tags: `HPCC`, `Nextflow`, `documentation`

Nextflow Job Guide

1. Introduction of Nextflow

> Nextflow is designed to address numerical instability, efficient parallel execution, error tolerance, execution provenance and traceability. It is a domain-specific language that enables rapid pipeline development through the adaptation of existing pipelines written in any scripting language. (Tommaso et al. 2017, Nextflow Documentation)

2. Prepration for Nextflow Installation

Nextflow can be installed and run on any POSIX compatible system (Linux, OS X, etc). It require Bash 3.2 (or later) and Java 8 (or later, up to 17) to be install. * check you bash location and version

```
which bash

bash --version
```

- check you java version

```
which java

java --version
```

3. Nextflow Install

3.1 Use conda install Nextflow (recommended)

conda as a environment and software management tool, will help prepare required environment of install nextflow. In general, HPCC recommend users using conda to install and manage software by themselves.

Nextflow is inside Bioconda. Set up Bioconda accoding to documnetation. * install conda If you haven't install conda, please consider it. Please reference to HPCC conda instructions and some other documents Getting started with conda, Managing environments -conda.

- set up channels

```
conda config --add channels defaults

conda config --add channels bioconda

conda config --add channels conda-forge
```

- `conda install` Nextflow

We recommend to install nextflow in a conda environment, could named nxtfl. You need to activate nxtfl every time run nexflow. For more detail information could check our HPCC conda environment related document.

```
conda create -n nxtfl_env

conda activate nxtfl_env

conda install nextflow
```

Or using one single command: create conda environment and install nextflow

```
conda create -n nxtfl_env nextflow
```

3.2 Or Install Nextflow from distributed package (if you install Nextflow with conda, skip this part)

- download the executable package and create the `nextflow` main executable file in the current directory.

```
wget -qO- https://get.nextflow.io | bash
```

- Make the binary executable on your system

```
chmod +x nextflow
```

- Optionally, move the `nextflow` file to a directory accessible to your `$PATH` variable. This is only required to avoid remembering and typing the full path to `nextflow` each time you need to run it.

```
pwd

# copy the path and paste it in your /.bashrc file:
nano ~/.bashrc
```

3.3 Validate Installation

Run a simple hello script to test Nextflow pipeline. The hello script come with Nextflow installation. After installation process, run following command:

```
nextflow run hello
```

4. Nextflow Job

The simple nextflow script job submission is quite straight forward. Using test/tutorial.nf as an example, please seed Appendix I of the script. More detail about job submissions on our system could refer to Our HPCC Job Submission Guide.

4.1 Non-interactive Job Submission

- write a job script, named sub_tutorial.sh (as below) to submit the job script (named tutorial.nf, Appendix I), which job script include three parts: shebang, SLURM commands and job commands.

```
#!/bin/bash
#SBATCH --job-name=nextflow-test
#SBATCH --output=%x.o%j
#SBATCH --error=%x.e%j
#SBATCH --partition nocona
#SBATCH --nodes=1
#SBATCH --ntasks=1

source /home/yannchen/miniconda3/etc/profile.d/conda.sh

conda activate nxtfl_env

cd ~

nextflow run test/tutorial.nf
```

- submit the job script using `sbatch`

```
sbatch test/sub_tutorial.sh
```

4.2 Interactive Job

- Inqury an interactive session.

```
interactive -p nocona -c 1
```

- Run nextflow job script in the interactive session

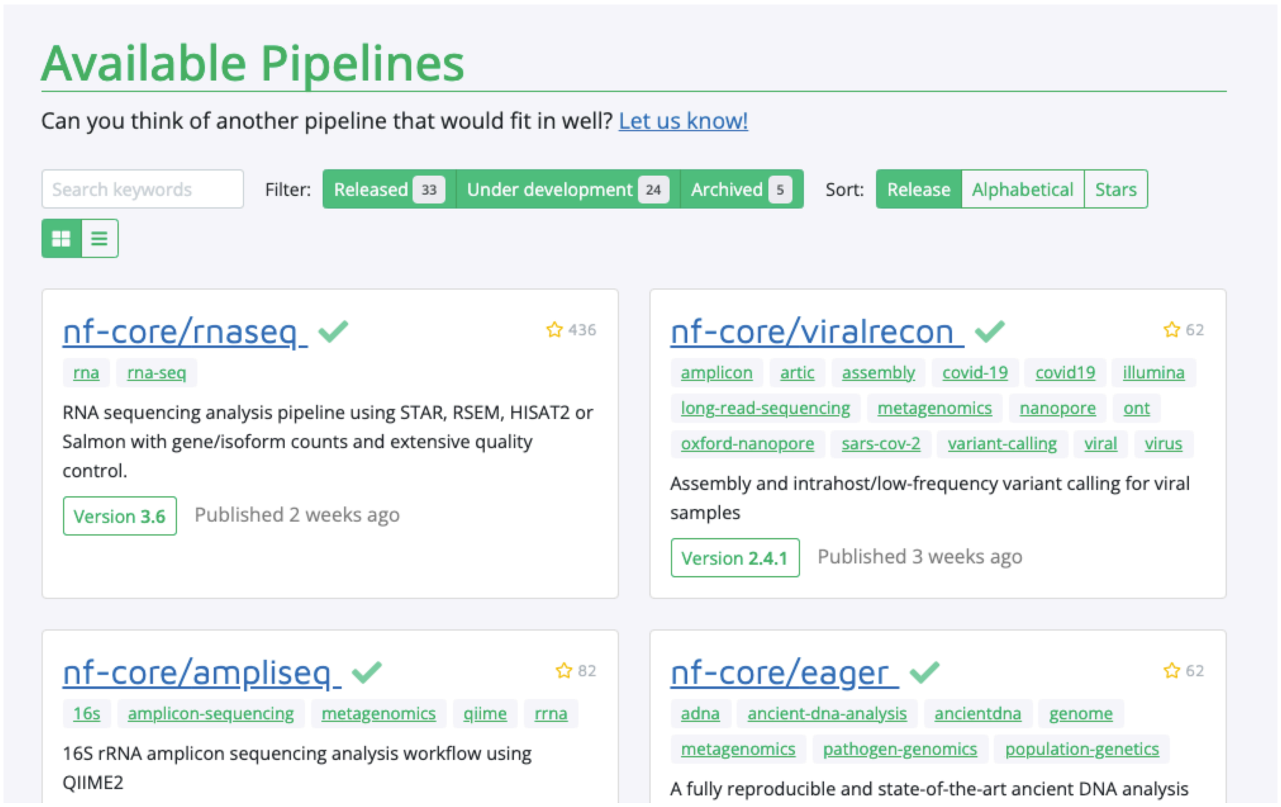
```
conda activate nxtfl_env

nextflow run test/tutorial.nf

nextflow run test/tutorial.nf --str 'Bonjour le monde'
```

5. nf-core pipelines

nf-core is a community effort to collect a curated set of analysis pipelines but using Nextflow. Available pipelines are listed on websites.


Available Pipelines

Can you think of another pipeline that would fit in well? Let us know!

Search keywords
Filter: Released 33 Under development 24 Archived 5
Sort: Release Alphabetical Stars

nf-core/rnaseq ✓ ☆ 436
rna rna-seq
RNA sequencing analysis pipeline using STAR, RSEM, HISAT2 or Salmon with gene/isoform counts and extensive quality control.
Version 3.6 Published 2 weeks ago

nf-core/viralrecon ✓ ☆ 62
amplicon artic assembly covid-19 covid19 illumina long-read-sequencing metagenomics nanopore ont oxford-nanopore sars-cov-2 variant-calling viral virus
Assembly and intrahost/low-frequency variant calling for viral samples
Version 2.4.1 Published 3 weeks ago

nf-core/ampliseq ✓ ☆ 82
16s amplicon-sequencing metagenomics qiime rrna
16S rRNA amplicon sequencing analysis workflow using QIIME2

nf-core/eager ✓ ☆ 62
adna ancient-dna-analysis ancientdna genome metagenomics pathogen-genomics population-genetics
A fully reproducible and state-of-the-art ancient DNA analysis

5.1 Using nf-core/tools to install, mange nf-core pipelines

- install nf-core/tools with conda

```
conda install nf-core
```

- check available pipelines and their status

```
nf-core list
```

- download exist pipelines

> It is highly recommended to use the nf-core download command to pre-download all of the required containers before running the pipeline and to set the NXF*SINGULARITY*CACHEDIR or singularity.cacheDir Nextflow options to be able to store and re-use the images from a central location for future pipeline runs.

The command will triger an user friendly interactive session to help you navigate the download process.

```
nf-core download
```

During the first download, the interactive session will ask about `NXF_SINGULARITY_CACHEDIR` and automatically update your `~/.bashrc`. The `~/.bashrc` will have lines similar to the following.

```
#######################################
## Added by `nf-core download` v2.1 ##
export NXF_SINGULARITY_CACHEDIR="/home/yannchen/nextflow"
#######################################
```

- run downloaded pipelines

```
nextflow run <pipeline>/workflow/ -c <local config file>

nextflow run <pipeline>/workflow/ -c <local config file> -resume

# example of running nf-core/rnaseq pipeline using local config file
nextflow run nf-core-rnaseq-3.4/workflow/ -c test/test.config
```

Currently, we didn't have a standard config file to run nf-core pipelines on the RedRaider cluster. Each running job need to have a local config file. The config file need to include three parts: define input files, enable container, define scheduler. Please see a sample config file for running nf-core/rnaseq (Appendix II).

- use nf-core template to develop pipelines

```
nf-core create
```

6. Appendix

6.1 Appendix I tutorial.nf

```
#!/usr/bin/env nextflow

params.str = 'Hello world!'

process splitLetters {

    output:
    file 'chunk_*' into letters

    """
    printf '${params.str}' | split -b 6 - chunk_
    """
}

process convertToUpper {

    input:
    file x from letters.flatten()

    output:
    stdout result

    """
    cat $x | tr '[a-z]' '[A-Z]'
    """
}

result.view { it.trim() }
```

6.2 Appendix II Sample config file (test.config) of running test run of nf-core/rnaseq pipelines

```
/*
============================================================================
    Nextflow config file for running minimal tests
============================================================================
    Defines input files and everything required to run a pipeline test.
    Use as follows:
        nextflow run nf-core/rnaseq -profile test,<docker/singularity>
----------------------------------------------------------------------------
*/

params {
    config_profile_name        = 'Test profile'
    config_profile_description = 'Minimal test dataset to check function'

    // Limit resources so that this can run on GitHub Actions
    max_cpus   = 2
    max_memory = '6.GB'
    max_time   = '6.h'

    // Input data
    input = 'https://raw.githubusercontent.com/nf-core/test-datasets/\
    rnaseq/samplesheet/v3.4/samplesheet_test.csv'

    // Genome references
    fasta               = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/genome.fa'
    gtf                 = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/genes.gtf.gz'
    gff                 = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/genes.gff.gz'
    transcript_fasta    = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/transcriptome.fasta'
    additional_fasta    = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/gfp.fa.gz'
    bbsplit_fasta_list = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/bbsplit_fasta_list.txt'
    hisat2_index        = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/hisat2.tar.gz'
    star_index          = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/star.tar.gz'
    salmon_index        = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/salmon.tar.gz'
    rsem_index          = 'https://github.com/nf-core/test-datasets/\
    raw/rnaseq/reference/rsem.tar.gz'

    // Other parameters
    skip_bbsplit        = false
    pseudo_aligner      = 'salmon'
    umitools_bc_pattern = 'NNNN'
}

// When using RSEM, remove warning from STAR whilst building tiny indices
process {
    withName: 'RSEM_PREPAREREFERENCE' {
        ext.args2 = "--genomeSAindexNbases 7"
    }
```

```
}

/*
===========================================================================
    Nextflow config file for RedRaider Cluster
===========================================================================
    Defines Singularity, scheduler type and cluster options
---------------------------------------------------------------------------
*/

singularity {
      enabled = true
      cacheDir = "/home/yannchen/nextflow"
}

process {
        executor = 'slurm'
        clusterOptions = '-p nocona -N 1 -n 2'

}
```