Introduction to the Workflow Management System
-- Nextflow & nf-core

Yanni Chen

Graduate Research Assistant
High Performance Computing Center

Fall 2022
Before Workshop

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

- Pre-workshop survey
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 bioinformatic pipelines
- Part IV: Getting Help
Computation Workflow Management System
Fourth paradigm is using the computational power to generate, curate, analyze, archive a massive scientific data

Gray 2009

Liew et al. 2016
Part I: Workflow Management System

Biological Experiment
Sequencing Data Prep and Data Size
(Illumina NovaSeq 6000)

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
The bioinformatic researchers are trying to automate the computational data processes to assist the data preparation process.
Bioinformatic Data Analysis Pipeline

Part I: Workflow Management System

Data Integration
- Read data from storage

Data Process
- Data trimming
- Quality control
- Data analysis

Data Summary
- Storage output
Part I: Workflow Management System

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
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.
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.
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 remaining data
- Visualizing the results
- Moving the results to a storage 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
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
Part I: Workflow Management System

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
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

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
A reactive workflow framework

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

Nextflow Highlights

• Domain specific language
• Work with cluster scheduler
• Error handling
• Reproducibility
Part II: Nextflow

Scripting Language

- Groovy
- Bash
- Python
- R
- Perl

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

Benefit of Work with Different Scheduler:

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

---

```bash
#!/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"){
    print("Usage: nextflow execute_joblist.nf --joblist [joblist file] --
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
}
```

Scatter Execution of Nextflow Examples
Part II: Nextflow

Version Control

- Docker
- Apptainer / Singularity
- Shifter
- Podman
Part II: Nextflow

Exercise #0 Install NextFlow
Part II: Nextflow

Exercise #0 Install Nextflow (Solution)

Check prerequisites
Java 8 or later is required

1

Set up
Dead easy to install

2

Launch
Try a simple demo

3

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.

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.

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

```plaintext
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

Your First Script:
- [tutorial.nf](#) (DSL1)
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)

```bash
nextflow run tutorial.nf
```
Exercise #1 Run first nextflow script (solution)

```bash
nextflow run tutorial.nf
nextflow run tutorial.nf
--resume
nextflow run tutorial.nf
--str 'Bonjour le monde'
```
Part II: Nextflow

DSL2

```plaintext
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
        """
}
```
A community effort to collect a curated set of analysis pipelines built using Nextflow.
Part III: nf-core pipelines

Use nf-core pipelines

check available nf-core pipeline, through website
Exercise #2 Check available pipelines

- using **nf-core list**

<table>
<thead>
<tr>
<th>Pipeline Name</th>
<th>Stars</th>
<th>Latest Release</th>
<th>Released</th>
</tr>
</thead>
<tbody>
<tr>
<td>fetchngs</td>
<td>45</td>
<td>1.5</td>
<td>6 days ago</td>
</tr>
<tr>
<td>eager</td>
<td>61</td>
<td>2.4.1</td>
<td>1 week ago</td>
</tr>
<tr>
<td>nanoseq</td>
<td>54</td>
<td>2.0.1</td>
<td>1 week ago</td>
</tr>
<tr>
<td>mag</td>
<td>69</td>
<td>2.1.1</td>
<td>2 weeks ago</td>
</tr>
<tr>
<td>cutandrun</td>
<td>16</td>
<td>1.0.0</td>
<td>1 months ago</td>
</tr>
<tr>
<td>ampliseq</td>
<td>75</td>
<td>2.1.1</td>
<td>1 months ago</td>
</tr>
<tr>
<td>rnaseq</td>
<td>399</td>
<td>3.4</td>
<td>2 months ago</td>
</tr>
<tr>
<td>mhcquant</td>
<td>16</td>
<td>2.0.0</td>
<td>3 months ago</td>
</tr>
<tr>
<td>bacass</td>
<td>33</td>
<td>2.0.0</td>
<td>3 months ago</td>
</tr>
<tr>
<td>viralrecon</td>
<td>51</td>
<td>2.2</td>
<td>4 months ago</td>
</tr>
<tr>
<td>bcellmagic</td>
<td>16</td>
<td>2.0.0</td>
<td>5 months ago</td>
</tr>
<tr>
<td>bactmap</td>
<td>25</td>
<td>1.0.0</td>
<td>6 months ago</td>
</tr>
<tr>
<td>smrnaseq</td>
<td>31</td>
<td>1.1.0</td>
<td>6 months ago</td>
</tr>
<tr>
<td>sarek</td>
<td>129</td>
<td>2.7.1</td>
<td>6 months ago</td>
</tr>
<tr>
<td>hic</td>
<td>32</td>
<td>1.3.0</td>
<td>7 months ago</td>
</tr>
</tbody>
</table>
Exercise #2 Check available pipelines (Solution)

- **Install nf-core**
  - `conda install nf-core` or
  - `conda update nf-core`

- **list all pipelines**
  - `nf-core list`
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
Part III: nf-core pipelines

Exercise #3

- Download nf-core/RNASeq
- Run nf-core/RNASeq
Exercise #3

- Download nf-core/RNASeq

- Run nf-core/RNASeq

```bash
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
Getting Help
Part IV: Getting Help

- slack: communicate with developers, trouble shooting
- GitHub: see source code
- Twitter: milestone announcement
- YouTube: Tutorial videos
Part IV: Getting Help

• Visit Our Website:
  • Nextflow
  • nf-core
  • TTU Training Material

• Submit a support ticket:
  • Send an email to hpccsupport@ttu.edu
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