(ARCify your) Computational Workflows

Automating and Reproducing Data Analysis Pipelines

Dominik Brilhaus, CEPLAS

2025-10-16

CEPLAS
Cluster of Excellence on Plant Sciences

DataPLANT

Goals for today

1. Write and execute a simple CWL workflow
2. Structure a demo workflow into an ARC
3. Convert an existing CLI tool or script into a CWL-wrapped workflow
4. …

What Are Computational Workflows?

- Define **steps** in data analysis (e.g., preprocessing → alignment → QC)
- Each step specifies:
  - **Inputs**
  - **Outputs**
  - **Tools / Commands**
- Enable:
  - Reproducibility
  - Portability
  - Scalability

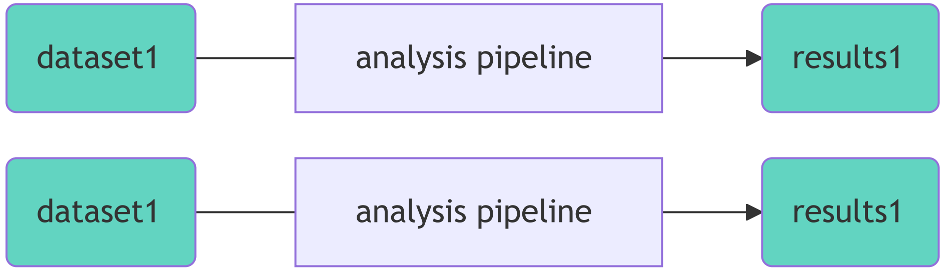
Materials & Methods ✅

```
fastqc assays/rnaseq/dataset/sample1.fastq.gz
fastqc assays/rnaseq/dataset/sample2.fastq.gz
fastqc assays/rnaseq/dataset/ ...
```

"*FastQC v0.12.1 was employed for read quality control using default parameters.*"

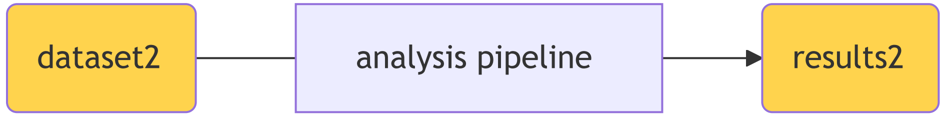
Why Workflows?

- **Reproducibility** of the data
- **Replicability** of the analysis



Re−running the **same** analysis on the **same** dataset

- **Reusability** of the analysis



Applying the same analysis on **another** dataset

Some factors affecting reproducibility & reusability

- Version of tool, software, package, or library
- Version of interpreter (python, R, F#, etc.)
- Operating system (linux, win, mac) and version
- ...

Workflow Languages

- CWL
  - https://www.commonwl.org/
  - Open standard for describing analysis workflows
  - Interoperable & portable
- Nextflow
  - https://nextflow.io
  - Domain-specific language for pipelines
- Snakemake
  - https://snakemake.github.io
  - Makefile-like workflows in Python
  - Easy syntax, flexible, local-friendly

CWL: Common Workflow Language

- Open community standard
- Describes:
  - Tools (command-line wrappers)
  - Workflows (combining tools)
- YAML-based description of:
  - Inputs & Outputs
  - Dependencies (e.g. Docker container)
  - Resource needs (e.g. RAM, cores)

https://www.commonwl.org

CWL is a time investment at first

There's a *tiny* learning curve and some dependencies

- Docker
- Conda and the cwltool (or other reference runner)
- JavaScript (good to know for file handling)
- …

… but it pays off!

COMMON
WORKFLOW
LANGUAGE

https://www.commonwl.org

CWL Resources

- CWL user guide: https://www.commonwl.org/user_guide/
- Specification v1.2: https://www.commonwl.org/v1.2/CommandLineTool.html
- CWL Discourse: https://cwl.discourse.group
- CWL tool: https://github.com/common-workflow-language/cwltool
- CWL tool docs: https://cwltool.readthedocs.io/en/latest/

CWL workflow repos

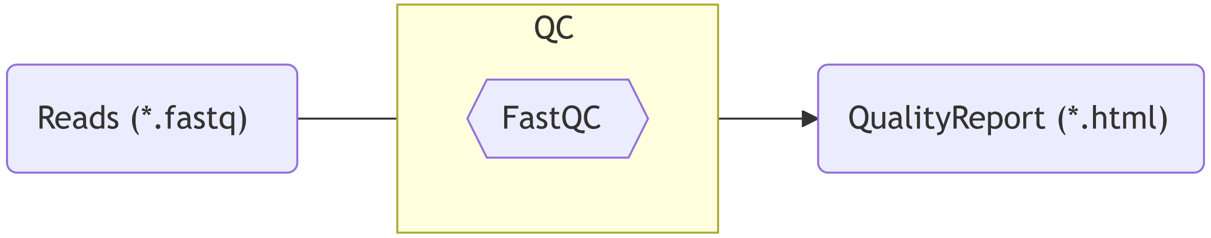
- Published CWL Workflows: https://view.commonwl.org/workflows
- CWL repos: https://www.commonwl.org/repos/
- Bio-cwl-tools: https://github.com/common-workflow-library/bio-cwl-tools/

Installing (bioinformatic) tools is fun (?)🎢

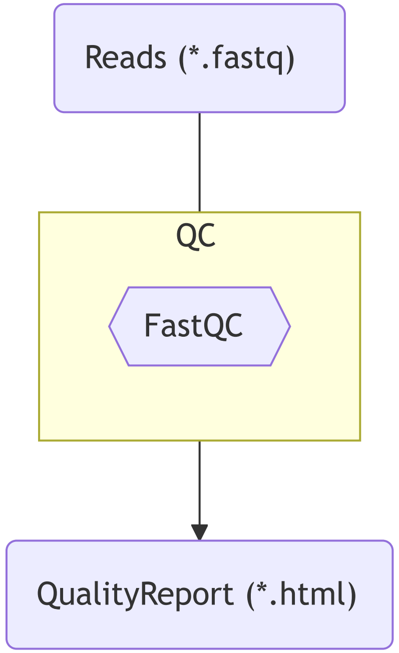
- From source: https://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc
- Docker: `docker pull quay.io/biocontainers/fastqc`
- Conda: `conda install fastqc`
- …

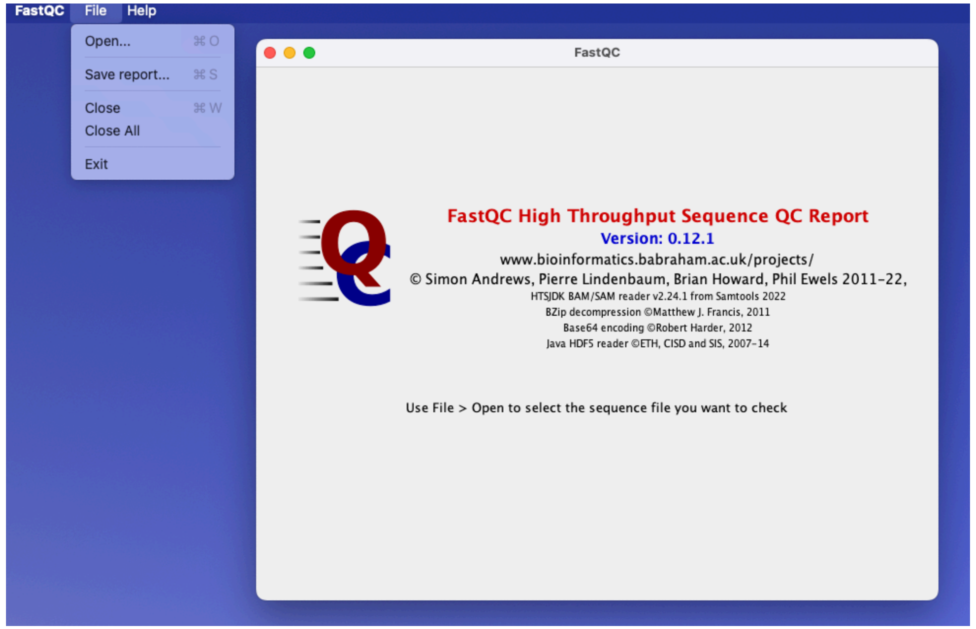
Example tool: FastQC

First step in RNASeq data analysis: QC of read files (e.g. *.fastq)



FastQC has a GUI





FastQC High Throughput Sequence QC Report
Version: 0.12.1
www.bioinformatics.babraham.ac.uk/projects/
© Simon Andrews, Pierre Lindenbaum, Brian Howard, Phil Ewels 2011-22,
HTSJDK BAM/SAM reader v2.24.1 from Samtools 2022
BZip decompression ©Matthew J. Francis, 2011
Base64 encoding ©Robert Harder, 2012
Java HDF5 reader ©ETH, CISD and SIS, 2007-14

Use File > Open to select the sequence file you want to check

Are we FAIR, yet?

- where did I click
- reproducibility
- record exactly what I’ve done
- history
- instruction
- tool version
- ...

Command line tool

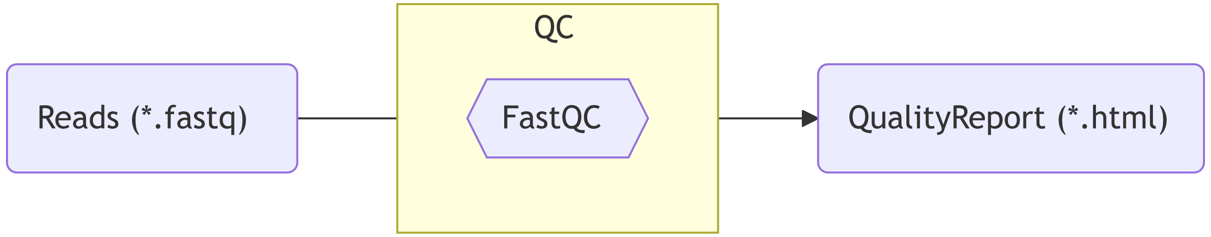
- Some tool that you can run … on the command line
- Example:
  - CLI: **ARC Commander**
  - (GUI: **ARCitect**)
- Takes arguments or parameters as **inputs**
- Generates **outputs**

FastQC via command line

```
fastqc --version
fastqc --help
```

FastQC via command line

```
fastqc assays/rnaseq/dataset/blau1_CGATGT_L005_R1_002.fastq.gz
```



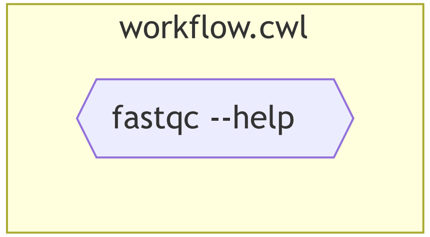
Demo: CWL-Wrapping the CommandLineTool FastQC

Step 1: Define CLI tool as CWL CommandLineTool

- Without in/out
- (Requires **local tool installed**)

workflow.cwl

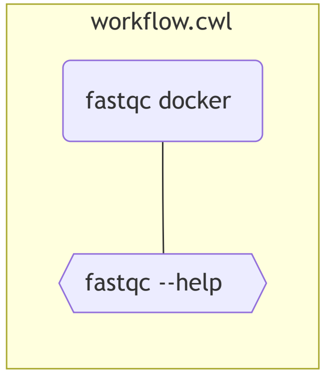
```
#!/usr/bin/env cwl-runner
cwlVersion: v1.2
class: CommandLineTool

baseCommand: ["fastqc", "--help"]

inputs: []

outputs: []
```



Step 2: Add a docker container

workflow.cwl

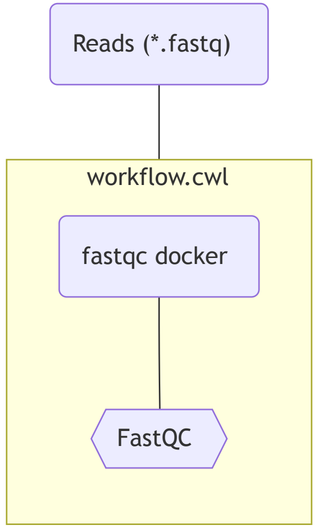
```
#!/usr/bin/env cwl-runner
cwlVersion: v1.2
class: CommandLineTool

hints:
  DockerRequirement:
    dockerPull: quay.io/biocontainers/fastqc:0.11.9--hdfd78af_1

baseCommand: ["fastqc", "--help"]

inputs: []

outputs: []
```



Step 3: Define inputs

workflow.cwl

```
#!/usr/bin/env cwl-runner
cwlVersion: v1.2
class: CommandLineTool

hints:
  DockerRequirement:
    dockerPull: quay.io/biocontainers/fastqc:0.11.9--hdfd78af_1

baseCommand: ["fastqc"]

inputs:
  reads:
    type: File[]
    inputBinding:
      position: 1

arguments:
  - valueFrom: $(runtime.outdir)
    prefix: "-o"

outputs: []
```



Data PLANT CEPLAS

Step 4: Define outputs

Run the workflow

You can provide arguments via another file:

Run the workflow

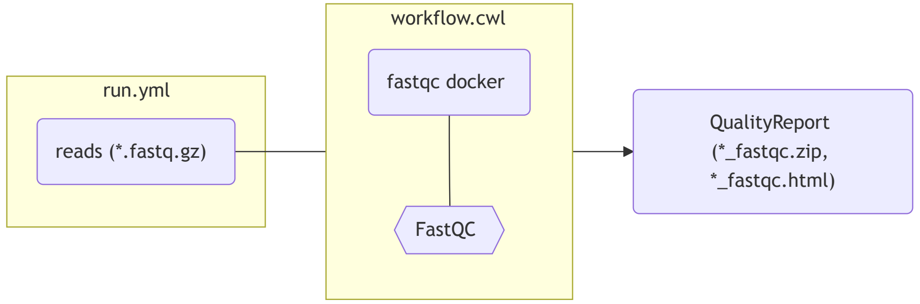
You can provide arguments via another file:

run.yml

```yaml
reads:
  - class: File
    path: ../../assays/rnaseq/dataset/blau1_CGATGT_L005_R1_002.fastq.gz
  - class: File
    path: ../../assays/rnaseq/dataset/blau2_TGACCA_L005_R1_002.fastq.gz
```

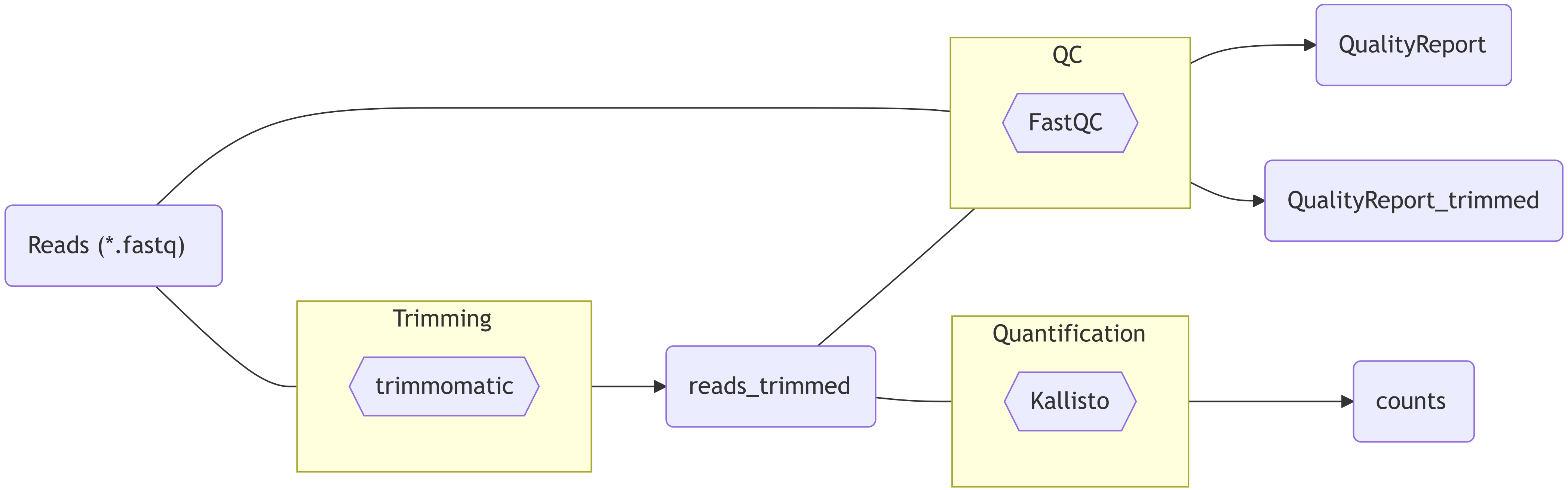
workflow.cwl

```yaml
#!/usr/bin/env cwl-runner
cwlVersion: v1.2
class: CommandLineTool

hints:
  DockerRequirement:
    dockerPull: quay.io/biocontainers/fastqc:0.11.9--hdfd78af_1

baseCommand: ["fastqc"]

inputs:
  reads:
    type: File[]
    inputBinding:
      position: 1
...
```



```
cwltool workflow.cwl run.yml
```

Growing pipeline: First steps RNASeq pipeline



Workflow Platforms

Galaxy

Galaxy Logo

- Web-based platform for **data-intensive research**
- Visual drag-and-drop workflow builder
- Integrates with CWL, Nextflow, Snakemake

https://galaxyproject.org

Data PLANT CEPLAS

WorkflowHub

- **FAIR registry** for describing, sharing and publishing scientific computational workflows
- Supports multiple workflow languages
- Provides metadata, versioning, and citation info
- Facilitates discovery and re-use of workflows in an accessible and interoperable way
- Encourages **reusability** and **collaboration**
- extensive use of **open standards and tools**:
  - CWL
  - RO-Crate
  - Bioschemas

WorkflowHub

https://workflowhub.eu

Data PLANT

CEPLAS

Setup for CWL

- Install Docker
- Install conda
- Install the CWL Runner cwltool

Conda

- Miniconda, Anaconda, Miniforge, …
- **Package manager** for scientific software
- Creates **isolated environments**
- Reproducible installation of tools

Install a conda distribution

https://docs.conda.io/projects/conda/en/latest/user-guide/install/

Create environment

```
conda create -n cwl_env
conda activate cwl_env
```

Install tool or package

```
conda install cwltool
```

Avoid using Anaconda and the "default" channel

Check the terms of services

- https://www.anaconda.com/pricing/terms-of-service-faqs
- https://docs.conda.io/projects/conda/en/latest/user-guide/configuration/settings.html#config-channels

Adapt .condarc to prevent using "default" channel

- Your `.condarc` (e.g. ~/miniconda3/.condarc) should look similar to this

.condarc

```
channels:
  - conda-forge
  - bioconda
```

Docker

- Containerization platform
- Bundles software + dependencies
- CWL can define Docker images for each tool

https://www.docker.com/

Install tool or package

```
docker pull commonworkflowlanguage/cwltool
```

cwltool

- Reference CWL runner
- Validates and executes workflows
- Supports Docker, Singularity, and Conda

```
pip install cwltool
cwltool --help
```

HPC HHU

- conda mirrors
- cwl–toil

CWL & HPC => toil-runner

CWL

```
cwlVersion: v1.0
class: CommandLineTool
baseCommand: echo
hints:
  ResourceRequirement:
    coresMin: 1
    ramMin: 100
stdout: output.txt
inputs:
  message:
    type: string
    inputBinding:
      position: 1
outputs:
  output:
    type: stdout
```

PBS job script

Approaches towards CWL in ARCs

1. Wrap a script
2. Wrap a CLI tool
3. Reuse an existing CWL document (command line tool or full workflow)
4. ...

Reusability: Simply import an existing CWL

- e.g. from one ARC to another

ARC–CWL tutorials

- Knowledge Base: https://nfdi4plants.github.io/nfdi4plants.knowledgebase/guides/arc-cwl/

Outlook

- Publish to WorkflowHub
- Execute Galaxy (.ga) workflows