

nf-core 

Elixir workshop

Gisela Gabernet, Friederike Hanssen

Workshop outline

- **Part 1:** Intro to the nf-core project, pipelines from the nf-core project, how to run an nf-core pipeline
- **Part 2:** Intro to nf-core tools: listing pipelines, creating a new pipeline from a template.
- **Part 3:** Intro to nf-core modules: what is a module, the nf-core/modules repositories, adding a module to a pipeline.



Friederike
Hanssen

friederike.hanssen@qbic.uni-tuebingen.de



@FriederikeHanssen



@RikeHanssen



Gisela
Gabernet

gisela.gabernet@qbic.uni-tuebingen.de



@ggabernet



@GGabernet

Part 1

Introduction to nf-core (20 min)

A community effort to collect a curated set of Nextflow pipelines



Stable releases



Packaged software



Documentation



Portable pipelines



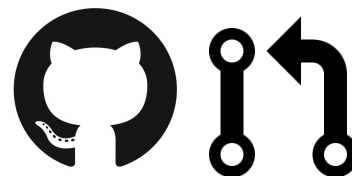
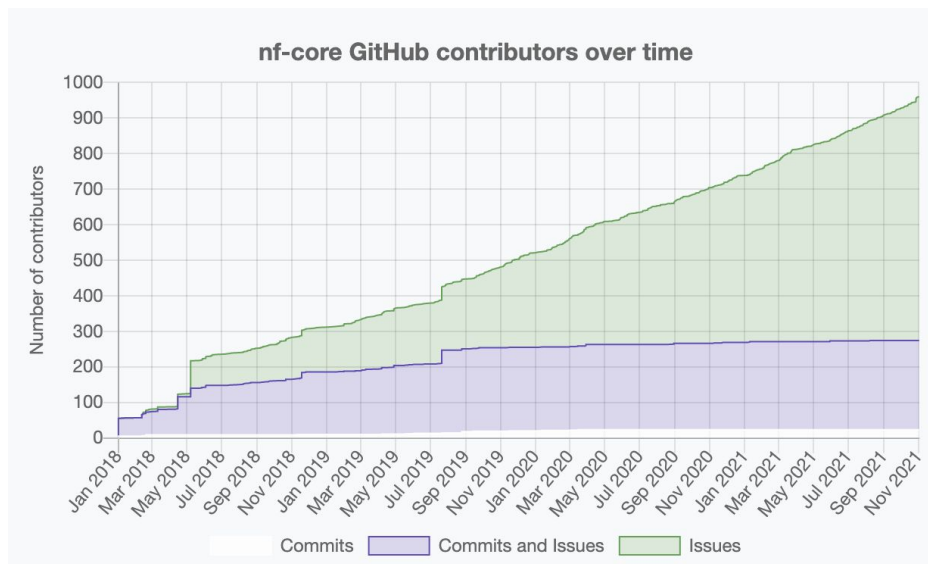
Continuous integration



Cloud ready

Community contributions

Best practice pipelines, peer reviewed by the community



Collaboration over
GitHub



Code review

Contribution guidelines



Develop with the community

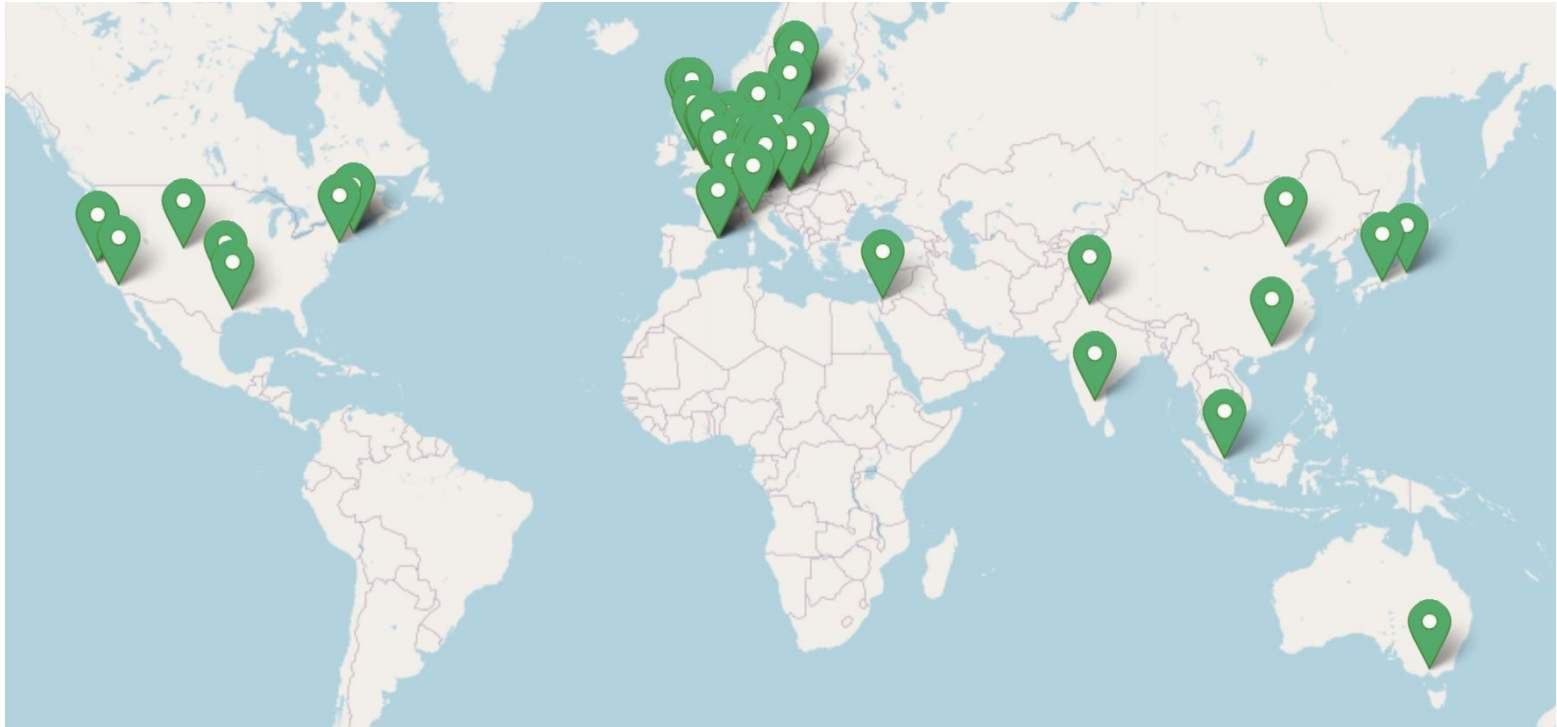
Join slack, communicate and contribute together to a pipeline



Cooperate, don't duplicate

One pipeline per analysis type, contribute by adding new tools, new features...

Organizations using nf-core



Running a pipeline

Prerequisites

- Java
- Nextflow
- Container engine
(Singularity / Docker)
- Unix system
(MacOS / Linux)

nextflow

+



Running a pipeline

- No need to clone a repo or download a pipeline to run it
- Automatically pulled from GitHub + container repository by Nextflow

```
nextflow pull nf-core/<pipeline> -r <version>
```

```
nextflow run nf-core/<pipeline> -r <version>
```

Configuration profiles

Core profiles

```
nextflow run nf-core/<pipeline> -r <version> -profile docker
```

Specify software packaging



Docker



Singularity



Conda

Specify test profile



Running pipeline tests

```
nextflow run nf-core/<pipeline> -r <version> -profile test,docker
```

- Pipeline is executed with a tiny test data set
- So tiny, that you are probably able to run it on your laptop

Summary: running a pipeline

1. Open i.e. a screen session to run pipeline in background / use **-bg**

```
screen -S my_project
```

2. Pull released pipeline version

```
nextflow pull nf-core/<pipeline> -r <ver>
```

3. Run pipeline

```
nextflow run nf-core/<pipeline> -r <ver> -profile <x> --input...>
```

Exercise

- Run the pipeline tests for the nf-core/mag pipeline:

```
nextflow pull nf-core/mag -r 2.1.1
```

```
nextflow run nf-core/mag -r 2.1.1 -profile test,docker
```

Exercise

```
[~ ] process > NFCORE_MAG:MAG:NANOPL0T_RAW -
[~ ] process > NFCORE_MAG:MAG:PORECHOP -
[~ ] process > NFCORE_MAG:MAG:NANOLYSE -
[~ ] process > NFCORE_MAG:MAG:FILTLONG -
[~ ] process > NFCORE_MAG:MAG:NANOPL0T_FILTERED -
[92/a5095a] process > NFCORE_MAG:MAG:CENTRIFUGE_DB_PREPARATI0N [100%] 1 of 1 ✓
[79/e6a7e6] process > NFCORE_MAG:MAG:CENTRIFUGE (test_minigut-m... [100%] 2 of 2 ✓
[24/7aa435] process > NFCORE_MAG:MAG:KRAKEN2_DB_PREPARATI0N [100%] 1 of 1 ✓
[76/d8debc] process > NFCORE_MAG:MAG:KRAKEN2 (test_minigut-mini... [100%] 2 of 2 ✓
[2a/239fc4] process > NFCORE_MAG:MAG:MEGAHIT (test_minigut) [100%] 2 of 2 ✓
[1f/70e216] process > NFCORE_MAG:MAG:SPADES (test_minigut) [100%] 2 of 2 ✓
[~ ] process > NFCORE_MAG:MAG:SPADESHYBRID -
[39/2faf4f] process > NFCORE_MAG:MAG:QUAST (SPAdes-test_minigut) [100%] 4 of 4 ✓
[be/5c1a09] process > NFCORE_MAG:MAG:METABAT2_BINNING:BOWTIE2_A... [100%] 4 of 4 ✓
[ad/076c73] process > NFCORE_MAG:MAG:METABAT2_BINNING:BOWTIE2_A... [100%] 8 of 8 ✓
[43/ce06ce] process > NFCORE_MAG:MAG:METABAT2_BINNING:METABAT2 ... [100%] 4 of 4 ✓
[a3/25f8ed] process > NFCORE_MAG:MAG:METABAT2_BINNING:MAG_DEPTH... [100%] 4 of 4 ✓
[31/8b7d26] process > NFCORE_MAG:MAG:METABAT2_BINNING:MAG_DEPTH... [100%] 4 of 4 ✓
[ec/8ab7ef] process > NFCORE_MAG:MAG:METABAT2_BINNING:MAG_DEPTH... [100%] 1 of 1 ✓
[95/8bd6fb] process > NFCORE_MAG:MAG:BUSCO_QC:BUSCO_DB_PREPARAT... [100%] 1 of 1 ✓
[43/e86378] process > NFCORE_MAG:MAG:BUSCO_QC:BUSCO (SPAdes-tes... [100%] 12 of 12 ✓
[0c/674a6e] process > NFCORE_MAG:MAG:BUSCO_QC:BUSCO_PLOT (MEGAH... [100%] 4 of 4 ✓
[6c/504576] process > NFCORE_MAG:MAG:BUSCO_QC:BUSCO_SUMMARY [100%] 1 of 1 ✓
[37/adf4d5] process > NFCORE_MAG:MAG:QUAST_BINS (MEGAHIT-test_m... [100%] 4 of 4 ✓
[6a/2164f7] process > NFCORE_MAG:MAG:QUAST_BINS_SUMMARY [100%] 1 of 1 ✓
[~ ] process > NFCORE_MAG:MAG:CAT -
[35/932dac] process > NFCORE_MAG:MAG:BIN_SUMMARY (1) [100%] 1 of 1 ✓
[4b/3f61b0] process > NFCORE_MAG:MAG:GET_SOFTWARE_VERSIONS [100%] 1 of 1 ✓
[51/6b9d11] process > NFCORE_MAG:MAG:MULTIQC [ 0%] 0 of 1
```


Part 2

The nf-core tools (30 min)

nf-core/tools: Overview

Helper tools:



```
pip install nf-core
```

```
pip install --upgrade --force-reinstall  
git+https://github.com/nf-core/tools.git@dev
```



```
conda install -c bioconda nf-core
```

- Live demo

```
> nf-core --help

NF-CORE 

nf-core/tools version 2.2.dev0

Usage: nf-core [OPTIONS] COMMAND [ARGS]...

Options:
  --version          Show the version and exit.
  -v, --verbose     Print verbose output to the console.
  -l, --log-file <filename> Save a verbose log to a file.
  --help            Show this message and exit.

Commands:
  list          List available nf-core pipelines with local info.
  launch       Launch a pipeline using a web GUI or command line prompts
  download     Download a pipeline, nf-core/configs and pipeline...
  licences     List software licences for a given workflow.
  create       Create a new pipeline using the nf-core template.
  lint        Check pipeline code against nf-core guidelines.
  modules     Tools to manage Nextflow DSL2 modules as hosted on...
  schema      Suite of tools for developers to manage pipeline schema.
  bump-version Update nf-core pipeline version number.
  sync        Sync a pipeline TEMPLATE branch with the nf-core template
```

Exercise: Install nf-core/tools

- Exercise
 - a. Create a new conda environment
 - b. Install dev version of nf-core/tools
 - c. Check installation by listing all pipelines: `nf-core list`



```
conda create -n nf-core
```

```
conda activate nf-core
```

```
pip install --upgrade --force-reinstall git+https://github.com/nf-core/tools.git@dev
```

```
nf-core list
```

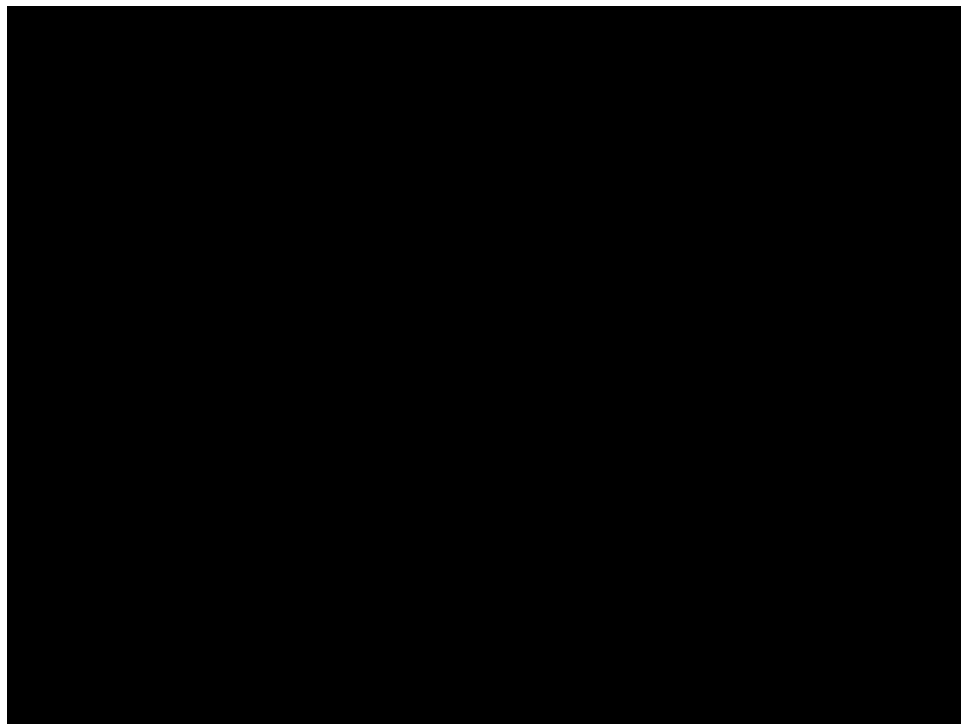


Take a break

nf-core: interactively launch a pipeline

Live demo

`nf-core launch`

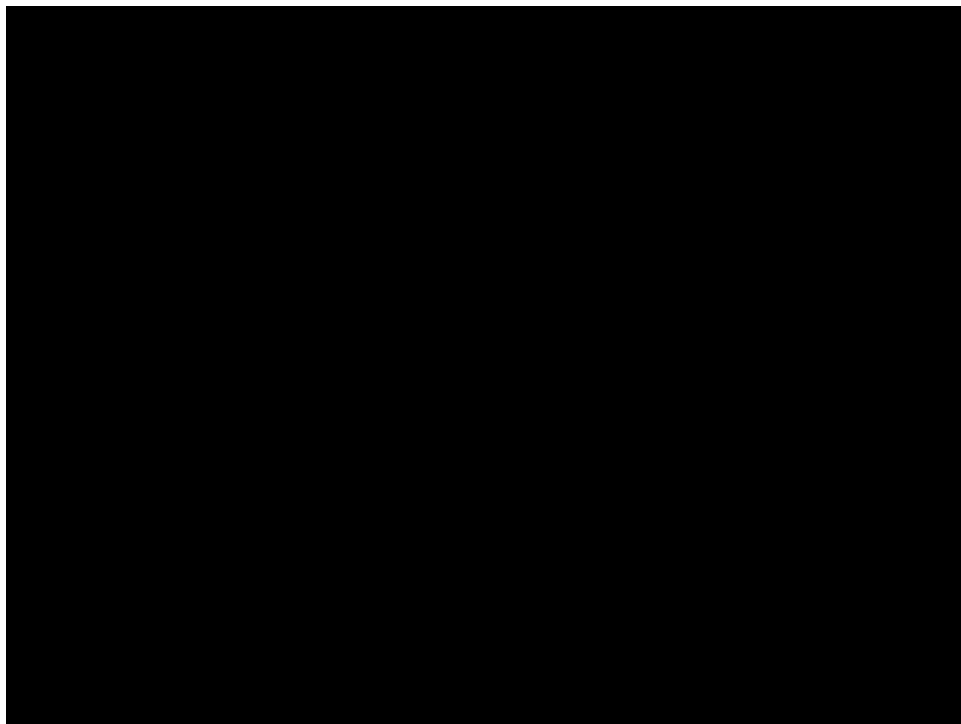


nf-core: creating a pipeline from template

Live demo

Exercise: Create a pipeline from
the template

```
nf-core create
```



nf-core: creating a pipeline from template

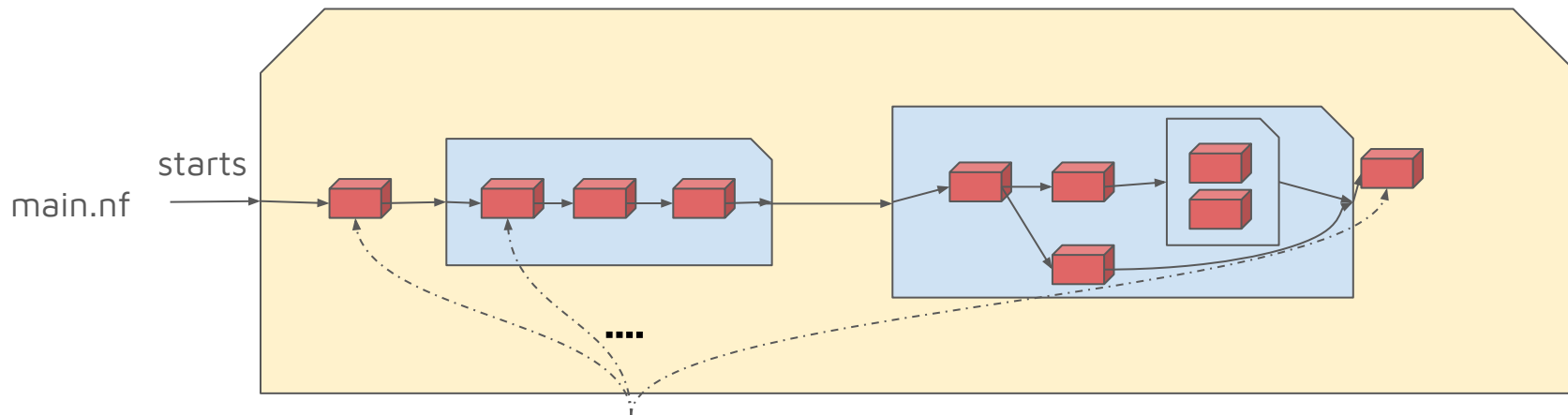
- The template is a functional mini-pipeline with already provided test data

Exercise: Check that the tests pass:

```
nextflow run . -profile test,docker
```

```
executor > local (7)
[d9/c19bb9] process > NFCORE_TESTPIPELINE:TESTPIPELINE:INPUT_CHECK:SAM... [100%] 1 of 1 ✓
[ce/f61c29] process > NFCORE_TESTPIPELINE:TESTPIPELINE:FASTQC (SAMPLE2... [100%] 4 of 4 ✓
[82/250329] process > NFCORE_TESTPIPELINE:TESTPIPELINE:CUSTOM_DUMPSOFT... [100%] 1 of 1 ✓
[94/d2b0dd] process > NFCORE_TESTPIPELINE:TESTPIPELINE:MULTIQC [100%] 1 of 1 ✓
-[nf-core/testpipeline] Pipeline completed successfully-
```

nf-core: pipeline structure



modules.config:
Provides parameters for each tool



Modules: Process that is as atomic as possible i.e. cannot be split into another module



Subworkflows: Collection of modules belonging to i.e. one analysis step



Workflows: Complete analysis runs from raw input to final output

nf-core: pipeline code structure

Detailed overview: [Pipeline code walkthrough \(nf-core/bytesize #3\)](#)

modules.config: Holds parameters for each tool in modules

main.nf: Invoked by nextflow, starts the pipeline

modules: Individually wrapped tools, local or nf-core

subworkflows: collection of modules belonging to one analysis step.

<workflow-name>.nf: contains subworkflows and modules for the pipeline

```
├── CHANGELOG.md
├── CITATIONS.md
├── CODE_OF_CONDUCT.md
├── LICENSE
├── README.md
├── assets
│   └── ...
├── bin
│   └── check_samplesheet.py
├── conf
│   ├── base.config
│   ├── igenomes.config
│   └── modules.config
│       ├── test.config
│       └── test_full.config
├── docs
│   ├── README.md
│   ├── images
│   ├── output.md
│   └── usage.md
├── lib
│   ├── NfcoreSchema.groovy
│   ├── NfcoreTemplate.groovy
│   ├── Utils.groovy
│   ├── WorkflowMain.groovy
│   └── WorkflowTest.groovy
├── nfcore_external_java_deps.jar
├── main.nf
├── modules
│   ├── local
│   └── nf-core
├── modules.json
├── nextflow.config
├── nextflow_schema.json
├── subworkflows
│   └── local
├── workflows
│   └── test.nf
```

What is a Nextflow module?

```
process TRIMGALORE {  
  tag "$meta.id"  
  label 'process_high'  
  
  conda (params.enable_conda ? 'bioconda::trim-galore=0.6.7' : null)  
  container "${ workflow.containerEngine == 'singularity' &&  
!task.ext.singularity_pull_docker_container ?  
  'https://depot.galaxyproject.org/singularity/trim-galore:0.6.7--hdfd78af_0' :  
  'quay.io/biocontainers/trim-galore:0.6.7--hdfd78af_0' }"  
  
  input:  
  tuple val(meta), path(reads)  
  
  output:  
  tuple val(meta), path("*.fq.gz")      , emit: reads  
  path "versions.yml"                    , emit: versions  
  
  script:  
  def args = task.ext.args ?: ''  
  """"  
  trim_galore $options.args --cores $task.cpus --gzip  
  
  trimgalore: \$(echo \$(trim_galore --version 2>&1) | sed 's/^.*.version //;  
  """"  
}
```

Tag, label

Conda env,
container definition

Process input,
output,
script



Take a break

Part 3

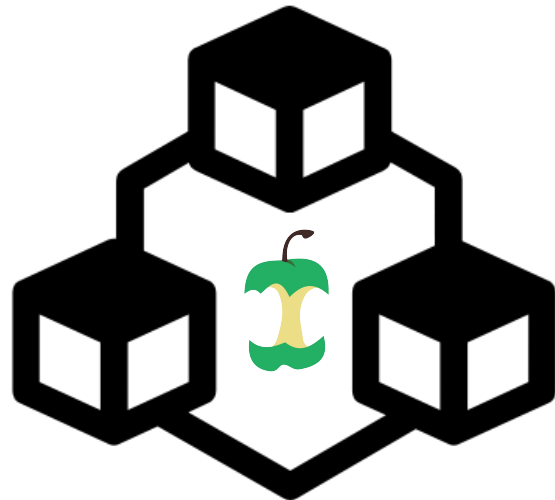
nf-core shared modules (30 min)

nf-core/modules: shared modules

- Shared modules across pipelines
<https://github.com/nf-core/modules>
- Shipped with documentation meta.yml.
- nf-core/modules include CI tests and small test data for all modules.

List available modules in nf-core/modules:

```
nf-core modules list remote
```



Exercise: Add an nf-core module

- Inside the newly created pipeline directory, add a trimgalore module for read adapter trimming.

```
nf-core modules list local
```

```
nf-core modules list remote trimgalore
```

```
nf-core modules install trimgalore
```

- Connect the new module to the rest of the workflow...

Exercise: Add an nf-core module

1. Add include statement for new module in workflow/<pipeline.nf>

```
workflows > elixirpipeline.nf
43  =====
44  */
45
46  //
47  // MODULE: Installed directly from nf-core/modules
48  //
49  include { FASTQC } from '../modules/nf-core/modules/fastqc/main'
50  include { MULTIQC } from '../modules/nf-core/modules/multiqc/main'
51  include { CUSTOM_DUMPSOFTWAREVERSIONS } from '../modules/nf-core/modules/custom/dumpsoftwareversions/main'
52  include { TRIMGALORE } from '../modules/nf-core/modules/trimgalore/main'
53
```

Exercise: Add an nf-core module

2. Add module inside <pipeline> workflow in workflows/<pipeline>.nf

```
workflows > elixirpipeline.nf
69     ...//
70     ... INPUT_CHECK (
71     ...     ... ch_input
72     ... )
73     ... ch_versions = ch_versions.mix(INPUT_CHECK.out.versions)
74
75     ...//
76     ...// MODULE: Run FastQC
77     ...//
78     ... FASTQC (
79     ...     ... INPUT_CHECK.out.reads
80     ... )
81     ... ch_versions = ch_versions.mix(FASTQC.out.versions.first())
82
83     ... TRIMGALORE (
84     ...     ... INPUT_CHECK.out.reads
85     ... )
86     ... ch_versions = ch_versions.mix(TRIMGALORE.out.versions.first())
87
```


Exercise: Add an nf-core module

3. Check modules.config and edit if necessary

```
conf > modules.config
26     ....]
27     ....}
28
29     ....withName: FASTQC {
30     ....     ....ext.args = '--quiet'
31     .... }
32
33     ....withName: TRIMGALORE {
34     ....     ....ext.args = ''
35     .... }
36
37     ....withName: CUSTOM_DUMPSOFTWAREVERSIONS {
38     ....     ....publishDir = [
39     ....         ....path: { "${params.outdir}/pipeline_info" },
40     ....         ....mode: 'copy',
41     ....         ....pattern: '*_versions.yml'
42     ....     ....]
43     .... }
44
45 }
```

Exercise: Add an nf-core module

4. Verify that the trimgalore process is running now by running the pipeline tests.

```
nextflow run . -profile test,docker
```

Exercise: Add an nf-core module

- Solve the warning by adding default parameter values for the new trimgalore parameters in nextflow.config.

```
WARN: Access to undefined parameter `clip_r1` -- Initialise it to a default value eg. `params.clip_r1 = some_value`  
WARN: Access to undefined parameter `clip_r2` -- Initialise it to a default value eg. `params.clip_r2 = some_value`  
WARN: Access to undefined parameter `three_prime_clip_r1` -- Initialise it to a default value eg. `params.three_prime_clip_r1 = some_value`  
WARN: Access to undefined parameter `three_prime_clip_r2` -- Initialise it to a default value eg. `params.three_prime_clip_r2 = some_value`
```

```
nextflow.config  
// Trimgalore options  
17 genome ..... = null  
18 igenomes_base ..... = 's3://ngi-  
19 igenomes_ignore ..... = false  
20  
21 // Trimgalore options You, seconds  
22 clip_r1 ..... = 0  
23 clip_r2 ..... = 0  
24 three_prime_clip_r1 ..... = 0  
25 three_prime_clip_r2 ..... = 0  
26
```

Exercise: Add an nf-core module

- Solve the warning by adding parameters in the nextflow_schema.config.

```
WARN: Found unexpected parameters:  
* --three_prime_clip_r2: 0  
* --three_prime_clip_r1: 0  
* --clip_r2: 0  
* --clip_r1: 0  
- Ignore this warning: params.schema_ignore_params = "three_prime_clip_r2,three_prime_clip_r1,clip_r2,clip_r1"
```

nf-core schema build

Exercise: Add an nf-core module


- Solution: check out the PR (files changed)

<https://github.com/ggabernet/elixir/pull/1>

Bonus: create a new local module

nf-core modules create

```
> nf-core modules create

NF-CORE 

nf-core/tools version 2.2.dev0

INFO     Press enter to use default values (shown in brackets) or type your own responses. ctrl+click underlined text to open links. create.py:82
Name of tool/subtool: samtools/merge
INFO     Using Bioconda package: 'bioconda::samtools=1.14' create.py:145
INFO     Using Docker container: 'quay.io/biocontainers/samtools:1.14--hb421002_0' create.py:171
INFO     Using Singularity container: 'https://depot.galaxyproject.org/singularity/samtools:1.14--hb421002_0' create.py:172
GitHub Username: (@FriederikeHanssen):
INFO     Provide an appropriate resource label for the process, taken from the nf-core pipeline template. create.py:198
For example: process_low, process_medium, process_high, process_long
? Process resource label: process_low
INFO     Where applicable all sample-specific information e.g. 'id', 'single_end', 'read_group' MUST be provided as an input via a Groovy Map called 'meta'. This information may not be required in some instances, for example indexing reference genome files. create.py:212
Will the module require a meta map of sample information? (yes/no) [y/n] (y): y
INFO     Created / edited following files: create.py:250
./modules/local/samtools/merge.nf
```

Bonus: create a new local module

Open `modules/local/samtools/merge.nf`:

```
// TODO nf-core: If in doubt look at other nf-core/modules to see how we are doing things! :) ...
process SAMTOOLS_MERGE {
  tag "$meta.id"
  label 'process_low'

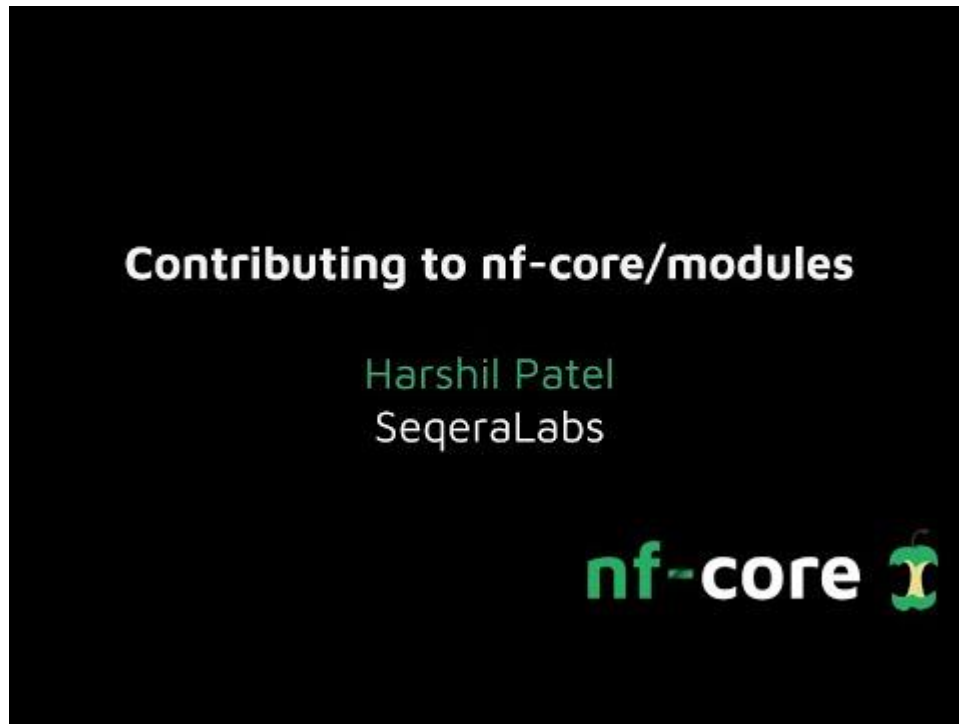
  conda (params.enable_conda ? 'bioconda::samtools=1.14' : null)
  container "${ workflow.containerEngine == 'singularity' && !task.ext.singularity_pull_docker_container ?
    'https://depot.galaxyproject.org/singularity/samtools:1.14--hb421002_0' :
    'quay.io/biocontainers/samtools:1.14--hb421002_0' }"

  input:
  // TODO: Where applicable all sample-specific information e.g. "id", "single_end", "read_group" ...
  tuple val(meta), path(bam)

  output:
  // TODO nf-core: Named file extensions MUST be emitted for ALL output channels ...
  tuple val(meta), path("*.bam"), emit: bam
  path "versions.yml" , emit: versions

  script:
  // TODO nf-core: It MUST be possible to pass additional parameters to the tool as a command-line string via the ...
  def args = task.ext.args ?: ''
  """
  // TODO nf-core: Please replace the example command below with your module's command:
  """
}
```


Bonus: adding a module to nf-core/modules



<https://www.youtube.com/watch?v=xuNYATGFuw4>

More resources

- Usage tutorial: https://nf-co.re/usage/nf_core_tutorial
- 'Tips & Tricks' by Phils Ewels: <https://www.youtube.com/watch?v=BlOtxLqI-hI>
- Bytesize talk series:
<https://www.youtube.com/playlist?list=PL3xpfTVZLcNiSvvPWORbO32S1WDJqKp1e>

Join us!



nf-core 

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

Thanks to our funding sources:



Bonus material

How to configure your pipeline run



Default 'base' config (always loaded)



Core profiles (e.g. singularity, conda, test)



Institutional profiles (nf-core/configs)



Your local config files (-c flag)

Config setup



Default 'base' config:

```
nextflow run nf-core/<pipeline>
```



Automatically loaded



Sensible default resource allocation



No software packaging specified



Runs locally, no job submission

Config setup



Institutional profiles

```
nextflow run nf-core/<pipeline> -profile mycluster
```

 Specifies job submission

 Specify software packaging

Works for:

   For all pipelines

 For all users on your system

 Single point to update

Config setup



Institutional-pipeline specific profiles

```
nextflow run nf-core/<pipeline> -profile mycluster
```

Works for:



Everyone using your system



and this specific pipeline

Config setup



Your local config files

```
nextflow run nf-core/<pipeline> -c myconfig.config
```



Specify resource requirements for your project

Don't have institutional profile yet?

- Max. resources
- Cluster scheduler
- Caching queues
- ...

Add to
nf-core/configs later

Config setup



Your local config files - Personal config



Customize `~/nextflow/config` with:

Email

Tower access token

....



Automatically loaded

Config setup



Your local config files - Parameters file

```
nextflow run nf-core/<pipeline> -params-file file.json
```



Save input parameters as `.json/YAML`



Good practice for reproducibility

➤ Stay tuned for 'Tools': `nf-core launch` (Thursday 1pm, Phil Ewels)

Sort out which configs

- ✓ Default configs
- ✓ Core profiles
- (✓) Institutional profiles
- Local Configs:
 - (✓) Personal config (e.g. email address)
 - (✗) Local config (e.g. project-specific pipeline resources)
 - (✗) Parameters file

