

Introduction to Nextflow



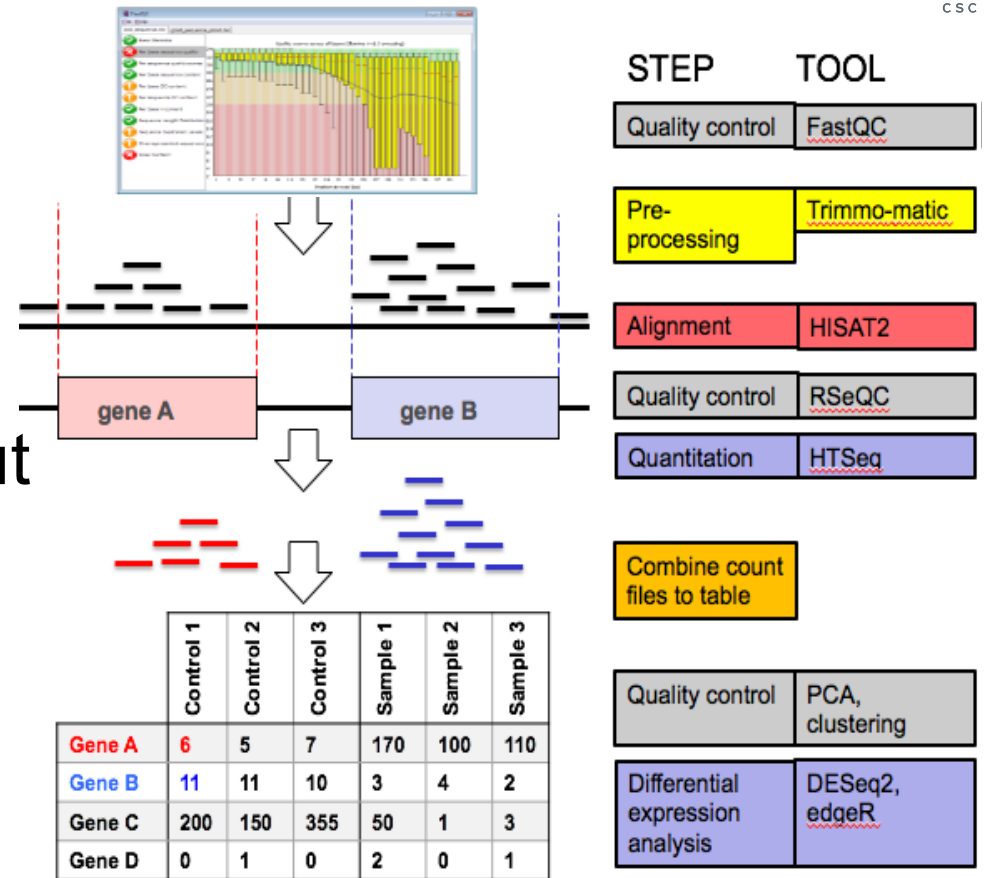
CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Outline

- Introduction
- Core features of Nextflow
- Nextflow essential building blocks
- Inspecting Hello-world Nextflow example
- Parameterisation in workflows

What is a Workflow ?

- A workflow is a collection of several analysis steps
- Steps are linked by input/output files
- One often needs to run the same workflow for several samples



RNAseq pipeline for differential gene regulation

Popular Choices for Bioinformatics Workflows



- Workflows
 - Snakemake
 - Cromwell
 - Nextflow
 - Galaxy



nextflow

What is Nextflow?



- A tool for managing scientific workflows, written in groovy, a language for java program



- A data driven language
 - Communication by dataflow variables
 - Processes (softwares/scripts) receiving (inputs) and emitting (outputs) through channels

Getting Started with Nextflow



- Required:
 - Posix file system (Linux/OS ...)
 - Java 8
- Software installation:
 - `curl get.nextflow.io | bash`
 - `mv nextflow ~/bin`
- Software stack you want:
 - Scripts available on PATH or under bin directory
 - Docker engine
 - Singularity
 - Conda

Core Features of Nextflow

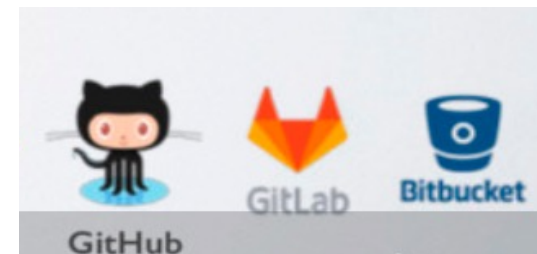
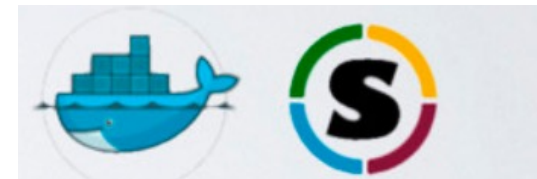


- A workflow manager
- Reproducibility
- Portability
- Parallelisation
- Easy to resume
- Easy prototyping

Core Features of Nextflow



- A workflow manager
- Reproducibility
- Portability
- Parallelisation
- Easy to resume
- Easy prototyping



Version
control

Supports integration with
containers and Github

Core Features of Nextflow



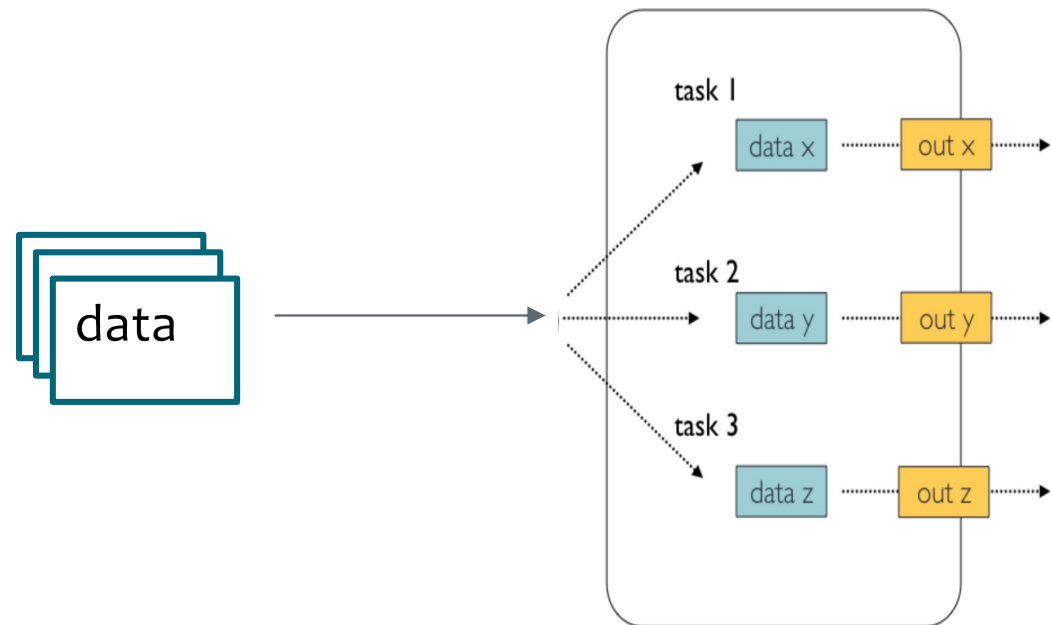
- A workflow manager
- Reproducibility
- **Portability**
- Parallelisation
- Easy to resume
- Easy prototyping



Core Features of Nextflow



- A workflow manager
- Reproducibility
- Portability
- **Parallelisation**
- Easy to resume
- Easy prototyping



Core Features of Nextflow



- A workflow manager
- Reproducibility
- Portability
- Parallelisation
- Easy to resume
- Easy prototyping

Call caching

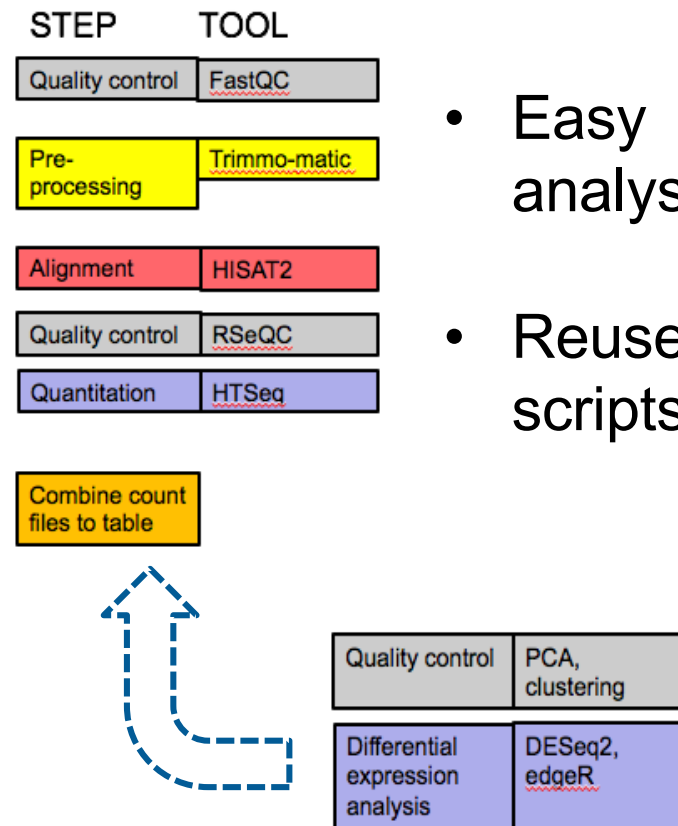
Monitors each chunk/file and process

Core Features of Nextflow



- A workflow management
- Reproducibility
- Portability
- Parallelisation
- Easy to resume
- Easy prototyping

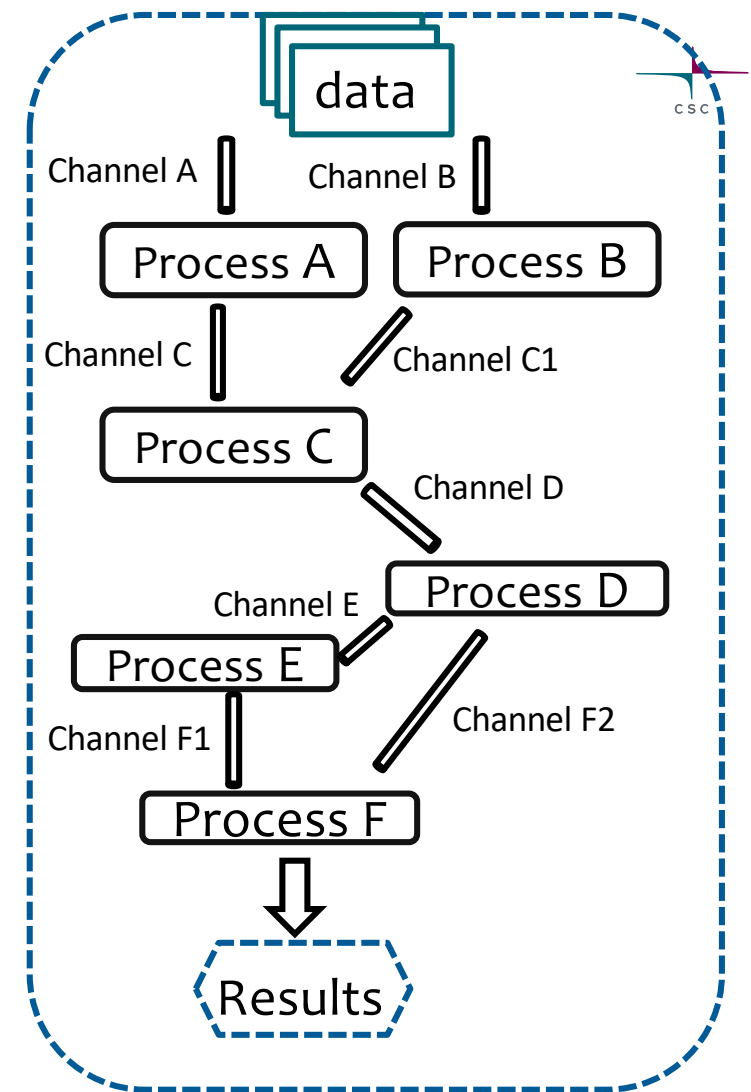
- Easy to add new analysis step
- Reuse your existing scripts and tools



RNAseq pipeline for differential gene regulation

NextFlow: Building Blocks

- **channel**: information flows from one process to another via 'channels' as defined in the input and output sections of each process
- **process**: one (independent) step in the pipeline block. This is where the execution of code happens



Bit more on Nexflow Channels



- Two different kinds of channels: Queue and Value channels.
- **Create a channel** : syntax - *Channel.<method>*
- **Value channel**: can be used multiple times in workflow
 - Channel.value ('single value/list object/map object')
- **Queue channel**: consumed when they are used by a process or an operator.
 - Channel.fromList (['salmon', 'kallisto'])
 - Channel.fromPath('data/*.fq.gz')
 - Channel.fromFilePairs('data/FA33*_{1,2}.fq.gz')
 - Channel.fromSRA('SRP043510')

NextFlow: Building Blocks



- **channel:** information flows from one process to another via 'channels' as defined in the input and output sections of each process
- **process:** one (independent) step in the pipeline block. This is where the execution of code happens

```
process /* <process_name> */ {  
    /* <config section> */  
  
    input:  
    /* <input channel> */  
  
    output:  
    /* <output channel> */  
  
    script: /* <task> */  
    """"  
    # some bash code  
    """"  
}
```

Nextflow : Hello-World Example



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

```
greet = Channel.from("Moi", "Ciao", "Hello", "Hola", "Bonjour")
```

```
process sayHello {
```

```
publishDir 'results'
```

```
input:
  val greet from greet
```

```
output:
  file "${greet}.txt" into greetingFiles
```

```
script:
  """
  echo ${greet} > ${greet}.txt
  """
```

```
}
```


Nextflow Help in Practice



- Help: `nextflow -h`
- Nextflow usage: `nextflow [options] COMMAND [arg...]`
- Commands:

Clean	Clean up project cache and work directories
clone	Clone a project into a folder
config	Print a project configuration
console	Launch Nextflow interactive console
drop	Delete the local copy of a project
help	Print the usage help for a command
info	Print project and system runtime information
kuberun	Execute a workflow in a Kubernetes cluster
list	List all downloaded projects
log	Print executions log and runtime info
pull	Download or update a project
run	Execute a pipeline project
self-update	Update nextflow runtime to the latest available
version	
view	View project script file(s)

Nextflow : Hello-World Example



```
(nextflow) [yetukuri@r07c49 ~]$ nextflow run hello
N E X T F L O W ~ version 20.07.1
Pulling nextflow-io/hello ...
downloaded from https://github.com/nextflow-io/hello.git
Launching `nextflow-io/hello` [berserk_mcclintock] - revision:
e6d9427e5b [master]
executor > local (4)
[99/a0a5ef] process > sayHello (3) [100%] 4 of 4 ✓
Bonjour world!

Ciao world!

Hola world!

Hello world!
```

Finding More Information About a Pipeline

```
(nextflow) [yetukuri@r07c49 ~]$ nextflow info hello
project name: nextflow-io/hello
repository   : https://github.com/nextflow-io/hello
local path   : /users/yetukuri/.nextflow/assets/nextflow-io/hello
main script  : main.nf
revisions    :
* master (default)
  mybranch
  testing
  v1.1 [t]
  v1.2 [t]
```

- Think of running above hello world example in a reproducible manner

Inspecting Nextflow Results



- Nextflow creates a work directory for each process (using some hash numbers)
- Each folder contains
 - Input files
 - Output files
 - Number of hidden files (useful for debugging)
 - Script used for the process
- You can publish results to a different folder