



Scaling standardisation

EMBL Scientific Workflow Club

Phil Ewels, Seqera



A community fostering collaboration and
building best practices for scientific analysis

Correspondence | Published: 13 February 2020

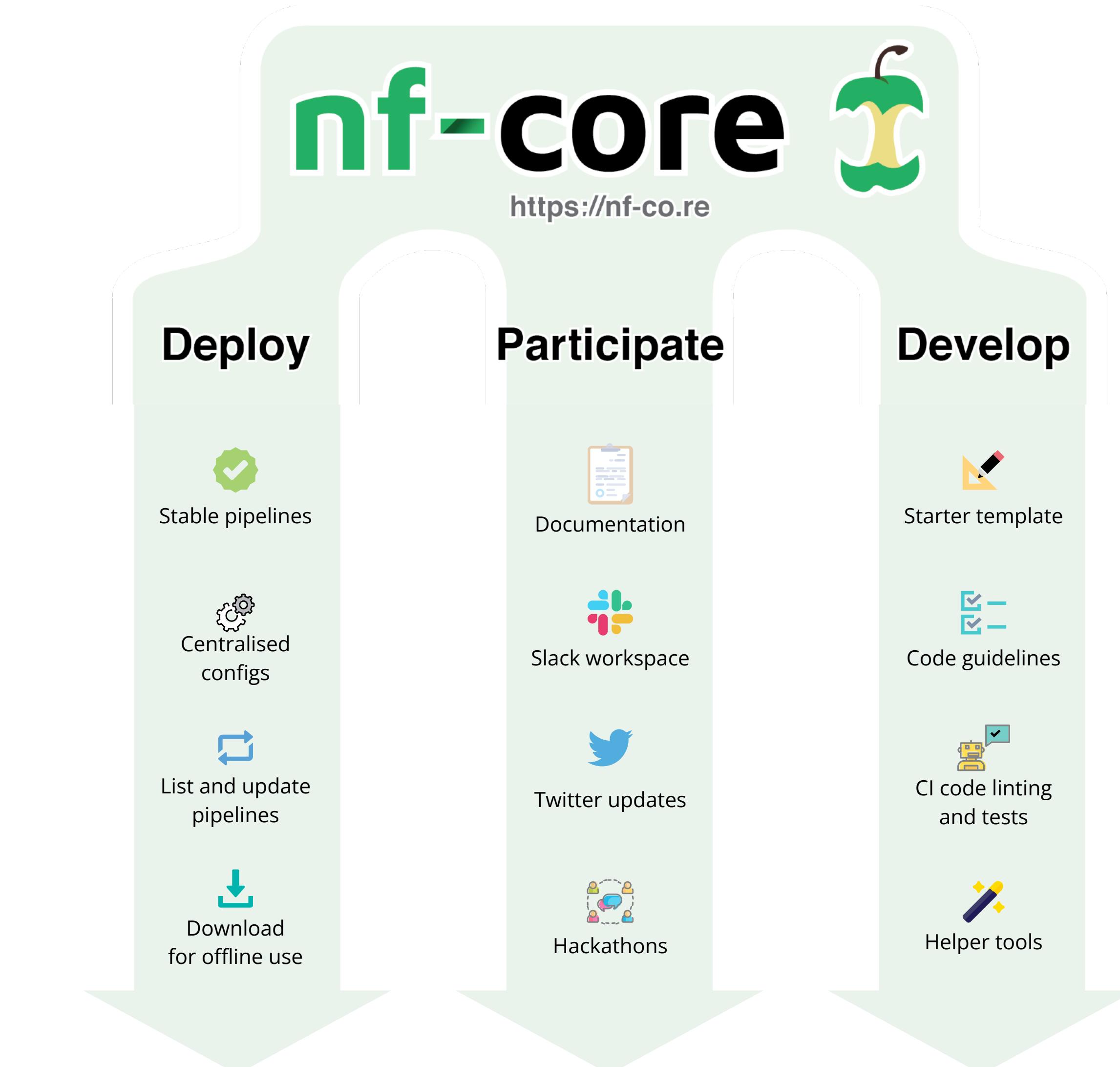
The nf-core framework for community-curated bioinformatics pipelines

Philip A. Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg,
Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso & Sven Nahnsen 

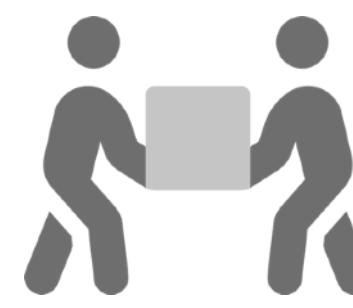
Nature Biotechnology 38, 276–278(2020) | Cite this article

3253 Accesses | 3 Citations | 172 Altmetric | Metrics

To the Editor — The standardization, portability and reproducibility of analysis pipelines are key issues within the bioinformatics community. Most bioinformatics pipelines are designed for use on-premises; as a result, the associated software dependencies and execution logic are likely to be tightly coupled with proprietary computing environments. This can make it difficult or even impossible for others to reproduce the ensuing results, which is a fundamental requirement for the validation of scientific findings. Here, we introduce the nf-core framework as a means for the development of collaborative, peer-reviewed, best-practice analysis pipelines (Fig. 1). All nf-core pipelines are written in Nextflow and so inherit the ability to be executed on most computational infrastructures, as well as having native support for container technologies such as Docker and Singularity. The nf-core community (Supplementary Fig. 1) has developed a suite of tools that automate pipeline creation, testing, deployment and synchronization. Our goal is to provide a framework for high-quality bioinformatics pipelines that can be used across all institutions and research facilities.



Introduction



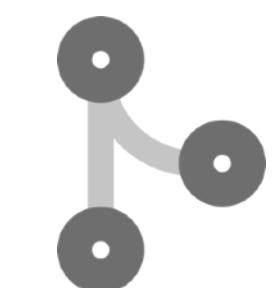
Cooperation

Develop with the community



Standards

Use a common template



Collaboration

Collaborate, don't duplicate



Helper Tools

Tools built for everyone



Compatibility

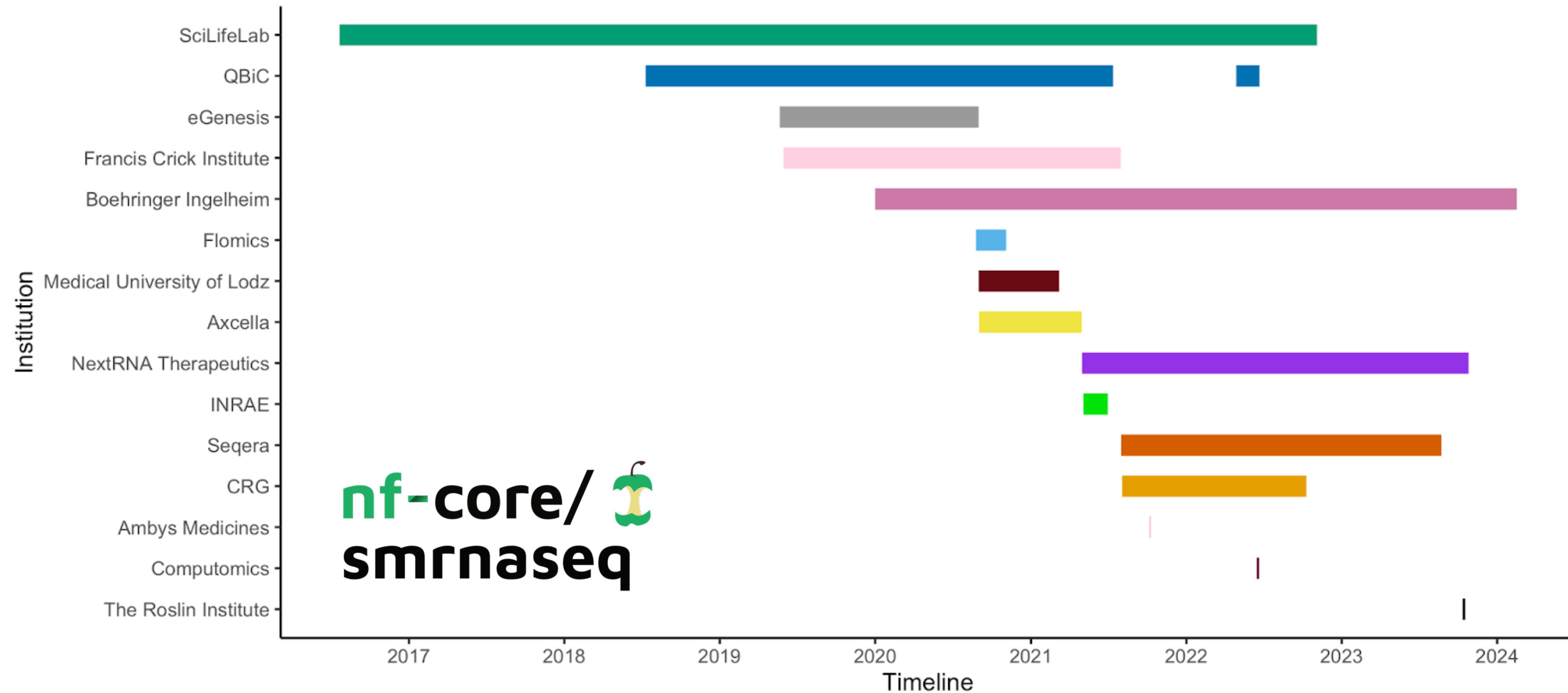
Works with any Nextflow pipeline



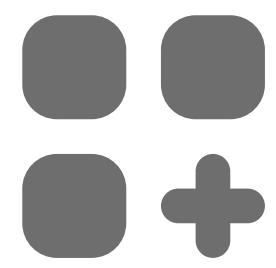
Components

Collaborate on components

Introduction

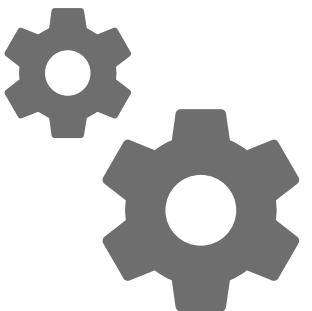


Introduction



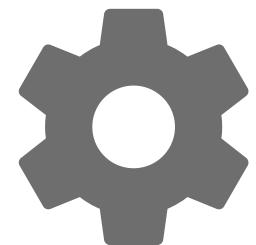
Pipelines

~140 pipelines and a base template



Subworkflows

>80 subworkflows



Modules

>1600 modules



Standardisation

Agreeing on best practices for the community



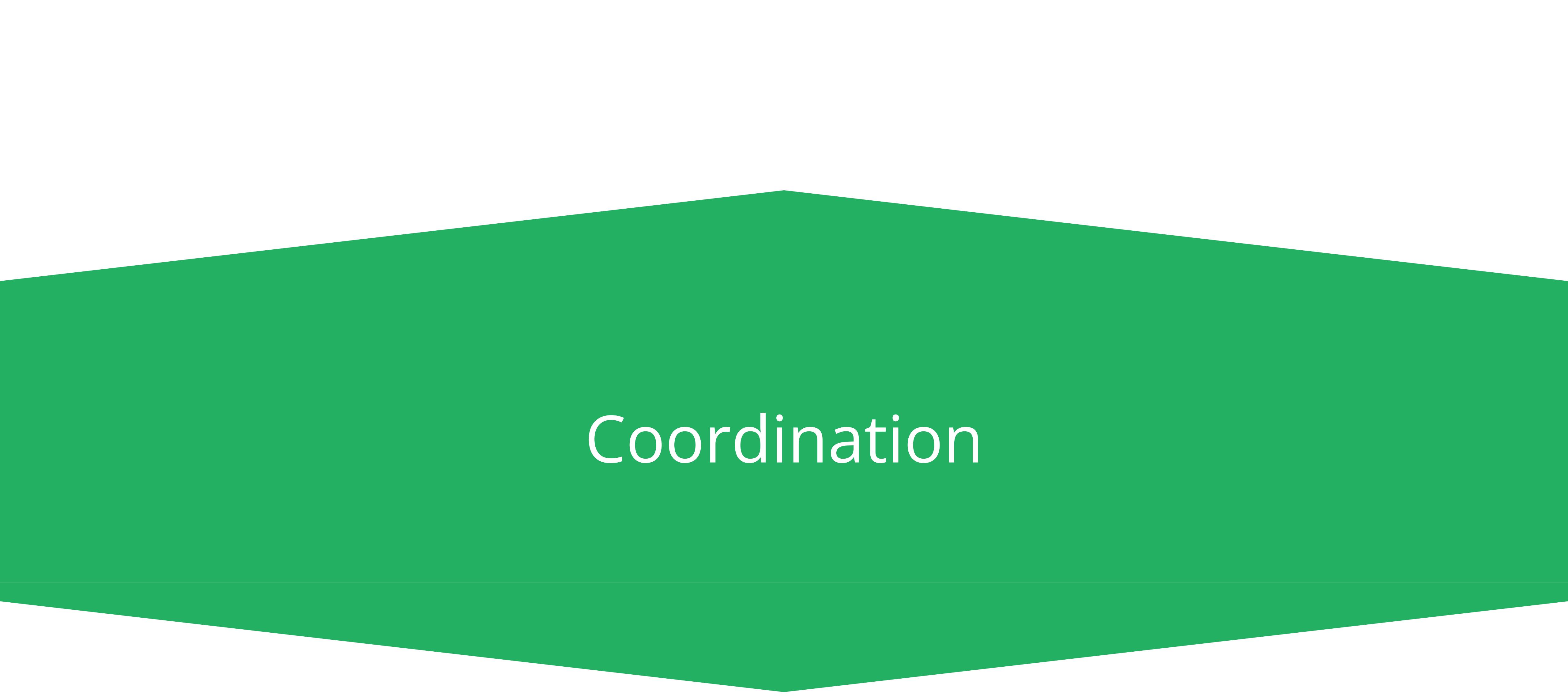
Governance

Coordination, guidelines and regulation



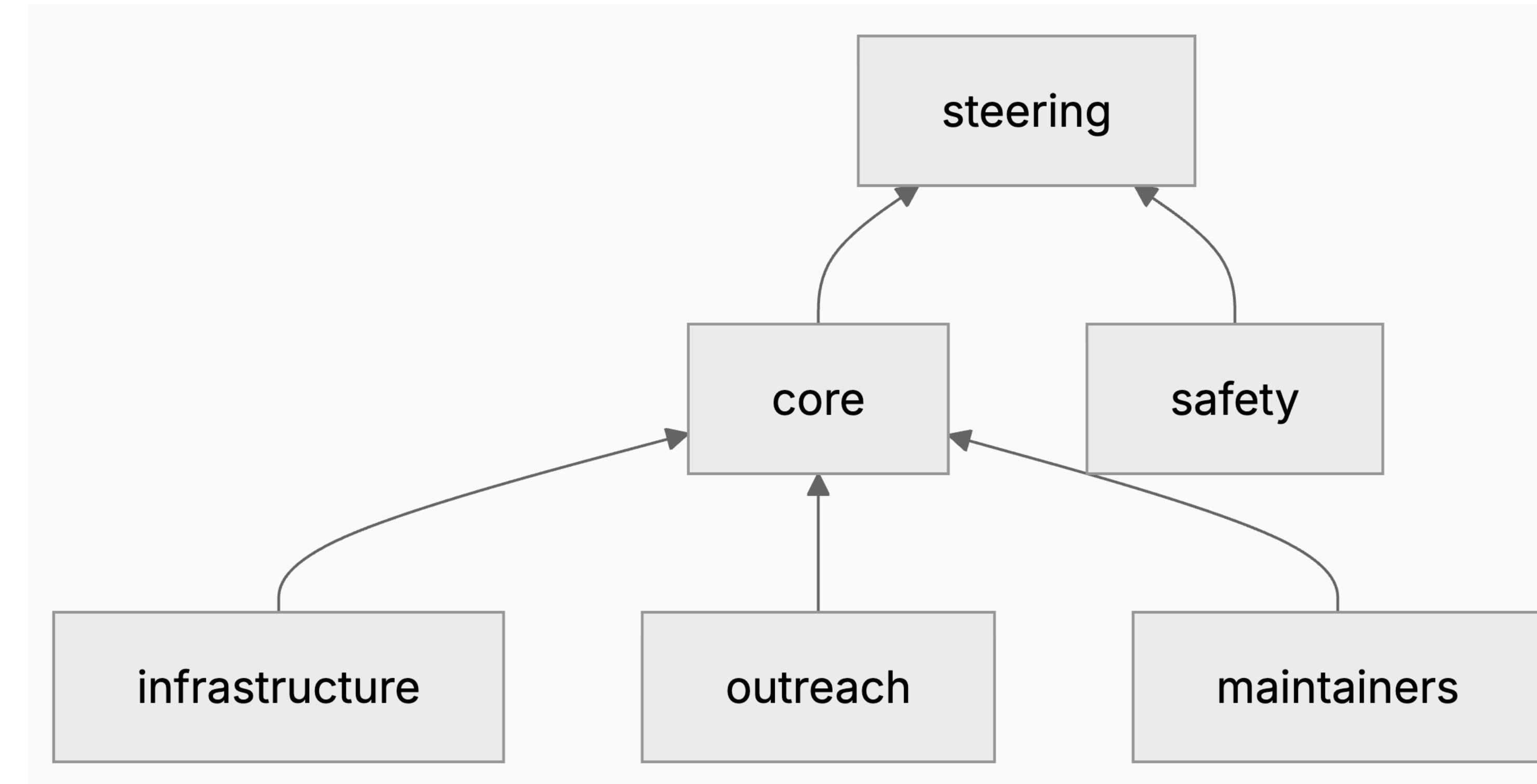
Tooling

Development and deployment



Coordination

Governance



<https://nf-co.re/governance>

Guidelines

Requirements

All nf-core pipelines *must* follow the following guidelines:

- Nextflow: Workflows must be built using Nextflow.
- Community owned: Pipelines are owned by the community.
- Identity and branding: Primary development must be on the nf-core organisation.
- Workflow specificity: There should only be a single pipeline per data / analysis type.
- Workflow size: Not too big, not too small.
- Workflow name: Names should be lower case and without punctuation.
- Use the template: All nf-core pipelines must be built using the nf-core template.
- Software license: Pipelines must open source, released with the MIT license.
- Bundled documentation: Pipeline documentation must be hosted on the nf-core website.
- Docker support: Software must be bundled using Docker and versioned.
- Continuous integration testing: Pipelines must run CI tests.
- Semantic versioning: Pipelines must use stable release tags.
- Standardised parameters: Strive to have standardised usage.
- Single command: Pipelines should run in a single command.
- Keywords: Excellent documentation and GitHub repository keywords.
- Pass lint tests: The pipeline must not have any failures in the `nf-core pipelines lint` tests.
- Credits and Acknowledgements: Pipelines must properly acknowledge prior work.
- Minimum inputs: Pipelines should be able to run with as little input as possible.
- Use nf-core git branches: Use `master`, `dev` and `TEMPLATE`.

Recommendations

All nf-core pipelines *should* follow the following guidelines, if possible / appropriate:

- Use Bioconda: Package software using bioconda and biocontainers.
- File formats: Use community accepted modern file formats such as `CRAM`.
- Testing: Use `nf-test` to test pipeline completes successfully with valid outputs using a minimal example.
- DOIs: Pipelines should have digital object identifiers (DOIs).
- Cloud compatible: Pipelines should be tested on cloud computing environments.
- Publication credit: Pipeline publications should acknowledge the nf-core community.

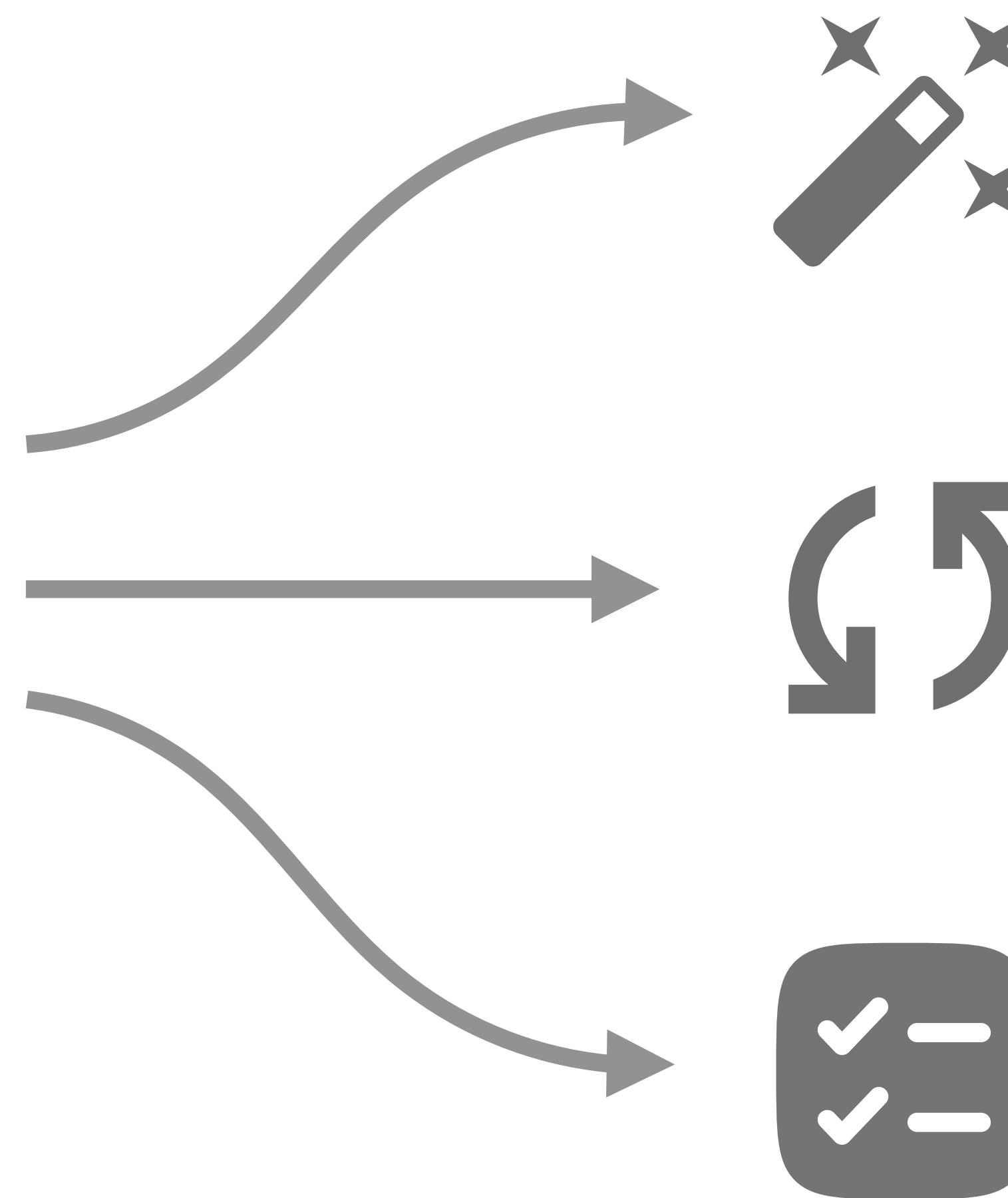
<https://nf-co.re/docs/guidelines/pipelines/overview>

Tooling



Templates

Pipeline template
Module template



Create

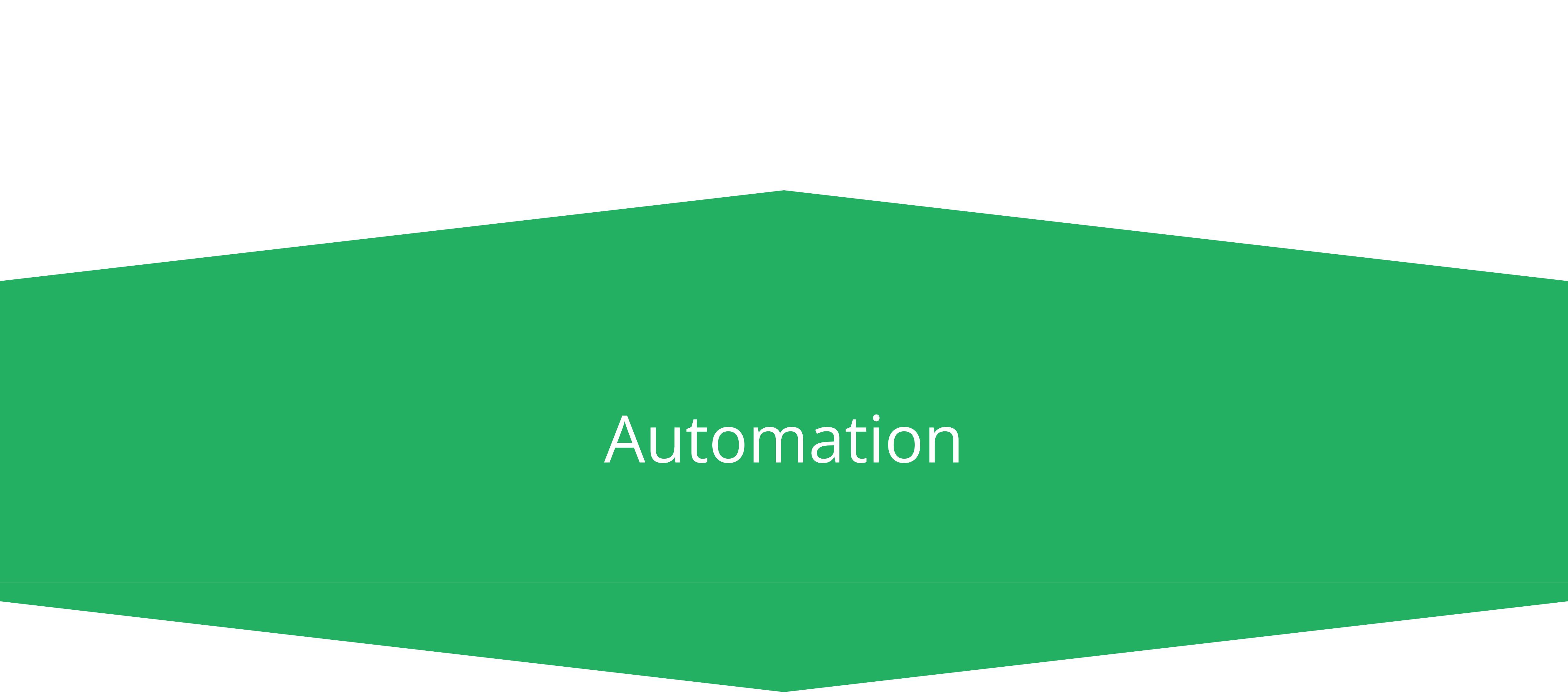
Make new, starting
from our template

Synchronise

Automated updates to pipelines
when the template is updated

Lint

Tests ensure template
is adhered to



Automation

Testing

nextflow run

Does the pipeline crash?

AWS megatests

Ok but does it crash with full-size data?

nf-test

Do the outputs look how I expect?



Testing

`nextflow run
AWS megatests
nf-test`

nf-core lint

Did you adhere to the template?

prettier

Did you format your code nicely?

nextflow lint ➔
soon

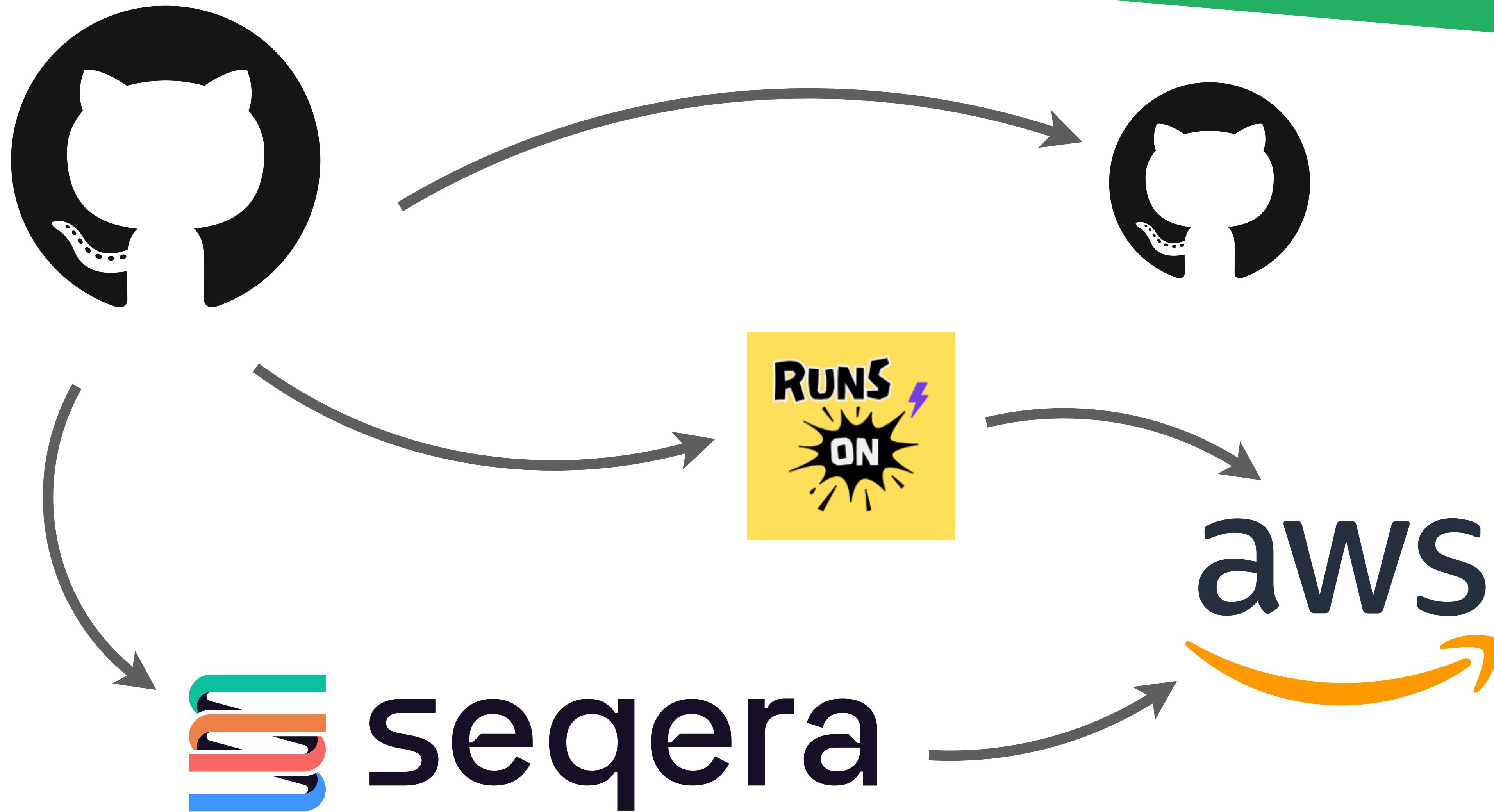
Is your Nextflow code valid / formatted?



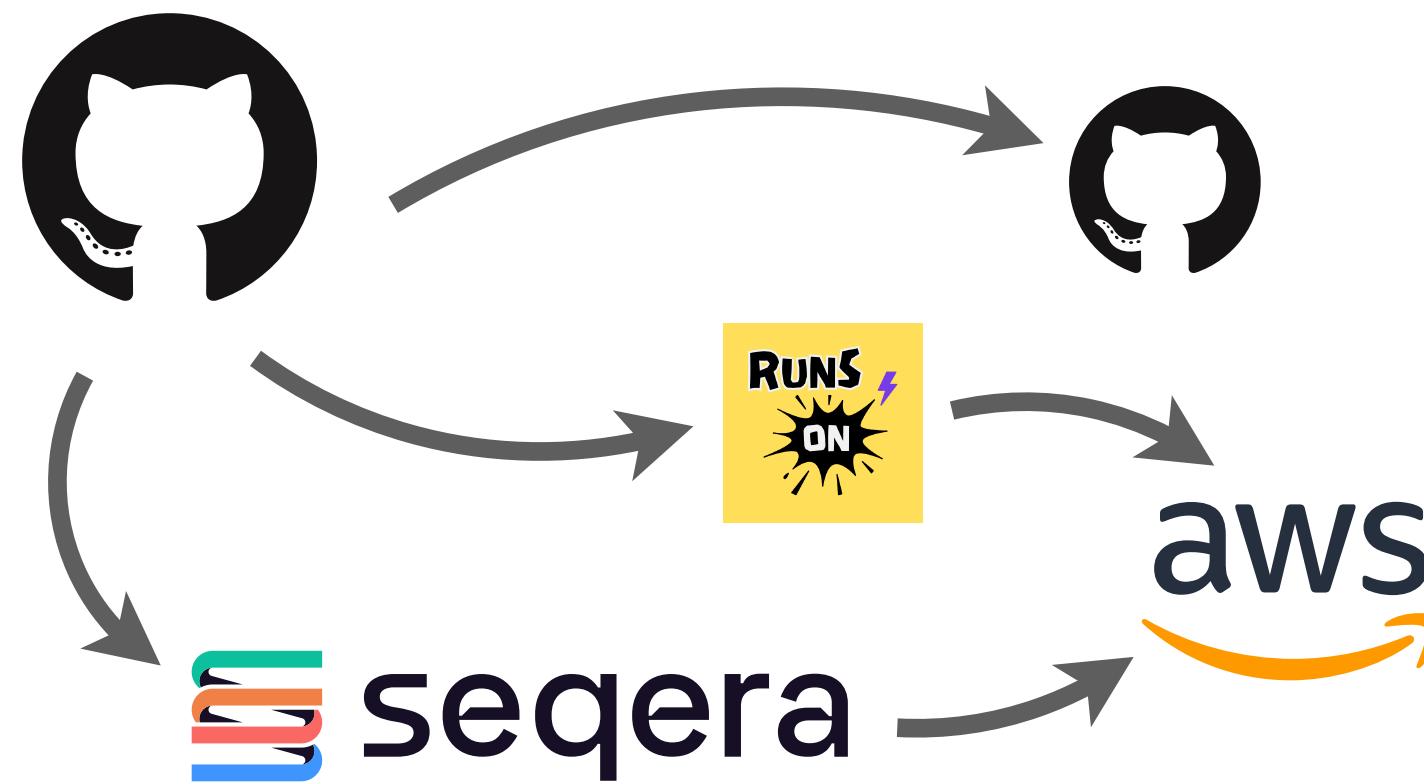
Compute



Compute



Compute



Compute capacity

Big instances for big tests

CPU architecture

linux/aarch64 and linux/arm64

Test sharding

Splitting up tests to run in parallel

Accelerators (GPU)

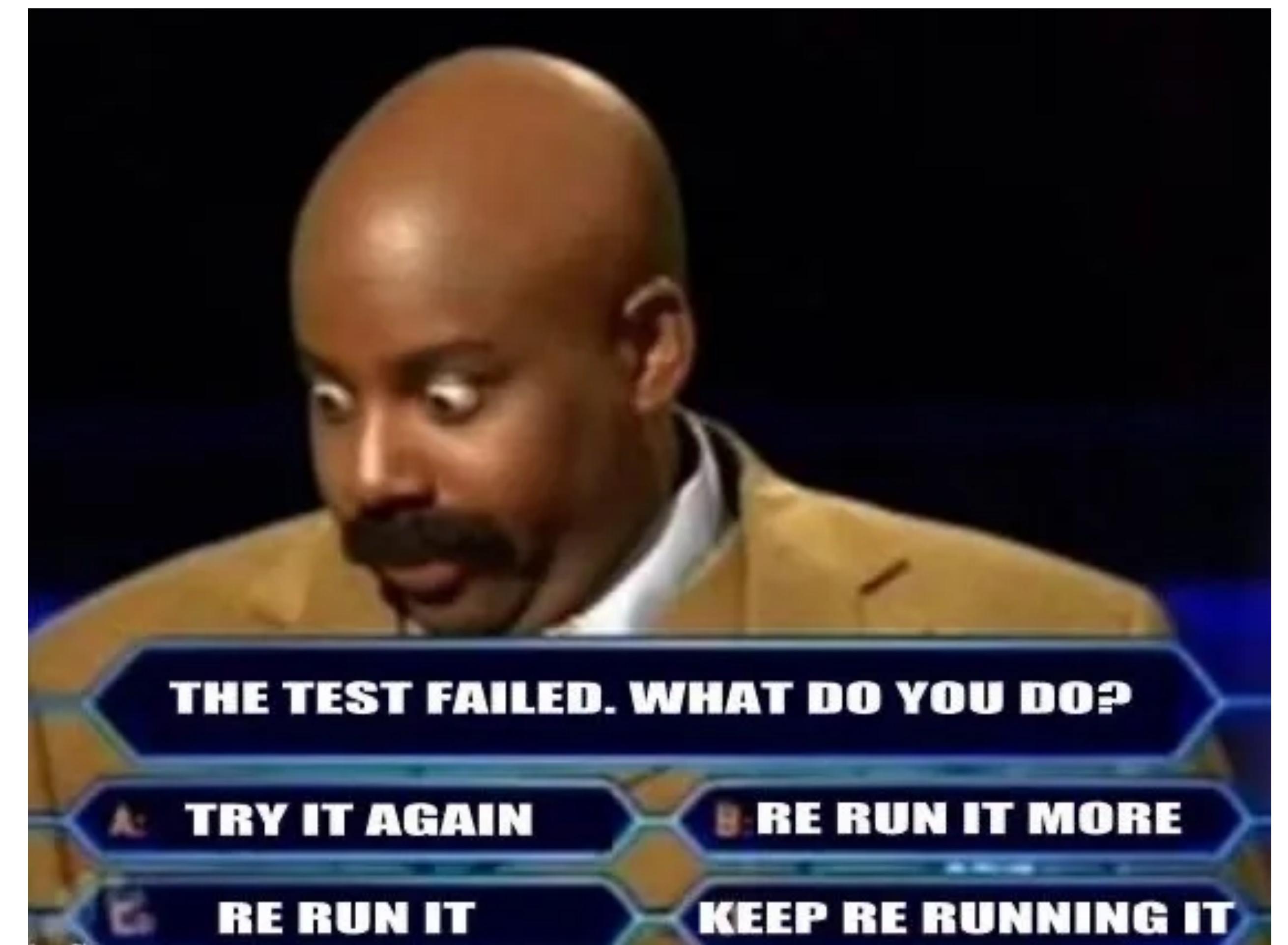
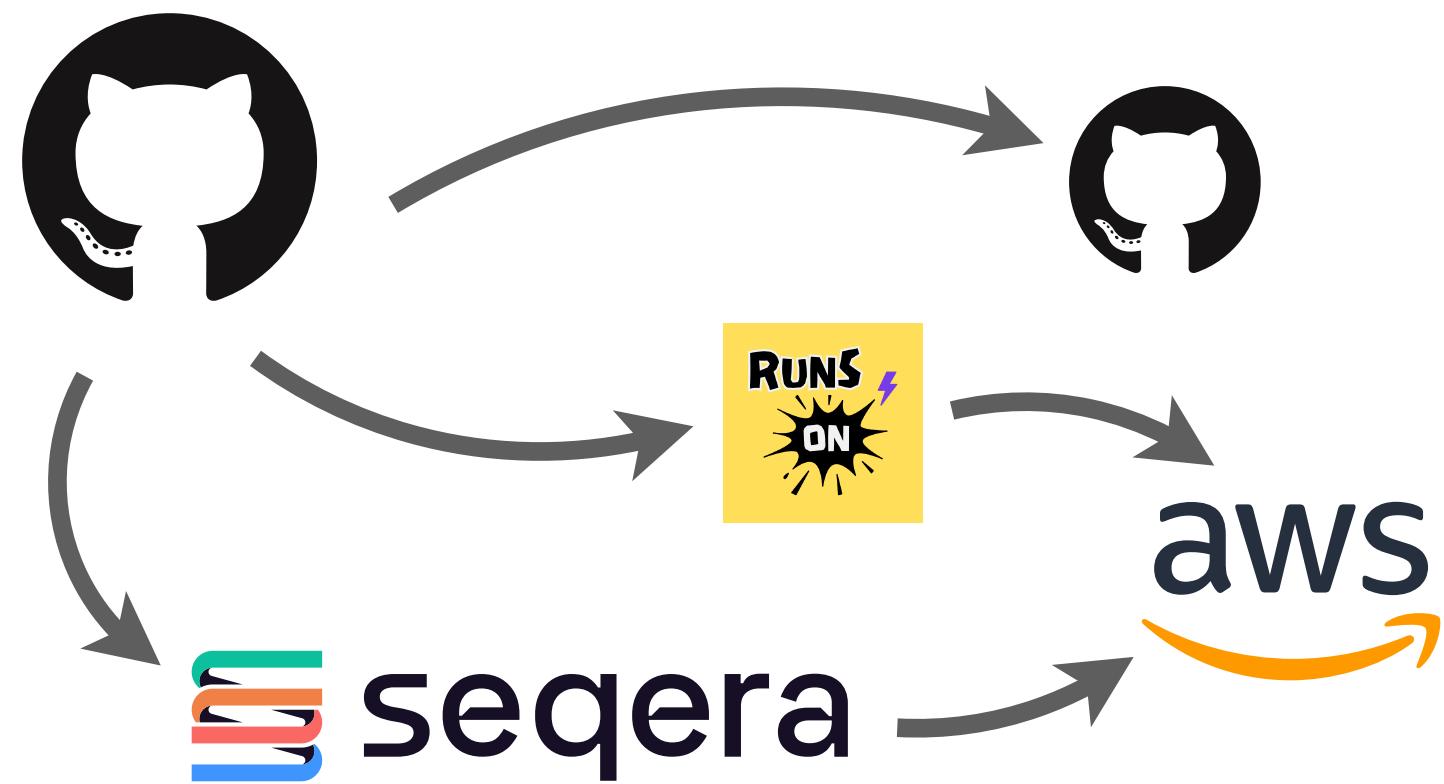
Keeping image analysis folks happy

Instance availability

All the tests, at once!

<https://nf-co.re/blog/2025/state-of-nf-core-ci>

Compute





@nf-core-bot fix linting

Automatically fix Prettier code formatting errors

@nf-core-bot update changelog

Add to CHANGELOG.md

@nf-core-bot update snapshots

Update nf-test / Textualize test snapshot files

nf-core lint --fix

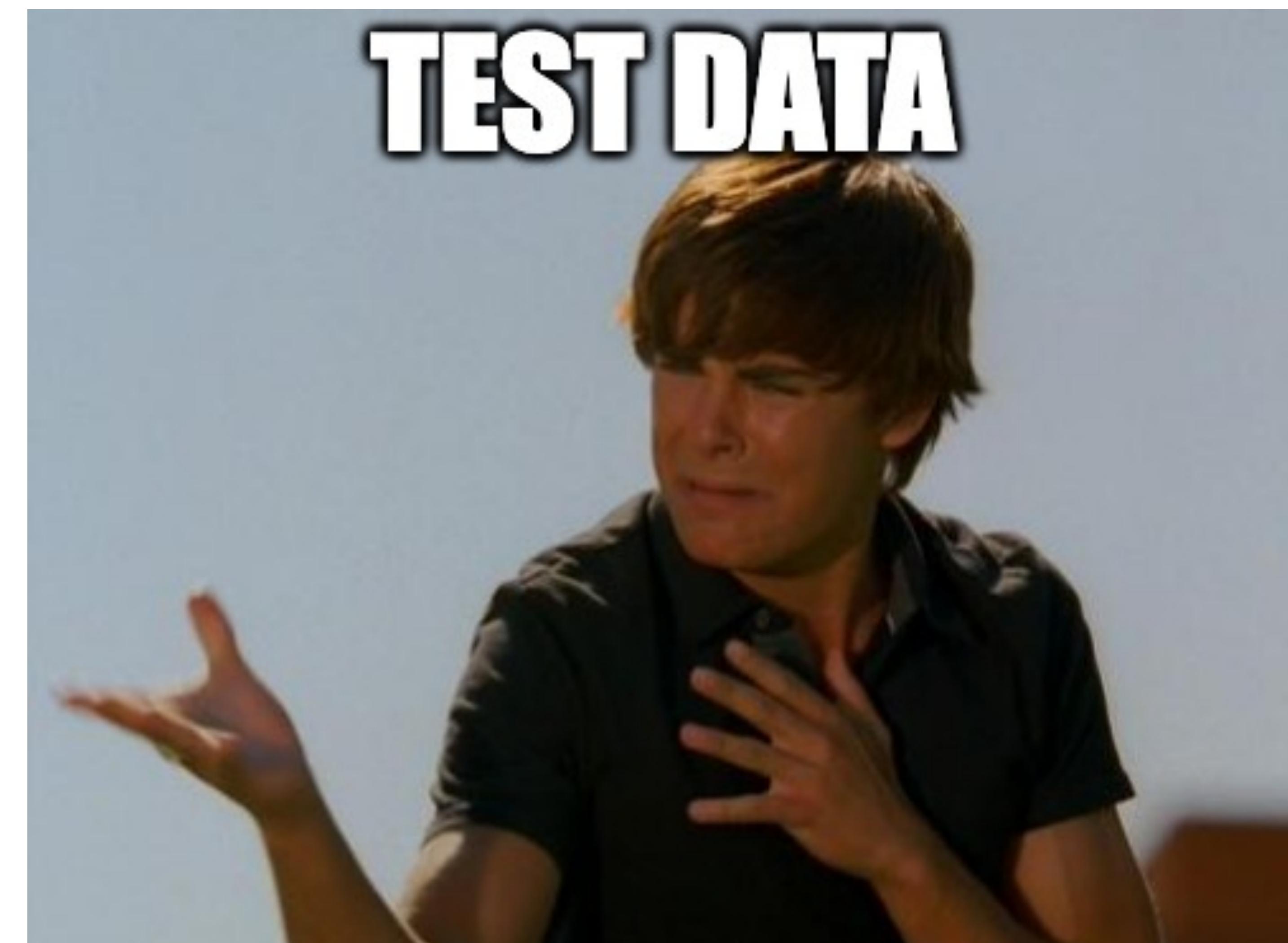
Auto-fix what we can

Pre-commit

Run tests before the code leaves your machine

<https://nf-co.re/docs/contributing/github/nf-core-bot>

Test data



Hosting

nf-core / test-datasets

<> **Code** (Issues 35) Pull requests 15

test-datasets Public

master 183 Branches 1 Tag

Switch branches/tags Find or create a branch...

Branches Tags

- ✓ master default
- msisensor
- porepatrol
- preseq
- proteinannotator
- proteinfamilies



Test data



```
$ nf-core test-datasets list --branch mag --generate-nf-path
```



nf-core/tools version 3.3.2 - <https://nf-co.re>

Nextflow Import

```
params.pipelines_testdata_base_path + "assemblies/MEGAHIT-test_minigut.contigs.fa.gz"
params.pipelines_testdata_base_path + "assemblies/MEGAHIT-test_minigut_sample2.contigs.fa.gz"
params.pipelines_testdata_base_path + "assemblies/SPAdes-test_minigut_contigs.fasta.gz"
params.pipelines_testdata_base_path + "assemblies/SPAdes-test_minigut_sample2_contigs.fasta.gz"
params.pipelines_testdata_base_path + "databases/busco/bacteria_odb10.2024-01-08.tar.gz"
params.pipelines_testdata_base_path + "databases/cat/minigut_cat.tar.gz"
params.pipelines_testdata_base_path + "databases/gtdbtk/gtdbtk_mockup_20250422.tar.gz"
params.pipelines_testdata_base_path + "host_reference/genome.hg38.chr21_10000bp_region.fa"
params.pipelines_testdata_base_path + "samplesheets/assembly_samplesheet.csv"
```

Unit tests

NEXTFLOW

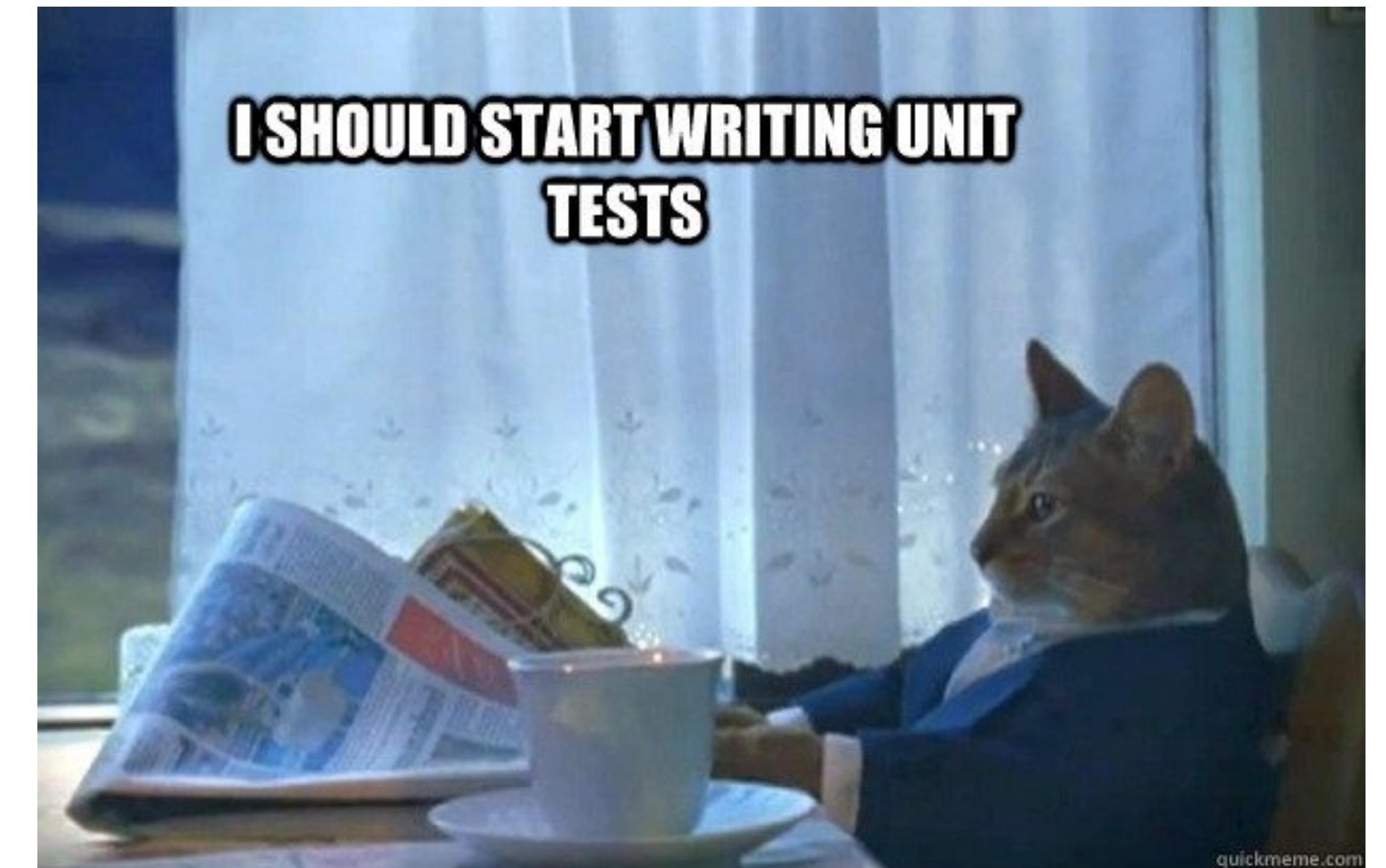
PROJECT

Search processes and wo

List view

Test coverage: 99.22%

- ABACAS
- ABRA2
- ABRICATE_RUN
- ABRICATE_SUMMARY
- ABRITAMR_RUN
- ABUNDANCE_DIFFERENTIAL_FILTER
- ADAPTERREMOVAL
- ADAPTERREMOVALPREFIX
- ADMIXTURE
- AFFY_JUSTRMA
- AGAT_CONVERTBED2GFF



★ Seqera AI

Beta

Conclusions

Conclusions

None of this happens overnight

Pay attention to best practices,
but build only what you need

People are lazy, but want to do the right thing.
Automation and tooling are key

Summit 2025

nextflow SUMMIT

 **Online Nextflow Summit (Free!)**

 **Barcelona Hackathon & Training**

<https://summit.nextflow.io>

Questions

nf-core 