



Targeted spatial transcriptomics data processing

## To learn

- What is nf-core/molkart
- How to set up nf-core/molkart for your targeted transcriptomic data
- Example of a pipeline run
- How to read the output
- Where to get more information and support for nf-core/molkart

# What is nf-core/molkart?

nf-core/molkart is a scalable and reproducible bioinformatics best-practise **pipeline for processing targeted spatial transcriptomics** data.

Originally developed for processing Molecular Cartography data, it can also be used for MERFISH, as well as any targeted spatial transcriptomic method that produces TIFF images and spot positions.

# What does the input data look like?

- **Nuclear image** as a TIFF
- Additional image to help with **segmentation** as a TIFF
- **Spot table** as either CSV, TSV or TXT  
with X, Y, (Z), and gene columns

# Advantages of Nextflow and nf-core

## Fully featured pipelines

### Documentation

Extensive documentation covering installation, usage and description of output files ensures that you won't be left in the dark.



### Portable and reproducible

Pipelines follow best-practices to ensure maximum portability and reproducibility. The large community makes the pipelines exceptionally well tested and easy to run.



### Stable Releases

nf-core pipelines use GitHub releases to tag stable versions of the code and software, making pipeline runs totally reproducible.



### CI Testing

Every time a change is made to the pipeline code, nf-core pipelines use continuous-integration testing to ensure that nothing has broken.



### Cloud-ready

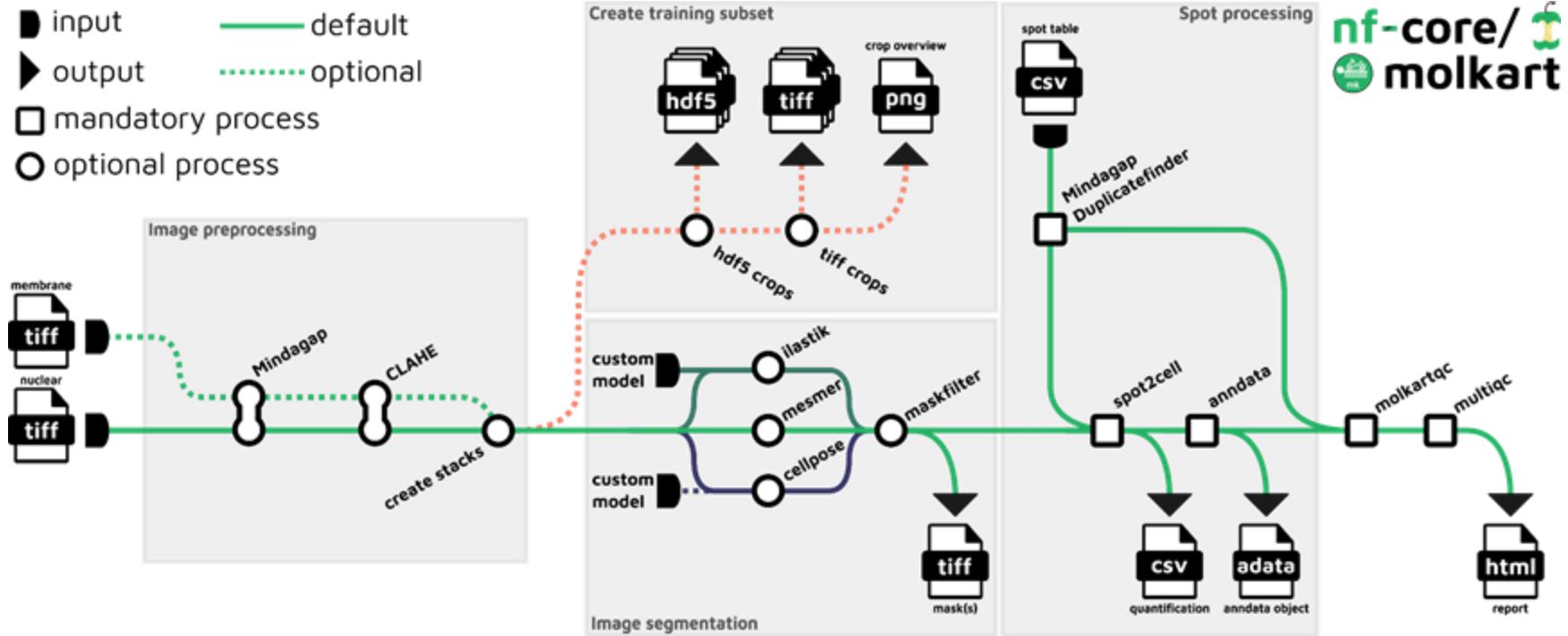
Pipelines are tested on AWS after every release. You can even browse results live on the website and use outputs for your own benchmarking.



### Packaged software

Pipeline dependencies are automatically downloaded and handled using Docker, Singularity, Conda or others. No need for any software installations.



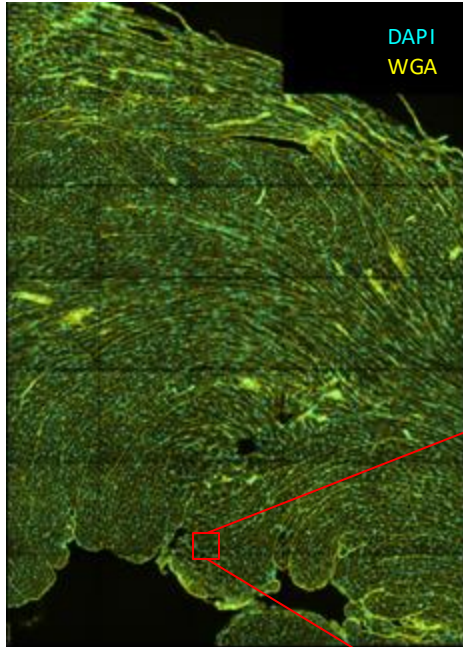


# Samplesheet

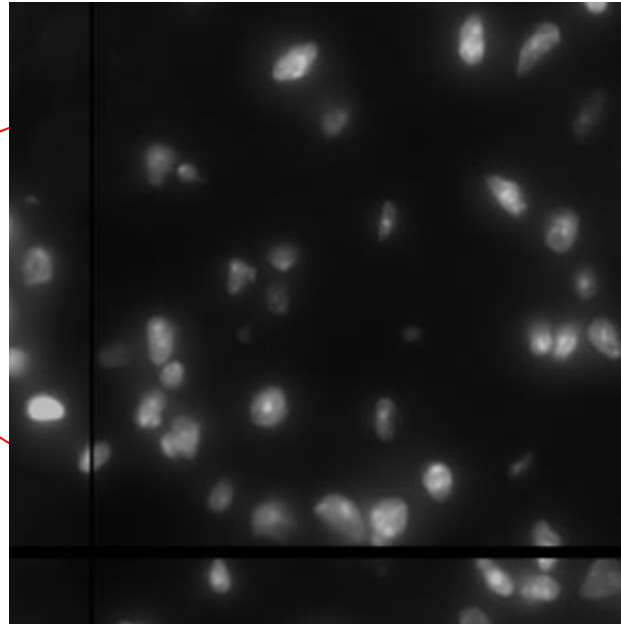
sample	nuclear_image	spot_table	membrane_image
sample_control_r1_s1	/images/sample_control_r1_s1.DAPI.tiff	/spots/sample_control_r1_s1.spots.tsv	/images/sample_control_r1_s1.WGA.tiff
sample_control_r2_s1	/images/sample_control_r2_s1.DAPI.tiff	/spots/sample_control_r2_s1.spots.tsv	/images/sample_control_r2_s1.WGA.tiff
sample_control_r1_s2	/images/sample_control_r1_s2.DAPI.tiff	/spots/sample_control_r1_s2.spots.tsv	/images/sample_control_r1_s2.WGA.tiff
...	...	...	...

Presenting on cardiac data from Wünnemann *et al.* 2024

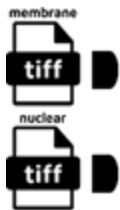
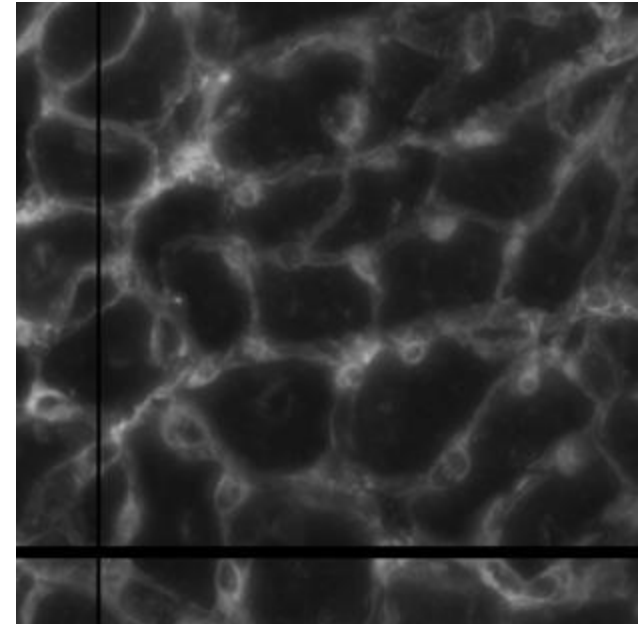
# Image-based inputs



DAPI

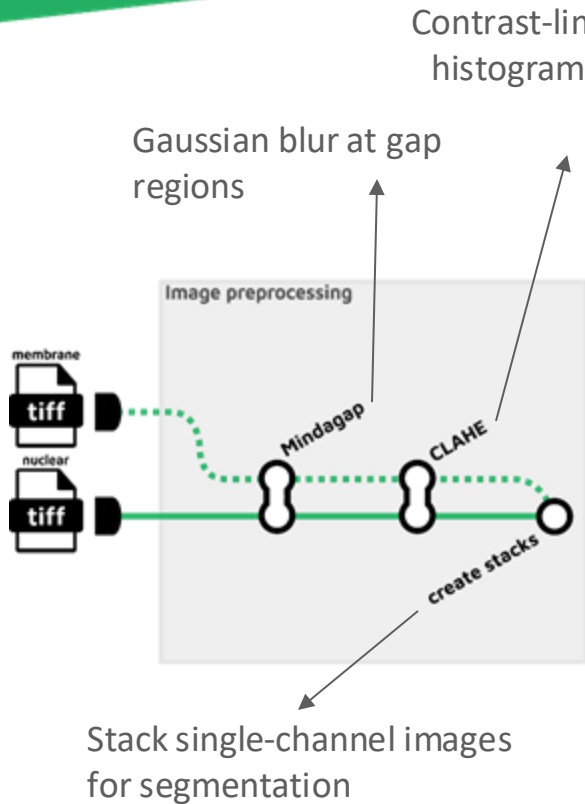


WGA

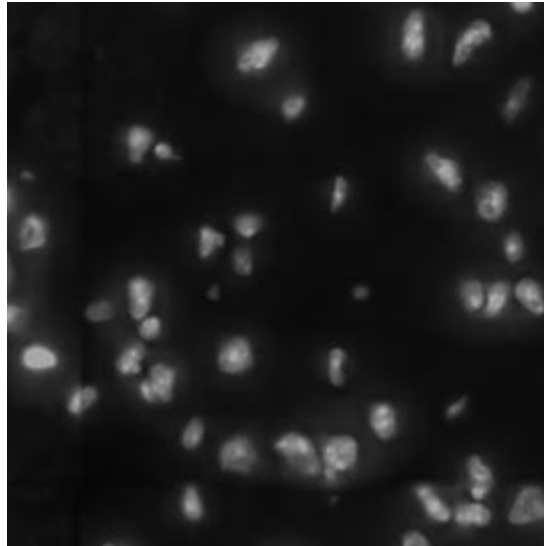




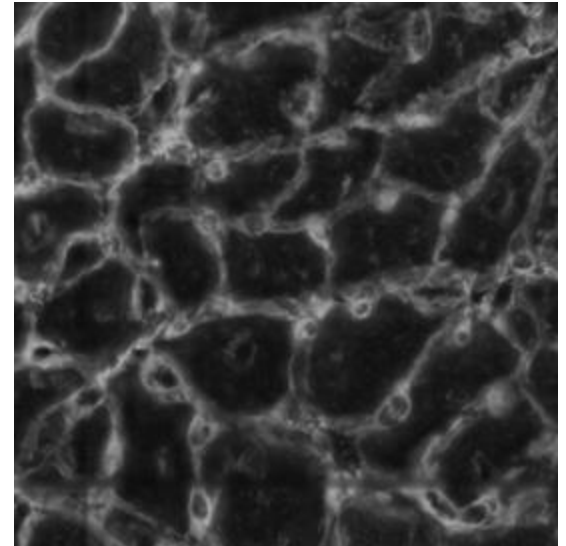
# Image preprocessing



DAPI

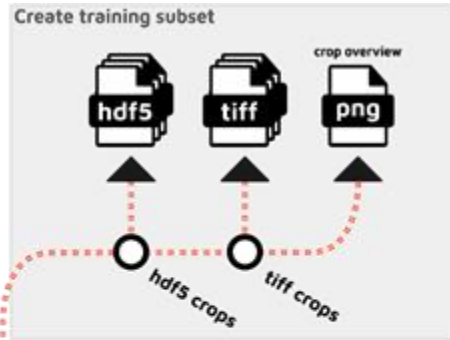


WGA

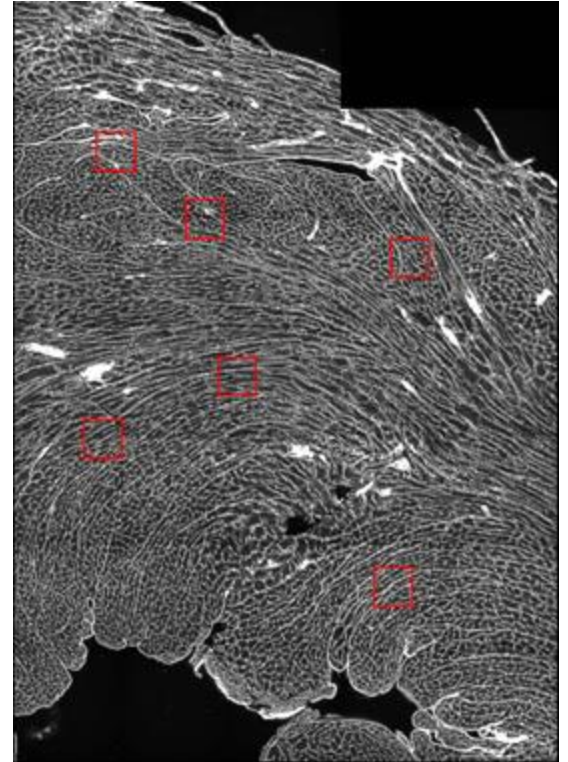


# Create training subset

For atypical cell shapes and sizes, a well-performing **segmentation model** might not be readily available.



Crop output in HDF5 and TIFF formats for interactive model training with ilastik and/or Cellpose 2.



# Segmentation options



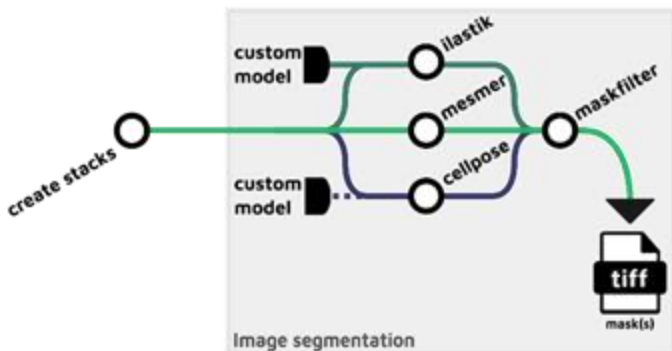
**ilastik**: pixel classifier + Multicut



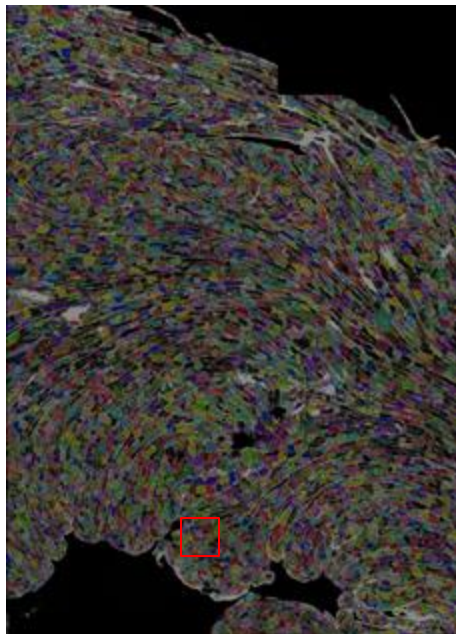
**DeepCell Mesmer**: pre-trained models



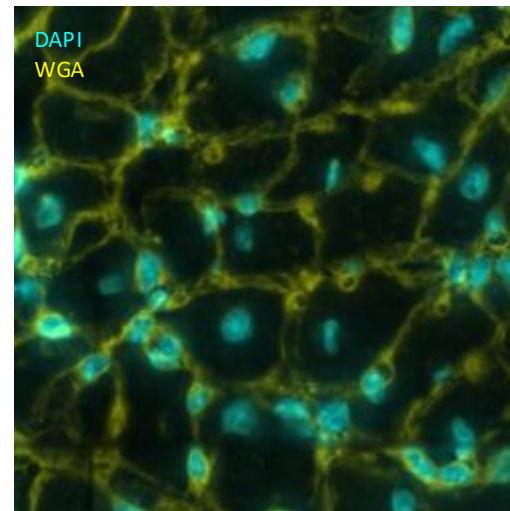
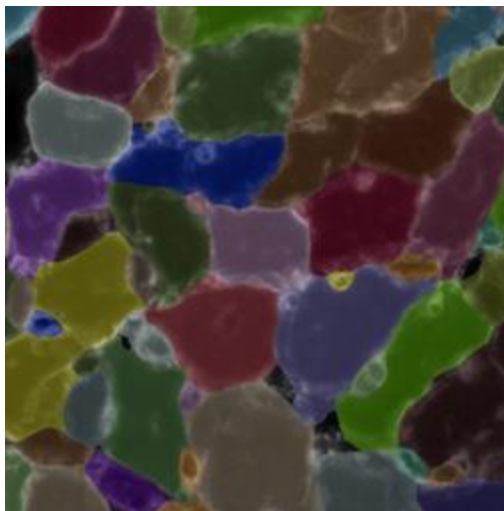
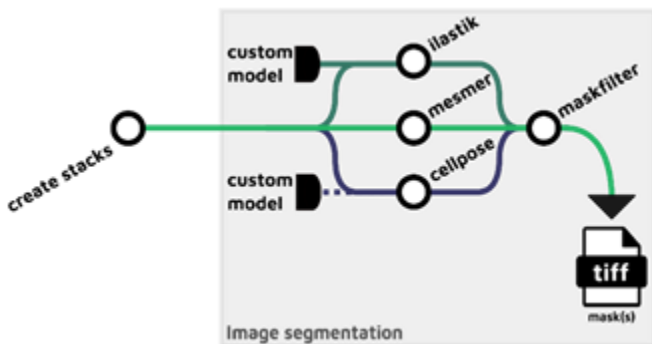
**Cellpose**: pre-trained models / custom model



# Segmentation output



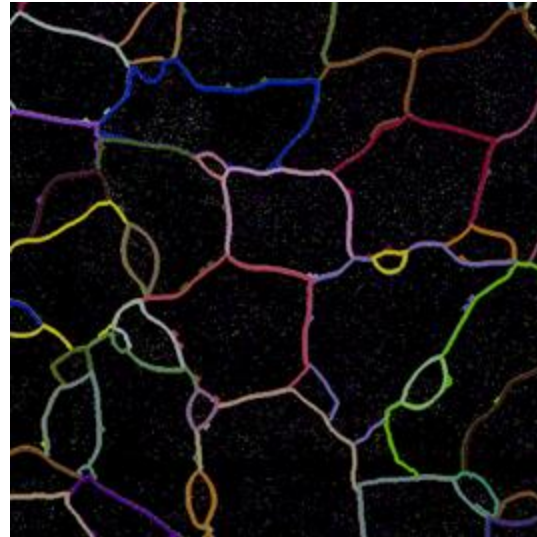
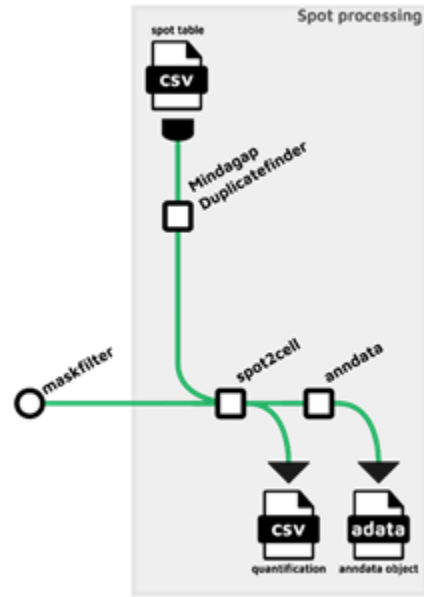
Cellpose 2 custom trained model



# Spot processing

Duplicated gene detection and filtering

Spot assignment to cells based on overlap



Cell-by-gene count matrix with region properties

# Quality control and report

The **MultiQC** report presents gathered sample-specific QC metrics interactively



multiqc v1.18

- QC statistics from segmentation
- nf-core/molkart Methods Description
- nf-core/molkart Software Versions
- nf-core/molkart Workflow Summary



A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

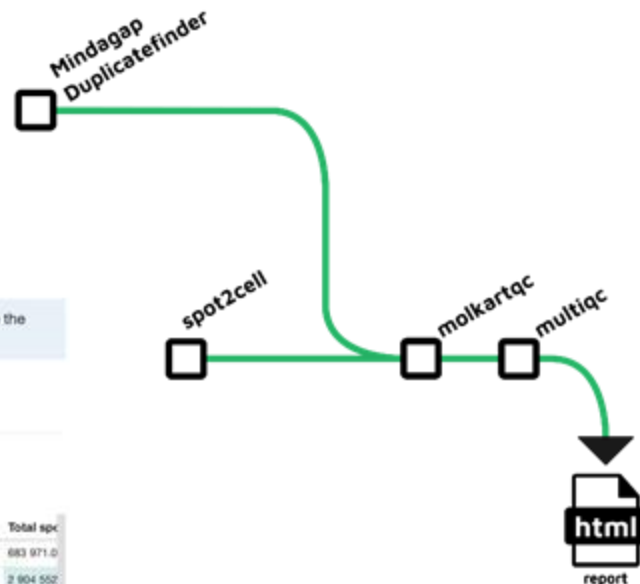
This report has been generated by the nf-core/molkart analysis pipeline. For information about how to interpret these results, please see the [documentation](#).

Report generated on 2024-01-08, 15:11 CET based on data in: /gfs/bwfer/work/vs/hd\_gr294-HIproject\_nfcore\_molkart/data/Molecular\_Cartography/work/1a/e8c36973f137218146082388a17192

## QC statistics from segmentation

Copy table | Configure Columns | Plot | Showing 14 rows and 7 columns.

sample_id	Segmentation method	Total Number of cells	Average cell area	Total number of spots	Average spots assigned per cell	Total spots
sample_4d_r2_s2_cellpose	cellpose	8 755.0	5 795.1	758 429.0	78.1	683 971.0
sample_4h_r1_s1_fastik	fastik	16 860.0	9 391.3	3 036 550.0	172.3	2 904 552
sample_4h_r1_s1_cellpose	cellpose	15 413.0	8 811.6	3 036 550.0	170.9	2 634 913
sample_2d_r1_s2_cellpose	cellpose	12 105.0	7 487.8	2 241 977.0	189.1	2 046 666
sample_2d_r1_s1_mesmer	mesmer	13 627.0	6 091.6	940 758.0	57.0	777 065.0
sample_2d_r2_s1_mesmer	mesmer	12 341.0	6 605.0	1 055 456.0	58.6	722 999.0
sample_control_r1_s2_cellpose	cellpose	14 202.0	7 544.9	5 187 141.0	326.0	4 629 258
sample_2d_r2_s1_fastik	fastik	23 350.0	5 708.8	1 055 456.0	45.0	1 050 927



# Pipeline outputs

```
├── anndata
│   └── {sample}_{segmentation}.adata
├── clahe
│   ├── {sample}.DAPI_gridfilled_clahe.tiff
│   └── {sample}.WGA_gridfilled_clahe.tiff
├── mindagap
│   ├── {sample}.DAPI_gridfilled.tiff
│   ├── {sample}.WGA_gridfilled.tiff
│   └── {sample}.spots_markedDups.txt
├── multiqc
│   ├── final_QC.all_samples.csv
│   ├── multiqc_data
│   │   ├── multiqc_citations.txt
│   │   ├── multiqc_data.json
│   │   ├── multiqc.log
│   │   ├── multiqc_segmentation_stats_table.txt
│   │   └── multiqc_sources.txt
│   ├── multiqc_plots
│   └── multiqc_report.html
├── pipeline_info
│   ├── execution_report_2024-01-08_15-05-25.html
│   ├── execution_timeline_2024-01-08_15-05-25.html
│   ├── execution_trace_2024-01-08_15-05-25.txt
│   ├── params_2024-01-08_15-11-59.json
│   ├── pipeline_dag_2024-01-08_15-05-25.html
│   └── software_versions.yml
├── segmentation
│   └── filtered_masks
│       └── {sample}_{segmentation}_filtered.tif
├── spot2cell
│   └── cellxgene_{sample}_{segmentation}.csv
└── stack
    └── {sample}_stack.ome.tif
```

Anndata object per sample and segmentation method

CLAHE-adjusted images

Gridfilled images and a spots table with marked duplicates

MultiQC report and relevant files

Information relevant to pipeline execution for reproducibility including report with resource usage, timeline, dag, parameters and software versions

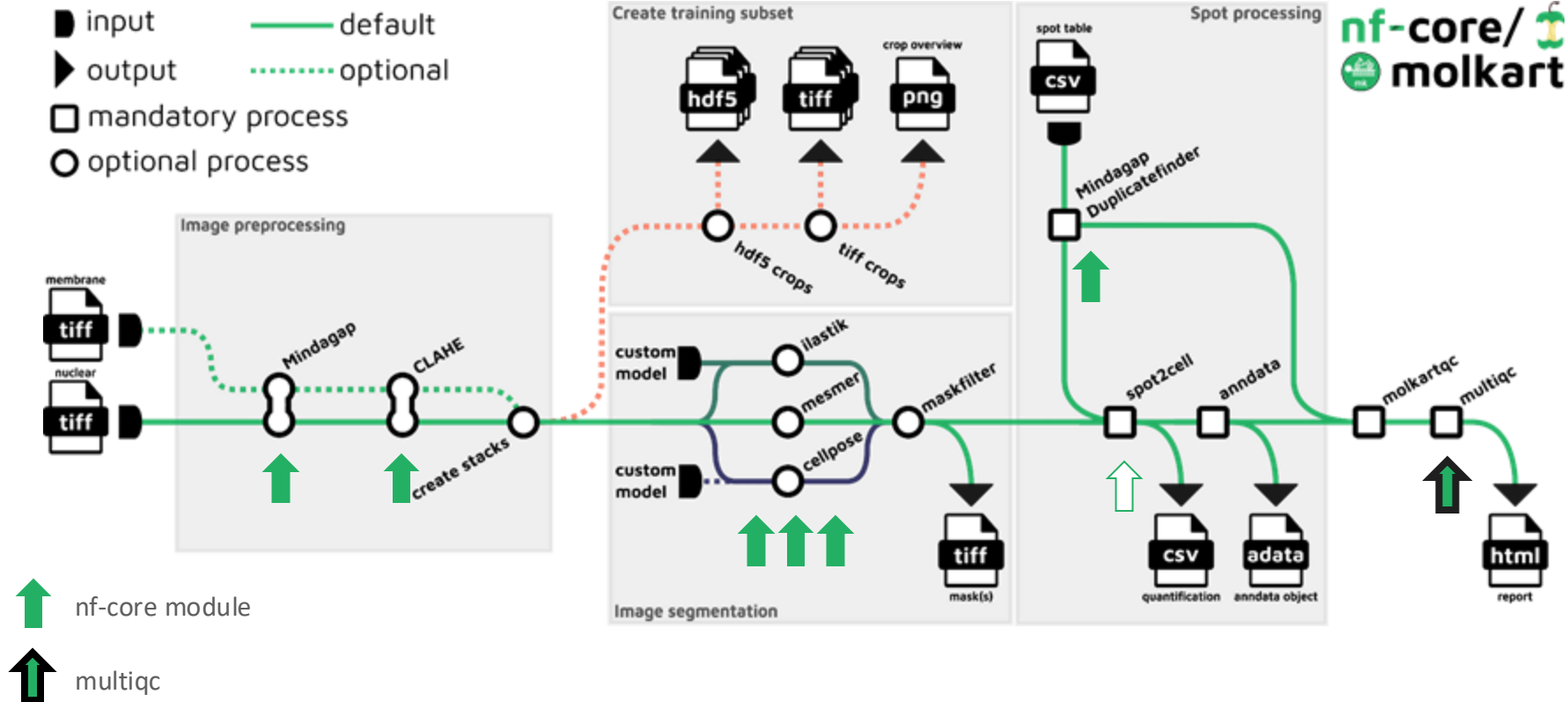
Segmentation masks named based on sample and segmentation method

Cell-by-gene quantification files

OME-TIF stacked processed input images



# Contributions to nf-core





## **SpatialData** support

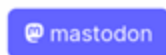
More segmentation options including spot-based methods such as **BOMS** and **Baysor**

**GPU** support for segmentation

# Slack and Github

For support feel free to open an issue on Github or ask questions in the #molkart Slack channel!

We use a few different tools to organise the nf-core community - you are welcome to join us at any or all!



⚠ All nf-core community members are expected to adhere to the nf-core [code of conduct](#)

✕ If your question is about Nextflow and not directly related to nf-core, please use the [Slack community chat](#) or the [discussion forum](#) on GitHub.

## Slack

💡 If you would like help with running nf-core pipelines, Slack is the best place to start.






Slack is a real-time messaging tool, with discussion split into channels and groups. We use it to provide help to people running nf-core pipelines, as well as discussing development ideas.

Once you have registered, you can access the nf-core slack at <https://nfcore.slack.com/> (NB: No hyphen!)

🔗 Get an invite to nf-core Slack



## To learn





-  What is nf-core/molkart
-  How to set up nf-core/molkart for your targeted transcriptomic data
-  Example of a pipeline run
-  How to read the output
-  Where to get more information and support for nf-core/molkart

# Before we finish...

Come and develop with us!



**Contributors** 4

-  **kbestak** Krešimir Beštak
-  **FloWuenne** Florian Wuenneemann
-  **nf-core-bot** nf-core bot
-  **maxulysse** Maxime U Garcia

## Contributors

The nf-core pipelines and community is driven by many individuals, listed below. This list updates automatically.

Want to see who's working on what? See the [contributor leaderboard](#) on the Statistics page.



# More Info & Acknowledgements



<https://nf-co.re>



[github.com/nf-core](https://github.com/nf-core)



[nfcore.slack.com](https://nfcore.slack.com)



[youtube.com/nf-core](https://youtube.com/nf-core)



[@nf\\_core@mstdn.science](mailto:@nf_core@mstdn.science)



[@nf\\_core](https://twitter.com/nf_core)

Some emojis designed by [OpenMoji](#) — the open-source emoji and icon project. License: [CC-BY-SA 4.0](#)

Financial &  
Infrastructure  
Support:



Gitpod



HackMD

Speaker's  
Support



nf-core/core &  
community



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386



Schapiro  
Lab