TEXAS TECH UNIVERSITY
Information Technology Division™

Introduction to the

# Workflow Management System -- Nextflow & nf-core


Yanni Chen

*Graduate Research Assistant*
*High Performance Computing Center*


*Fall 2022*

# Before Workshop

TEXAS TECH UNIVERSITY
Information Technology Division™

- **Check your conda environment**
  - **Get yourself start with conda**
  - **Manage your conda environment**

- **Pre-workshop survey**

# Target Audience



**This session is intended for an audience who …**

- currently runs, plans to run workflow / pipeline / series analysis
- has to analyze data using different platform (personal computer, lab cluster, HPC, or cloud computing)
- tries to improve reproducibility of data analysis
- writes proposals for workflow development
- is simply curious about workflow management

# Purposes of Training



**Purposes of this training are …**

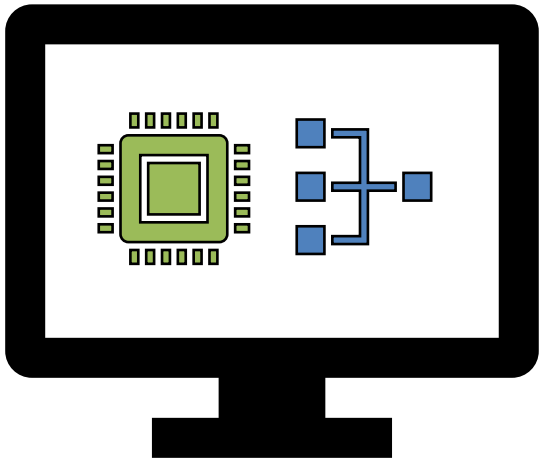
- To introduce workflow management system to cluster users
- To introduce some existing workflows and their development community
- To demonstrate how to run the workflow on RedRaider Cluster

# Outline



- Part I: Workflow Management System
- Part II: Nextflow, data-driven computational pipelines
- Part III: nf-core pipelines, a community curation of bioinfomatic pipelines
- Part IV: Getting Help

# Computation Workflow Management System



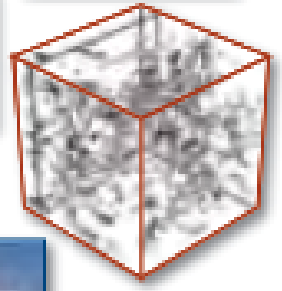
# Part I: Workflow Management System

TEXAS TECH UNIVERSITY
Information Technology Division™

## Science Paradigms

- Thousand years ago:
  science was **empirical**
  *describing natural phenomena*
- Last few hundred years:
  **theoretical** branch
  *using models, generalizations*
- Last few decades:
  a **computational** branch
  *simulating complex phenomena*
- Today: **data exploration** (eScience)
  *unify theory, experiment, and simulation*
  - Data captured by instruments or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files using data management and statistics

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G\rho}{3} - K\frac{c^2}{a^2}$$



Gray 2009

**Fourth paradigm** is using the computational power to generate, curate, analyze, archive a massive scientific data

Liew et al. 2016

# Part I: Workflow Management System

TEXAS TECH UNIVERSITY
Information Technology Division™

## Biological Experiment Sequencing Data Prep and Data Size

## (Illumina NovaSeq 6000)

| Prep | Sequence | Analyze |
|---|---|---|
| 3–5hr | ≤44hr | <25min* |

*Time per sample, assuming 30x coverage/sample

System specifications range

| | |
|---|---|
| Output range | 80 - 6000 Gb |
| Paired end reads per run | 1.6 - 40B |
| Max read length | 2 × 250 bp |
| Run time | 13 - 44 hours |

In a single sequencing run on NovaSeq 6000:

about 4 days library prep and sequencing work could generate 6000 Gb data.

# Part I: Workflow Management System



**Challenges of scientific data explosion**

- Large amount of data are generated from sequencing steps for different experiments
- Data need to go through a series of processing steps based on analysis
- Data processing may be executed on different platforms

# Part I: Workflow Management System



The bioinformatic researchers are trying to automate the computational data processes to assist the data preparation process

# Part I: Workflow Management System

TEXAS TECH UNIVERSITY
Information Technology Division™

## Bioinformatic Data Analysis Pipeline

**Data Integration**

- Read data from storage

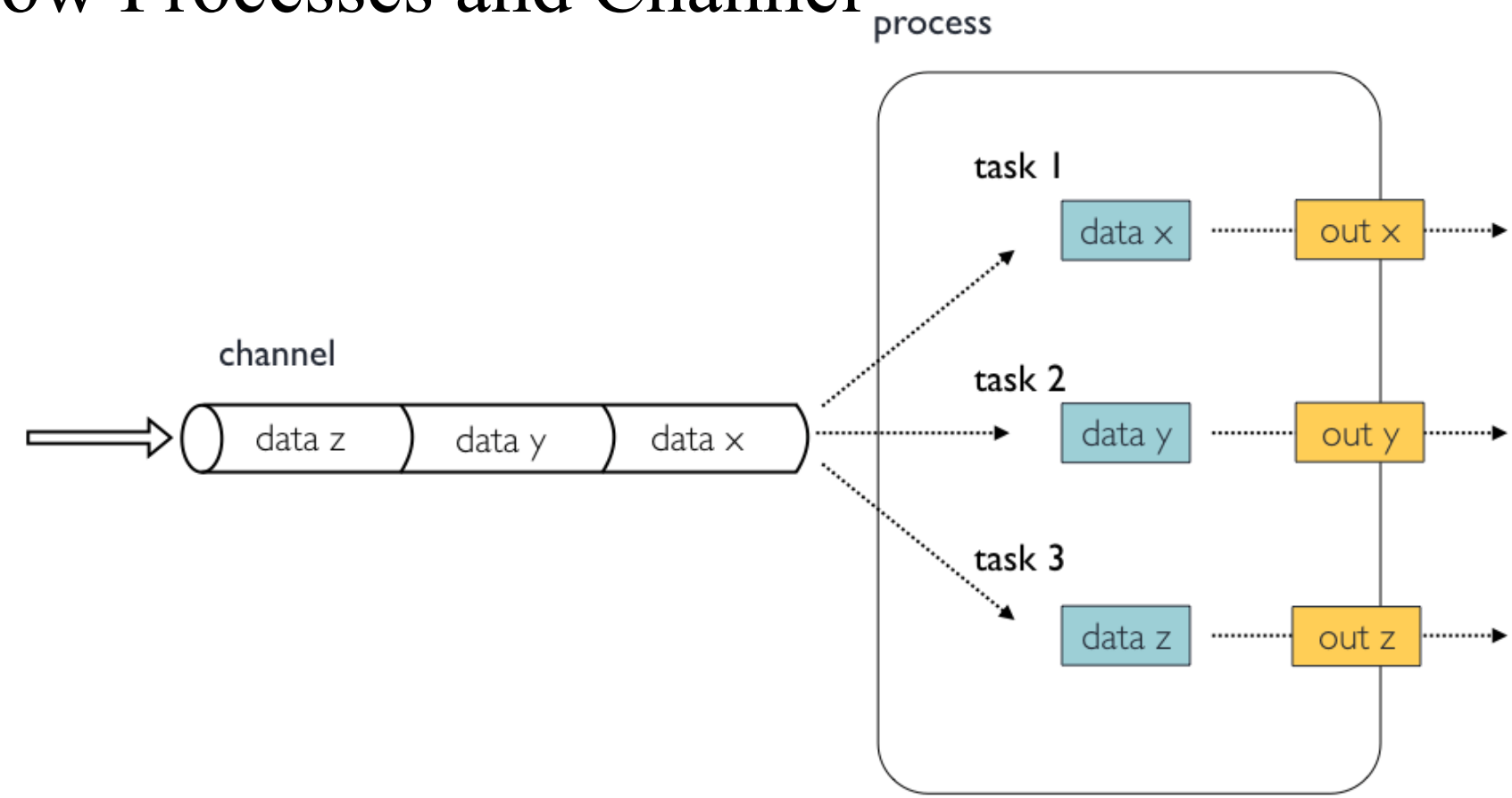
**Data Process**

- Data trimming
- quality control
- data analysis

**Data Summary**

- Storage output

# Nextflow Processes and Channel



# Part I: Workflow Management System



**Advance Solution**

Model biological data as workflows and use a workflow management system to organize their execution.

**Pipeline + HPC infrastructure -> workflow management**

# Part I: Workflow Management System



Dictionary: the sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.

Wikipedia:

Workflows may be viewed as one fundamental building block to be combined with other parts of an organization's structure such as information technology, teams, projects and hierarchies.

# Part I: Workflow Management System



A workflow management system (WfMS) is a software system for setting up, performing, and monitoring of a defined sequence of processes and tasks, with the broad goals of increasing productivity, reducing costs, becoming more agile, and improving information exchange within an organization.

# Part I: Workflow Management System



**Data intensive workflow often involves:**

- Moving data from data sources to computational resources
- Cleaning, calibrating and normalizing data
- Constructing a model using part of that preprocessed data
- Validating the model with the remining data
- Visualizing the results
- Moving the results to a storage system

**Liew et al. 2016**

# Part I: Workflow Management System



**Common roles for workflows**

- Support for collaborative research by enabling scientific communities to share automated and formalized processes such as data analysis
- Construction free from distracting details about workflow management and execution
- The ability to automat workflow steps, that is, their mapping and execution, and to repeat in silicon experiments

**Liew et al. 2016**

# Part I: Workflow Management System



**Common roles for workflows (cont.)**

- Integrating resources from distributed and heterogeneous enactment platforms
- Handling large volumes of data and complex computations
- Improving the execution through various optimization strategies

**Liew et al. 2016**

# The benefit of using workflow

## Use Domain Language

- handling I/O file between flows
- provide standard log file along with analysis

## Work with scheduler

- Each process get its job submission
- Build-in checkpoint

## Work with container

- Reproducibility / version control
- Avoid install conflicts

# Part I: Workflow Management System

TEXAS TECH UNIVERSITY
Information Technology Division™

OPEN ACCESS

EDUCATION

## Using prototyping to choose a bioinformatics workflow management system

Michael Jackson, Kostas Kavoussanakis, Edward W. J. Wallace

Published: February 25, 2021 • https://doi.org/10.1371/journal.pcbi.1008622

Article | Authors | Metrics | Comments | Media Coverage



### Abstract

Workflow management systems represent, manage, and execute multistep computational analyses and offer many benefits to bioinformaticians. They provide a common language for describing analysis workflows, contributing to reproducibility and to building libraries of reusable components. They can support both incremental build and re-entrancy—the ability to selectively re-execute parts of a workflow in the presence of additional inputs or changes in configuration and to resume execution from where a workflow previously stopped. Many workflow management systems enhance portability by supporting the use of containers, high-performance computing (HPC) systems, and clouds. Most importantly, workflow management systems allow bioinformaticians to delegate how their workflows are run to the workflow management system and its developers. This frees the bioinformaticians to focus on what these workflows should do, on their data analyses, and on their science.

# Part II



nextflow

# Part II: Nextflow



A reactive workflow framework

A programming DSL /DSL2 that eases the writing of data-intensive computational pipelines.

## Nextflow Highlights

- Domain specific language
- Work with cluster scheduler
- Error handling
- Reproducibility

# Scripting Language

- Groovy
- Bash
- Python
- R
- Perl

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

params.range = 100

/*
 * A trivial Perl script producing a list of numbers pair
 */
process perlTask {
    output:
    stdout into randNums

    shell:
    '''
    #!/usr/bin/env perl
    use strict;
    use warnings;

    my $count;
    my $range = !{params.range};
    for ($count = 0; $count < 10; $count++) {
        print rand($range) . ', ' . rand($range) . "\n";
    }
    '''
}

/*
 * A Python script task which parses the output of the previous script
 */
process pyTask {
    echo true

    input:
    stdin from randNums

    '''
    #!/usr/bin/env python
    import sys

    x = 0
    y = 0
    lines = 0
    for line in sys.stdin:
        items = line.strip().split(",")
        x = x+ float(items[0])
        y = y+ float(items[1])
        lines = lines+1

    print "avg: %s - %s" % ( x/lines, y/lines )
    '''
}
```

Mixing scripting language of Nextflow:
https://www.nextflow.io/example2.html

# Part II: Nextflow



## HPC Systems

- Slurm
- SGE
- LSE

## Cloud Systems

- AW Batch
- Google Life Science

# Part II: Nextflow

TEXAS TECH UNIVERSITY
Information Technology Division™

Benefit of Work with Different Scheduler:

- adaptability;
- parallelism of jobs:
  - create job channels
  - split job using scheduler

[Scatter Execution of Nextflow Examples]

```
#!/usr/bin/env nextflow
// Author: Bogdan Kirilenko, 2020
// Nextflow procedure to execute chain feature extraction jobs
// Joblist contains a file where each line is a separate command
// We just call these lines in parallel

// params section: basically command line arguments
params.joblist = 'NONE'  // file containing jobs

// if still default -> nothing assigned: show usage message and quit
if (params.joblist == "NONE"){
    printf("Usage: nextflow execute_joblist.nf  --joblist [joblist file] -c
    System.exit(2);
}

// create channel lines -> we need to execute lines in parallel
joblist = file(params.joblist)
lines = Channel.from(joblist.readLines())

process execute_jobs {

    // allow each process to fail 3 times
    errorStrategy 'retry'
    maxRetries 3

    input:
    val line from lines

    "${line}"  // one line represents an independent command
}
```

# Part II: Nextflow



## Version Control

- Docker
- Apptainer / Singualrity
- Shifter
- Podman

# **Exercise #0** Install NextFlow

# **Exercise #0** Install Nextflow (Solution)

## Check prerequisites

Java 8 or later is required

1

Make sure 8 or later is installed
on your computer by using the command:
`java -version`

Note: version numbers "1.8.y_z" and "8" identify the same Java release.

## Set up

Dead easy to install

2

Enter this command in your terminal:
`curl -s https://get.nextflow.io | bash`
(it creates a file `nextflow` in the current dir)

Note: it can also be downloaded from GitHub or installed by using Bioconda package manager.

## Launch

Try a simple demo

3

Run the classic *Hello world*
by entering the following command:
`./nextflow run hello`

# Part II: Nextflow



## **Exercise #0** Install Nextflow (Solution)

- `curl -s https://get.nextflow.io | bash`

Or

- `conda install -c bioconda nextflow`

Test Installation

- `./nextflow run hello`

Or

- `nextflow run hello`

## **Exercise #0** Install Nextflow (Solution)

- `Recommendation:`

`Install Nextflow in your conda environment`

# Part II: Nextflow



**Potential Problem:**

1. nextflow installation needs java 8+ version:
   - `Java -version`
2. conda install needs Bioconda package manager:
   - `conda config --add channels defaults`
   - `conda config --add channels bioconda`
   - `conda config --add channels conda-forge`
3. Install miniconda

# Part II: Nextflow



Your First Script:

- tutorial.nf (DSL2)

```
params.str = 'Hello world!'

process splitLetters {
  output:
    path 'chunk_*'

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

process convertToUpper {
  input:
    path x
  output:
    stdout

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

workflow {
  splitLetters | flatten | convertToUpper | view { it.trim() }
}
```

# Part II: Nextflow

TEXAS TECH UNIVERSITY
Information Technology Division™

Your First Script:

- tutorial.nf (DSL1)

```
#!/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() }
```

# Part II: Nextflow



**Exercise #1** Run first nextflow script

- Copy existing script to personal directory
- Run the script
- Modify script and run

# **Exercise #1** Run first nextflow script (solution)

```
nextflow run 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() }
```

## **Exercise #1** Run first nextflow script (solution)

```
nextflow run tutorial.nf
```

```
nextflow run tutorial.nf
-resume
```

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

```
#!/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() }
```

# Part II: Nextflow



## DSL2

```
nextflow.enable.dsl=2

process foo {
    output:
      path 'foo.txt'
    script:
      """
      your_command > foo.txt
      """
}

 process bar {
    input:
      path x
    output:
      path 'bar.txt'
    script:
      """
      another_command $x > bar.txt
      """
}
```

# Part III



nf-core

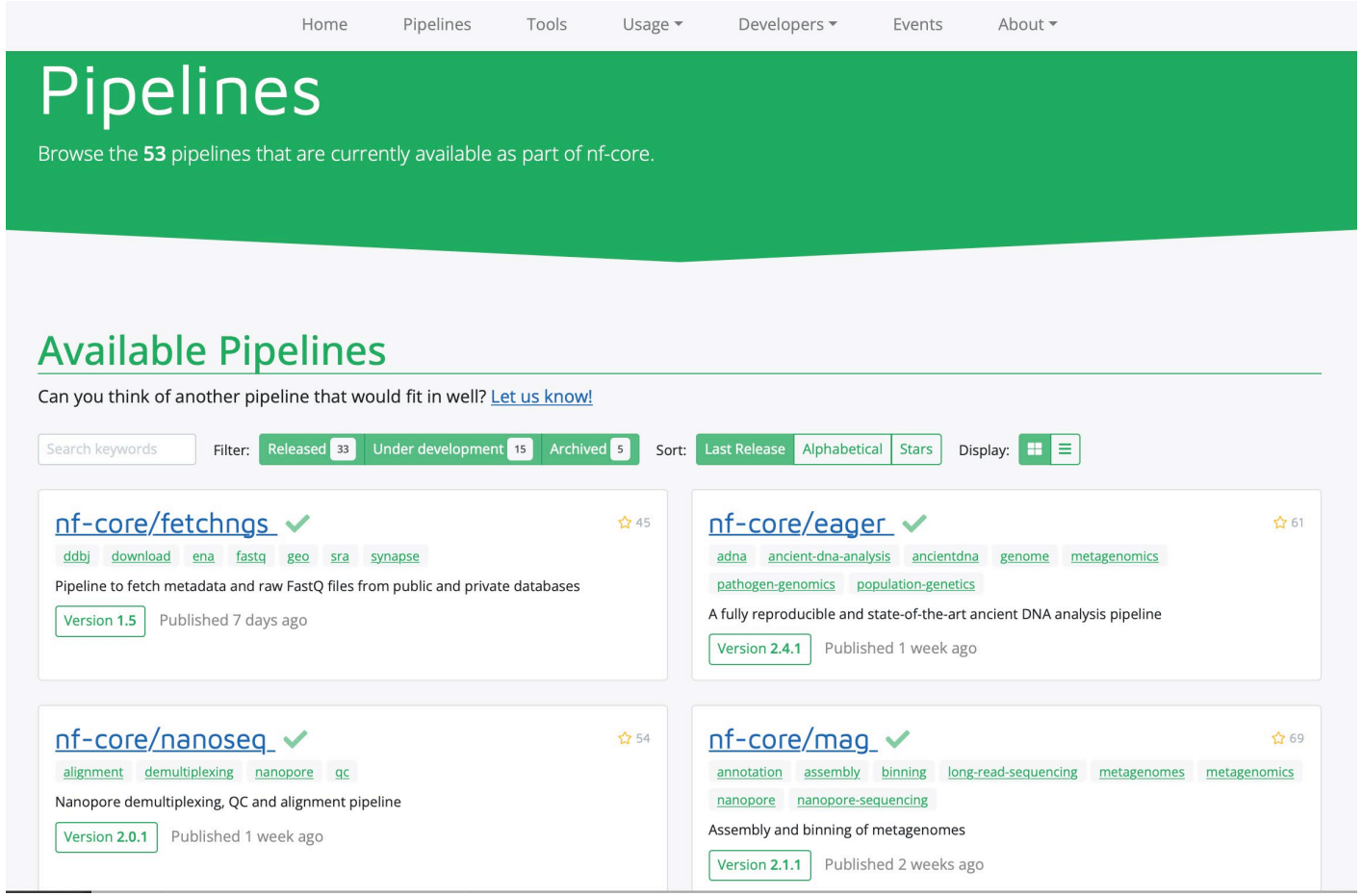
A community effort to collect a curated set of analysis pipelines built using Nextflow.

# Part III: nf-core pipelines

TEXAS TECH UNIVERSITY
Information Technology Division™

## Use nf-core pipelines

check available nf-core pipeline, through [website]



# **Exercise #2** Check available pipelines

- using `nf-core list`

| Pipeline Name | Stars | Latest Release | Released |
|---|---|---|---|
| fetchngs | 45 | 1.5 | 6 days ago |
| eager | 61 | 2.4.1 | 1 week ago |
| nanoseq | 54 | 2.0.1 | 1 week ago |
| mag | 69 | 2.1.1 | 2 weeks ago |
| cutandrun | 16 | 1.0.0 | 1 months ago |
| ampliseq | 75 | 2.1.1 | 1 months ago |
| rnaseq | 399 | 3.4 | 2 months ago |
| mhcquant | 16 | 2.0.0 | 3 months ago |
| bacass | 33 | 2.0.0 | 3 months ago |
| viralrecon | 51 | 2.2 | 4 months ago |
| bcellmagic | 16 | 2.0.0 | 5 months ago |
| bactmap | 25 | 1.0.0 | 6 months ago |
| smrnaseq | 31 | 1.1.0 | 6 months ago |
| sarek | 129 | 2.7.1 | 6 months ago |
| hic | 32 | 1.3.0 | 7 months ago |

**Exercise #2** Check available pipelines (Solution)

- **Install nf-core**
  - `conda install nf-core` or
  - `conda update nf-core`
- **list all pipelines**
  - `nf-core list`

# Part III: nf-core pipelines



## Run nf-core pipelines on RedRaider Cluster

- Ask scheduler for an interactive session
- Download nf-core pipelines
- Adapt the config file
  - Enable singularity container
  - Define the executor
- Run the job

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

or

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

# Config File

- Enable singularity container

```
singularity {
enabled = true
cacheDir = ”<singularity image location>" }
```

- Define the executor

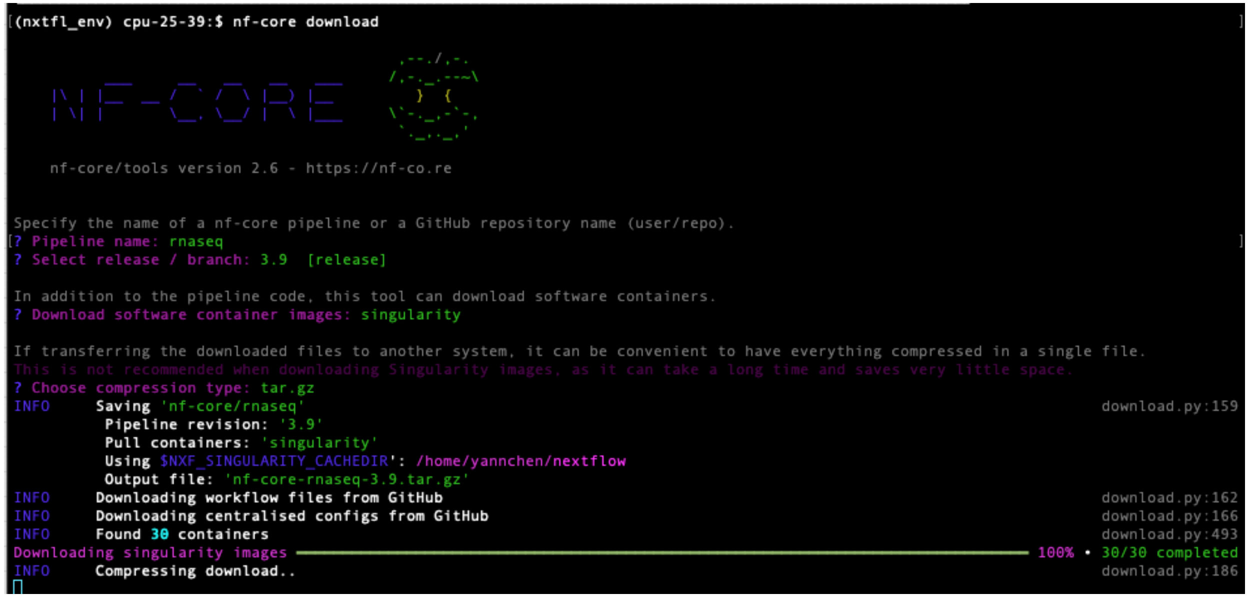
```
process {
executor = 'slurm’
clusterOptions = '-p nocona -N 1 -n 2’ }
```

- Define the input files

# Part III: nf-core pipelines

## Exercise #3

- Download nf-core/RNASeq
- Run nf-core/RNASeq



```
(nxtfl_env) cpu-25-39:$ nf-core download

                                          ,--./,-.
          ___     __   __   __   ___     /,-._.--~\
    |\ | |__  __ /  ` /  \ |__) |__         }  {
    | \| |       \__, \__/ |  \ |___     \`-._,-`-,
                                          `._,._,'

    nf-core/tools version 2.6 - https://nf-co.re


Specify the name of a nf-core pipeline or a GitHub repository name (user/repo).
? Pipeline name: rnaseq
? Select release / branch: 3.9  [release]

In addition to the pipeline code, this tool can download software containers.
? Download software container images: singularity

If transferring the downloaded files to another system, it can be convenient to have everything compressed in a single file.
This is not recommended when downloading Singularity images, as it can take a long time and saves very little space.
? Choose compression type: tar.gz
INFO     Saving 'nf-core/rnaseq'                                                                          download.py:159
          Pipeline revision: '3.9'
          Pull containers: 'singularity'
          Using $NXF_SINGULARITY_CACHEDIR': /home/yannchen/nextflow
          Output file: 'nf-core-rnaseq-3.9.tar.gz'
INFO     Downloading workflow files from GitHub                                                           download.py:162
INFO     Downloading centralised configs from GitHub                                                      download.py:166
INFO     Found 30 containers                                                                              download.py:493
Downloading singularity images ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% • 30/30 completed
INFO     Compressing download..                                                                           download.py:186
```

# Part III: nf-core pipelines



**Exercise #3**

- Download nf-core/RNASeq
- Run nf-core/RNASeq

- ```
  nf-core download <rnaseq> -r <3.9> --outdir <nf-core-rnaseq-3.9> -x <none> -c <singularity> <--force>
  ```
- ```
  nextflow run <nf-core-rnaseq-3.9/workflow/> -c <test/test.config> --outdir <rnaseq-3.9-outdir>
  ```

# Part III: nf-core pipelines

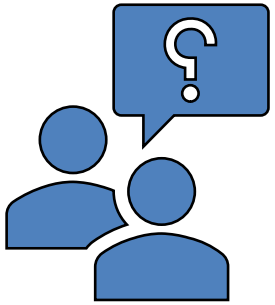


## Develop nf-core pipelines

- Discuss with nf-core
- Download nf-core template
- Check available modules
- Write your own pipelines
- Push to nf-core github

```
nf-core create
```

# Getting Help



# Part IV: Getting Help



nf-core

- slack : communicate with developers, trouble shooting
- GitHub : see source code
- twitter : milestone announcement
- YouTube : Tutorial videos

# Part IV: Getting Help

TEXAS TECH UNIVERSITY
Information Technology Division™

High Performance Computing Center

- Visit Our Website:
  - Nextflow
  - nf-core
  - TTU Training Material
- Submit a support ticket:
  - Send an email to hpccsupport@ttu.edu

# Part IV: Getting Help

TEXAS TECH UNIVERSITY
Information Technology Division™

**Please help us improve this workshop and develop further training sessions:**

- Ask questions at the end of this session
- Fill out survey forms
- Email any questions you have
- Post-workshop survey

# News



- A Long run a partition RedRaider cluster
  - Hardware: cores, clock speed, RAM
  - Scheduler: job length
- Other workflow management systems

snakemake

COMMON WORKFLOW LANGUAGE

TEXAS TECH UNIVERSITY

# Information Technology Division™