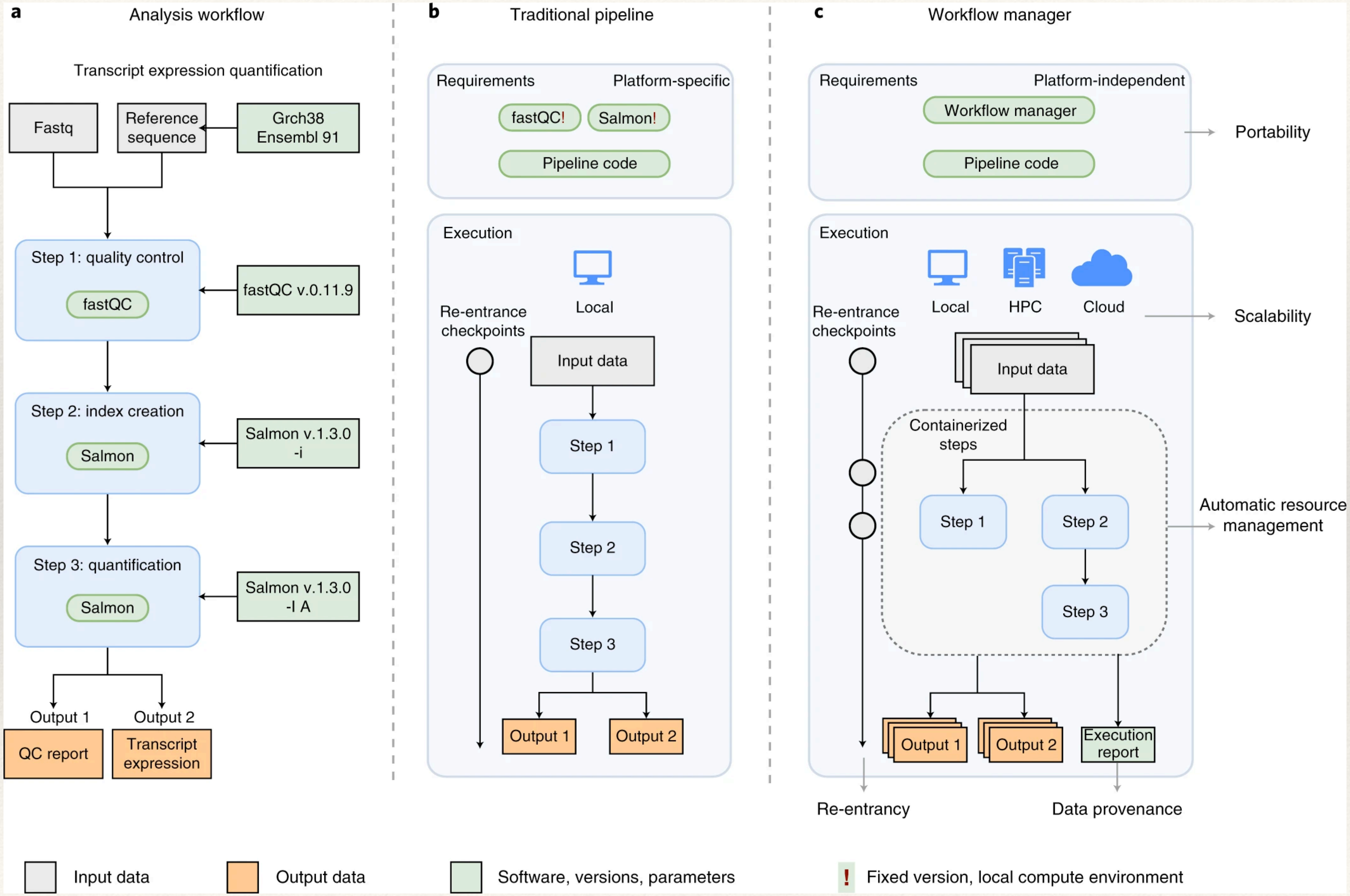
UiO IN-BIOS5000/9000

Nextflow - pipelines

Arvind Sundaram
Oct 30, 2024

Norwegian Sequencing Centre
OUS, Ullevål, Oslo

Bioinformatics workflow managers



https://www-nature-com.ezproxy.uio.no/articles/s41592-021-01254-9

Requirements

- Data provenance
- Portability
- Scalability
- Re-entrancy

Requirements

- Graphical workflow managers: point-and-click pipeline development
- Domain-specific language workflow managers (DSL): rapid and flexible development
- Programming-library-based workflow managers
- Workflow specifications: portability across workflow systems
- Ready-to-use pipelines provide easy access to complex workflows

Bioinformatics workflow managers

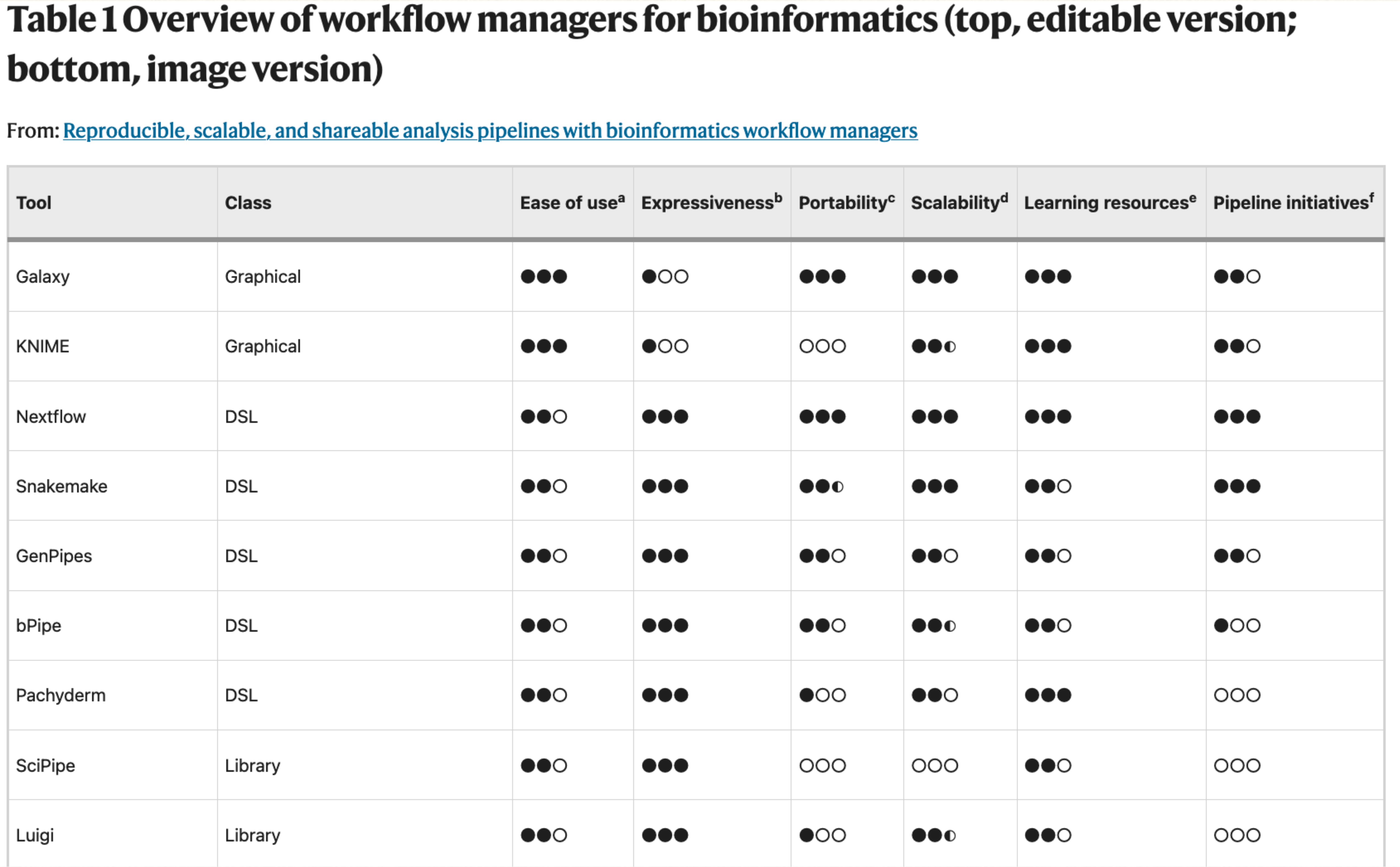


Table 1 Overview of workflow managers for bioinformatics (top, editable version; bottom, image version)

From: Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers

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	●●○	●●●	●○○	●●◐	●●○	○○○

https://www-nature-com.ezproxy.uio.no/articles/s41592-021-01254-9

Bioinformatics workflow managers

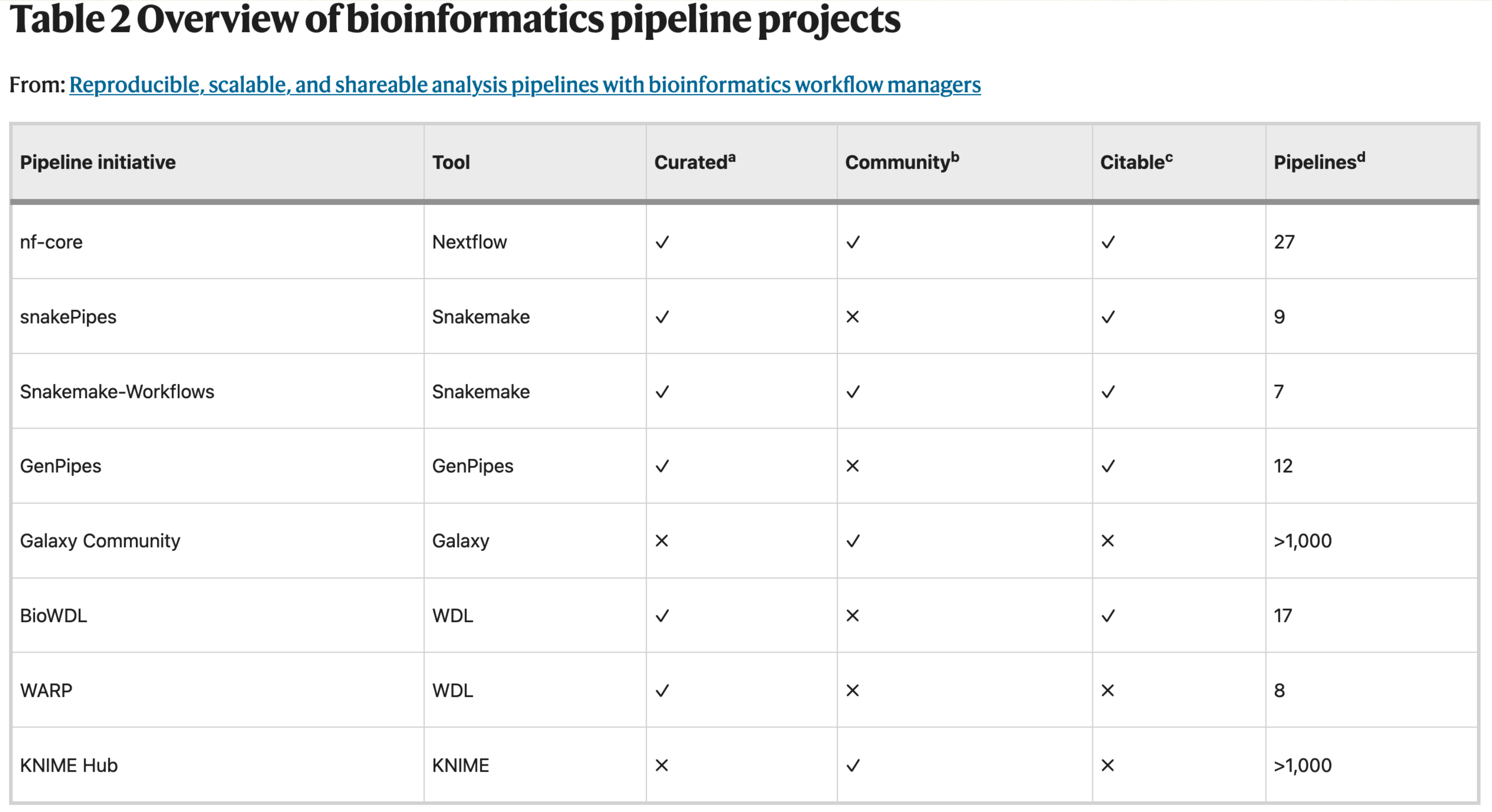


Table 2 Overview of bioinformatics pipeline projects

From: Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers

Pipeline initiative	Tool	Curated[a]	Community[b]	Citable[c]	Pipelines[d]
nf-core	Nextflow	✓	✓	✓	27
snakePipes	Snakemake	✓	×	✓	9
Snakemake-Workflows	Snakemake	✓	✓	✓	7
GenPipes	GenPipes	✓	×	✓	12
Galaxy Community	Galaxy	×	✓	×	>1,000
BioWDL	WDL	✓	×	✓	17
WARP	WDL	✓	×	×	8
KNIME Hub	KNIME	×	✓	×	>1,000

https://www-nature-com.ezproxy.uio.no/articles/s41592-021-01254-9

Nextflow

nextflow

https://www.nextflow.io/

- Built for bioinformatics
- Uses “processes”, “channels” and “workflows”
- Written in Groovy
- Out of the box support
  - for several coding repositories
  - containers
  - Execution environments

Parallelisation

Reentrancy

Reusability

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

  """
  echo $cheers
  """
}

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

Nextflow

Software

Language

Compute

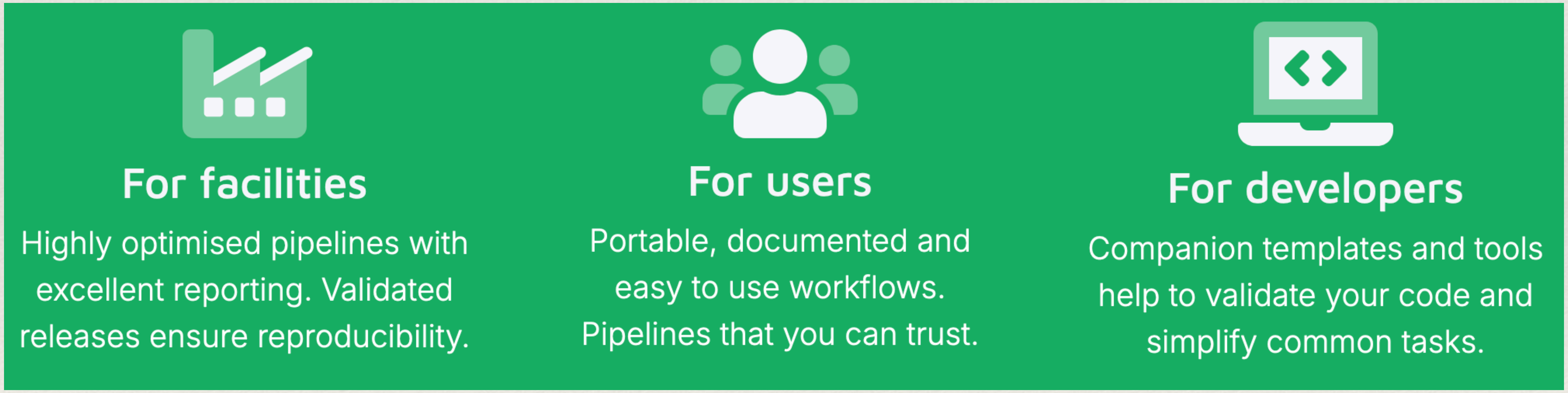
Quick hands-on

```
cd
cd in-biosx000
mkdir nextflow
cd nextflow

rsync -rauPW /projects/ec34/in-biosx000/nextflow .

module load Nextflow/24.04.2

cat main.nf

nextflow run main.nf

singularity pull fastqc_v1.sif docker://arvindsundaram/fastqc:v1

nextflow run main.nf -with-apptainer fastqc_v1.sif
```

Nextflow community

nf-core

https://nf-co.re/



For facilities

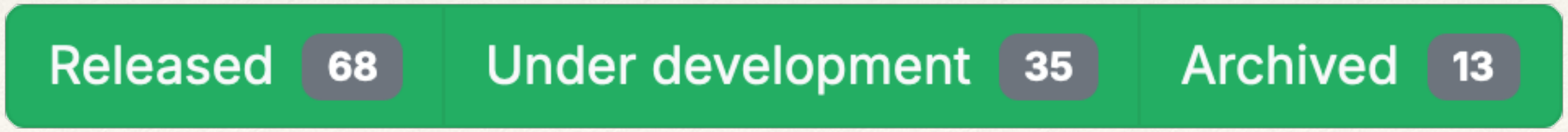
Highly optimised pipelines with excellent reporting. Validated releases ensure reproducibility.

For users

Portable, documented and easy to use workflows. Pipelines that you can trust.

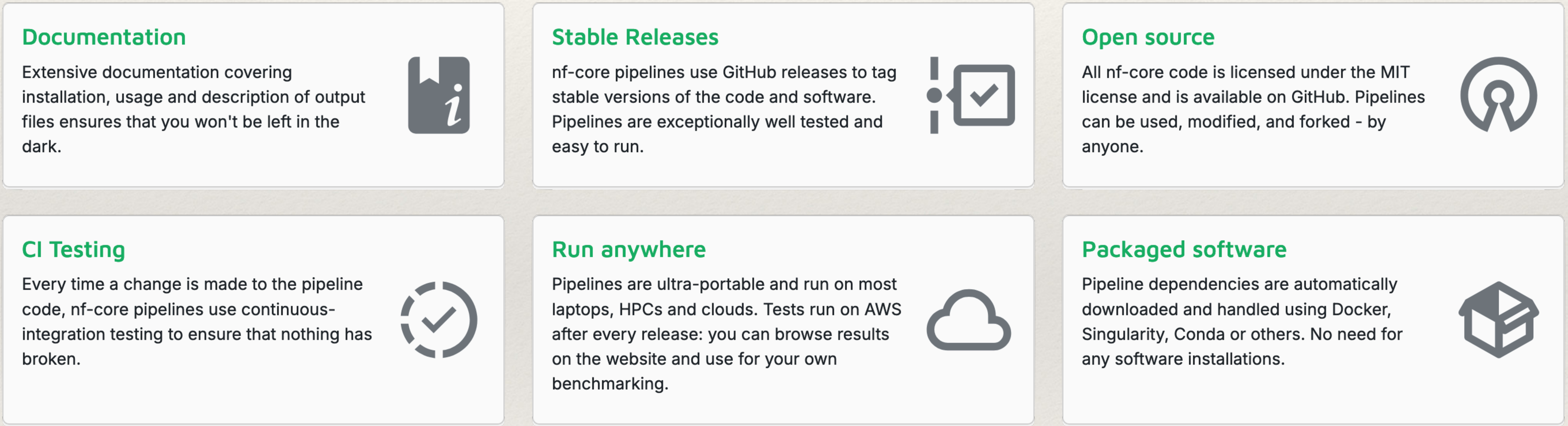
For developers

Companion templates and tools help to validate your code and simplify common tasks.



Released	Under development	Archived
68	35	13

nf-core



Documentation

Extensive documentation covering installation, usage and description of output files ensures that you won't be left in the dark.

Stable Releases

nf-core pipelines use GitHub releases to tag stable versions of the code and software. Pipelines are exceptionally well tested and easy to run.

Open source

All nf-core code is licensed under the MIT license and is available on GitHub. Pipelines can be used, modified, and forked - by anyone.

CI Testing

Every time a change is made to the pipeline code, nf-core pipelines use continuous-integration testing to ensure that nothing has broken.

Run anywhere

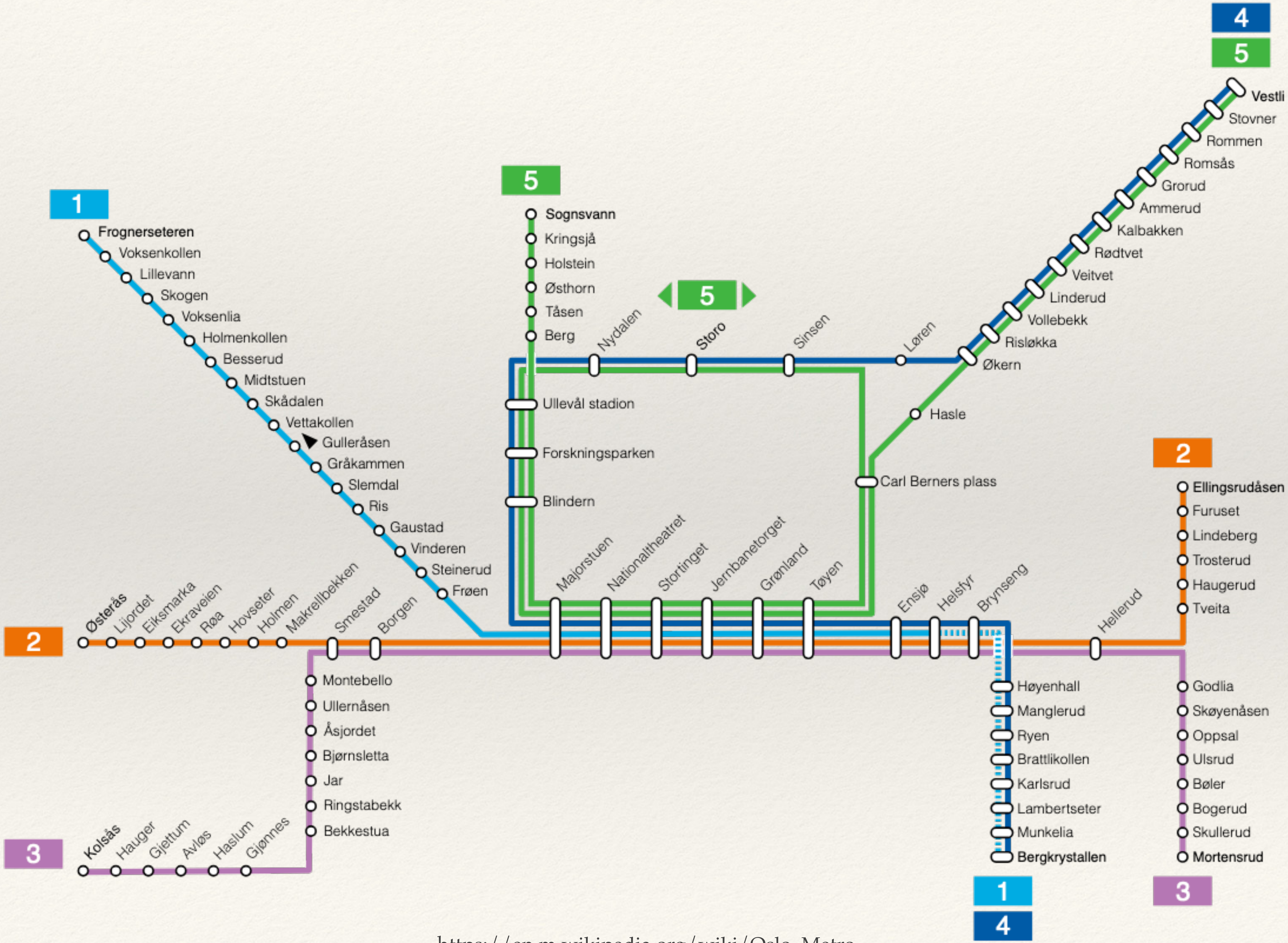
Pipelines are ultra-portable and run on most laptops, HPCs and clouds. Tests run on AWS after every release: you can browse results on the website and use for your own benchmarking.

Packaged software

Pipeline dependencies are automatically downloaded and handled using Docker, Singularity, Conda or others. No need for any software installations.

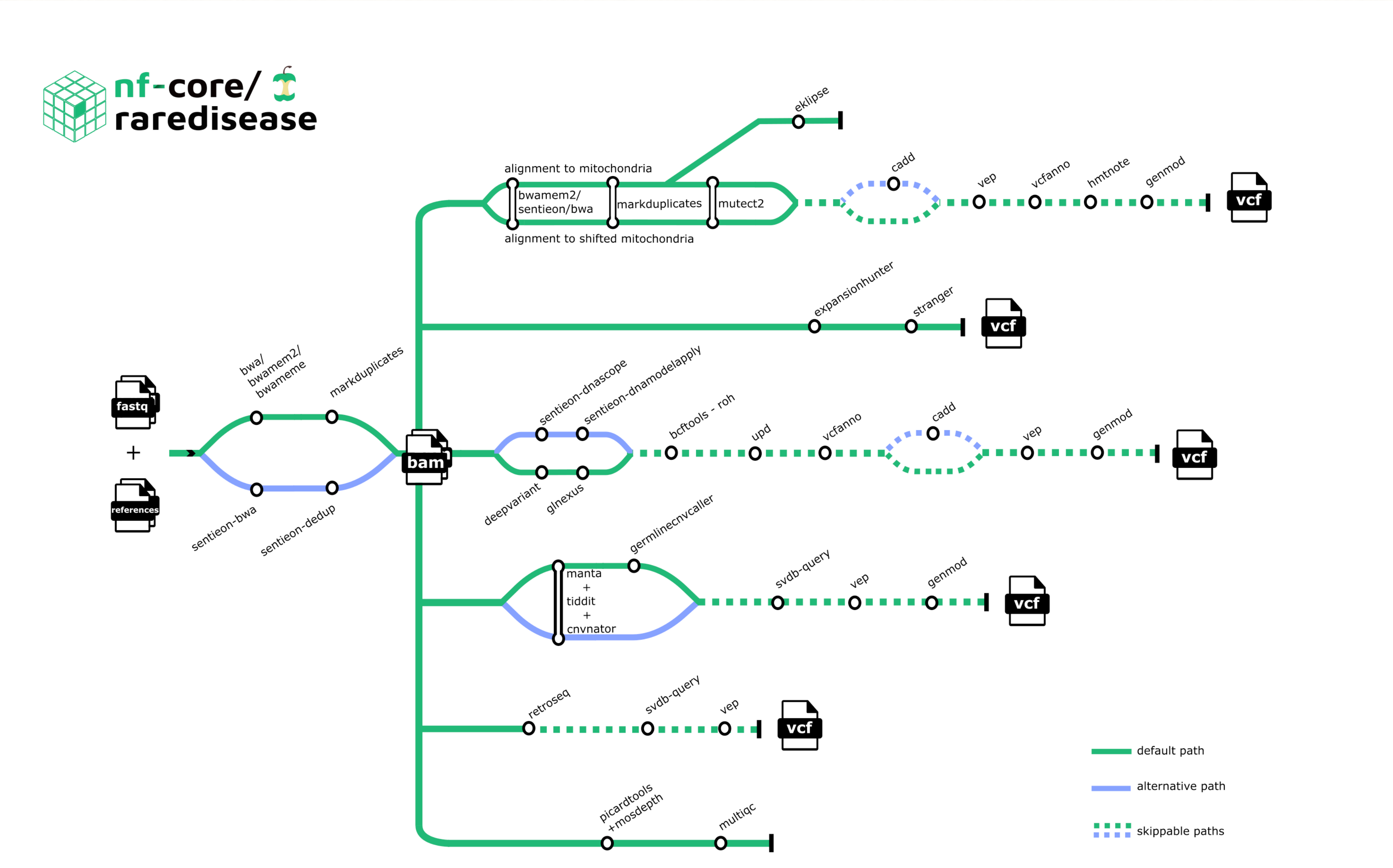
https://nf-co.re/stats

Oslo metro network



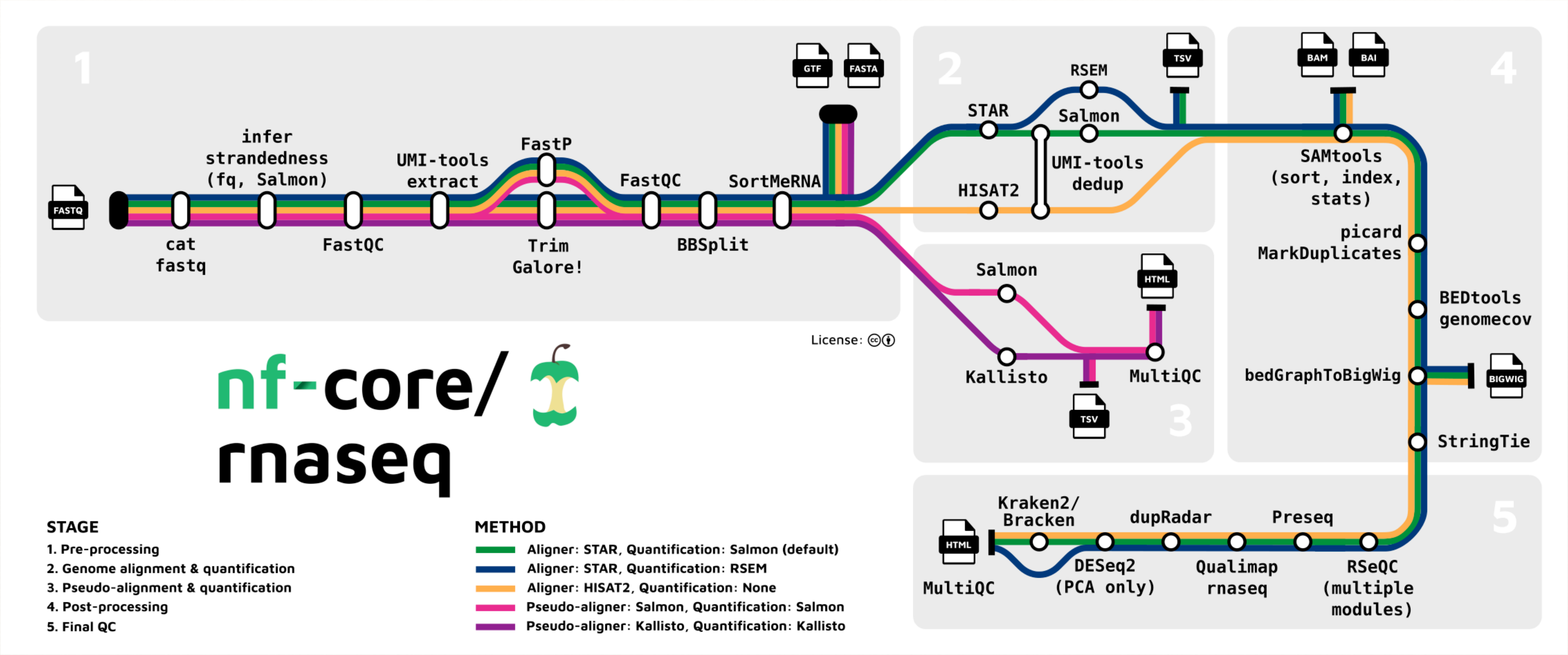
https://en.m.wikipedia.org/wiki/Oslo_Metro

nf-core pipelines



https://nf-co.re/raredisease/2.2.0/

nf-core pipelines

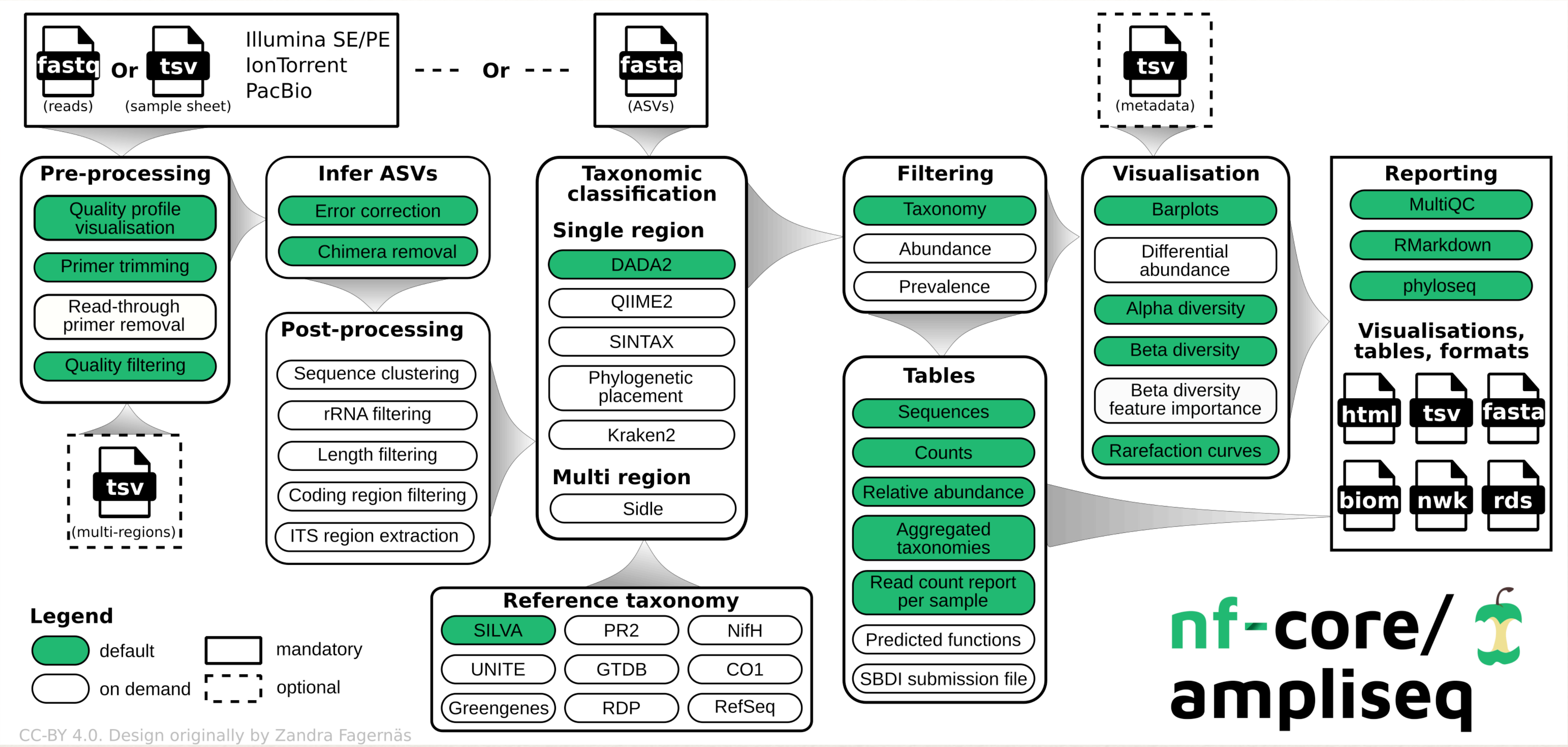


STAGE
1. Pre-processing
2. Genome alignment & quantification
3. Pseudo-alignment & quantification
4. Post-processing
5. Final QC

METHOD
- 🟩 Aligner: STAR, Quantification: Salmon (default)
- 🟦 Aligner: STAR, Quantification: RSEM
- 🟧 Aligner: HISAT2, Quantification: None
- 🟥 Pseudo-aligner: Salmon, Quantification: Salmon
- 🟪 Pseudo-aligner: Kallisto, Quantification: Kallisto

https://nf-co.re/rnaseq/3.17.0/

nf-core pipelines



fastq (reads) Or tsv (sample sheet)
Illumina SE/PE
IonTorrent
PacBio

Or

fasta (ASVs)

tsv (metadata)

Pre-processing
- Quality profile visualisation
- Primer trimming
- Read-through primer removal
- Quality filtering

tsv (multi-regions)

Infer ASVs
- Error correction
- Chimera removal

Post-processing
- Sequence clustering
- rRNA filtering
- Length filtering
- Coding region filtering
- ITS region extraction

Taxonomic classification

Single region
- DADA2
- QIIME2
- SINTAX
- Phylogenetic placement
- Kraken2

Multi region
- Sidle

Reference taxonomy
- SILVA
- PR2
- NifH
- UNITE
- GTDB
- CO1
- Greengenes
- RDP
- RefSeq

Filtering
- Taxonomy
- Abundance
- Prevalence

Tables
- Sequences
- Counts
- Relative abundance
- Aggregated taxonomies
- Read count report per sample
- Predicted functions
- SBDI submission file

Visualisation
- Barplots
- Differential abundance
- Alpha diversity
- Beta diversity
- Beta diversity feature importance
- Rarefaction curves

Reporting
- MultiQC
- RMarkdown
- phyloseq

Visualisations, tables, formats
html tsv fasta
biom nwk rds

Legend
- 🟩 default
- ▢ on demand
- ☐ mandatory
- ⬚ optional

nf-core/
ampliseq

https://nf-co.re/ampliseq/2.11.0/

Quick hands-on

```
cd
mkdir nextflow_rna-seq

cd nextflow_rna-seq

nextflow run nf-core/rnaseq -r 3.17.0 \
          -profile test,singularity \
          —outdir result
```