

nf-core Cheat Sheet

Overview

nf-core is a community effort to develop standardized, reproducible bioinformatics pipelines using Nextflow. All pipelines adhere to strict guidelines, use containers, and include extensive documentation.

Getting Started

Installation

Install Nextflow and nf-core tools:

```
# Install Nextflow
curl -s https://get.nextflow.io | bash
# Install nf-core/tools
pip install nf-core
```

Running an nf-core Pipeline

```
nextflow run nf-core/<pipeline> -profile <profile> # e.g., nf-core/rnaseq
nextflow run nf-core/<pipeline> -r <version>      # Specific version
nextflow run nf-core/<pipeline> -resume           # Resume from cache
```

Key Commands

Pipeline Management

```
nf-core list                # List available pipelines
nf-core download <pipeline> # Download pipeline for offline use
nf-core pipelines lint      # Lint pipeline code
nf-core pipelines bump-version 1.2.0 # Update pipeline version
```

Creating a Pipeline

```
nf-core pipelines create --name mypipeline --description "My pipeline" --org nf-core
```

Configuration

Profiles

Use configuration profiles for different environments (e.g., Docker, Singularity, HPC):

```
nextflow run nf-core/<pipeline> -profile docker
nextflow run nf-core/<pipeline> -profile singularity,slurm
```

nextflow.config

Example configuration for containers and resources:

```

process {
  executor = 'slurm' // Options: 'local', 'slurm', 'sge', 'lsf'
  cpus = 4
  memory = '8 GB'
}

docker {
  enabled = true
  runOptions = '-u $(id -u):$(id -g)'
}

singularity {
  enabled = true
  autoMounts = true
  cacheDir = '/path/to/cache'
}

```

Containers

Docker

nf-core pipelines use Docker by default:

```

process example {
  container 'nfcore/<tool>:<version>' // e.g., nfcore/fastqc:0.11.9
  input:
    path input_file
  output:
    path "output/*"
  script:
    """
    fastqc $input_file -o output
    """
}

```

Enable in nextflow.config:

```
docker.enabled = true
```

Singularity

For HPC environments:

```

process example {
  container 'docker://nfcore/<tool>:<version>'
  input:
    path input_file
  output:
    path "output/*"
  script:
    """
    fastqc $input_file -o output
    """
}

```

Enable in nextflow.config:

```
singularity.enabled = true  
singularity.cacheDir = '/path/to/cache'
```

Modules

nf-core provides reusable modules:

```
nf-core modules list          # List available modules  
nf-core modules install <module> # Install module to pipeline
```

Example module inclusion:

```
include { FASTQC } from 'nf-core/modules/fastqc'
```

Testing

Use `nf-test` for pipeline and module testing:

```
nf-core pipelines lint --release # Check pipeline for release  
nf-core modules test <module>   # Test a module
```

Best Practices

- **Use the Template:** Always start with `nf-core pipelines create`.
- **Linting:** Run `nf-core pipelines lint` to ensure compliance.
- **Test Data:** Use minimal test data from `nf-core/test-datasets`.
- **Versioning:** Follow semantic versioning (e.g., 1.0.0).
- **Containers:** Use specific container tags for reproducibility.
- **Contributing:** Fork, branch, and submit PRs via GitHub.

Resources

- nf-core Website
- Nextflow Documentation
- nf-core Slack
- GitHub Repository