

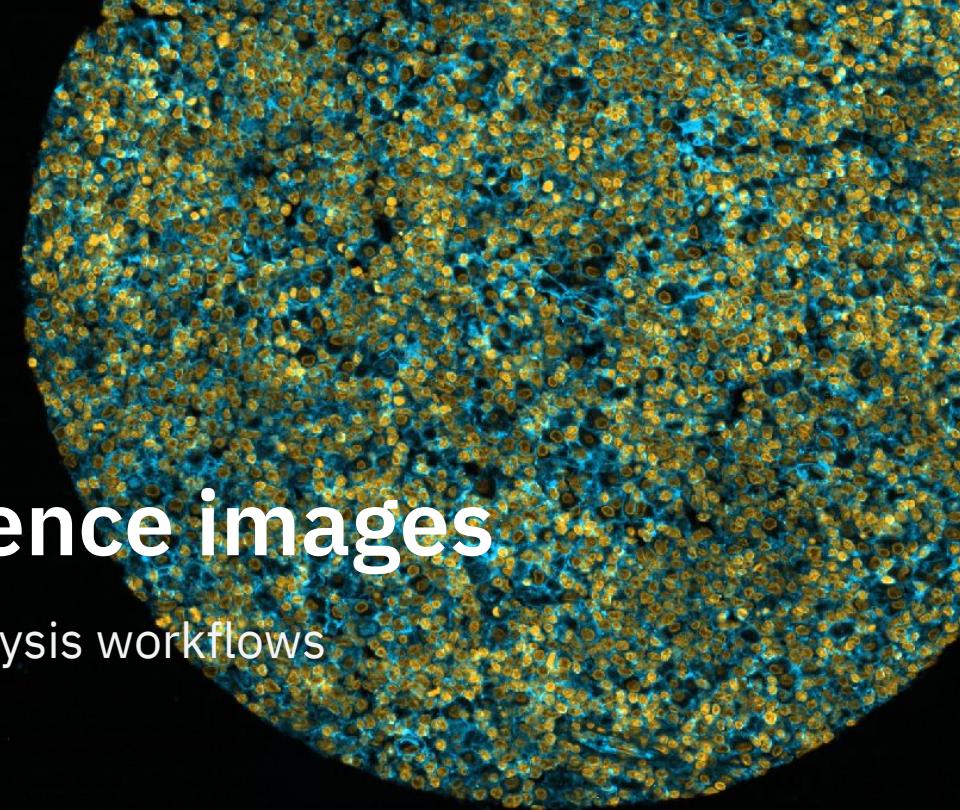
Analysis of highly multiplexed fluorescence images

Developing spatialproteomics data analysis workflows

ECCB Workshop

Matthias Meyer-Bender & Harald Voehringer

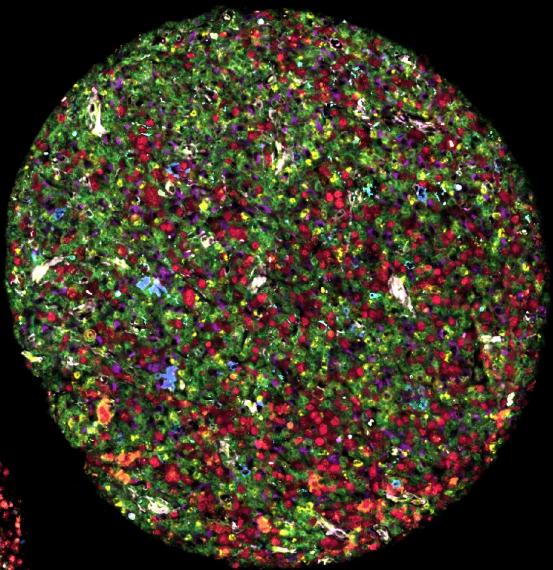
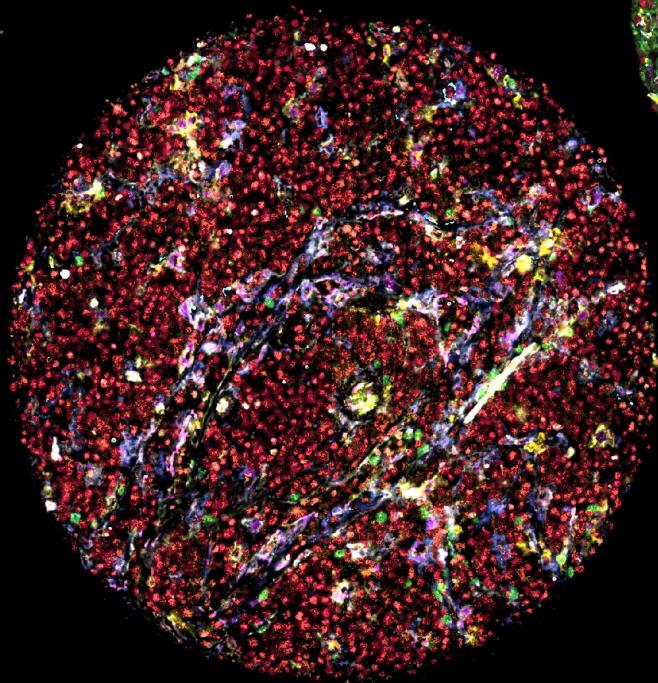
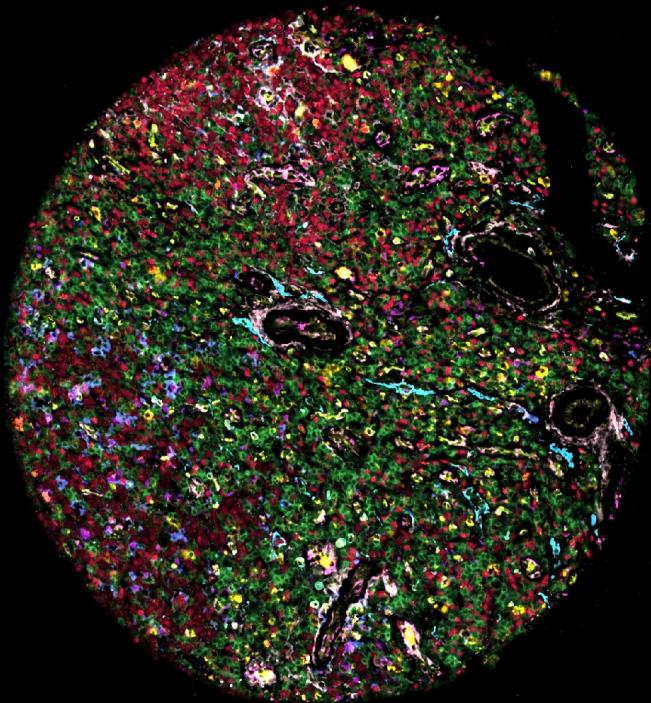
September 16th, 2024



eccb2024

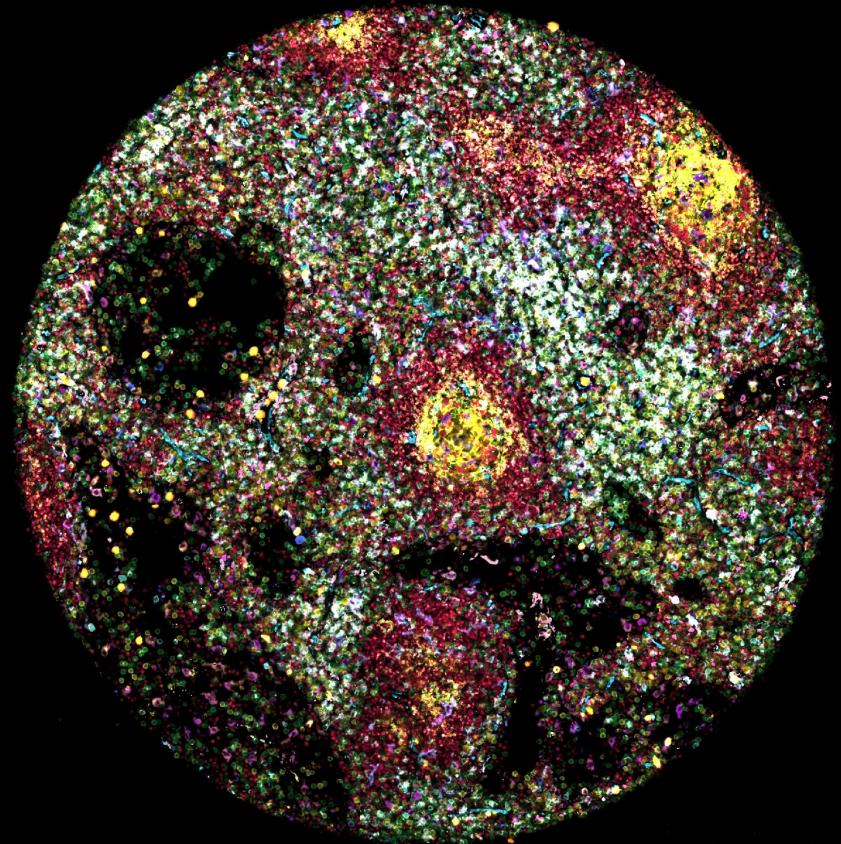
UKD Universitätsklinikum
Düsseldorf





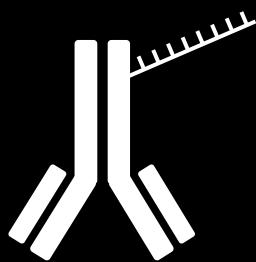
Why spatial omics?

- Biological processes happen in a spatial context
- Examples
 - Developmental biology
 - Physiology
 - Oncology

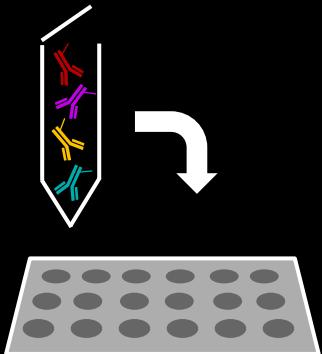


Highly multiplexed fluorescence imaging

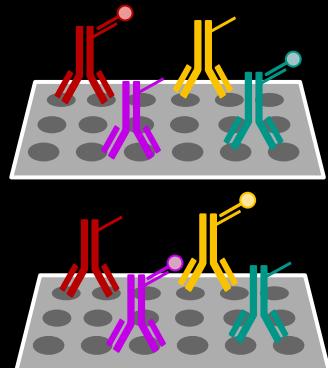
The CODEX technology



Antibody with
conjugated
oligonucleotide



All-in-one staining
with antibody
cocktail



Cyclic imaging using
complementary reporter
oligonucleotides

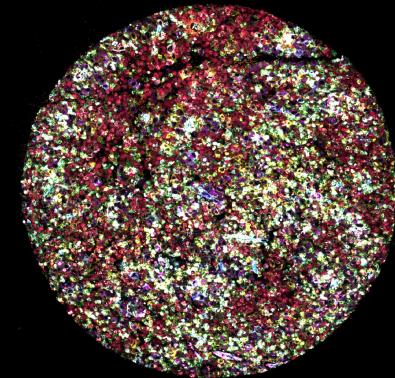
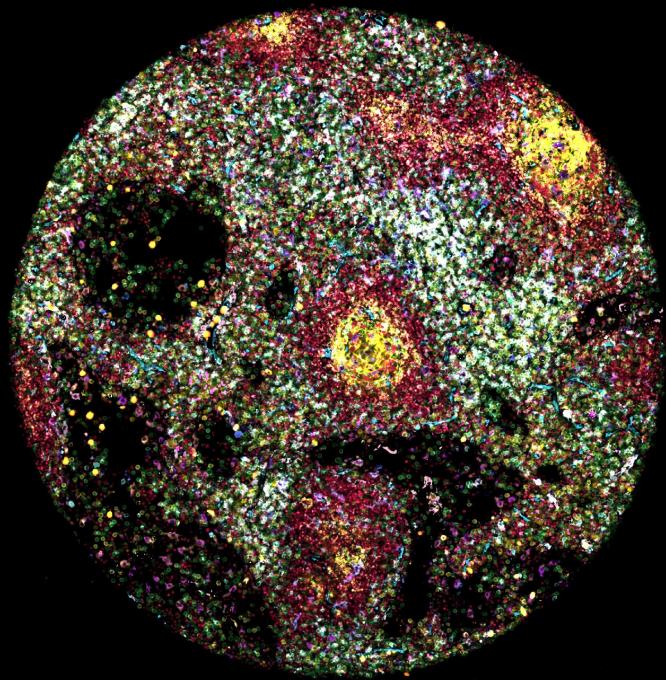


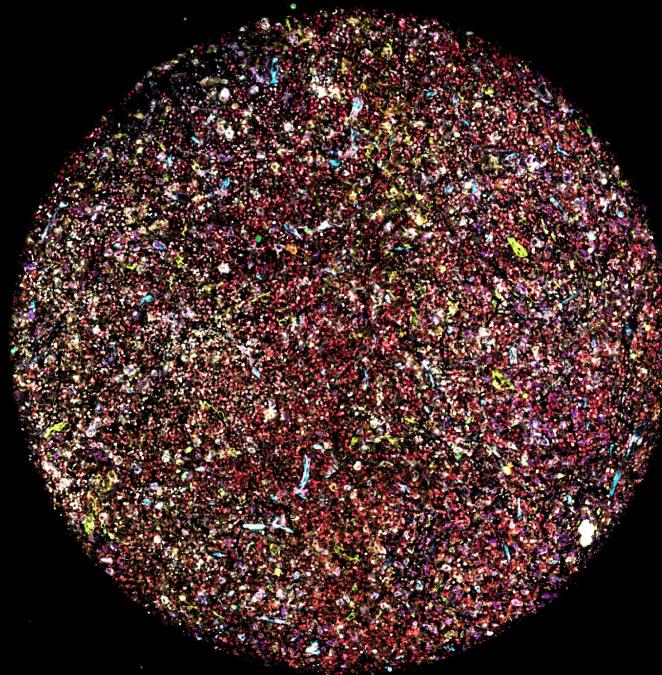
Image with 56
channels

Lymphomas structurally differ from healthy lymph nodes

Lymph node



Diffuse large B cell lymphoma



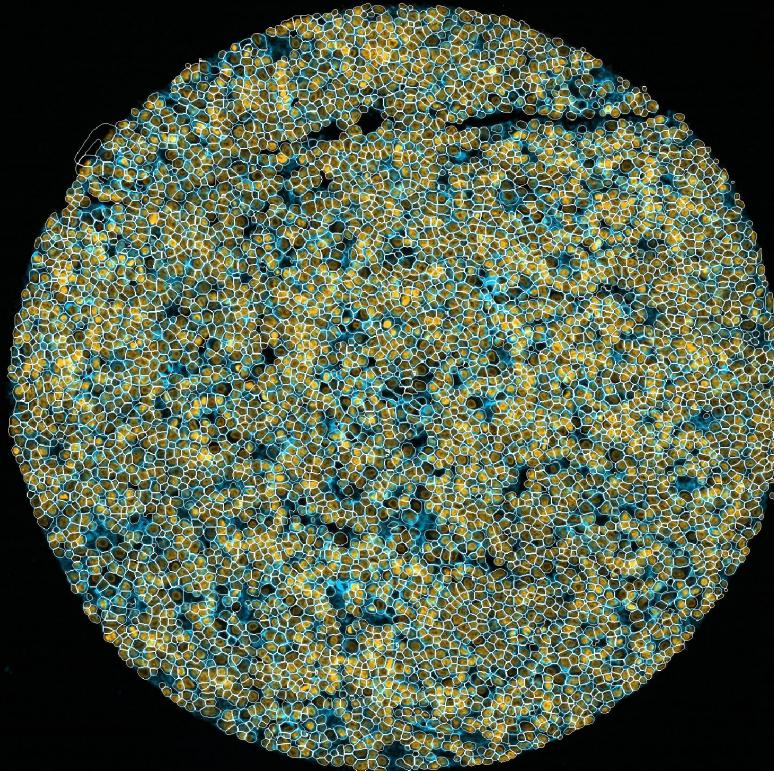
General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
4. Cell type assignment
5. Downstream analysis



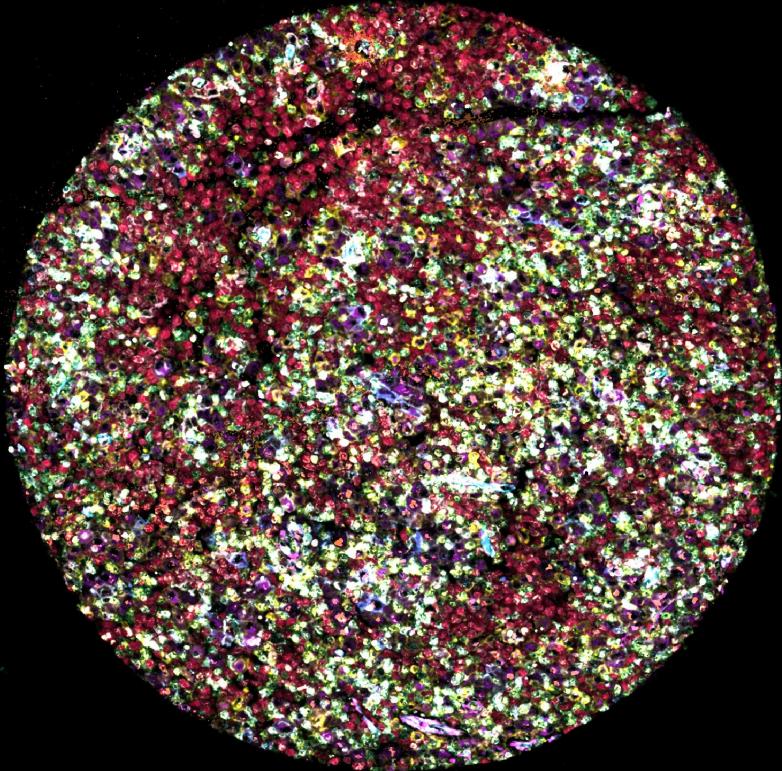
General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
4. Cell type assignment
5. Downstream analysis



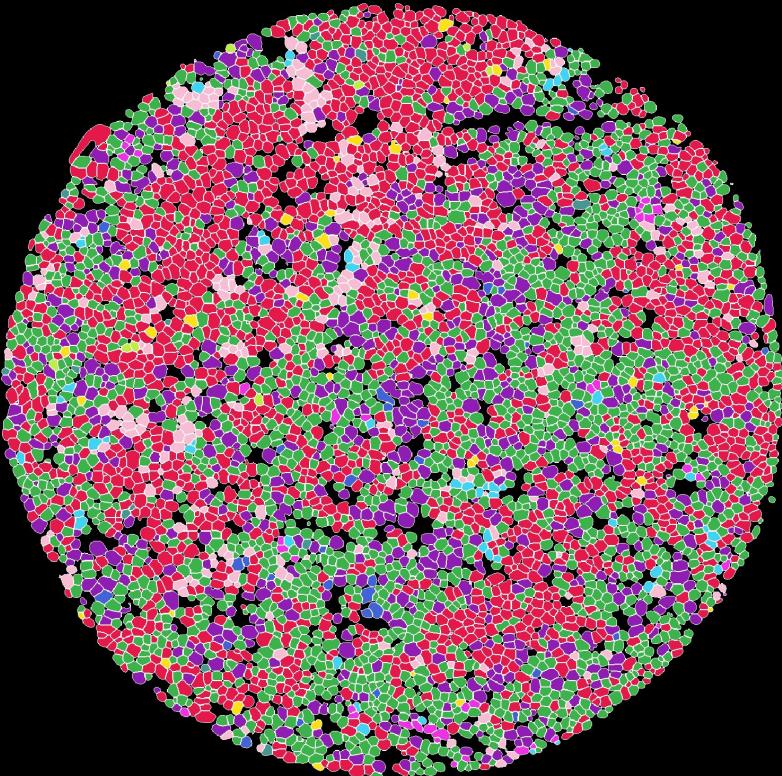
General Analysis Pipeline

1. Cell segmentation
2. **Image processing**
3. **Expression profiling**
4. Cell type assignment
5. Downstream analysis



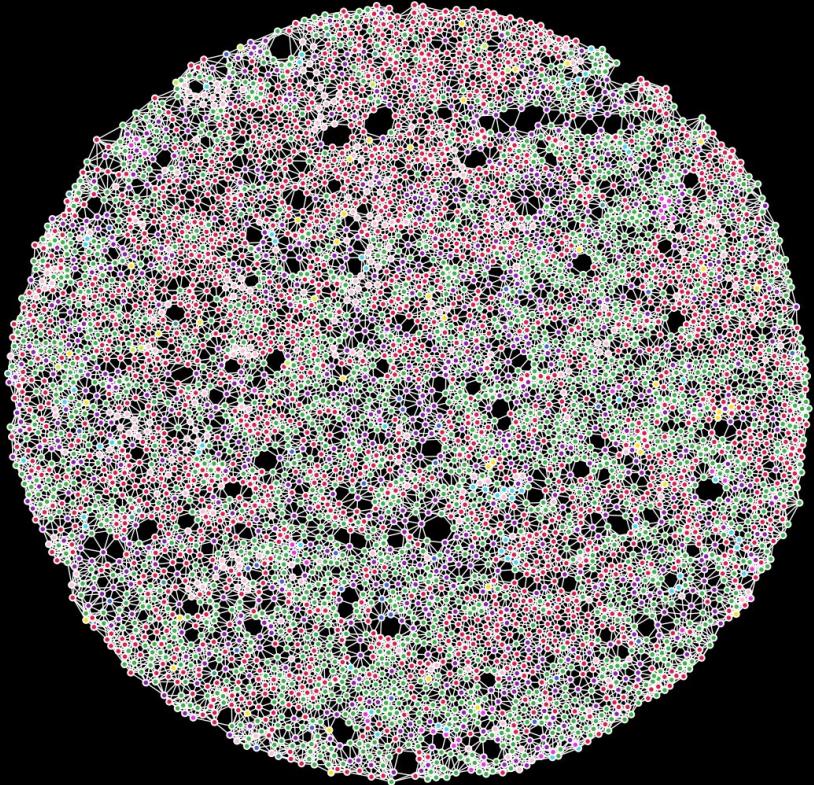
General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
- 4. Cell type assignment**
5. Downstream analysis



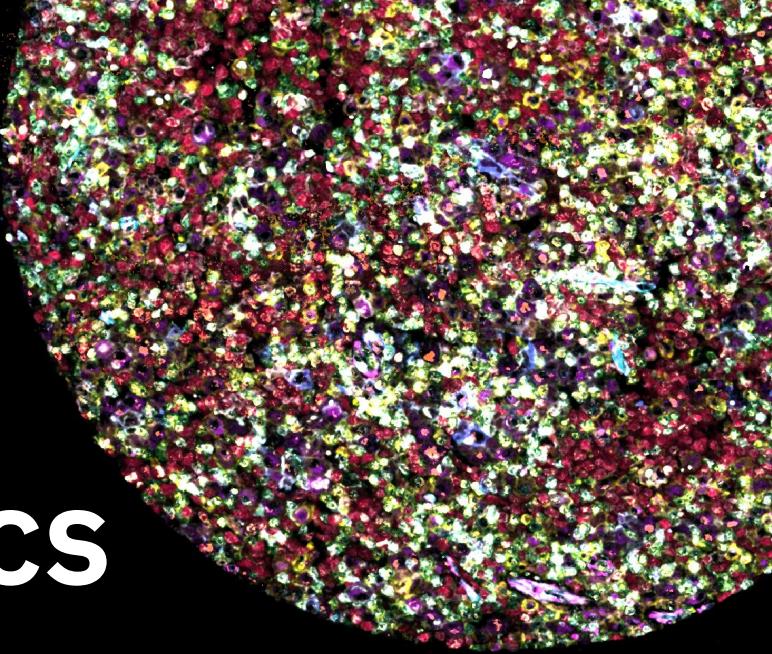
General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
4. Cell type assignment
5. **Downstream analysis**



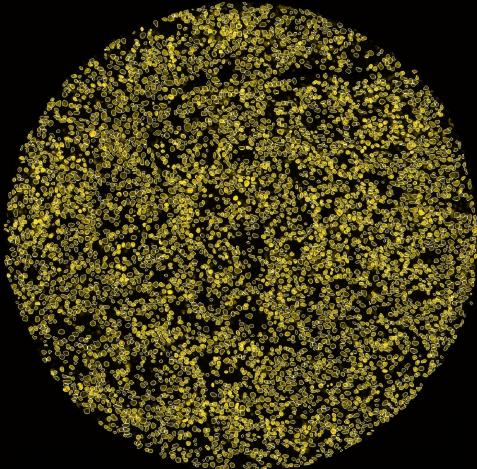
Spatialproteomics

A unified framework for the development of analysis pipelines

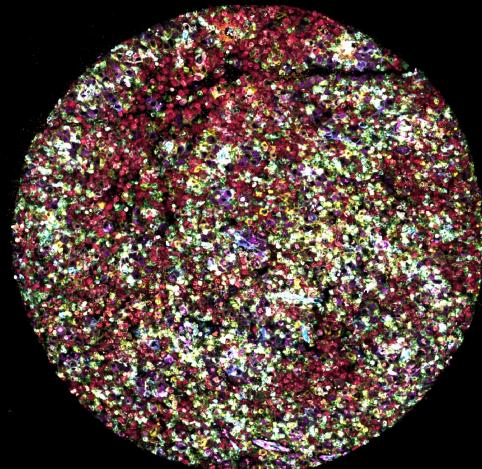


Spatialproteomics streamlines the analysis and facilitates experimentation

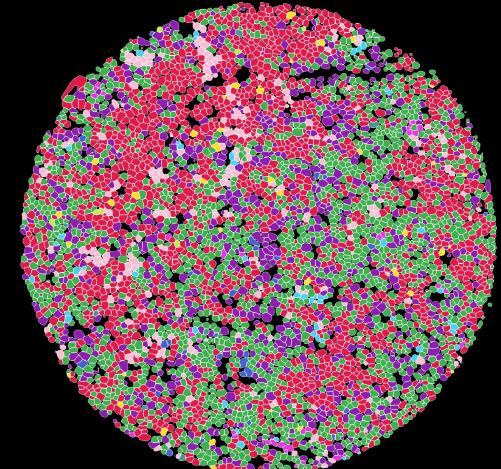
```
image.tl.cellpose(channel='DAPI')
```



```
image.pp.threshold(quantiles)  
.pp.apply(medfilt2d)
```

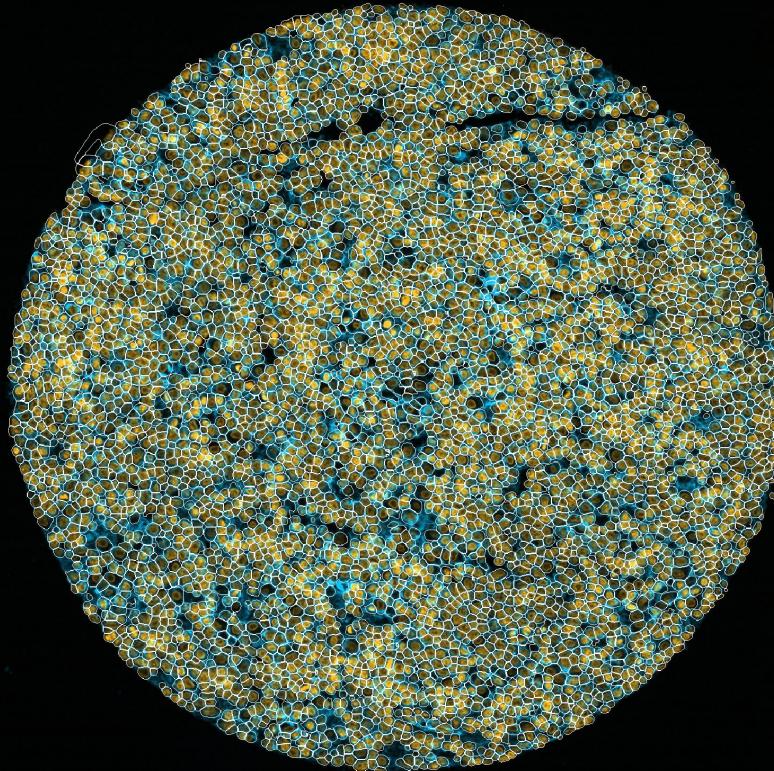


```
image.pp.add_quantification()  
.la.predict_cell_types_argmax(ct_marker_dict)
```



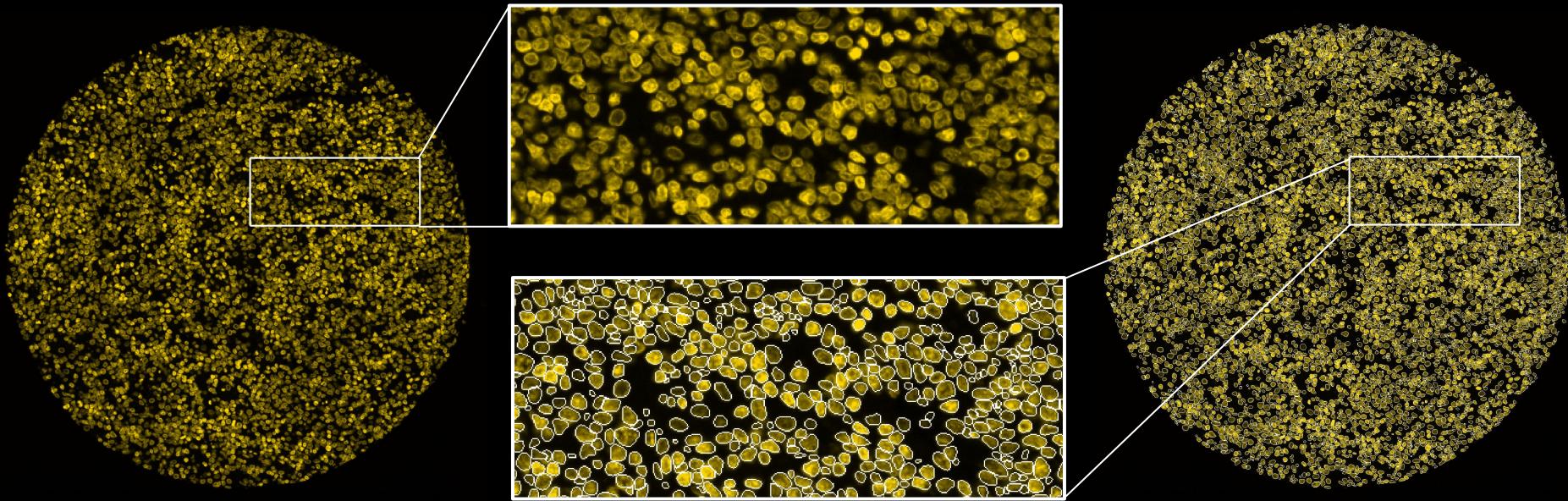
General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
4. Cell type assignment
5. Downstream analysis



Segmentation

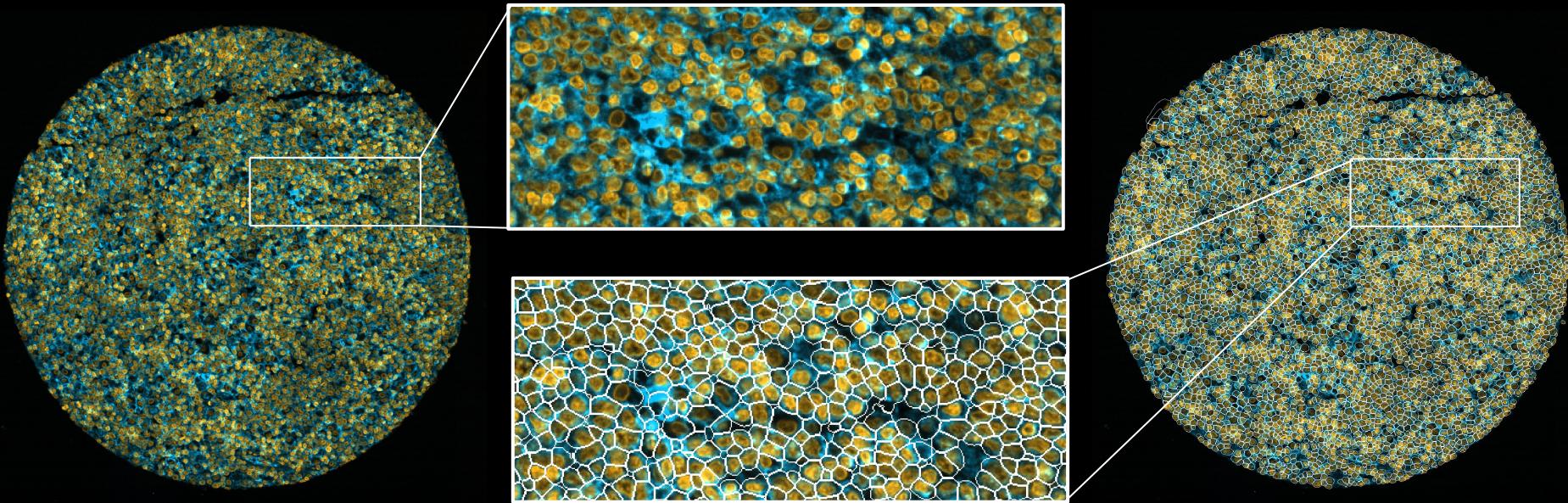
Nuclear segmentation with StarDist



DAPI (nuclear)

Segmentation

Whole cell segmentation with cellpose



General Analysis Pipeline

1. Cell segmentation
2. **Image processing**
3. **Expression profiling**
4. Cell type assignment
5. Downstream analysis

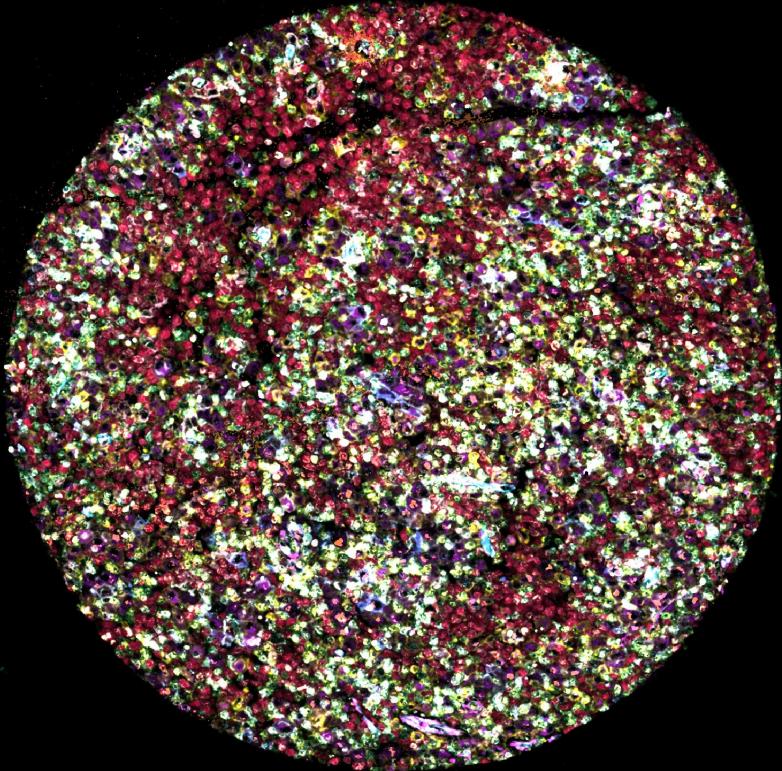


Image processing

Thresholding & median filter

```
image.pp[['PAX5', ... , 'CD56']]  
    .pp['Podoplanin']  
    .pp.threshold(quantile=0.95)  
    .pp.apply(func=medfilt2d)  
    .pl.show()
```

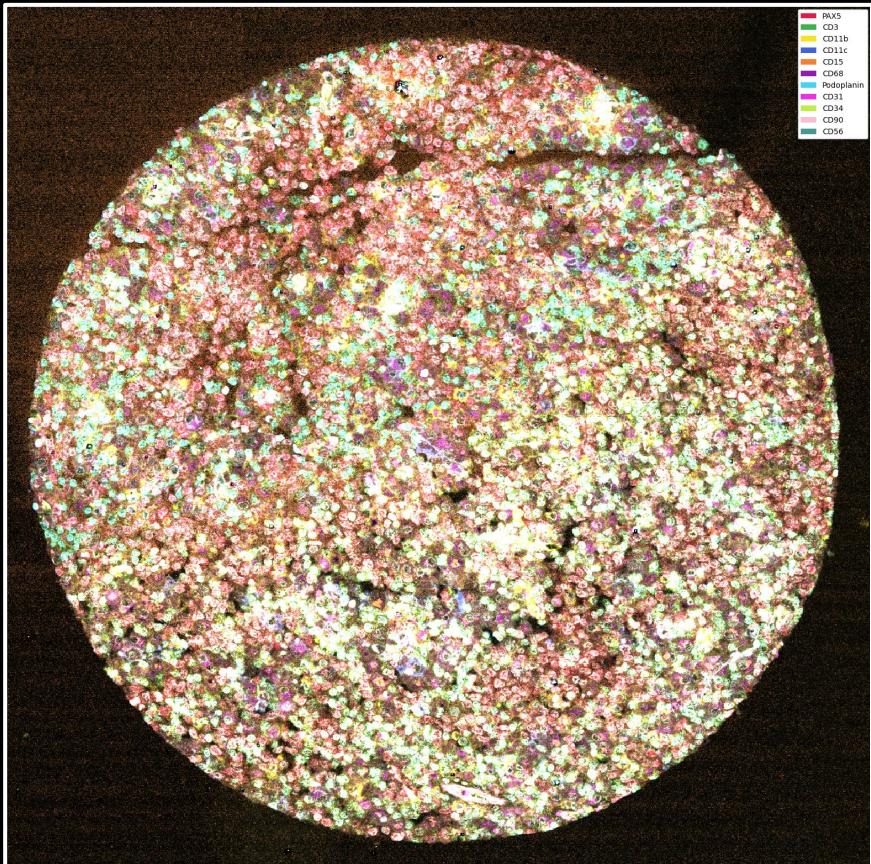


Image processing

Thresholding & median filter

```
image.pp[['PAX5', ... , 'CD56']]  
    .pp['Podoplanin']  
    .pp.threshold(quantile=0.95)  
    .pp.apply(func=medfilt2d)  
    .pl.show()
```

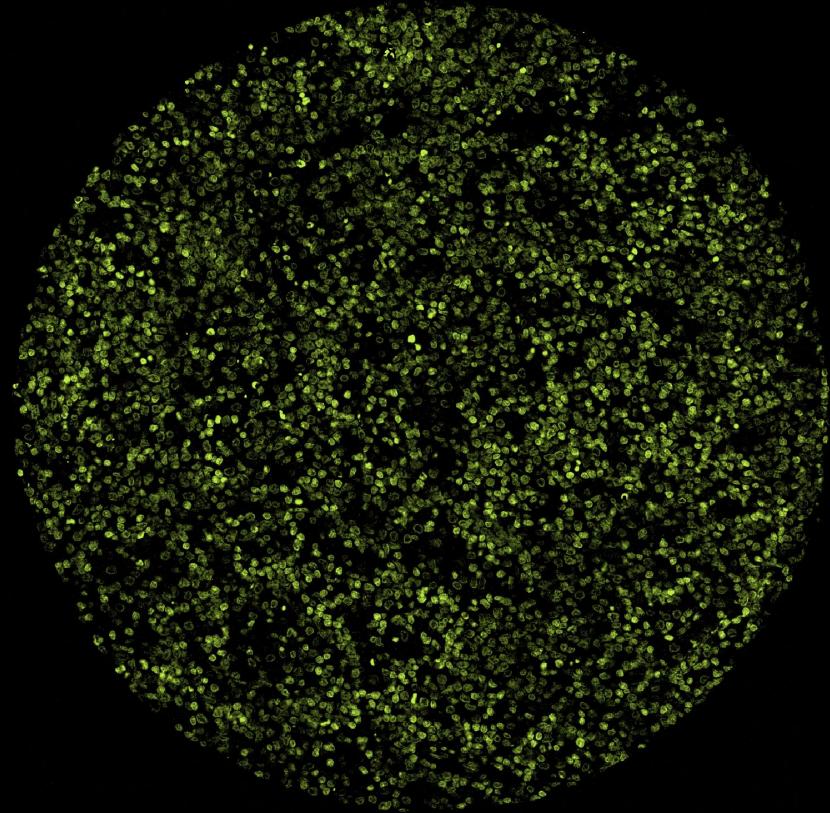


Image processing

Thresholding & median filter

```
image.pp[['PAX5', ... , 'CD56']]  
    .pp['Podoplanin']  
    .pp.threshold(quantile=0.95)  
    .pp.apply(func=medfilt2d)  
    .pl.show()
```

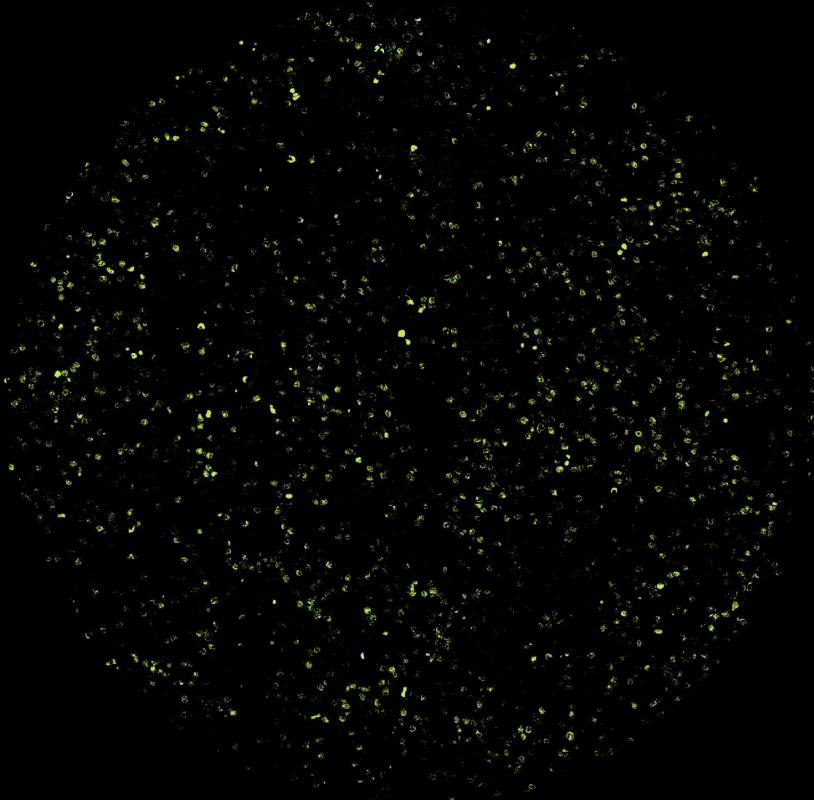


Image processing

Thresholding & median filter

```
image.pp[['PAX5', ... , 'CD56']]  
    .pp['Podoplanin']  
    .pp.threshold(quantile=0.95)  
    .pp.apply(func=medfilt2d)  
    .pl.show()
```

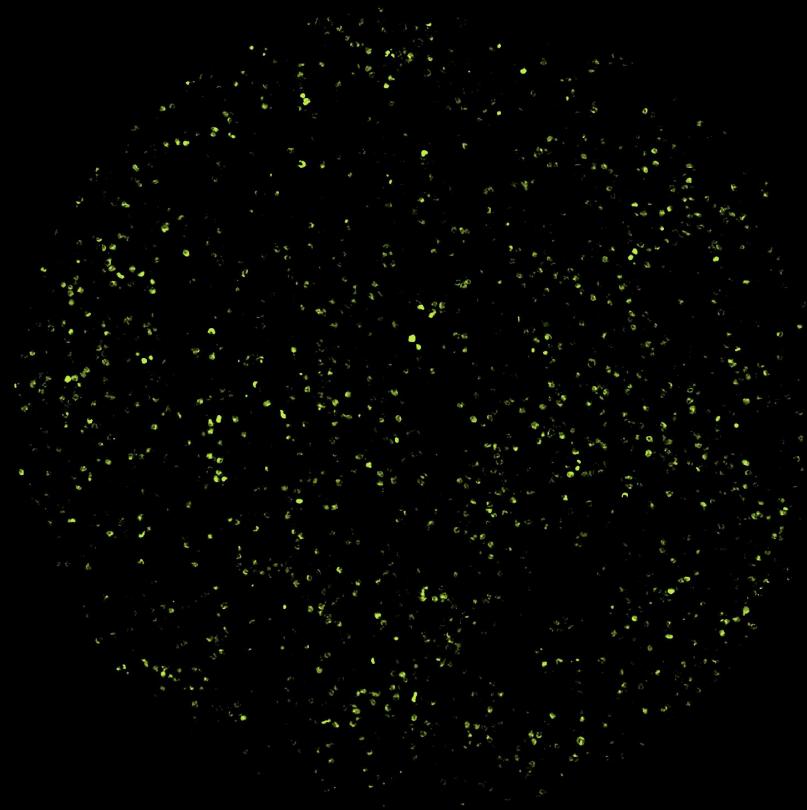
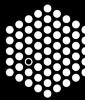
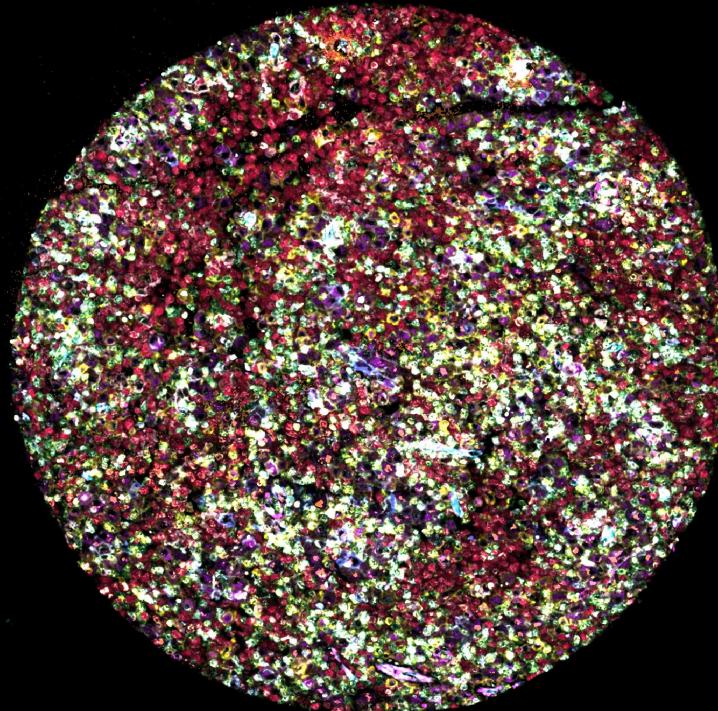


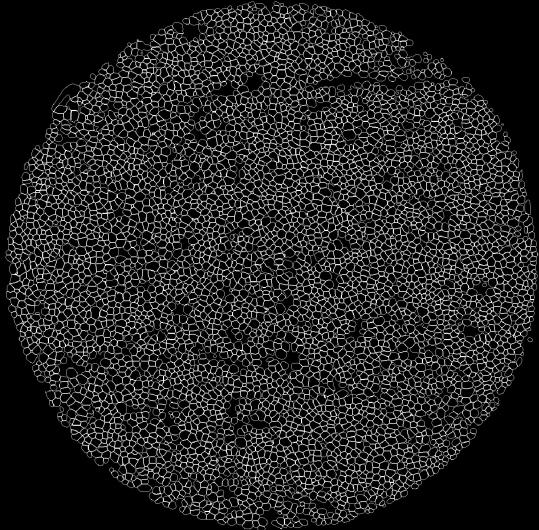
Image processing

Thresholding & median filter



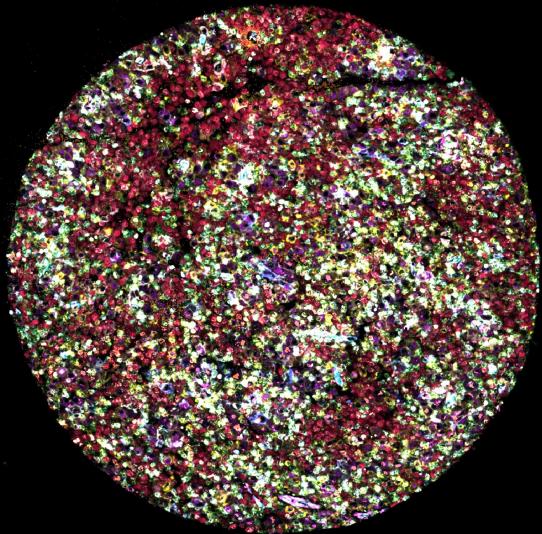
Protein quantification

Segmentation

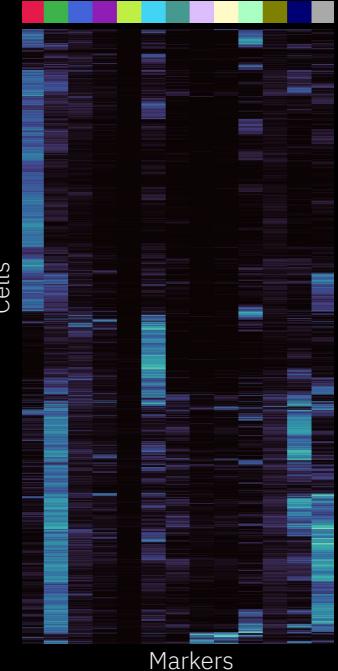


+

Processed image

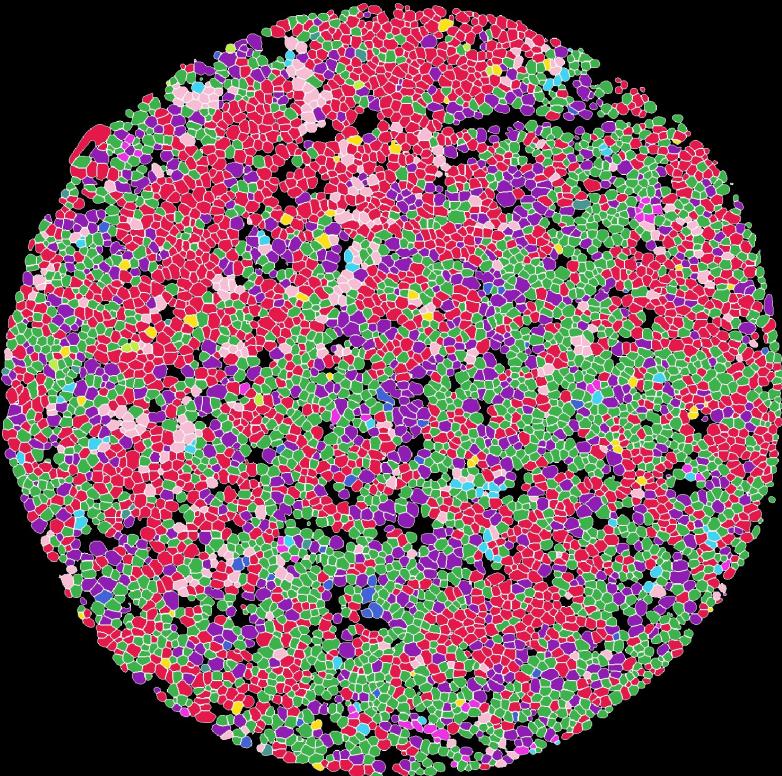


Expression matrix

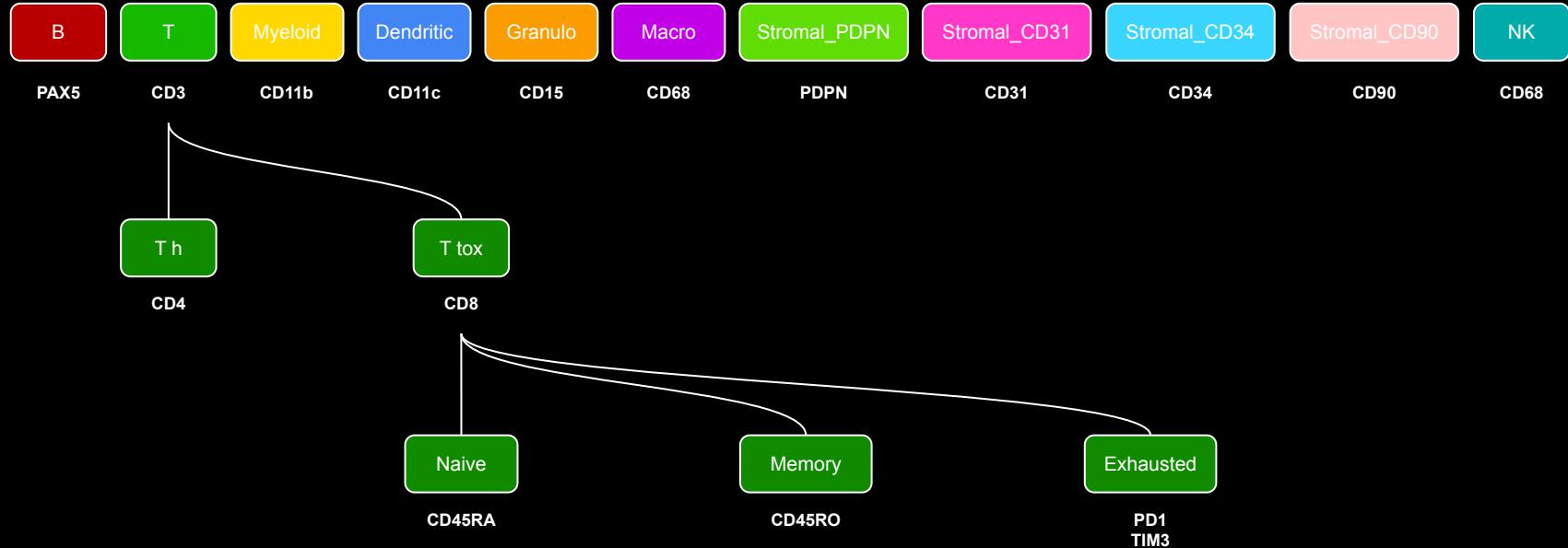


General Analysis Pipeline

1. Cell segmentation
2. Image processing
3. Expression profiling
- 4. Cell type assignment**
5. Downstream analysis



Biologically inspired cell type classification

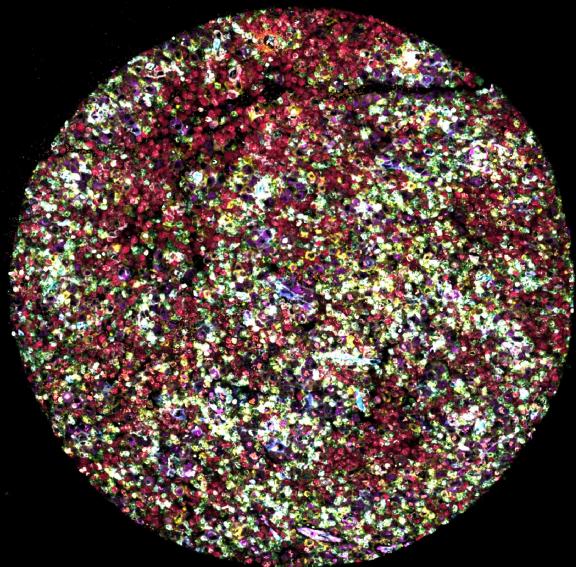


Classes of markers

Mutually exclusive markers vs. functional markers

Mutually exclusive markers

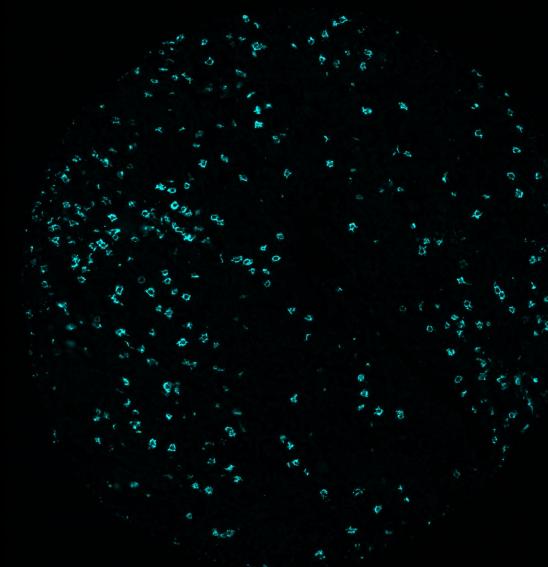
Prediction of major cell types



PAX5
CD3
CD11b
CD11c
CD15
CD68
PDPN
CD31
CD34
CD90
CD56

Functional markers

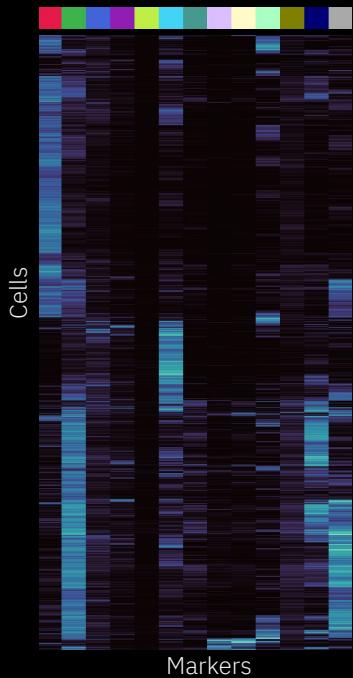
Subtype prediction



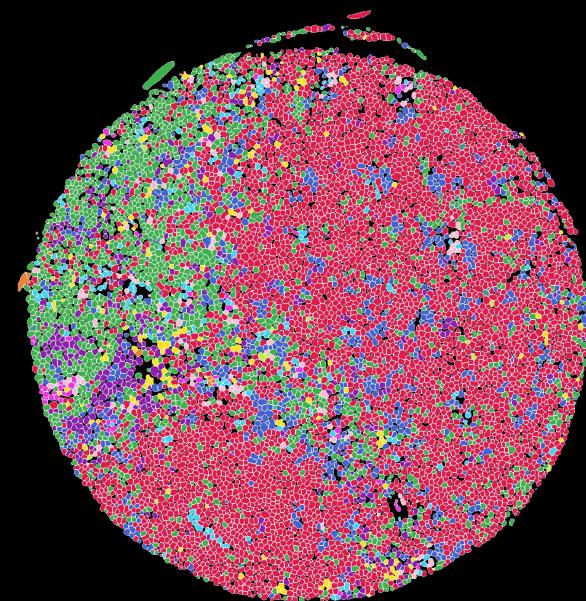
CD8

Cell type prediction

Expression matrix

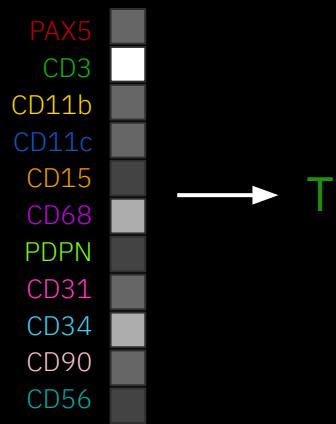


Cell type predictions



Cell type prediction

Argmax



Astir

Interpretable statistical model

$$\mathbb{E}[\mathbf{y}_n | z_n = c] = \exp(\boldsymbol{\mu} + \boldsymbol{\rho}_c \odot \boldsymbol{\delta}_c)$$

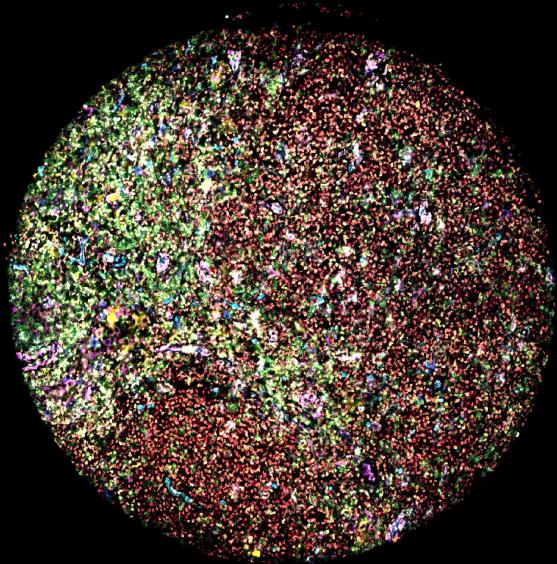
Base expression of a marker + overexpression of that marker if cell n is of cell type c

Deep recognitional neural network for fast inference

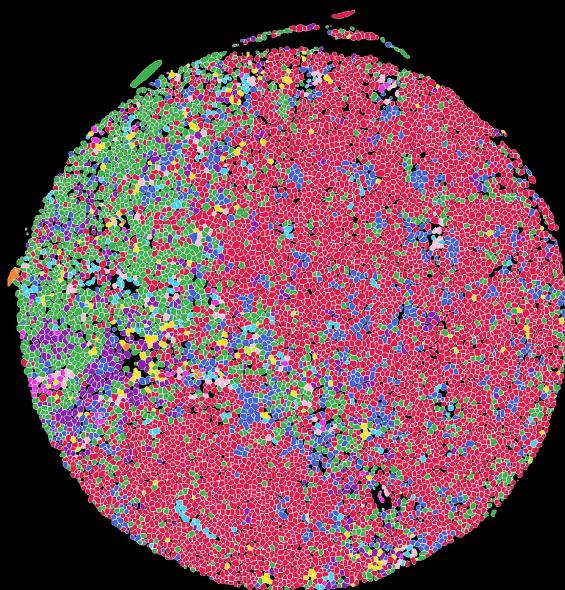
$$q(\mathbf{z}_n | \mathbf{x}_n) = \text{nn}(\mathbf{x}_n)$$

Cell type prediction

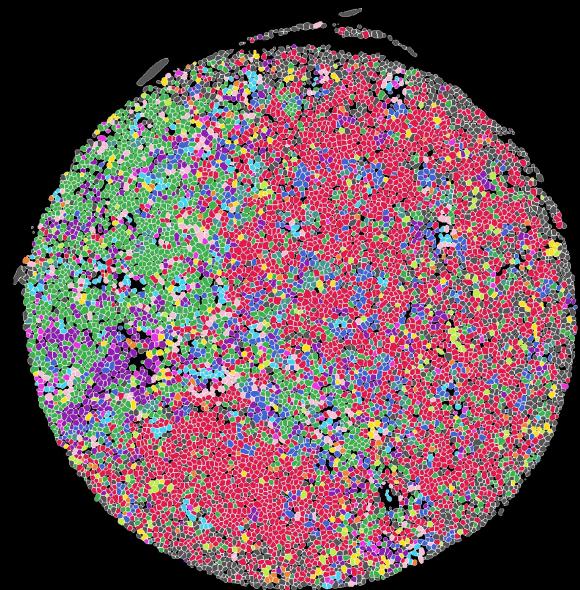
Processed intensities



Argmax prediction

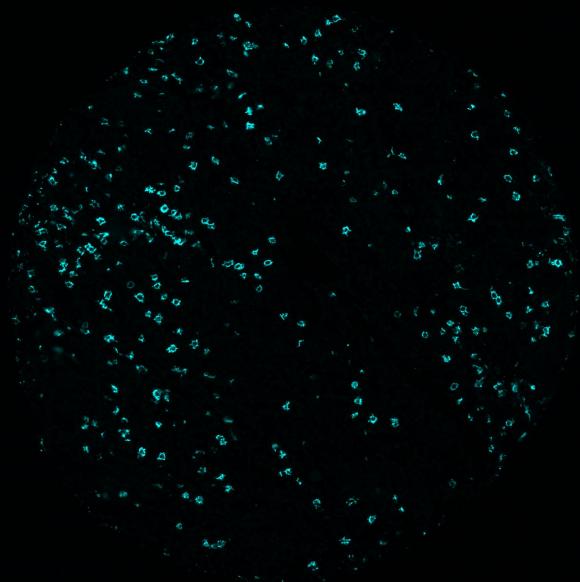


Astir prediction

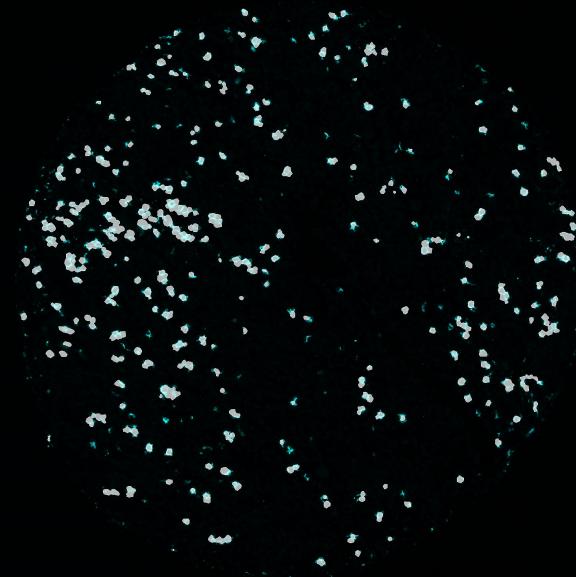


Binarization of functional markers in suitable samples

Intensities



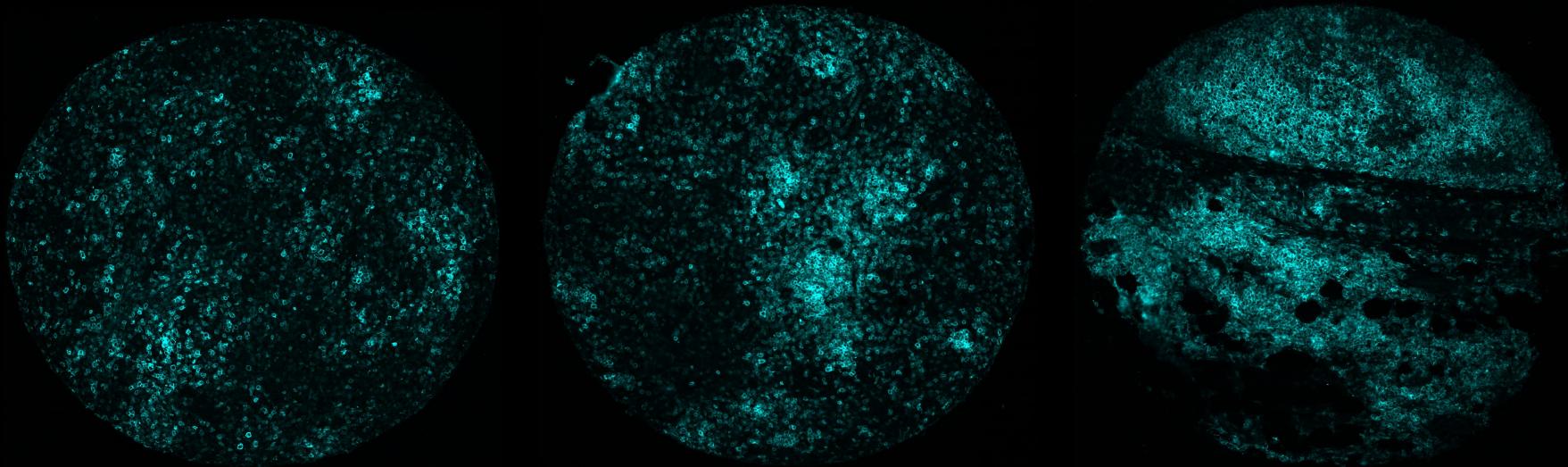
Positive cells



CD8 (CD8+ T cells)

Markers behave heterogeneously across samples

Finding cutoffs automatically is an open challenge



Summary - Spatialproteomics

Complex spatial analyses in “one line of code”

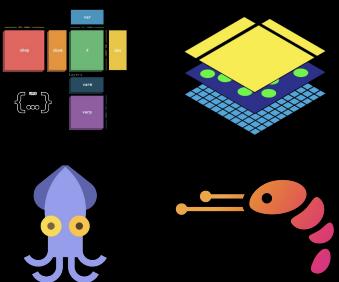


```
img.tl.cellpose(channel='DAPI', key='cellpose_segmentation') # segmentation
    .pp.add_segmentation('cellpose_segmentation')
    .pp.threshold([0.8, ..., 0.95])                                # channel specific thresholding
    .pp.apply(func=medfilt2d)                                         # median filter
    .pp.add_quantification(func=sp.pp.mean_intensity)                # marker quantification
    .pp.transform_expression_matrix(method='arcsinh')                 # arcsinh transform
    .pp.predict_cell_types_argmax(ct_marker_dict)                   # celltype prediction
    .pl.show()                                                       # plotting
```

Summary - Spatialproteomics

Complex spatial analyses in “one line of code”

```
pip install spatialproteomics
```



```
sp.tl.convert_to_anndata()  
sp.tl.convert_to_spatialdata()
```

Screenshot of the GitHub repository for `spatialproteomics`:

- README:** MIT license, 0.4.0 version, published 5 days ago, + 6 releases.
- Packages:** No packages published, Publish your first package.
- Contributors:** MeyerBender (Matthias Meyer-Bender), sagar87 (Harald Vöhringer).
- Deployments:** 91, latest: github-pages 52 minutes ago, + 90 deployments.
- Languages:** Python 100.0%.

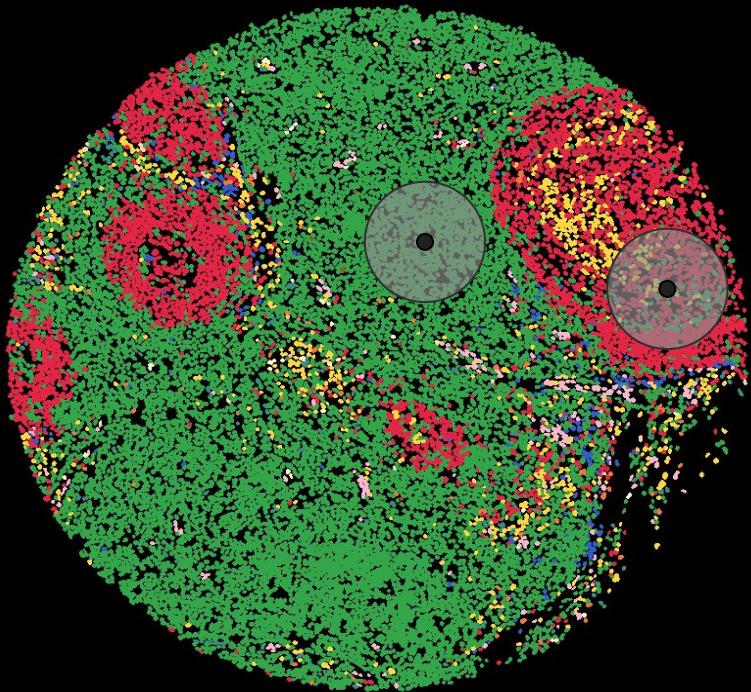
Principles: Multiplexed imaging data comprises at least 3 dimensions (i.e. channels, x, and y) and has often additional data such as segmentation masks or cell type annotations associated with it. In `spatialproteomics`, we use `xarray` to create a data structure that keeps all of these data dimension in sync. This data structure can then be used to apply all sorts of operations to the data. Users can segment cells, perform different image processing steps, quantify protein expression, predict cell types, and plot their data in various ways. By providing researchers with those tools, `spatialproteomics` can be used to quickly explore highly multiplexed spatial proteomics data directly within jupyter notebooks.

Installation:

```
pip install spatialproteomics
```

Check the documentation for further information
<https://sagar87.github.io/spatialproteomics/>

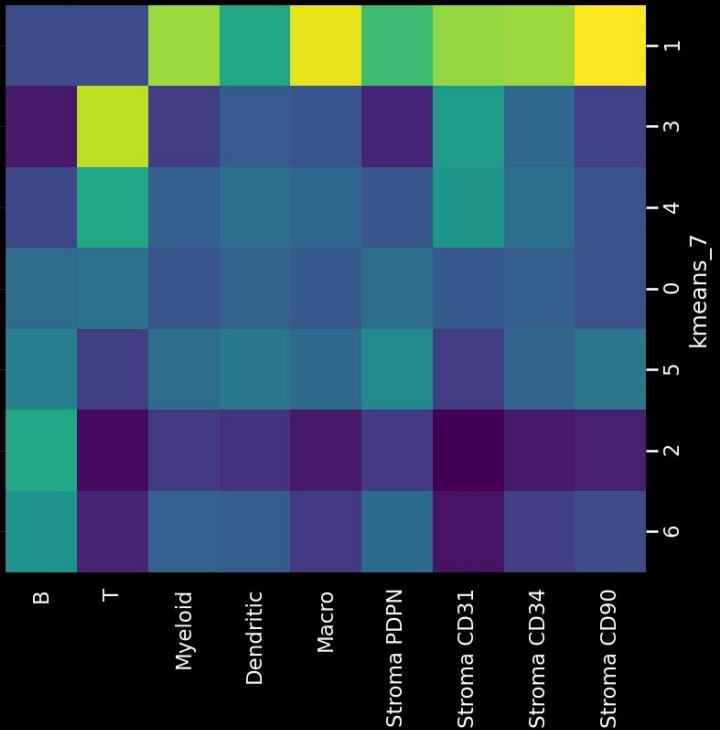
Cellular Neighborhood



B	T	Myeloid
0	30	0
20	10	10
...		

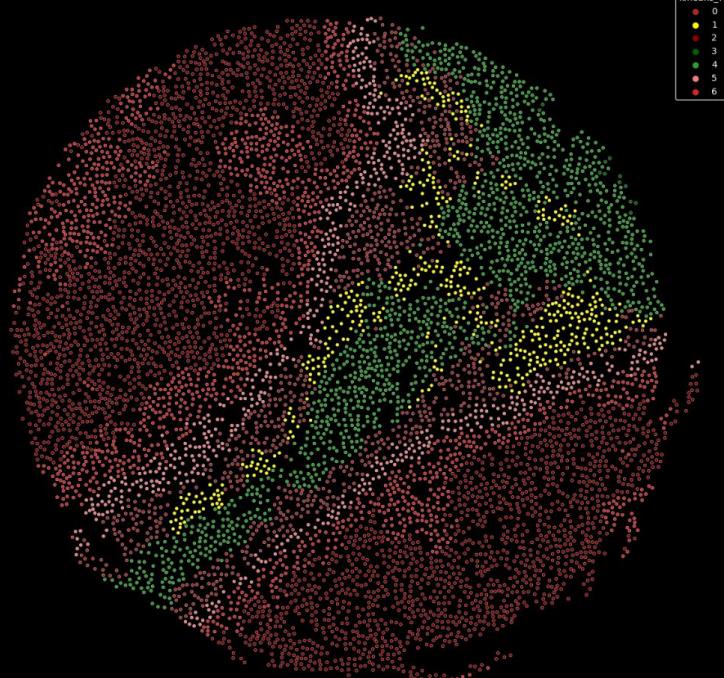
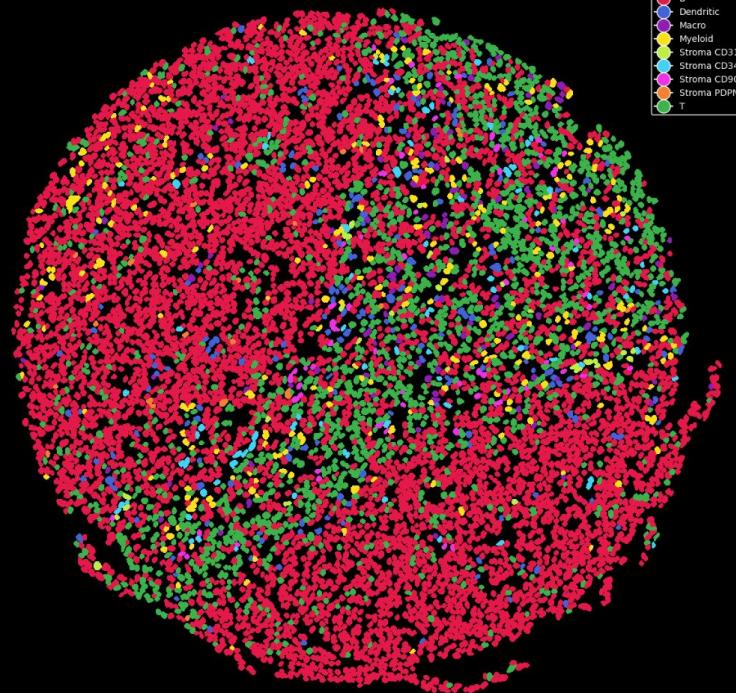
Clustering Neighborhoods

Mean cluster composition for k-means

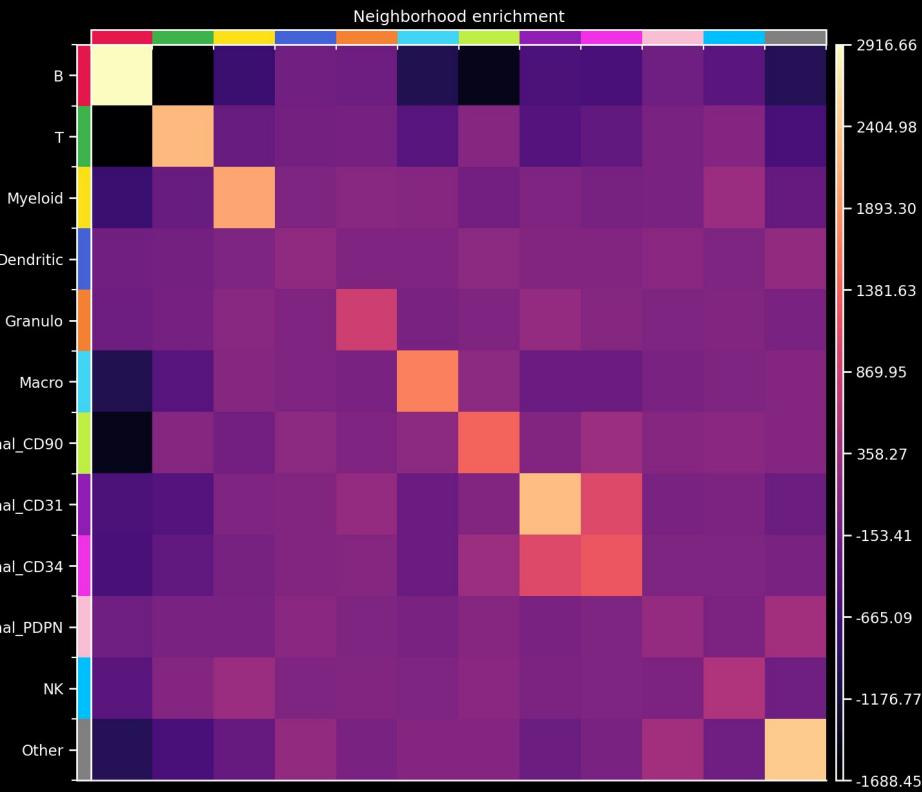


- Many clustering methods
 - k-means clustering
 - DBSCAN
 - hierarchical clustering
 - ...
- Each cluster is defined by a composition of cell types

Clustering Neighborhoods

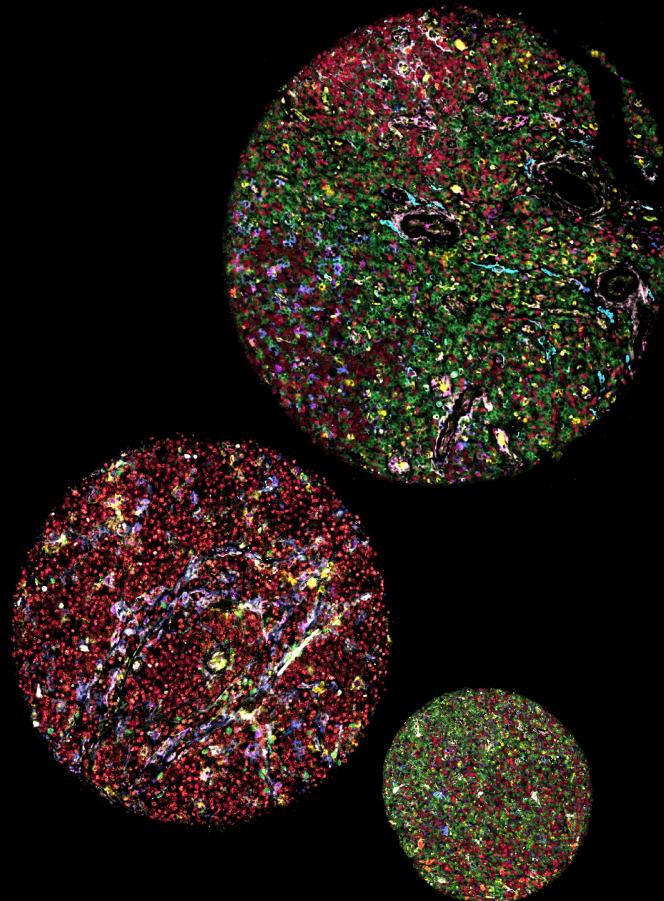


Neighborhood Enrichment with Squidpy



Summary

- Processing and analysis of highly multiplexed immunofluorescence data with spatialproteomics and other packages
- Segmentation → Image Processing → Cell Type Assignment → Downstream Analysis
- Download workshop material and get coding:
www.huber.embl.de/users/matthias/eccb_workshop.tar.gz



Workshop Material



MeyerBender / spatialproteomics_workshop



Workshop



sagar87 / spatialproteomics



Documentation

Acknowledgements

Huber Group

Wolfgang Huber
Anastasiia Horlova
Constantin Ahlmann-Eltze
Donnacha Fitzgerald
Erin Chung
Harald Vöhringer
Hosna Baniadam
Julia Lazoriak
Julia Philipp
Junyan Lu
Michelle Chan
Mike Smith
Petr Smirnov
Sarah Kaspar
Sascha Dietrich
Simone Bell
Stefan Peidl
Thomas Naake
Tümay Capraz

Dietrich Group

Sascha Dietrich
Peter-Martin Bruch
Christina Schniederjohann
Sarah Koziel
Lisa Held
Felix Czernilofsky

Collaborators

Marc Seifert
Marc Baertsch
Kristy Ou
Valeriy Pak

Thesis Advisory Committee

Wolfgang Huber
Sascha Dietrich
Virginie Uhlmann
Simon Anders
Shantanu Singh
Anna Kreshuk

