

# Introduction to Workflows with Nextflow



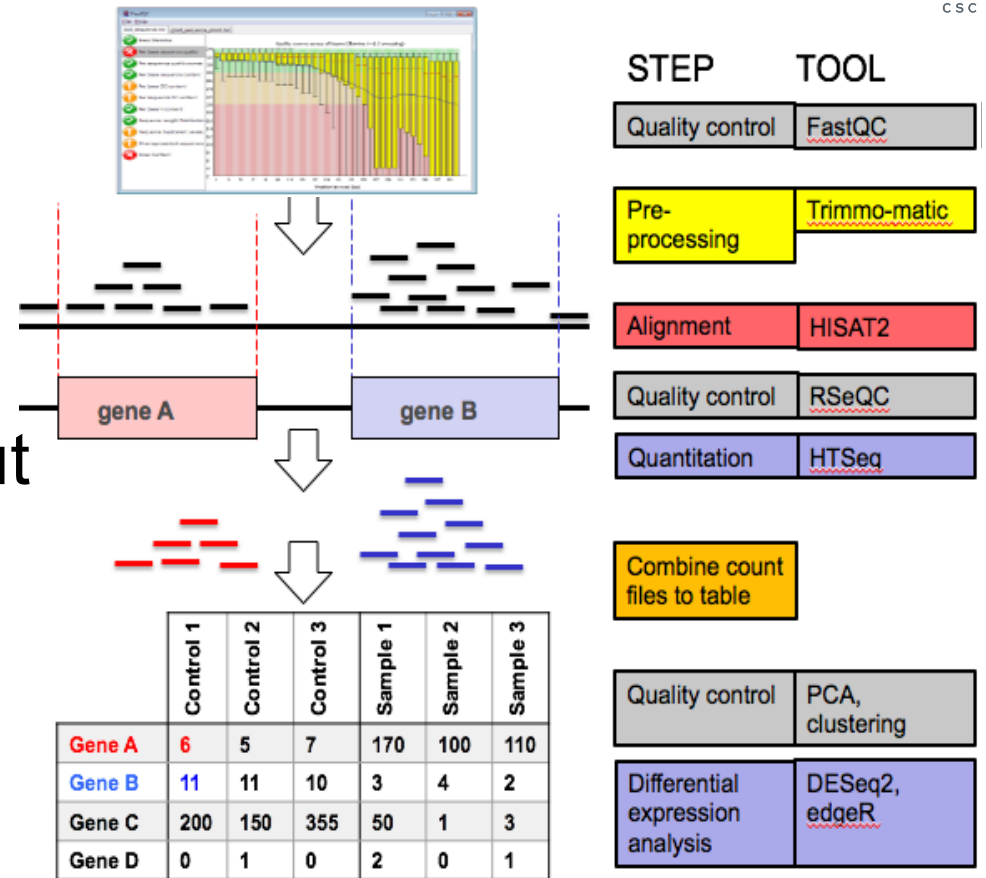
*CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus*

# Outline

- Introduction to Nextflow
- Core features of Nextflow
- Nextflow essential building blocks
- Inspecting Hello-world Nextflow example
- Parameterization in workflows
- Hands-on tutorials

# What is a Workflow ?

- A pipeline 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



nature  
biotechnology

Access provided by Universitat Pompeu Fabra

Altmetric: 117 Citations: 7

Correspondence

**Nextflow enables reproducible computational workflows**

Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo & Cedric Notredame 

*Paolo Di Tommaso, Nextflow Lead*

The image shows a screenshot of a Nature Biotechnology article. At the top left is the journal logo. To the right is a circular profile picture of Paolo Di Tommaso. Below the journal name, it says 'Access provided by Universitat Pompeu Fabra'. A horizontal bar contains 'Altmetric: 117 Citations: 7'. Underneath is the word 'Correspondence' followed by the article title 'Nextflow enables reproducible computational workflows'. At the bottom, the authors are listed: 'Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo & Cedric Notredame' with an envelope icon next to the last name. A separate line identifies 'Paolo Di Tommaso, Nextflow Lead'.

**nextflow**



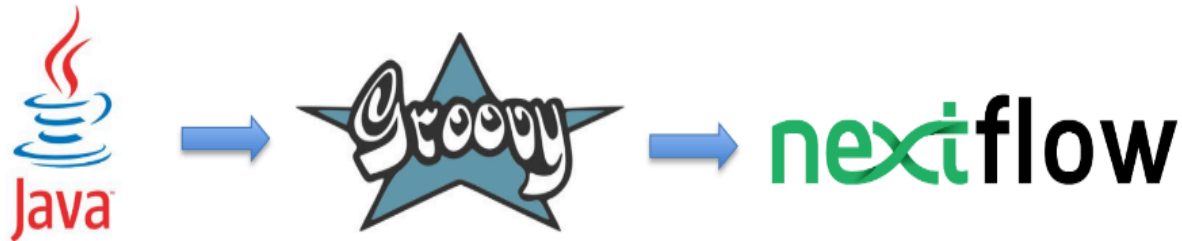
# Nextflow is Getting Popular



# What is Nextflow?



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



- A dataflow programming model
  - Communication by dataflow variables
  - Processes (softwares/scripts) receiving (inputs) and emitting (outputs) through channels

# Getting Started with Nextflow

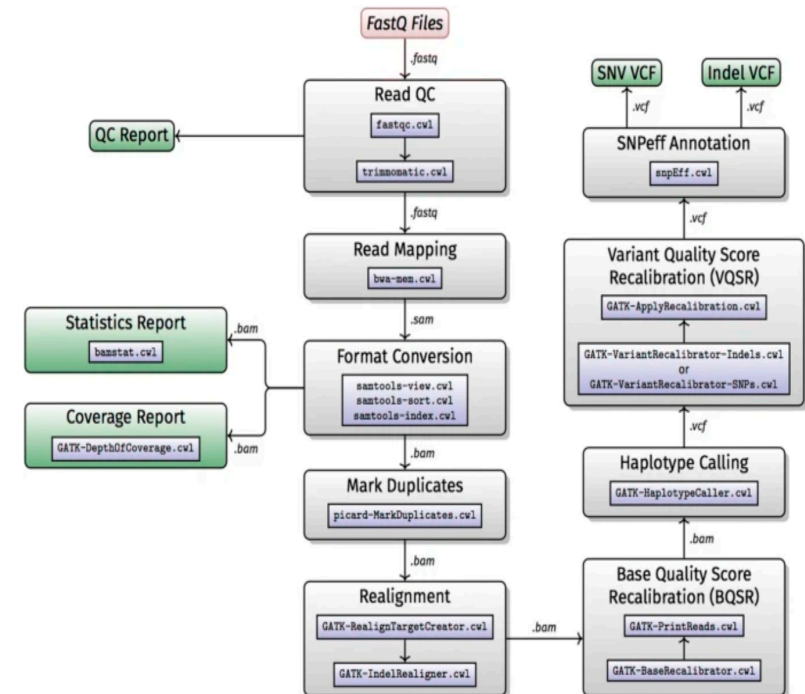


- Required:
  - Posix file system (Linux/OS ...)
  - Java 8
- Software installation:
  - `curl get.nextflow.io | bash`
  - `mv nextflow ~/bin`
- Software stack you want:
  - Scripts available on PATH or under bin directory
  - Docker engine
  - Singularity
  - Conda

# Core Features of Nextflow



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



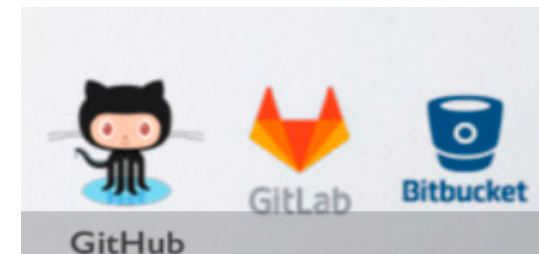
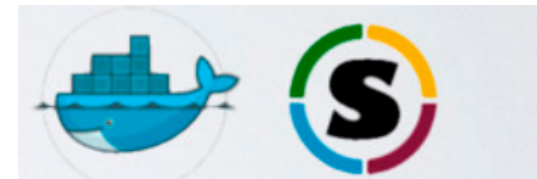
WGS/WES example from  
Baichoo Et.al., BMC Bioinformatics,19,457 (2018)



# 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



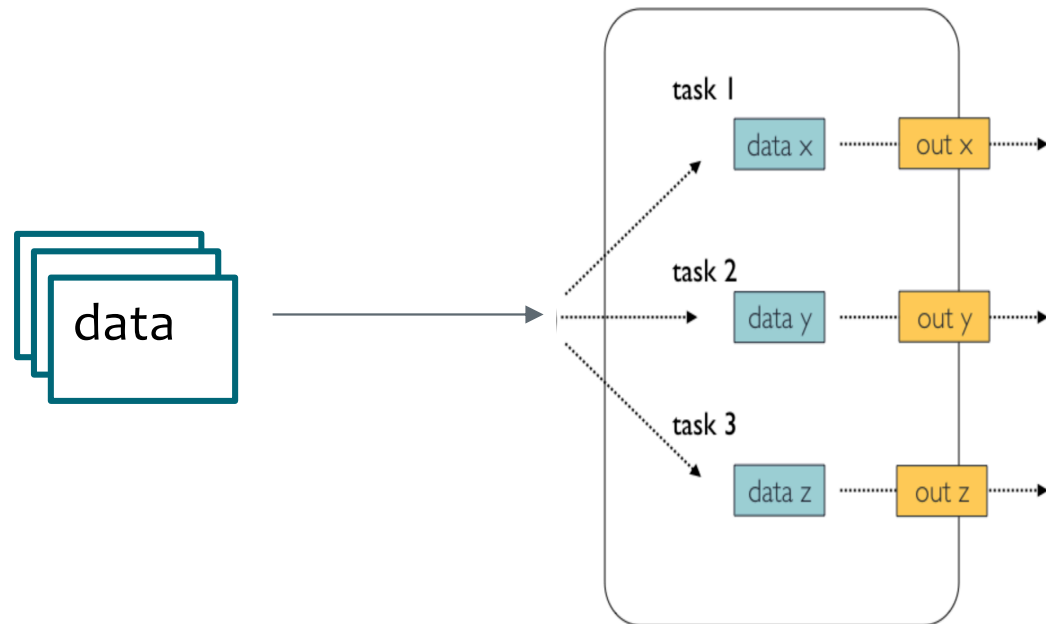
- A workflow manager
- Reproducibility
- **Portability**
- Parallelisation
- Easy to resume
- Easy prototyping



# 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

STEP	TOOL
Quality control	FastQC
Pre-processing	Trimmo-matic
Alignment	HISAT2
Quality control	RSeQC
Quantitation	HTSeq

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

Combine count files to table

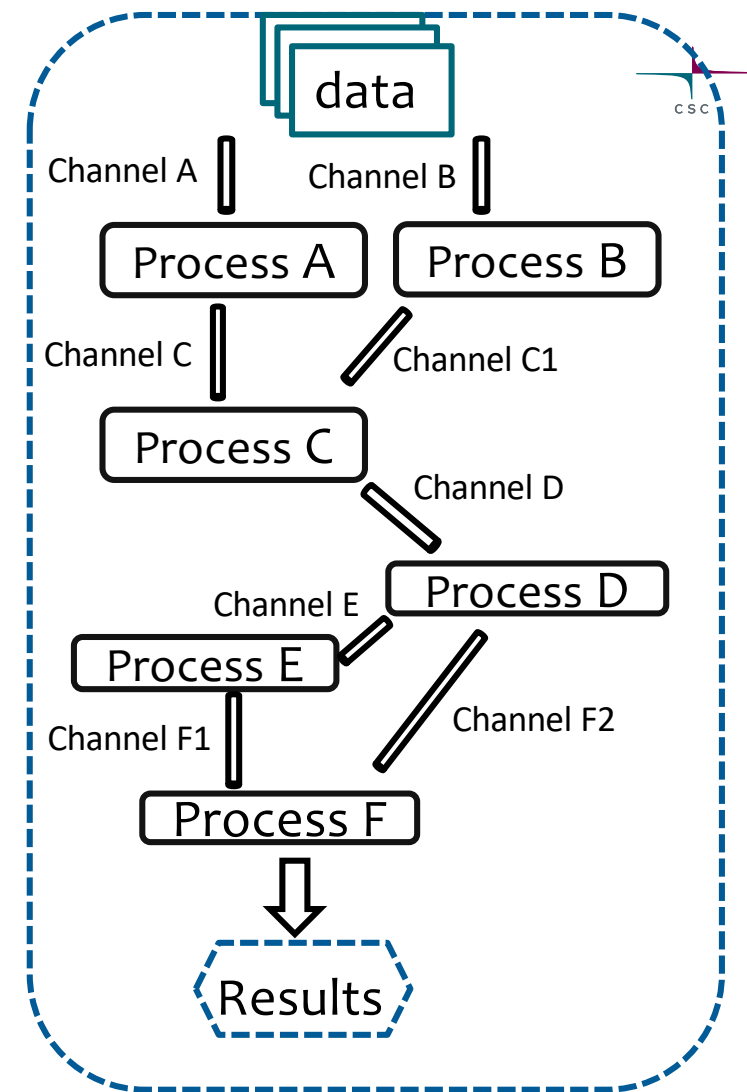


Quality control	PCA, clustering
Differential expression analysis	DESeq2, edgeR

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
```

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

```
process sayHello {
```

```
publishDir 'results'
```

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

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

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

```
}
```

# Nextflow Help in Practice



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

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

# Nextflow : Hello World *run* from github



```
(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!
```

# Nextflow : Hello World info



```
(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 folder (i.e., inside *work* directory) for each process (using some hash numbers)
- Each folder contains
  - Links to input files
  - Output files
  - Number of hidden files
  - Script used for the process
- You can publish results to a different folder

# Nextflow Parameterisation



- Enables you to change input to workflow at runtime
- Declare in Nextflow script by prepending the prefix *params.* to a variable (e.g., *params.greetings = "Moi"*)
- Pass a pipeline parameter to script on commandline using *--variable name* (e.g., *nextflow run hello-world.nf --greetings "Hei"*)
- Use a file (yaml/json) to pass many parameters using option *-params-file*

# Time for practicals !!!



- **Where to run practicals:** Interactive nodes on Puhti
- **Tutorials:** Hello-world and (close to) real-world tutorials
- **Expected outcome from tutorials:**
  - Learn to run a nextflow pipeline interactively (locally)
  - Able to Inspect default output files
  - Move resulting files to a convenient place