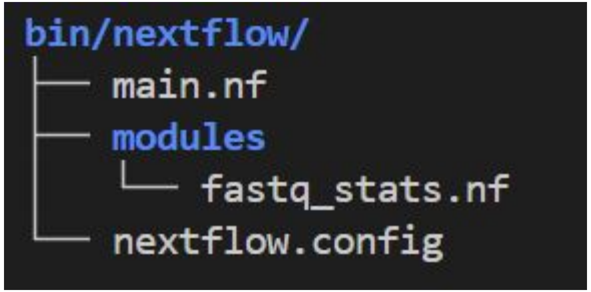
```
// Define default settings
params {
    input = null
}

// Configure process settings
process {
    cpus = 2
    memory = '2 GB'
    time = '1h'
}

profiles {
    standard {
        // Default profile for local execution
        process.executor = 'local'
    }

    docker {
        process {
            executor = 'docker'
            container = 'ubuntu:jammy'
        }
    }
}
```

Nextflow Files

Putting everything together

This is not obligatory, but following this structure is highly recommended.

The modules live in a `modules` folder where the `main.nf` and `nextflow.config` files are.



```
bin/nextflow/
├── main.nf
├── modules
│   └── fastq_stats.nf
└── nextflow.config
```

3. Running Nextflow Workflows

Running Nextflow Workflows

```
nextflow run main.nf -c nextflow.config --input sample.fastq
```

To run a pipeline:

- All software dependencies must be installed (Nextflow + Docker / Singularity / Conda).
- Configure Nextflow to run on your system (-profile)
- Run the tests for your pipeline in the terminal to confirm everything is working (-profile test)

Core Nextflow command-line options use one (-), whereas pipeline-specific parameters use two (--).

Use -resume to restart pipelines that did not complete. This uses cached results for successful tasks from the previous run, instead of executing all tasks from scratch.

5. Nf-Core

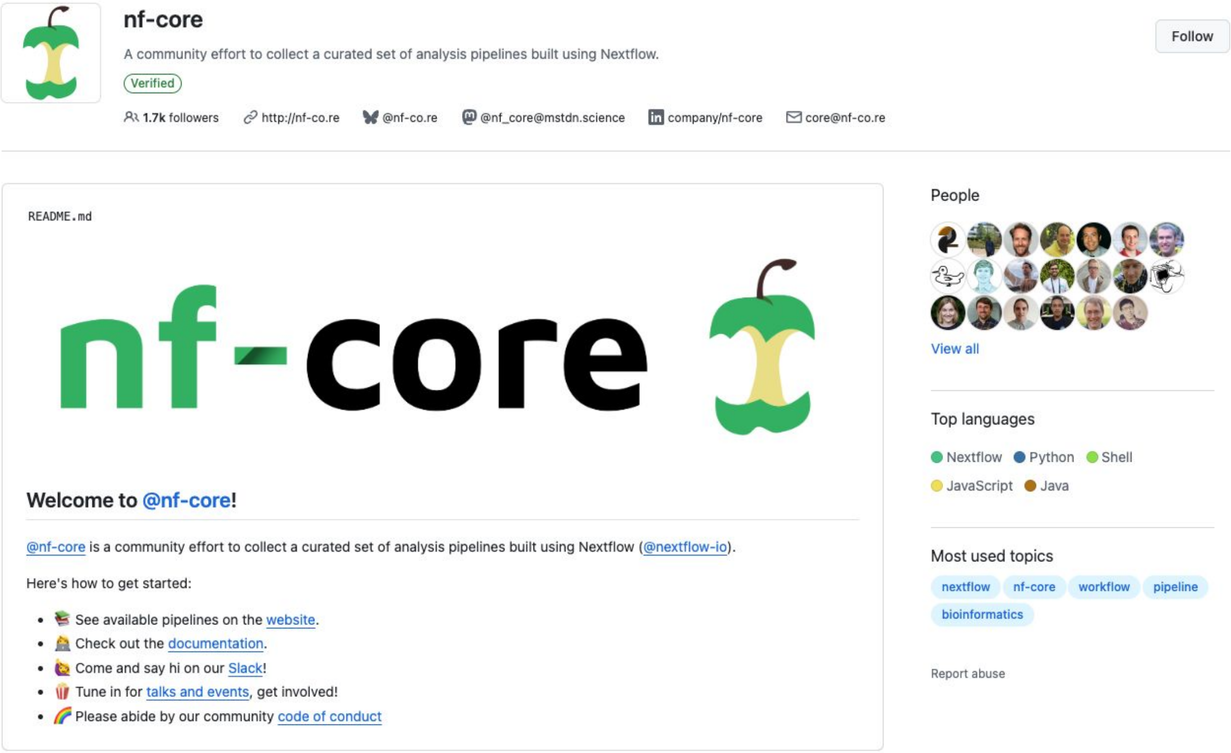
nf-core

Develop with a community

Use a common template

Collaborate, don’t duplicate

Nf-Core Github Repository



nf-core

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

1.7k followers · http://nf-co.re · @nf-co.re · @nf_core@mstdn.science · company/nf-core · core@nf-co.re

Welcome to @nf-core!

@nf-core is a community effort to collect a curated set of analysis pipelines built using Nextflow (@nextflow-io).

Here's how to get started:

- 📚 See available pipelines on the website.
- 👩‍🏫 Check out the documentation.
- 🙋 Come and say hi on our Slack!
- 🍿 Tune in for talks and events, get involved!
- 🌈 Please abide by our community code of conduct

People

View all

Top languages

● Nextflow ● Python ● Shell
● JavaScript ● Java

Most used topics

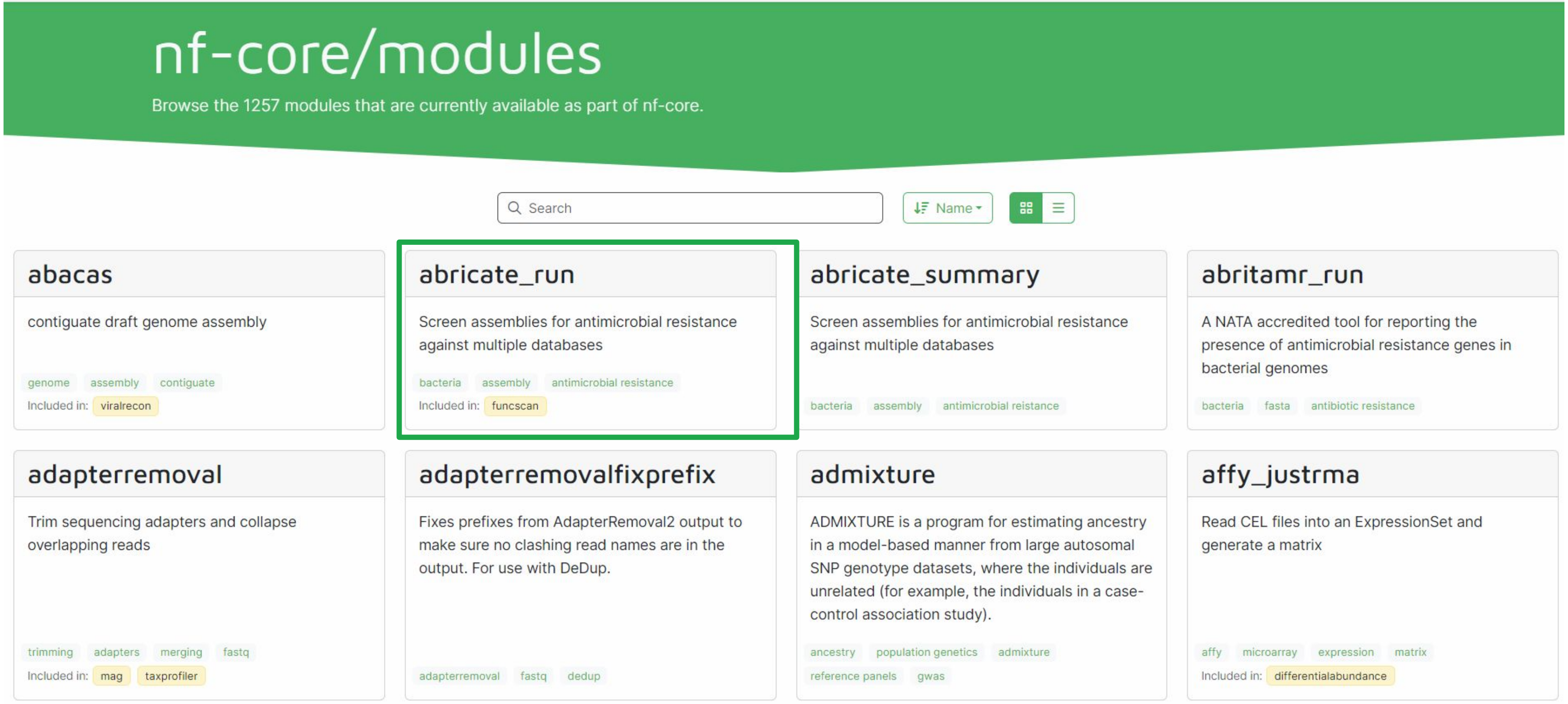
nextflow · nf-core · workflow · pipeline · bioinformatics

Report abuse

nf-core

- nf-core/modules has 500+ ready to use DSL2 modules
  - Rapid prototyping and pipeline building
  - Version controlled, extensive logging, supports Conda, Docker and Singularity

Nextflow DSL2 Module



nf-core/modules

Browse the 1257 modules that are currently available as part of nf-core.

abacas

contiguate draft genome assembly

genome assembly contiguate

Included in: viralrecon

abricate_run

Screen assemblies for antimicrobial resistance against multiple databases

bacteria assembly antimicrobial resistance

Included in: funcscan

abricate_summary

Screen assemblies for antimicrobial resistance against multiple databases

bacteria assembly antimicrobial reistance

abritamr_run

A NATA accredited tool for reporting the presence of antimicrobial resistance genes in bacterial genomes

bacteria fasta antibiotic resistance

adapterremoval

Trim sequencing adapters and collapse overlapping reads

trimming adapters merging fastq

Included in: mag taxprofiler

adapterremovalfixprefix

Fixes prefixes from AdapterRemoval2 output to make sure no clashing read names are in the output. For use with DeDup.

adapterremoval fastq dedup

admixture

ADMIXTURE is a program for estimating ancestry in a model-based manner from large autosomal SNP genotype datasets, where the individuals are unrelated (for example, the individuals in a case-control association study).

ancestry population genetics admixture reference panels gwas

affy_justrma

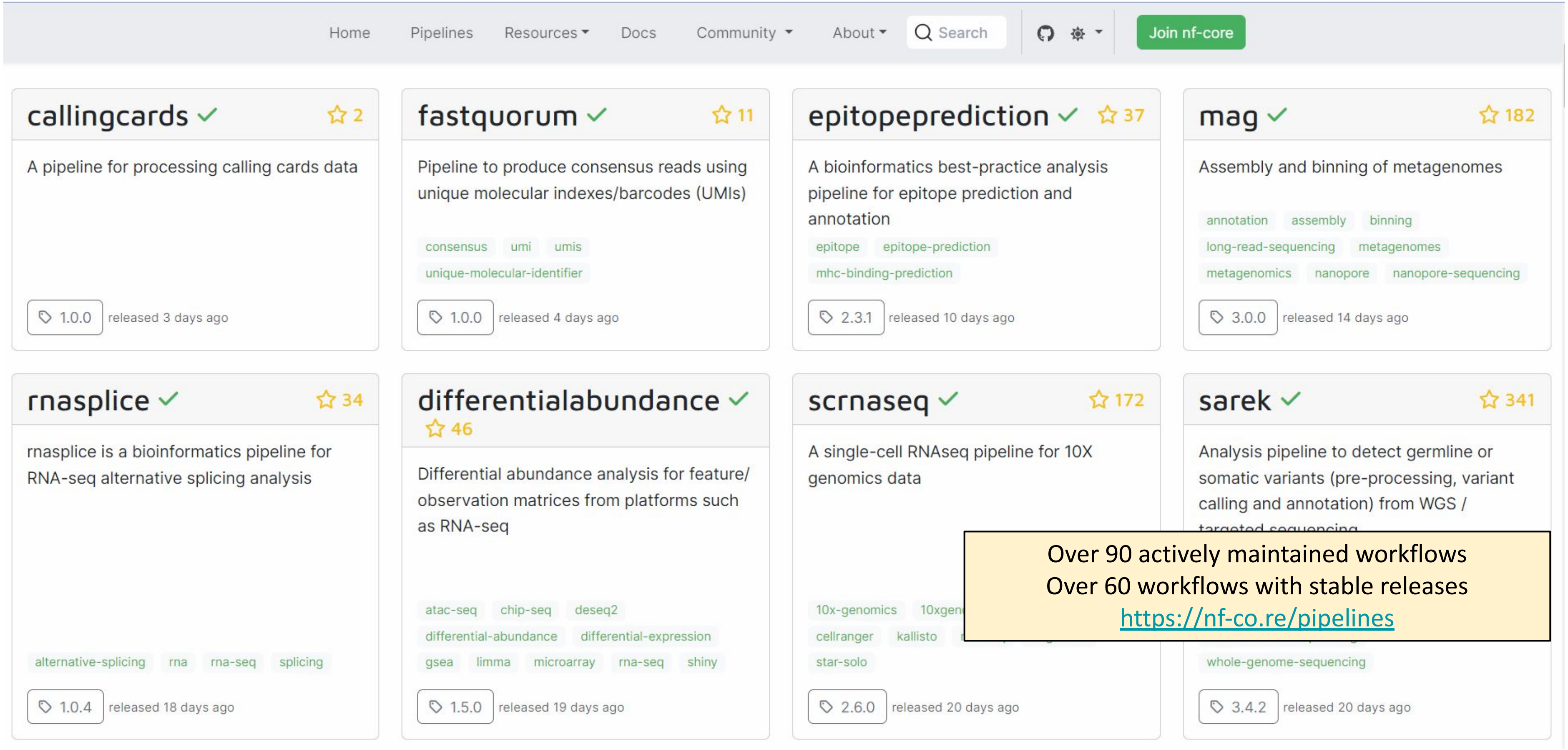
Read CEL files into an ExpressionSet and generate a matrix

affy microarray expression matrix

Included in: differentialabundance

nf-core

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



callingcards ✓ ☆ 2

A pipeline for processing calling cards data

1.0.0 released 3 days ago

fastquorum ✓ ☆ 11

Pipeline to produce consensus reads using unique molecular indexes/barcodes (UMIs)

consensus umi umis unique-molecular-identifier

1.0.0 released 4 days ago

epitopeprediction ✓ ☆ 37

A bioinformatics best-practice analysis pipeline for epitope prediction and annotation

epitope epitope-prediction mhc-binding-prediction

2.3.1 released 10 days ago

mag ✓ ☆ 182

Assembly and binning of metagenomes

annotation assembly binning long-read-sequencing metagenomes metagenomics nanopore nanopore-sequencing

3.0.0 released 14 days ago

rnasplice ✓ ☆ 34

rnasplice is a bioinformatics pipeline for RNA-seq alternative splicing analysis

alternative-splicing rna rna-seq splicing

1.0.4 released 18 days ago

differentialabundance ✓ ☆ 46

Differential abundance analysis for feature/observation matrices from platforms such as RNA-seq

atac-seq chip-seq deseq2 differential-abundance differential-expression gsea limma microarray rna-seq shiny

1.5.0 released 19 days ago

scrnaseq ✓ ☆ 172

A single-cell RNAseq pipeline for 10X genomics data

10x-genomics 10xgen cellranger kallisto star-solo

2.6.0 released 20 days ago

sarek ✓ ☆ 341

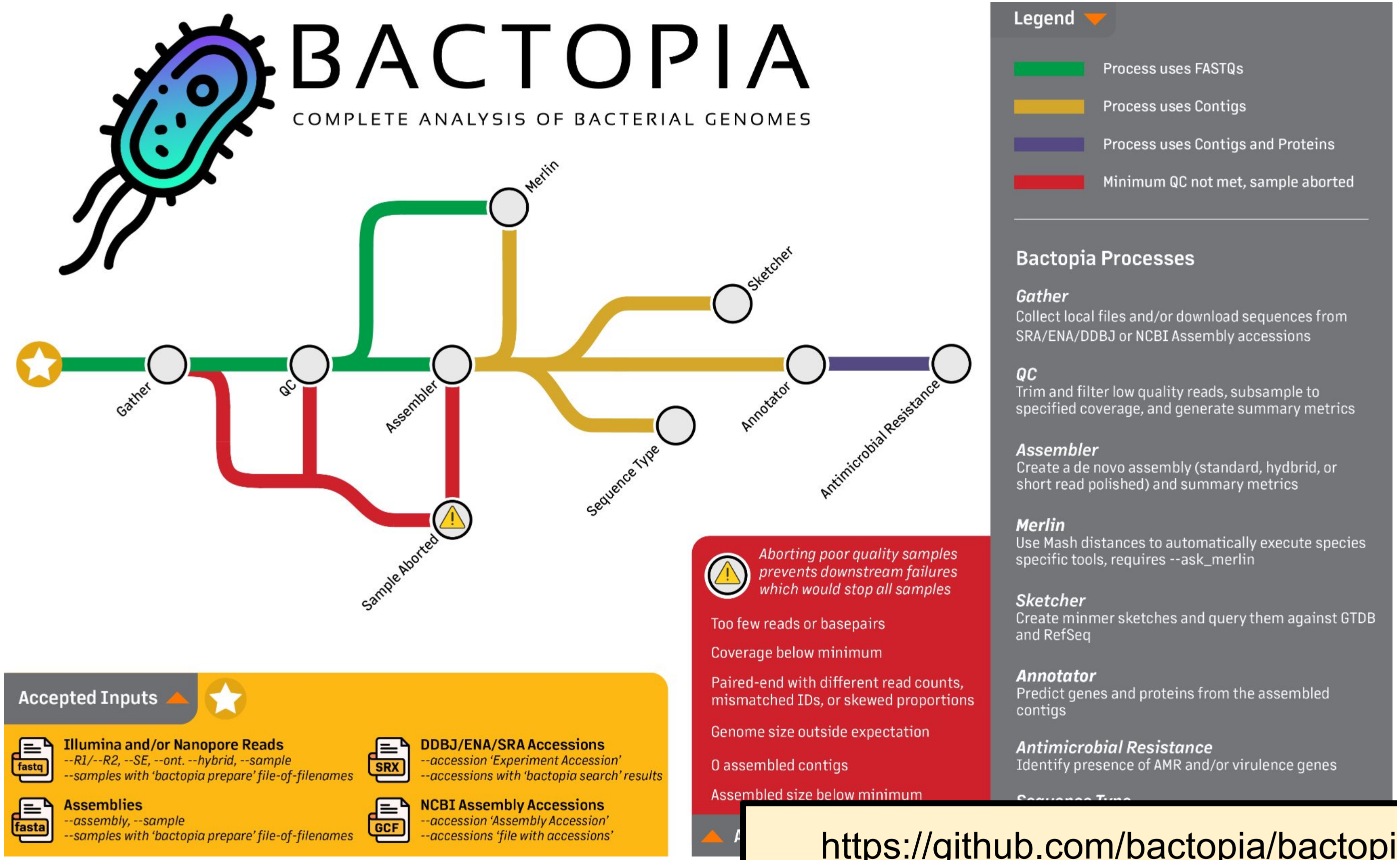
Analysis pipeline to detect germline or somatic variants (pre-processing, variant calling and annotation) from WGS / targeted sequencing

whole-genome-sequencing

3.4.2 released 20 days ago

Over 90 actively maintained workflows
Over 60 workflows with stable releases
https://nf-co.re/pipelines

6. Awesome Workflows



BACTOPIA

COMPLETE ANALYSIS OF BACTERIAL GENOMES

Legend

- 🟩 Process uses FASTQs
- 🟨 Process uses Contigs
- 🟪 Process uses Contigs and Proteins
- 🟥 Minimum QC not met, sample aborted

Bactopia Processes

Gather
Collect local files and/or download sequences from SRA/ENA/DDBJ or NCBI Assembly accessions

QC
Trim and filter low quality reads, subsample to specified coverage, and generate summary metrics

Assembler
Create a de novo assembly (standard, hydbrid, or short read polished) and summary metrics

Merlin
Use Mash distances to automatically execute species specific tools, requires --ask_merlin

Sketcher
Create minmer sketches and query them against GTDB and RefSeq

Annotator
Predict genes and proteins from the assembled contigs

Antimicrobial Resistance
Identify presence of AMR and/or virulence genes

⚠ *Aborting poor quality samples prevents downstream failures which would stop all samples*

- Too few reads or basepairs
- Coverage below minimum
- Paired-end with different read counts, mismatched IDs, or skewed proportions
- Genome size outside expectation
- 0 assembled contigs
- Assembled size below minimum

Accepted Inputs ★

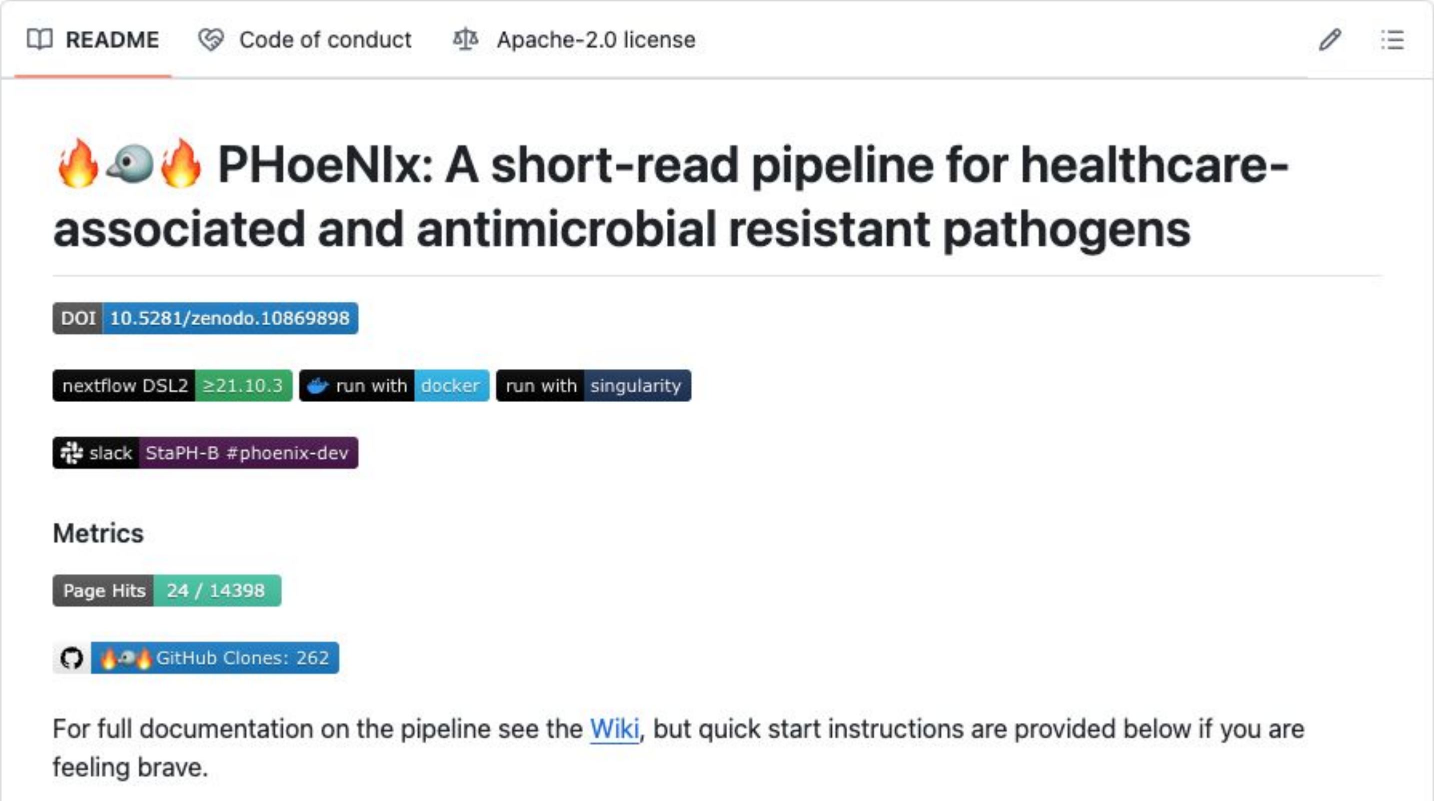
Illumina and/or Nanopore Reads
--R1/--R2, --SE, --ont. --hybrid, --sample
--samples with 'bactopia prepare' file-of-filenames

DDBJ/ENA/SRA Accessions
--accession 'Experiment Accession'
--accessions with 'bactopia search' results

Assemblies
--assembly, --sample
--samples with 'bactopia prepare' file-of-filenames

NCBI Assembly Accessions
--accession 'Assembly Accession'
--accessions 'file with accessions'

https://github.com/bactopia/bactopia


README
Code of conduct
Apache-2.0 license

🔥🐦🔥 PHoeNIx: A short-read pipeline for healthcare-associated and antimicrobial resistant pathogens

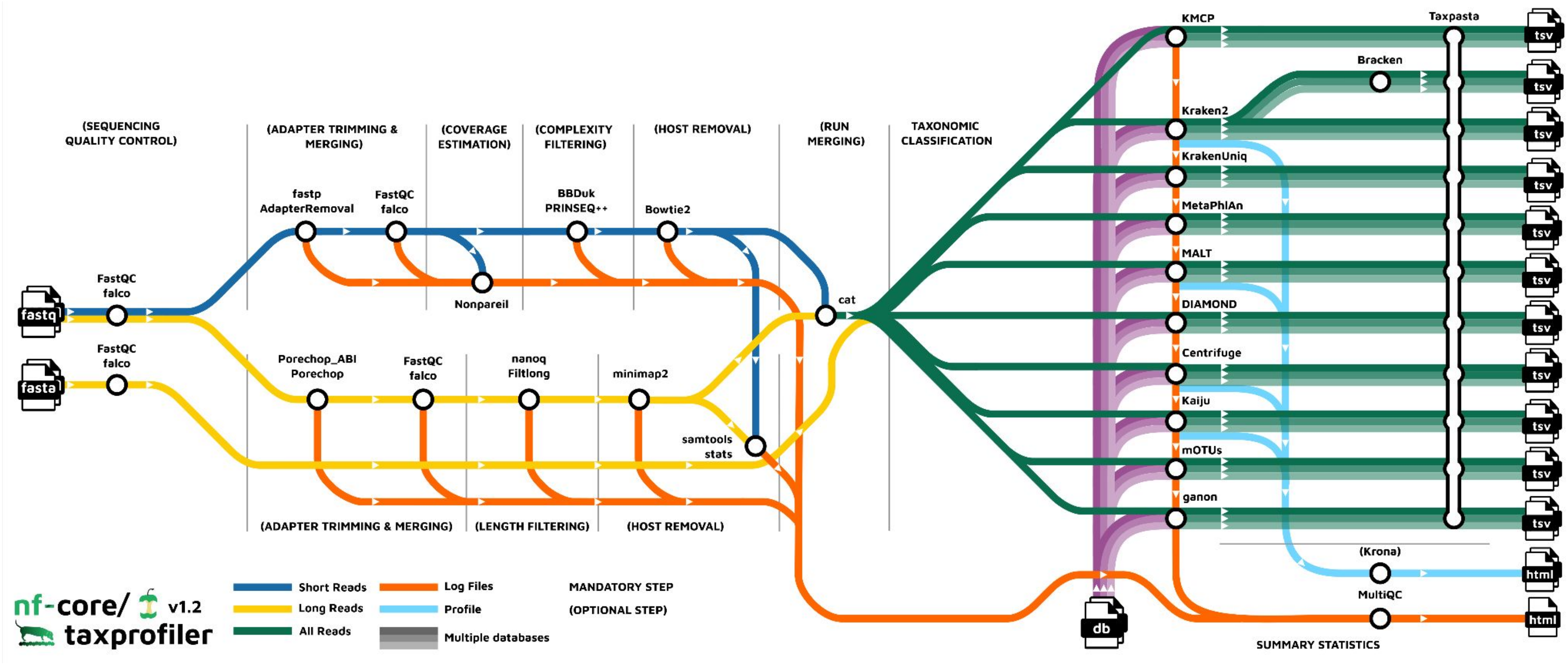
DOI 10.5281/zenodo.10869898
nextflow DSL2 ≥21.10.3
run with docker
run with singularity
slack StaPH-B #phoenix-dev

Metrics

Page Hits 24 / 14398
🔥🐦🔥 GitHub Clones: 262

For full documentation on the pipeline see the Wiki, but quick start instructions are provided below if you are feeling brave.

https://github.com/CDCgov/phoenix


(SEQUENCING QUALITY CONTROL)
(ADAPTER TRIMMING & MERGING)
(COVERAGE ESTIMATION)
(COMPLEXITY FILTERING)
(HOST REMOVAL)
(RUN MERGING)
TAXONOMIC CLASSIFICATION
fastq
fasta
FastQC
falco
FastQC
falco
fastp
AdapterRemoval
FastQC
falco
Nonpareil
BBDuk
PRINSEQ++
Bowtie2
Porechop_ABI
Porechop
FastQC
falco
nanoq
Filtlong
minimap2
samtools
stats
cat
(ADAPTER TRIMMING & MERGING)
(LENGTH FILTERING)
(HOST REMOVAL)
KMCP
Kraken2
KrakenUniq
MetaPhlAn
MALT
DIAMOND
Centrifuge
Kaiju
mOTUs
ganon
Bracken
Taxpasta
db
tsv
tsv
tsv
tsv
tsv
tsv
tsv
tsv
tsv
tsv
tsv
(Krona)
html
MultiQC
html
SUMMARY STATISTICS
Short Reads
Long Reads
All Reads
Log Files
Profile
Multiple databases
MANDATORY STEP
(OPTIONAL STEP)
nf-core/ v1.2
taxprofiler

https://github.com/nf-core/taxprofiler

README
Code of conduct
Apache-2.0 license
nf-core/
mycosnp
nextflow DSL2 ≥21.10.3
run with conda
run with docker
run wi

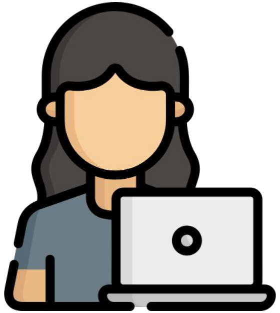
https://github.com/CDCgov/mycosnp-nf

Hands-On Exercise

Exercise 04: Writing Your First Nextflow Workflow

Exercise Goal

1. **Use GitHub and your dev environment to:**
    a. Install miniconda
    b. Install and test Nextflow
    c. Wrap fastq-peek.sh into a Nextflow workflow



Theiagen®