GCB 2025

# From a Collection of Scripts to a Pipeline
## – Writing Nextflow Workflows with nf-core Best Practices –

Workshop by Famke Bäuerle and Mark Polster

Install dependencies:

```
conda create --name nf-core nf-core nextflow
```

FASTQ → bwa_mem → SAM → samtools → BAM

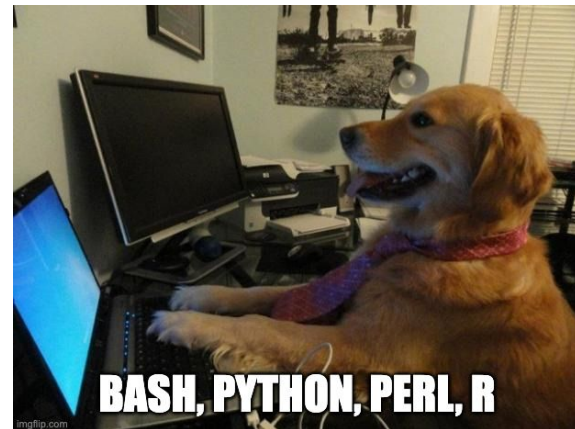FASTQ → (bwa_mem) → SAM → (samtools) → BAM

```bash
#!/bin/bash

# Download software, reference genome, and FASTQ files
wget http://downloads.sourceforge.net/project/bio-bwa/bwakit/bwakit-0.7.12_x64-
linux.tar.bz2
wget -O refGenomeFiles.zip https://ndownloader.figshare.com/files/4841809
wget -O FASTQ.zip https://ndownloader.figshare.com/articles/3114454/versions/1

# Extract software and reference genome files
tar -jxvf bwakit-0.7.12_x64-linux.tar.bz2
unzip refGenomeFiles.zip
unzip FASTQ.zip

# Align FASTQ data to reference genome and save to SAM file
bwa.kit/bwa mem  KJ660346.fa SRR1972917_1.fastq SRR1972917_1.fastq >
SRR1972917_aligned.sam

# Convert SAM file to BAM file
bwa.kit/samtools view -bS SRR1972917_aligned.sam > SRR1972917_aligned.bam

# Sort BAM file
bwa.kit/samtools sort SRR1972917_aligned.bam SRR1972917_aligned_sorted

# Index BAM file
bwa.kit/samtools index SRR1972917_aligned_sorted.bam
```
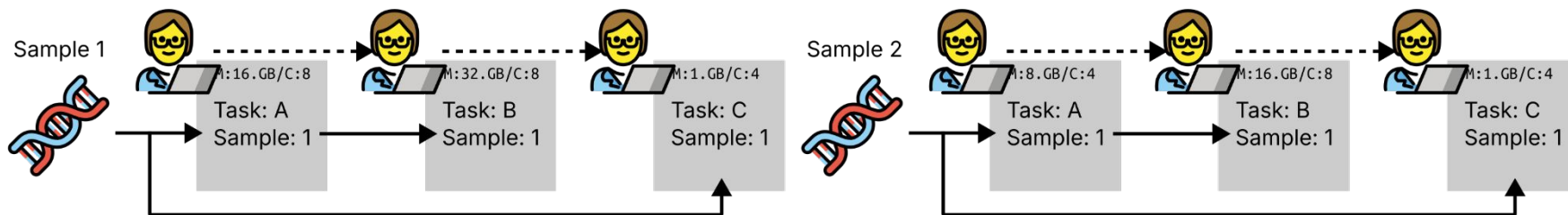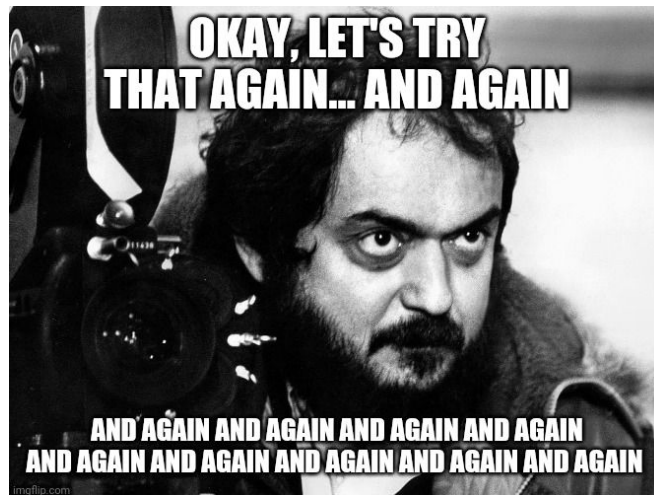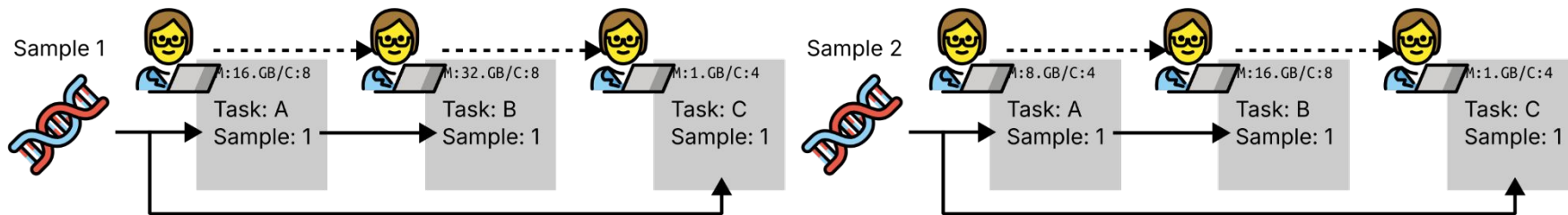


Piccolo & Frampton 2016

- Custom scripts

  - Bash, Python, Groovy, R, Perl

- Today we want to scale our raw scripts into a reproducible, maintainable and extendable workflow using **Nextflow**

- you don't have to reinvent the wheel:
  **let's make use of nf-core components and best practices**

| Tool | Class | Ease of use[a] | Expressiveness[b] | Portability[c] | Scalability[d] | Learning resources[e] | Pipeline initiatives[f] |
|---|---|---|---|---|---|---|---|
| Galaxy | Graphical | ●●● | ●○○ | ●●● | ●●● | ●●● | ●●○ |
| KNIME | Graphical | ●●● | ●○○ | ○○○ | ●●◐ | ●●● | ●●○ |
| Nextflow | DSL | ●●○ | ●●● | ●●● | ●●● | ●●● | ●●● |
| Snakemake | DSL | ●●○ | ●●● | ●●◐ | ●●● | ●●○ | ●●● |
| GenPipes | DSL | ●●○ | ●●● | ●●○ | ●●○ | ●●○ | ●●○ |
| bPipe | DSL | ●●○ | ●●● | ●●○ | ●●◐ | ●●● | ●○○ |
| Pachyderm | DSL | ●●○ | ●●● | ●○○ | ●●○ | ●●● | ○○○ |
| SciPipe | Library | ●●○ | ●●● | ○○○ | ○○○ | ●●○ | ○○○ |
| Luigi | Library | ●●○ | ●●● | ●○○ | ●●◐ | ●●○ | ○○○ |
| Cromwell + WDL | Execution + workflow specification | ●○○ | ●●○ | ●●● | ●●◐ | ●●○ | ●●○ |
| cwltool + CWL | Execution + workflow specification | ●○○ | ●●○ | ●●◐ | ○○○ | ●●● | ●●○ |
| Toil + CWL/WDL/Python | Execution + workflow specification | ●○○ | ●●● | ●◐○ | ●●● | ●●○ | ●●○ |

Fastq

**Quality control**

**command**: "fastqc -t 2 -q reads_1.fastq reads_2.fastq"

Filtered fastq

**Read mapping**

**command**: "bwa mem -p -C -M filtered_reads_1.fastq filtered_reads_2.fastq"

BAM

**Variant calling**

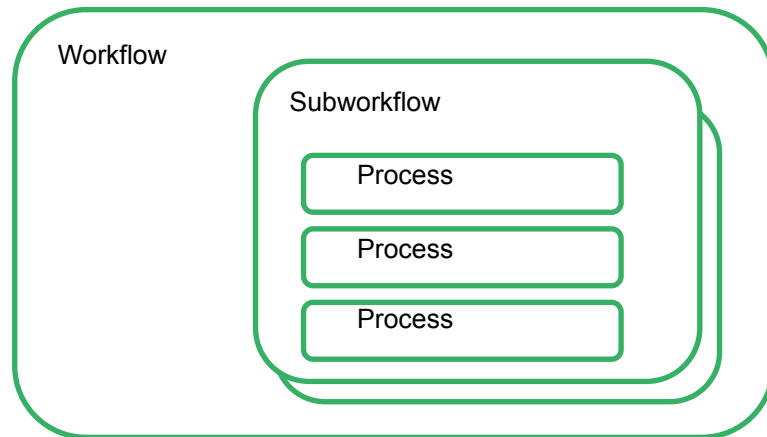**command**: "gatk Mutect2 -R genome.fa -I tumor.bam -I normal.bam -O somatic_variants.vcf"

vcf
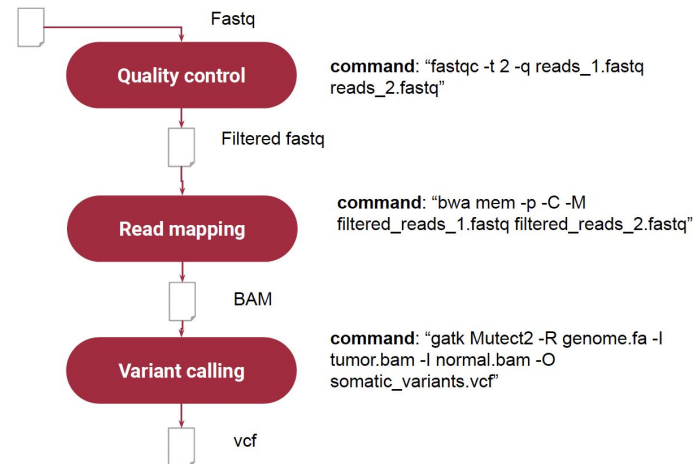
*A Nextflow pipeline consists of modularized processes,*

*channels and (sub)workflows*

- **Process:**

  A process is be written in any scripting language

- **Channel:**

  Provides inter process communication by using

  first-in first-out queues

- **Workflow:**

  A joint collection of one or more processes

# nf-core A community effort to collect a curated set of analysis pipelines built using Nextflow

Correspondence | Published: 13 February 2020

## The nf-core framework for community-curated bioinformatics pipelines

Philip A. Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso & Sven Nahnsen ✉
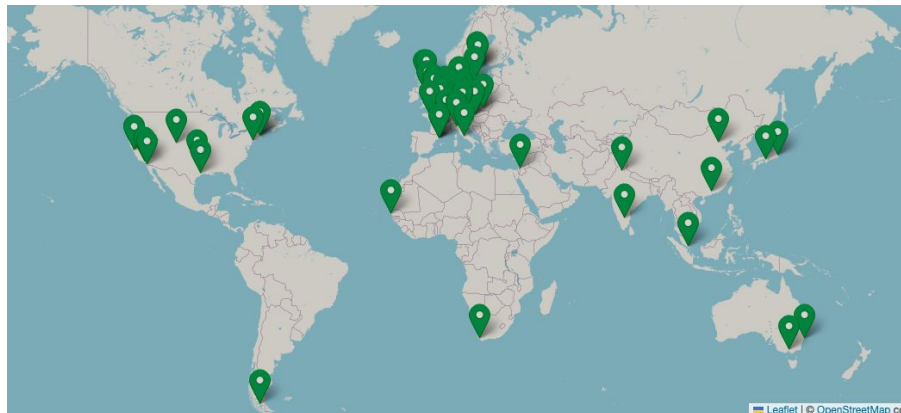
Best-practice guidelines

nf-core tools

Curated set of pipelines

> 12.500 Slack users     > 2.200 GitHub organisation members     83 released pipelines ✓

# nf-core pipelines follow strict best-practice guidelines

Documentation

Packaged software

Continuous Integration Testing

Portable and reproducible

Stable releases
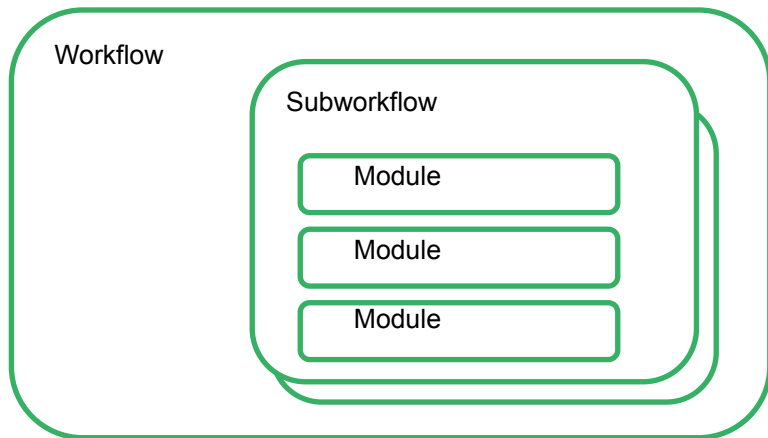
Cloud-ready

Tests - development

CI tests

nf-test

Full-size tests

Validation – user side

nf-schema plugin:

validation of pipeline parameters

and sample sheets

nextflow

nf-core

Workflow

Subworkflow

Module

Module

Module

Repositories

- nf-core **modules** (1620)

- nf-core **subworkflows** (8v4)

FASTQ — bwa_mem → SAM — samtools → BAM

```bash
#!/bin/bash

# Download software, reference genome, and FASTQ files
wget http://downloads.sourceforge.net/project/bio-bwa/bwakit/bwakit-0.7.12_x64-linux.tar.bz2
wget -O refGenomeFiles.zip https://ndownloader.figshare.com/files/4841809
wget -O FASTQ.zip https://ndownloader.figshare.com/articles/3114454/versions/1

# Extract software and reference genome files
tar -jxvf bwakit-0.7.12_x64-linux.tar.bz2
unzip refGenomeFiles.zip
unzip FASTQ.zip

# Align FASTQ data to reference genome and save to SAM file
bwa.kit/bwa mem  KJ660346.fa SRR1972917_1.fastq SRR1972917_1.fastq > SRR1972917_aligned.sam

# Convert SAM file to BAM file
bwa.kit/samtools view -bS SRR1972917_aligned.sam > SRR1972917_aligned.bam

# Sort BAM file
bwa.kit/samtools sort SRR1972917_aligned.bam SRR1972917_aligned_sorted

# Index BAM file
bwa.kit/samtools index SRR1972917_aligned_sorted.bam
```

Piccolo & Frampton 2016

FASTQ    **bwa_mem** →    **SAM**    samtools →    BAM

## nf-core/modules

Browse the 1620 modules that are currently available as part of nf-core.
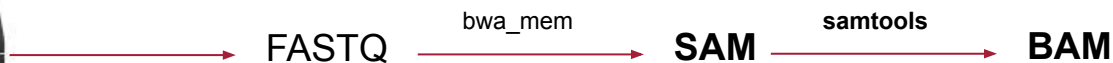
bwa_mem

### bwa_mem

Performs fastq alignment to a fasta reference using BWA

mem   bwa   alignment   map   fastq   bam   sam

Included in: atacseq   chipseq   circrna

and 13 more pipelines

# **What you could do today**

## **From Bash to pipeline**

- find our set of bash scripts and turn them into a nice nf-core based pipeline

- make use of **https://nf-co.re/modules** to find the modules you need

- find examples at **https://nf-co.re/pipelines**

## **From Python script to pipeline**

- under *py_scripts* we stored a couple python scripts

- you can add them to the *bin* folder of your pipeline and create modules for each within your workflow

- see **https://github.com/nf-core/epitopeprediction** for a pipeline where this is done successfully

> clone https://github.com/qbic-projects/gcb-2025-scripts-to-pipeline to start

# Some links

https://nf-co.re

github.com/nf-core

www.nextflow.io/docs/latest/
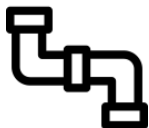
nfcore.slack.com

nf-co.re/pipelines/
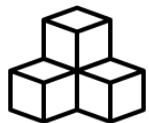
midnighter.github.io/nextflow-gotchas/

nf-co.re/modules/