Building reusable bioimaging workflows with Nextflow

Sebastian Gonzalez Tirado

GloBIAS
Global BioImage Analysts' Society

neubias
network of european bioimage analysts

German BioImaging
Gesellschaft für Mikroskopie und Bildanalyse

CZI

RIKEN

Shout-outs

- James Fellow Yates (Leibniz-HKI / MPI-EVA)
- Christian Tischer (EMBL)

Outcomes of this talk

- What are workflow managers and why should I care?
- Examples of complex bioimaging workflows orchestrated with Nextflow.
- Practical recommendations to create your own workflows.

Have you ever faced one of these problems?

Discovering an exciting new tool

S: “Oh cool, let’s go to the repo and check it out!”

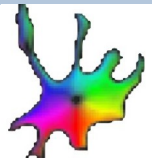

nature biotechnology ARTICLES

Cell segmentation in imaging-based spatial transcriptomics

Viktor Petukhov[1,2], Rosalind J. Xu[3,4,5], Ruslan A. Soldatov[1], Paolo Cadinu[3,4], Konstantin Khodosevich[1], Jeffrey R. Moffitt[3,4] and Peter V. Kharchenko[3,4,6]

...could not even install it.

Repetitive use of tools


Make sure your work is reusable

h-index from 0 to hero

Adapting slowly into standards

nf-core

Only one rule for bioimage analysts club: be lazy

What are workflow managers?

Like orchestra directors managing the flow of your pipeline/symphony.

What do they do?

Workflow managers provide a framework for the creation, execution, and monitoring of a pipeline. <...>

They simplify pipeline development, optimize resource usage, handle software installation and versions, and run on different compute platforms, enabling workflow portability and sharing.

Wratten et al. (2021) Nature Methods

Other benefits

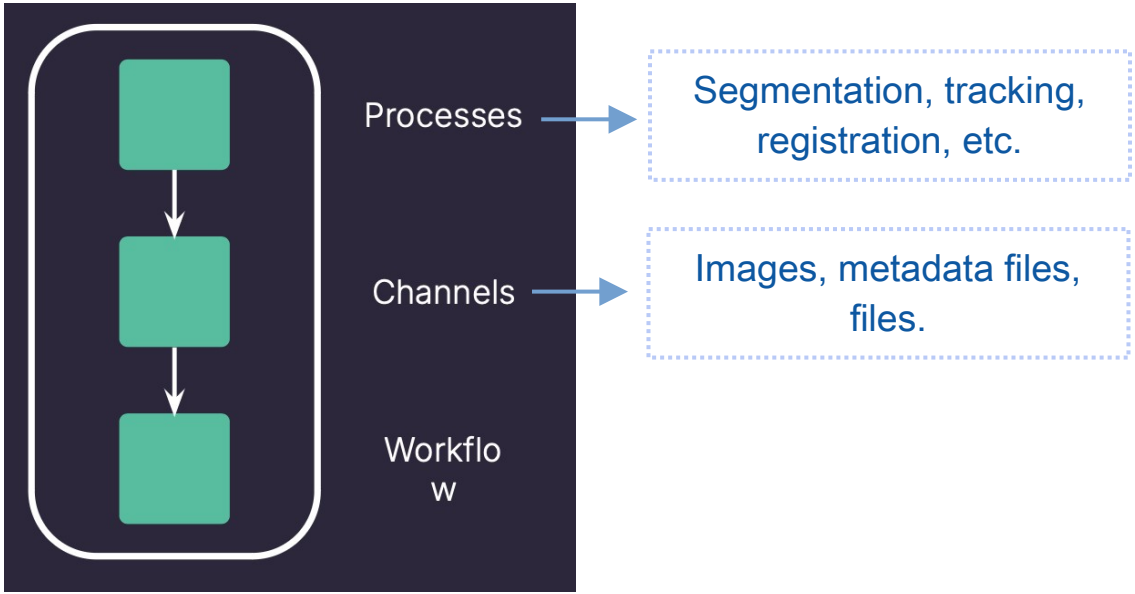
You also benefit from

REUSABLE

- Increased **portability**
- More **efficiency:**
  - Less headaches from installation
  - Better resource management and monitoring
  - Extremely parallelised

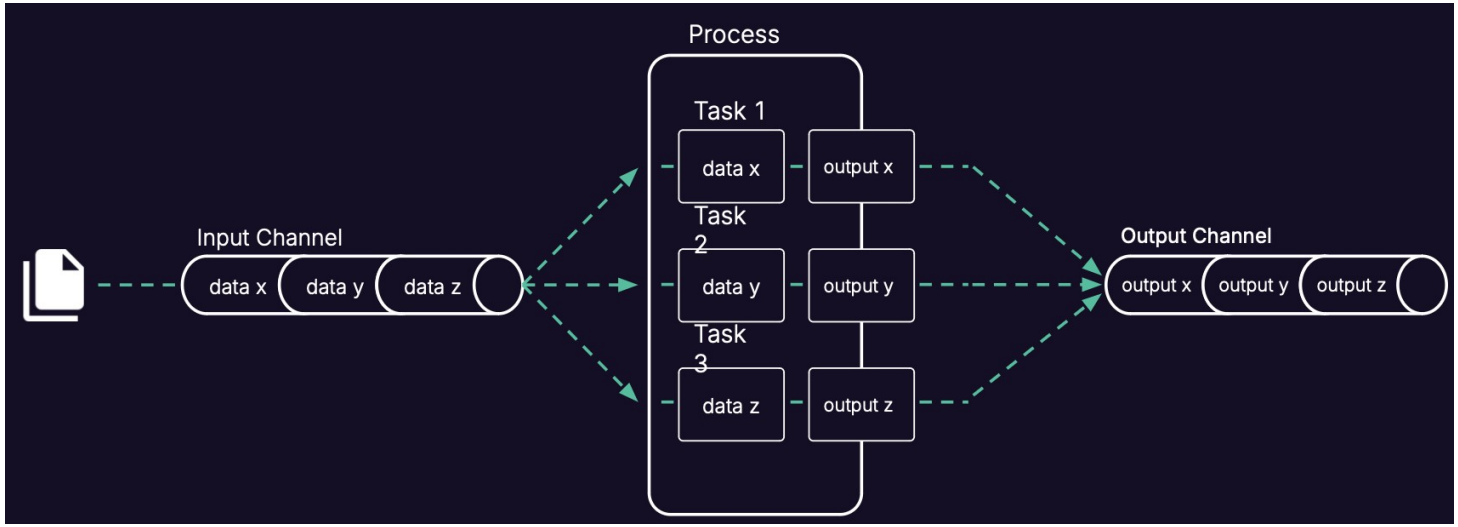
Getting more technical

Nextflow is a reactive workflow framework and a programming DSL



Getting more technical

Nextflow is a reactive workflow framework and a programming DSL



In line with our first rule: do not reinvent the wheel

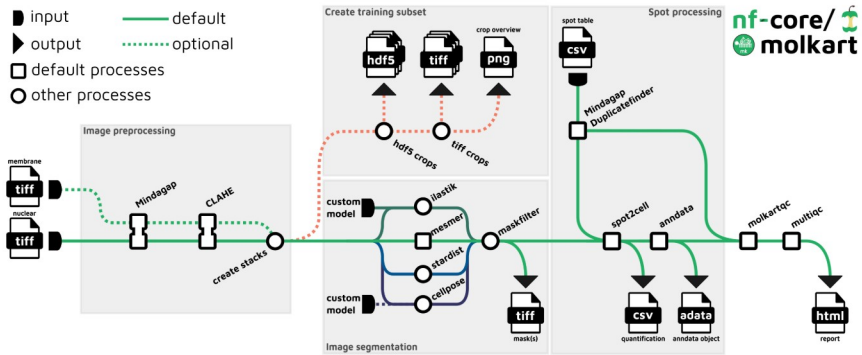
nf-core

A **community** effort to collect a curated set of analysis pipelines built using Nextflow.

nf-core

Introduction

nf-core/molkart is a pipeline for processing Molecular Cartography data from Resolve Bioscience (combinatorial FISH). It takes as input a table of FISH spot positions (x,y,z,gene), a corresponding DAPI image (`TIFF` format) and optionally an additional staining image in the `TIFF` format. nf-core/molkart performs end-to-end processing of the data including image processing, QC filtering of spots, cell segmentation, spot-to-cell assignment and reports quality metrics such as the spot assignment rate, average spots per cell and segmentation mask size ranges.



>_ run with

```
nf-core pipelines launch nf-cor
```

nf-core | Nextflow | Seqera Platform

subscribers	stars
163	13
open issues	**open PRs**
11	2
last release	**last update**
8 months ago	8 months ago

included modules

cellpose | deepcell_mesmer | ilastik_multicut | ilastik_pixelclassification | mindagap_duplicatefinder | and 3 more modules

included subworkflows

utils_nextflow_pipeline | utils_nfcore_pipeline | utils_nfschema_plugin

contributors



Adapting to nf-core specs takes time

You can also just write your own pipline:

https://github.com/BioImageTools/iss-nf

iss-nf: A Nextflow-based end-to-end *in situ* sequencing decoding workflow

Nima Vakili, Sebastián González-Tirado, Nils Kurzawa, Dmytro Dvornikov, Zeinab Mokhtari, Frank Wippich, Giovanna Bergamini, Rainer Pepperkok, Christian Tischer, Luis A. Vale-Silva

doi: https://doi.org/10.1101/2025.10.16.682795

Main components of a nextflow script

```
images = Channel.fromPath("${params.inputDir}/*.tif")
    .map { file -> tuple(file.baseName, file) }

workflow = {
    labels = SEGMENT ( images )
    MEASURE ( images.join( labels ) )
}
```

```
workflow = {
    labels = SEGMENT ( images )
    MEASURE ( images.join( labels ) )
}
```

```
process SEGMENT {
    input:
    path(image)
    output:
    path("*.tif")

    script:
    """
    cellpose --input ${image} --pretrained_model cyto --save_tif
    """
}

process MEASURE {
    publishDir '${params.outputDir}', mode: 'copy'
    input:
    tuple path(image), path(labels)
    output:
    path("*.csv")

    script:
    """
    fiji --run measure.groovy "${image}, ${labels}, ${imageID}.csv"
}
```

How do we run it?

$ nextflow run your_nextflow_script.nf –inputDir /path/2/images

```
❯ nextflow run practice_nextflow.nf --inputDir /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2
Nextflow 25.04.8 is available - Please consider updating your version to it

N E X T F L O W  ~  version 24.10.4

Launching `practice_nextflow.nf` [curious_lavoisier] DSL2 - revision: 4063a8a561

executor >  local (10)
[1b/91e239] SEGMENT (10) [100%] 10 of 10 ✔
Completed at: 18-Oct-2025 08:48:55
Duration    : 16m 16s
CPU hours   : 1.6
Succeeded   : 10
```

Some of the most important aspects to consider

Configuration files (my_file.config):

```
process {
    executor = 'slurm'
    queue = 'htc-el8'
    cpus = 8
    memory = 16.GB
    time = 20.min
    container = "docker://segonzal/fish_analysis:0.0.3"

    withLabel: 'long' {
        cpus = 8
        memory = 32.GB
        container = "docker://segonzal/fish_analysis:0.0.3"
    }

    withLabel: 'registration' {
        cpus = 4
        memory = 32.GB
    }
    withLabel: 'min' {
        cpus = 1
        memory = 400.MB
    }
}
```

That is it for the theory!

Before moving on to the exercises, questions???

Familiarize yourself with it

1. Create your first nextflow script ‘practice_nextflow.nf’, define the channel with the input images and print this channel (tip: check the entry for the .view() operator on the **nextflow documentation**).

What you should see:

```
❯ nextflow run practice_nextflow.nf --inputDir /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2
Nextflow 25.04.8 is available - Please consider updating your version to it

 N E X T F L O W  ~  version 24.10.4

Launching `practice_nextflow.nf` [reverent_wiles] DSL2 - revision: 8e3c76088e

[649_img, /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2/649_img.png]
[731_img, /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2/731_img.png]
[659_img, /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2/659_img.png]
[721_img, /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2/721_img.png]
[596_img, /Users/sebgoti/Documents/PhD/Riken_Globias/training_repo/train_cyto2/596_img.png]
```

Familiarize yourself with it

2. Make your first process: use cellpose to segment the images from the channel you created in 1.

Familiarize yourself with it

3. Quantify the average intensity of the cells in the masks. You can create your own script or macro!