xarray data structures in Biology

Ian Hunt-Isaak · SciPy 2026



Talk was presented at SciPy 2026 - watch it on YouTube here:
youtube.com/watch?v=AvdASvcOn1M

SUPPORTED BY

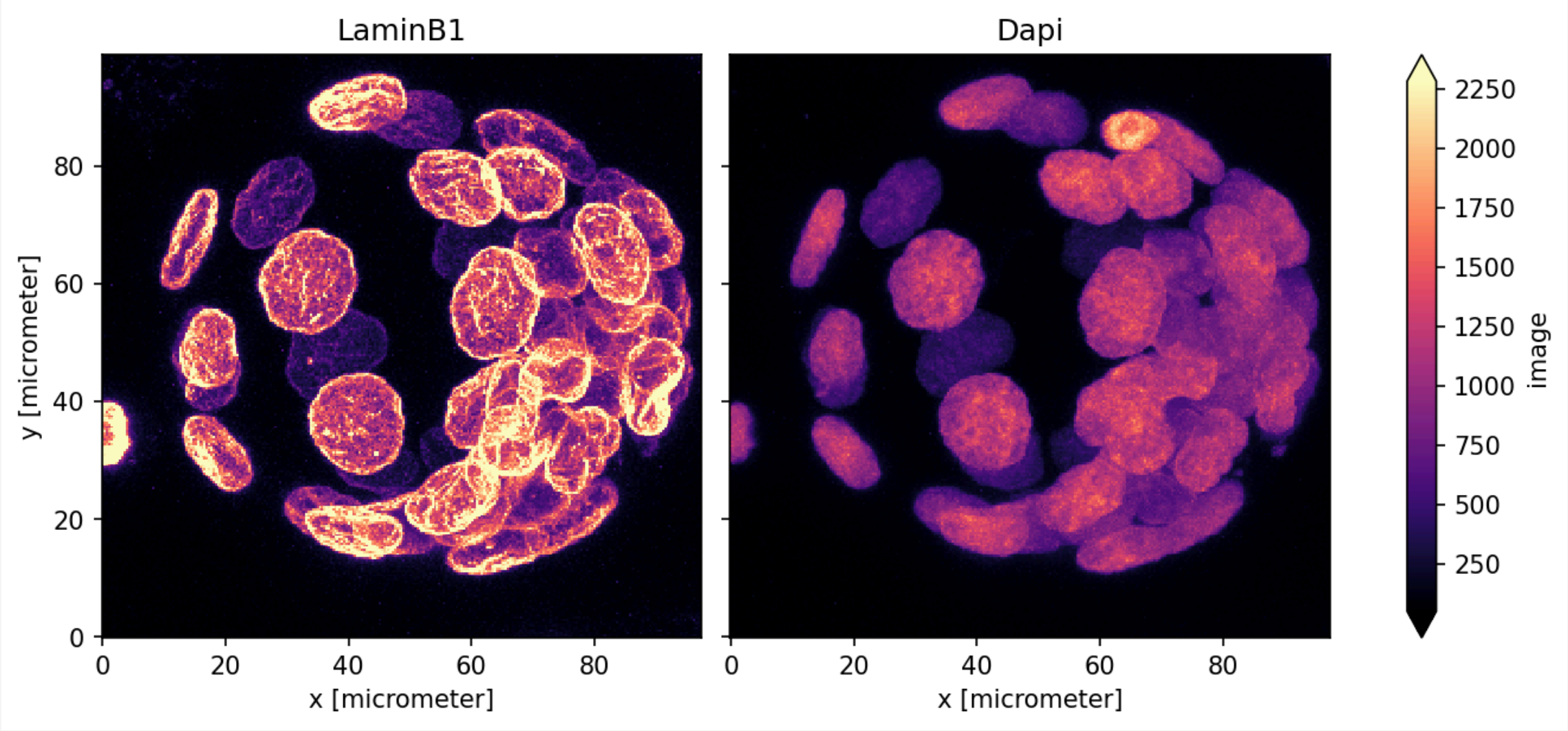


Chan Zuckerberg Initiative

outline

- Introduction
- Power of Xarray for Biology
  - Ergonomics
  - Indexes
- OME/NGFF and Xarray
  - Premade Indexes
  - Data Container
- OMICS
- Getting Data into Xarray (when you can't use `ngff-zarr`)
  - Virtualization
- Conclusion

introducing the data



IDR0062A · BLIN ET AL. 2019 · LAMINB1 + DAPI Z-STACK 2 CHANNELS · Z 236 × Y 275 × X 271 · UINT16 IDR.GITHUB.IO/OME-NGFF-SAMPLES · IDR0062

the loaded dataset

xarray.Dataset

▸ Dimensions: (**c**: 2, **z**: 236, **y**: 275, **x**: 271)

▾ Coordinates:

c	(c)	object	'LaminB1' 'Dapi'
z	(z)	float64	0.0 0.5002 1.0 ... 117.0 117.5
y	(y)	float64	0.0 0.3604 0.7208 ... 98.39 98.75
x	(x)	float64	0.0 0.3604 0.7208 ... 96.95 97.31

▾ Data variables:

image	(c, z, y, x)	uint16	dask.array<chunksize=(1, 1, 275, 271), met…
label_0	(z, y, x)	int8	dask.array<chunksize=(59, 69, 136), meta=…

▸ Indexes: (3)

▾ Attributes:

ome : {'creator': {'name': 'omero-zarr', 'version': '0.4.0'}, 'multiscales': [{'axe

DataTree.ngff v0.4

channel (c)

channel	color	window
LaminB1	#0000FF	0–1500
Dapi	#FFFF00	0–1500

space

axis	unit	transform	extent (world)
z	micrometer	scale 0.500203, +0	0 … 117.548
y	micrometer	scale 0.360398, +0	0 … 98.7491
x	micrometer	scale 0.360398, +0	0 … 97.3075

data: `image: uint16, label_0: int8 (labels)`

finest shape: `c=2 z=236 y=275 x=271`

levels: `3 (0, 1, 2)`

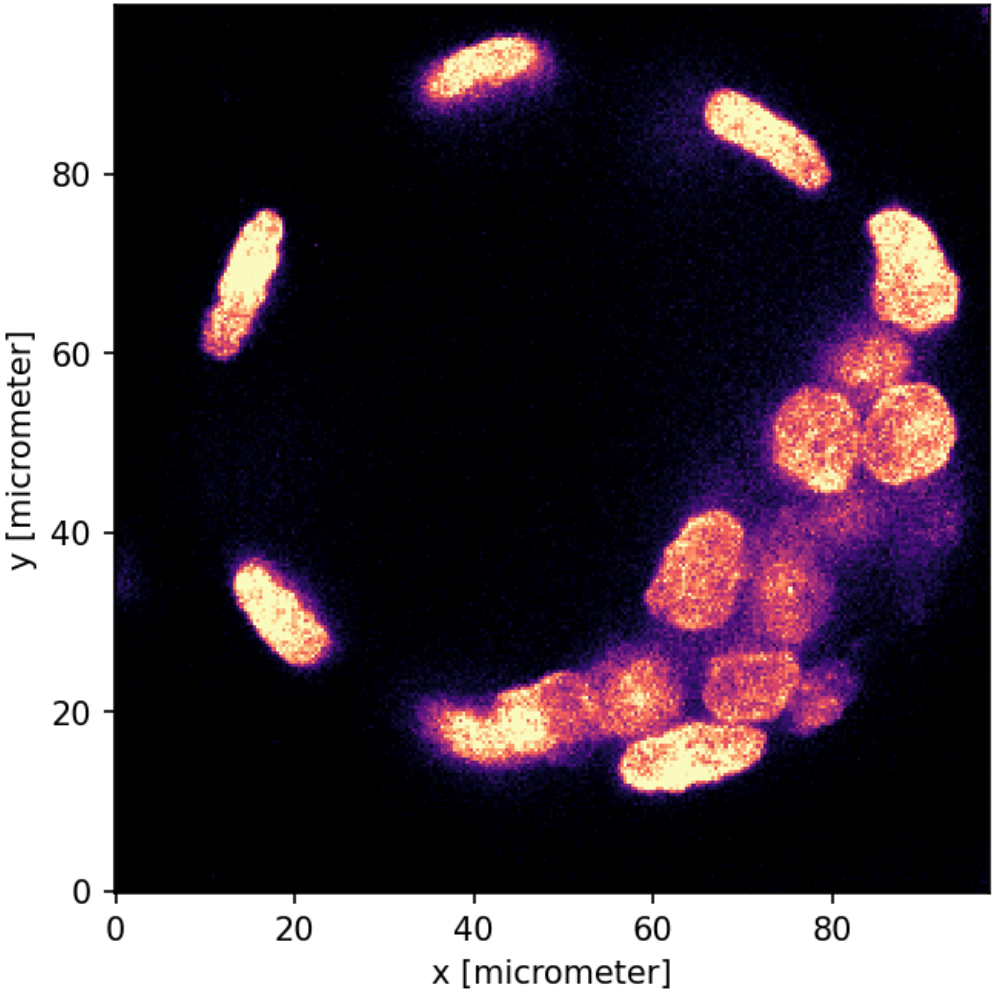
Selection in Physical Coords

No Xarray

```
1  img[1, 30]
```

With Xarray

```
1  img.sel(c="Dapi", z=59.0)
```



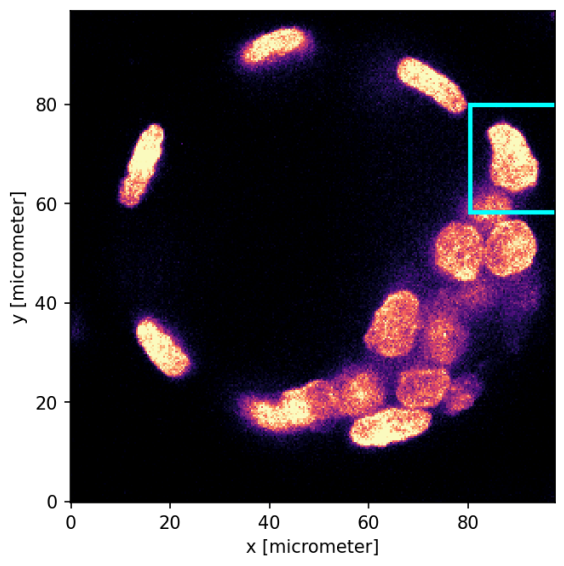
Selection in Physical Coords

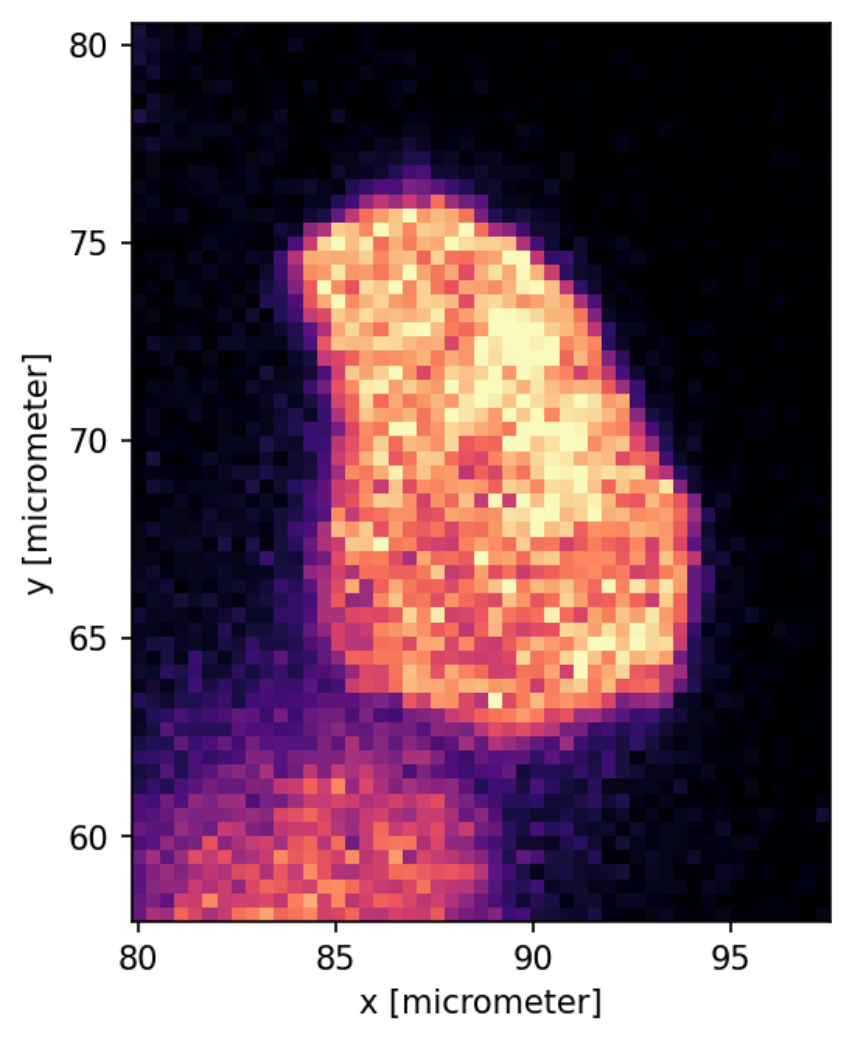
No Xarray

```
y0 = np.argmin(np.abs(y_um - 58))
y1 = np.argmin(np.abs(y_um - 80))
x0 = np.argmin(np.abs(x_um - 80))
x1 = np.argmin(np.abs(x_um - 98))
img[1, 30, y0:y1, x0:x1]
```

With Xarray

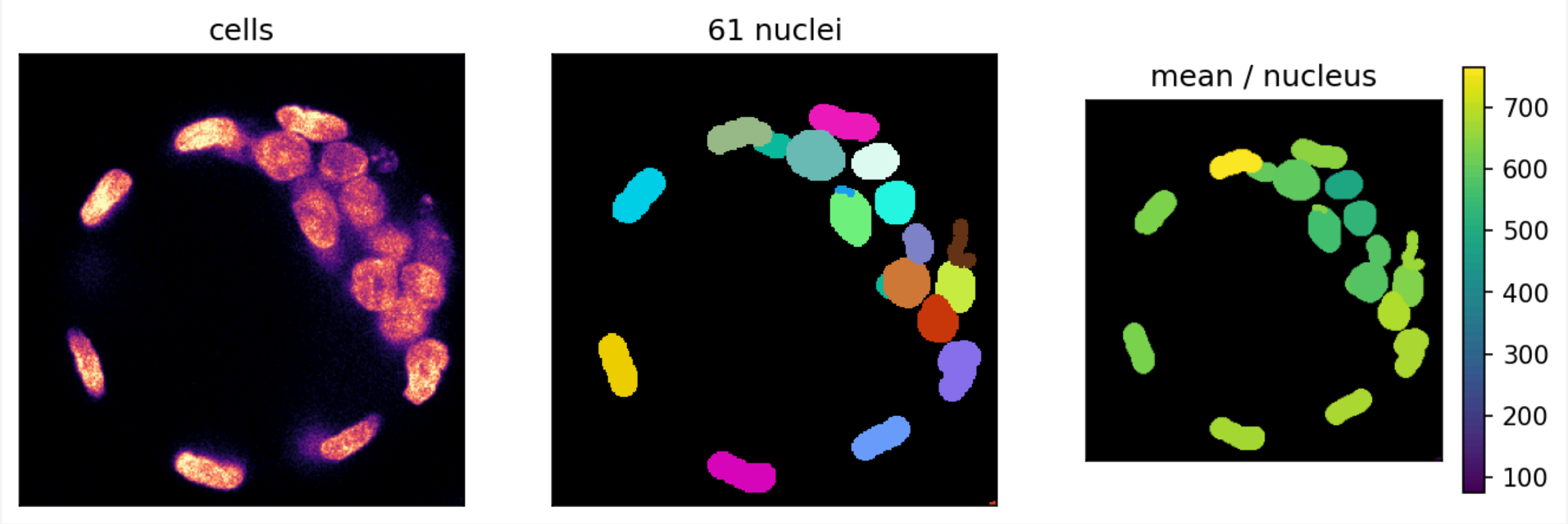
```
img.sel(c="Dapi", z=59.0,
        y=slice(58, 80), x=slice(80, 98))
```





per-object analysis

```
1  ds.ngff.groupby_labels().mean()
```

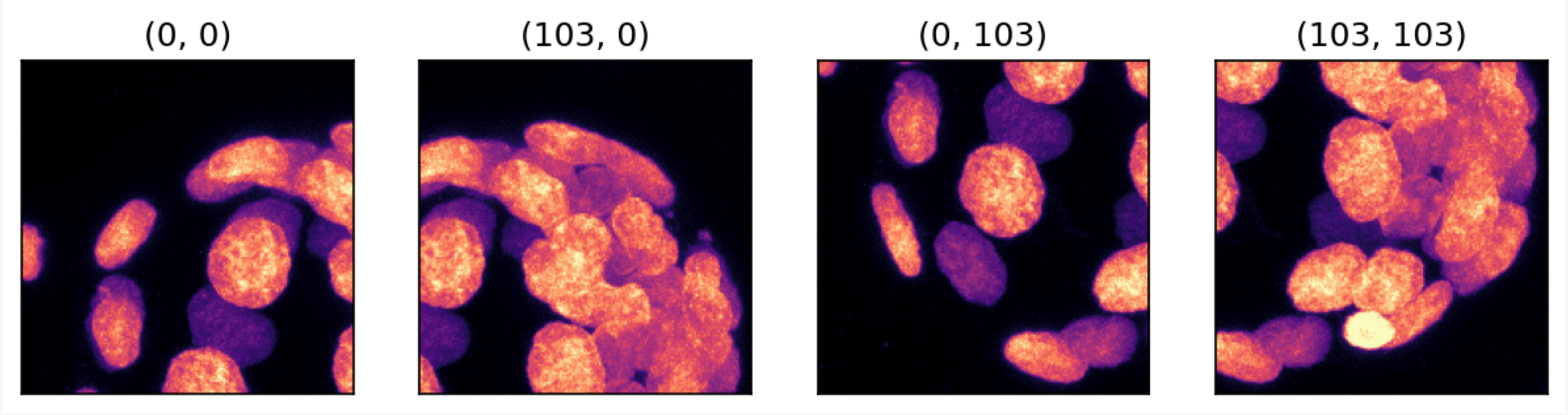


what’s an index?

encode knowledge of the structure of our data

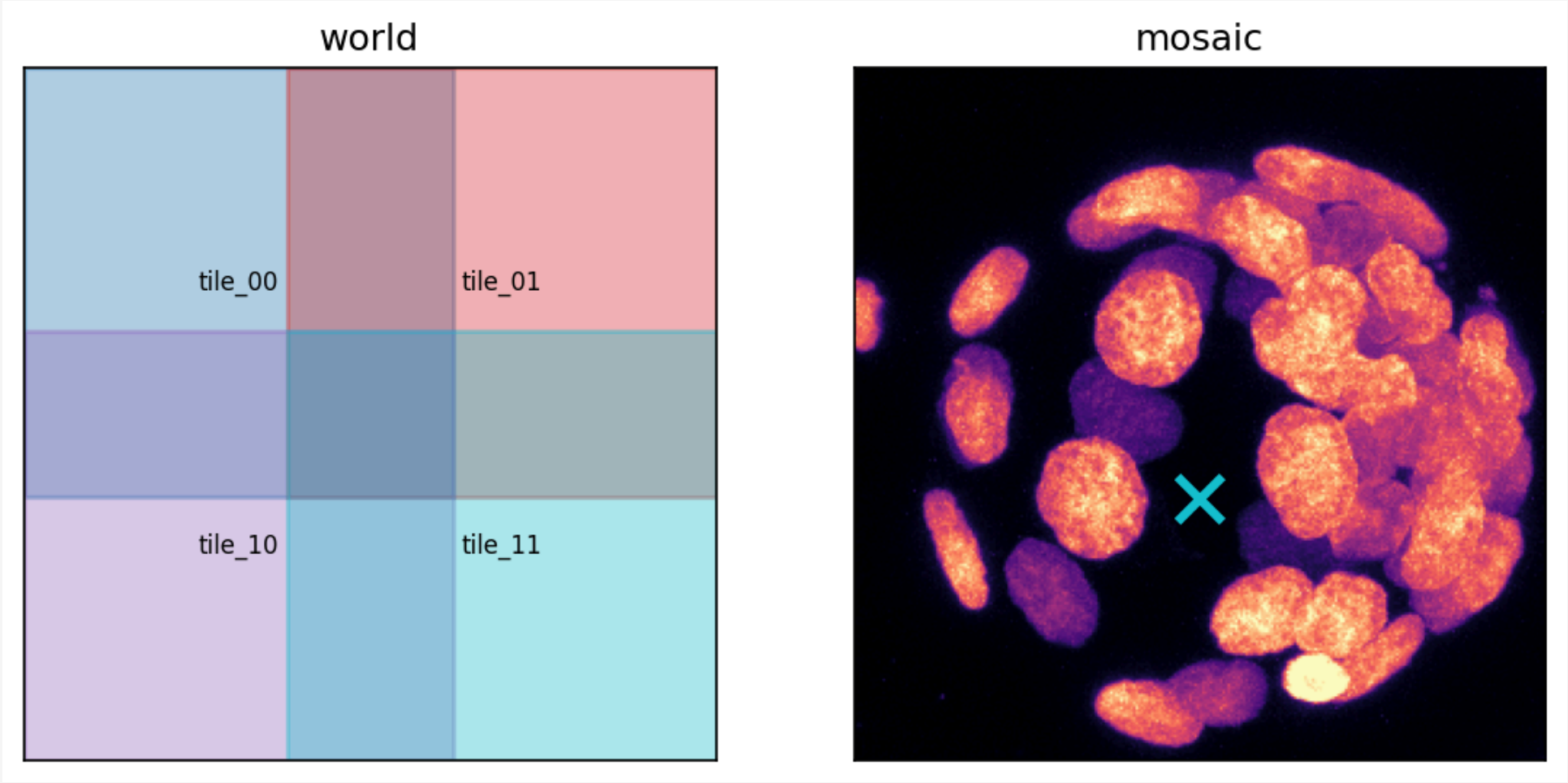
- select in physical coordinates (even complex ones)
- efficiently — a precomputed lookup, not a full scan

many parts, each with an offset



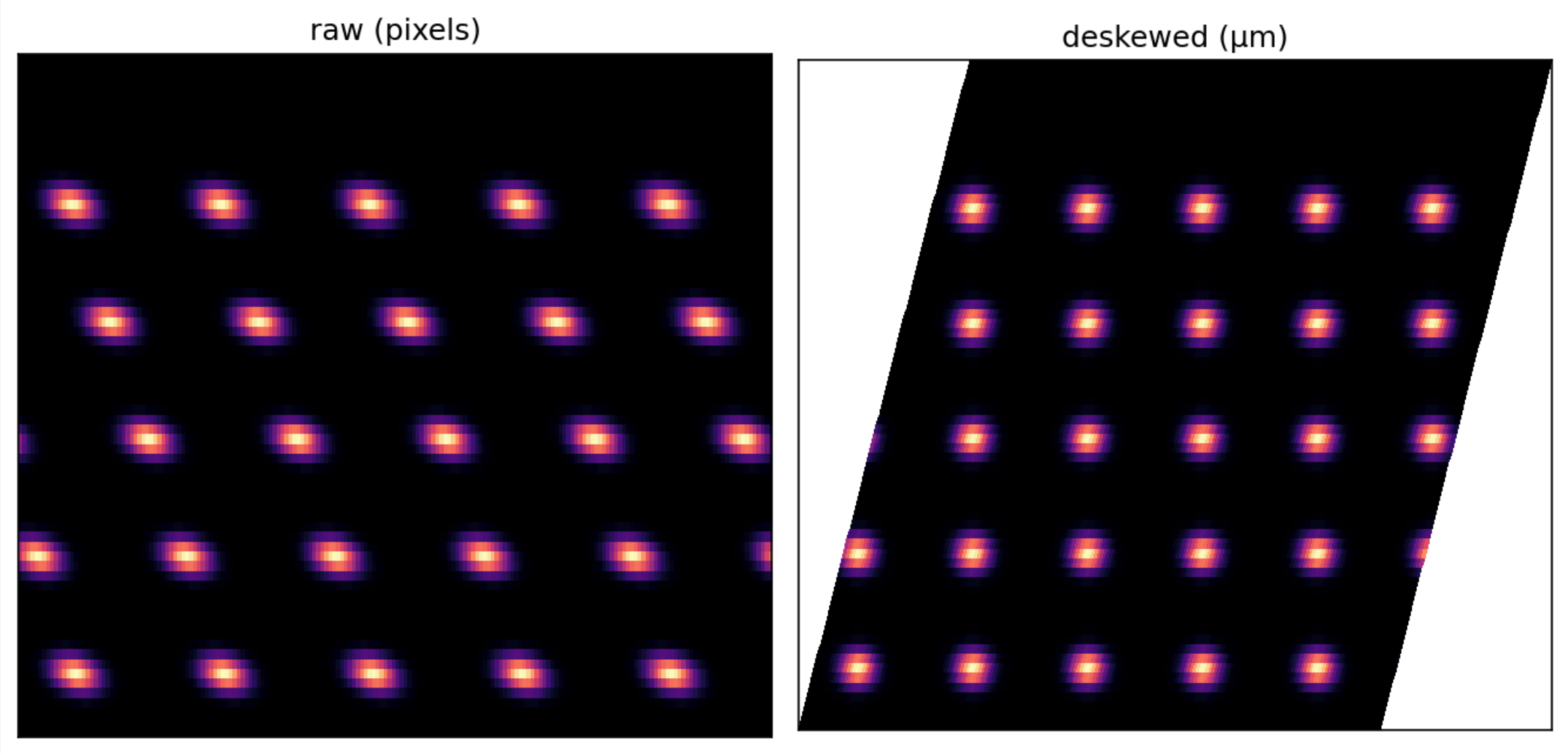
reconstruct from the translations

```
1  tiles  = [Tile(arr, translation={"y": ty, "x": tx}) for ...]
2  mosaic = place_tiles(tiles)
3  mosaic.sel(x=256, y=320)
```



deskew

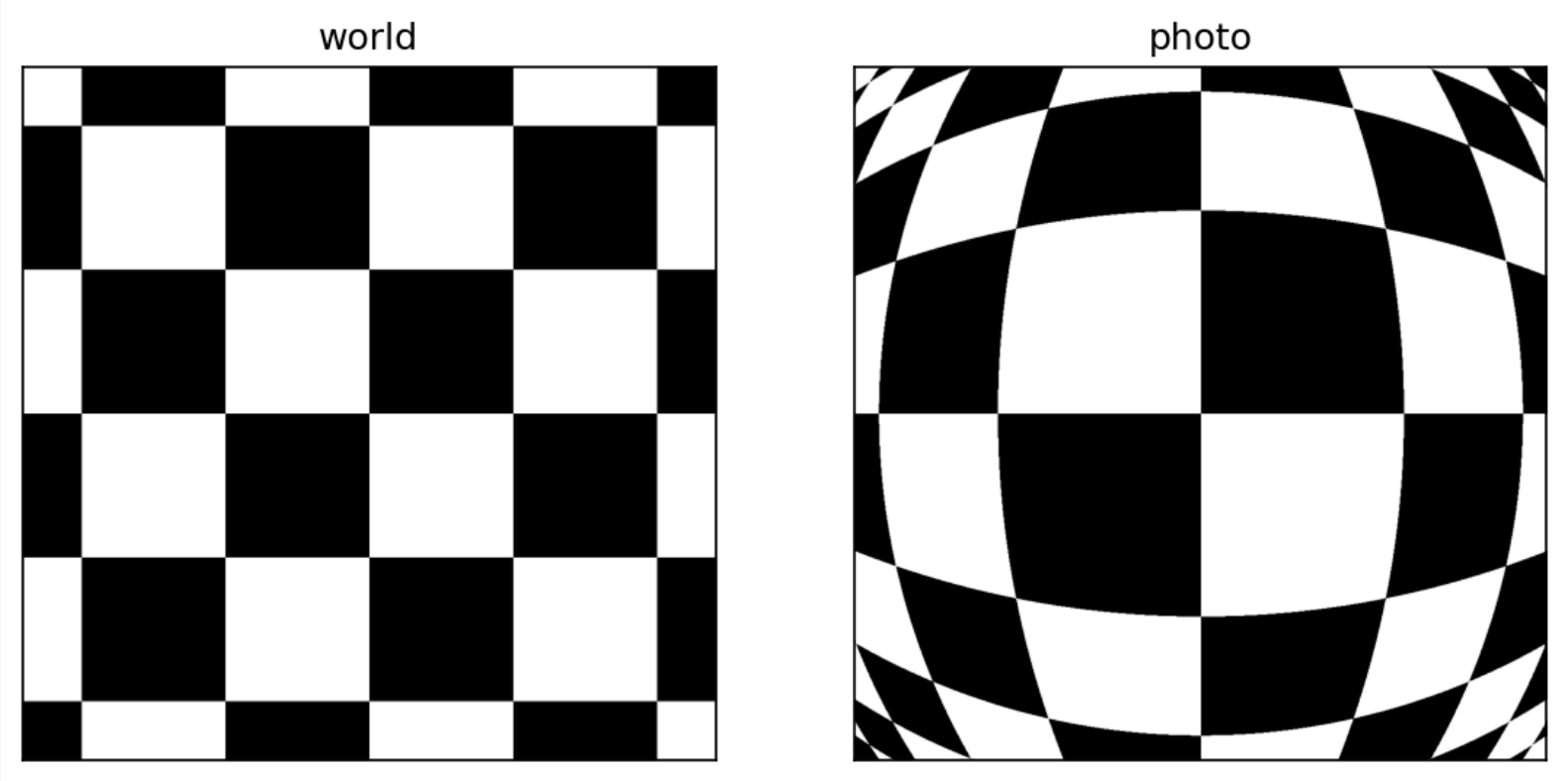
(synthetic)



lens distortion

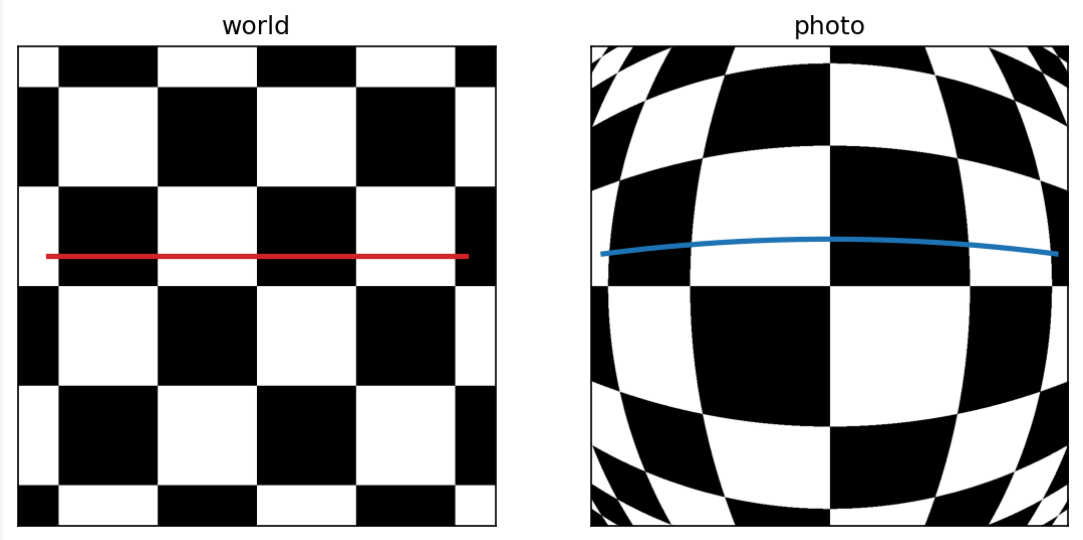
(synthetic)

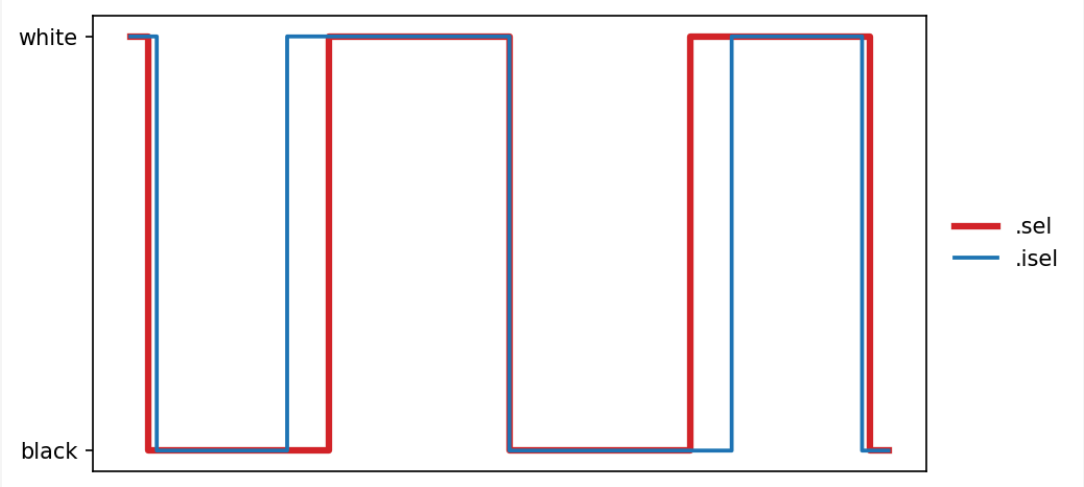
```
fish = xr.DataArray(
    photo,
    dims=("y_pixel", "x_pixel"),
    coords=Coordinates.from_xindex(
        CoordinateTransformIndex(lens)),
)
```



select in world space

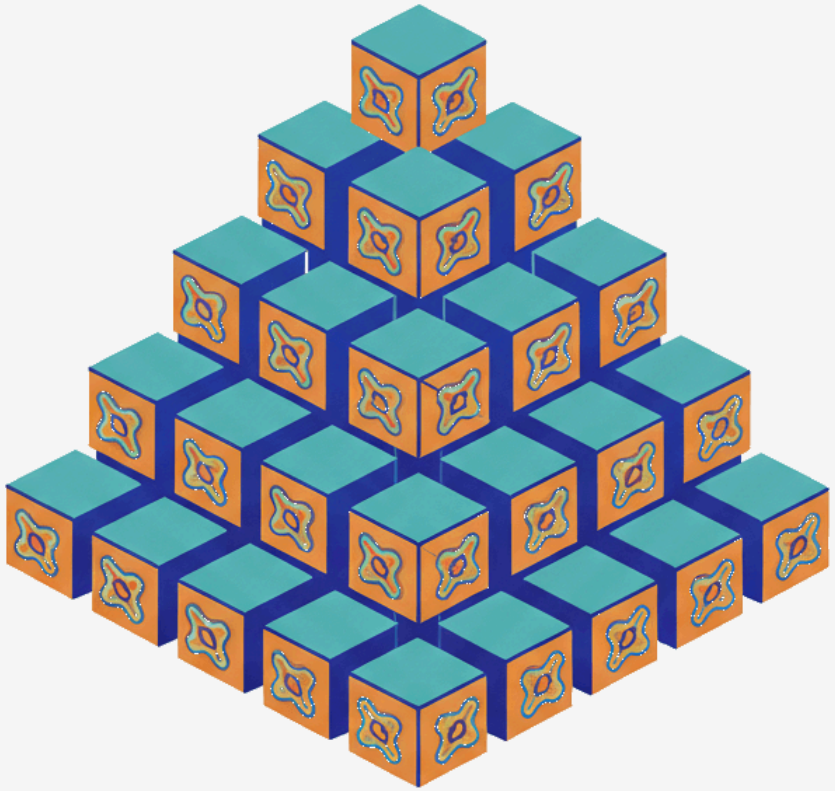
```
fish.sel(x_world=line, y_world=0.2,
         method="nearest")

fish.isel(y_pixel=512)
```



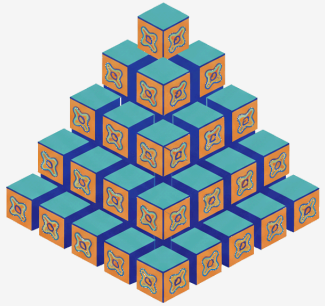


But Wait!

- I don’t know how to load my OME-Zarr into Xarray
- Writing a custom Index is a mystery to me
- Where on the Xarray object should I put the metadata



xarray-ome



- loads OME-Zarr via **ngff-zarr**
- metadata in consistent places — **CF conventions**
- a structured `.ngff` **accessor** for downstream tools

Goals

- Thoughtful translation from OME<-> Xarray
- Prototype implementation
- Cross Package metadata container

Non-Goals

- Do this alone

xarray.Dataset

▸ Dimensions: (**c**: 2, **z**: 236, **y**: 275, **x**: 271)

▾ Coordinates:

c	(c)	object	'LaminB1' 'Dapi'
z	(z)	float64	0.0 0.5002 1.0 ... 117.0 117.5
y	(y)	float64	0.0 0.3604 0.7208 ... 98.39 98.75
x	(x)	float64	0.0 0.3604 0.7208 ... 96.95 97.31

▾ Data variables:

image	(c, z, y, x)	uint16	dask.array<chunksize=(1, 1, 275, 271), meta=np.n…
label_0	(z, y, x)	int8	dask.array<chunksize=(59, 69, 136), meta=np.ndar…

▸ Indexes: (3)

▾ Attributes:

ome : {'creator': {'name': 'omero-zarr', 'version': '0.4.0'}, 'multiscales': [{'axes': [{'na

DataTree.ngff v0.4

channel (c)

channel	color	window
LaminB1	#0000FF	0–1500
Dapi	#FFFF00	0–1500

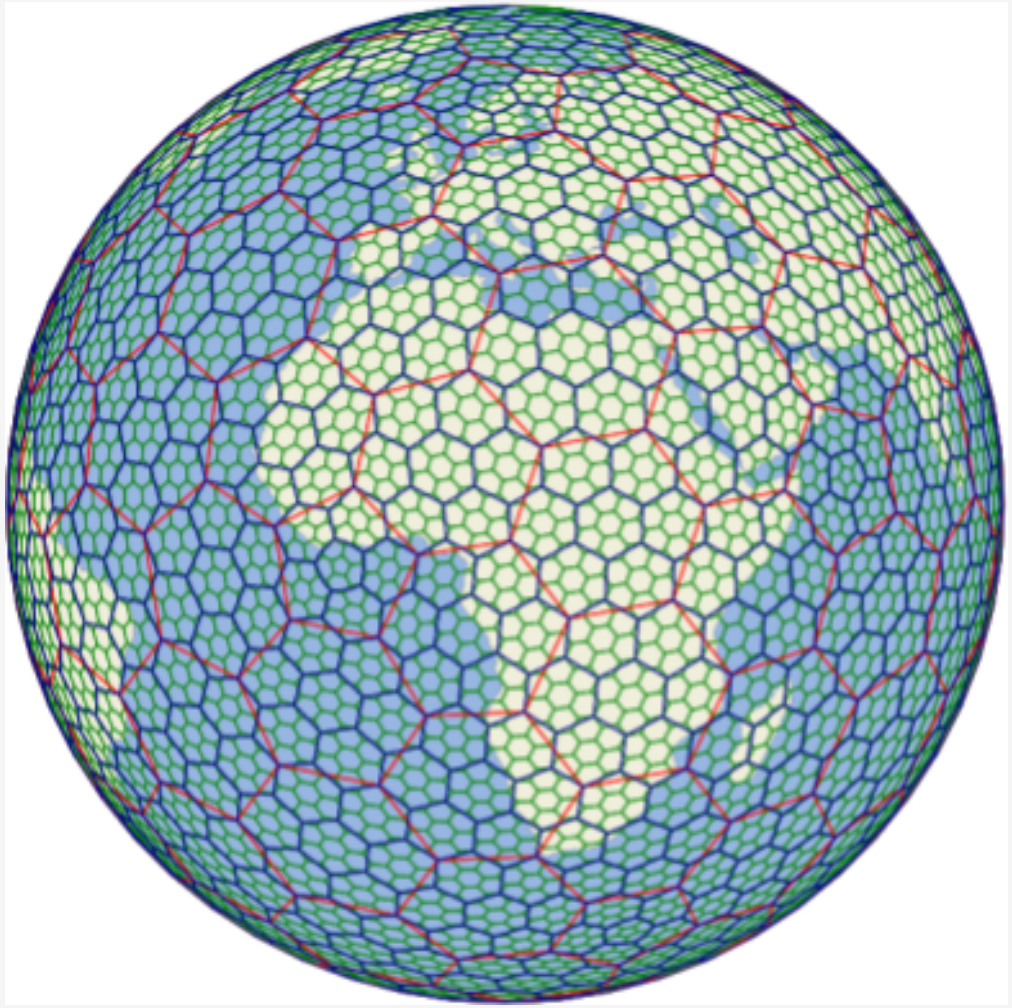
space

axis	unit	transform	extent (world)
z	micrometer	scale 0.500203, +0	0 … 117.548
y	micrometer	scale 0.360398, +0	0 … 98.7491

supported transform (coordinate system/indexes)

- **scale · translation** — physical units
- **affine** — rotation, shear, deskew
- **displacement field** — per-pixel warp (lens)

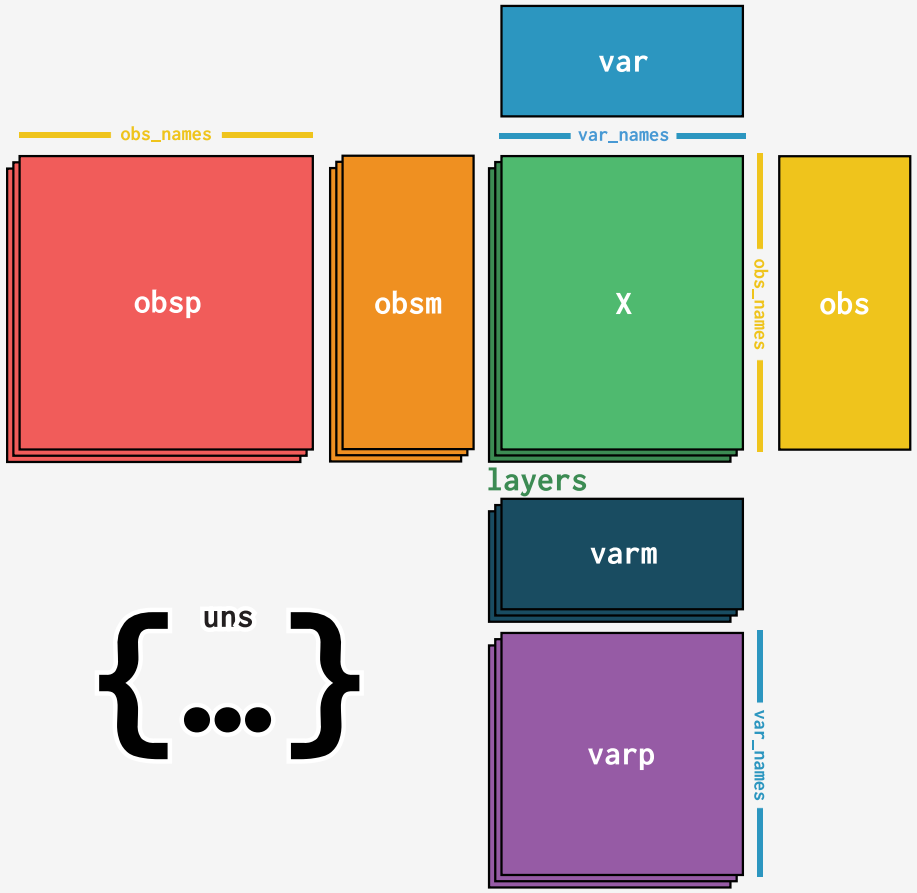
XDGGS — xarray-indexes.readthedocs.io/earth/xdggs.html



everything so far assumes
the data is already **loaded**

OMICS

anndata + spatial data



SpatialData

How to load the data in?

- BIG DATASETS
- not cloud native

File Name	N Subjects	N Samples	N Cells	File Size
SoundLife_OlderAdult_Female_CMVneg.h5ad	10	91	1,350,748	25 GB
SoundLife_OlderAdult_Female_CMVpos.h5ad	17	163	2,596,111	49 GB
SoundLife_OlderAdult_Male_CMVneg.h5ad	12	116	1,750,565	34 GB
SoundLife_OlderAdult_Male_CMVpos.h5ad	8	80	1,322,061	25 GB
SoundLife_YoungAdult_Female_CMVneg.h5ad	18	162	2,616,824	49 GB
SoundLife_YoungAdult_Female_CMVpos.h5ad	10	76	1,234,234	16 GB
SoundLife_YoungAdult_Male_CMVneg.h5ad	12	107	1,712,244	34 GB
SoundLife_YoungAdult_Male_CMVpos.h5ad	9	73	1,206,761	16 GB

https://apps.allenimmunology.org/aifi/insights/dynamics-imm-health-age/downloads/scrna/

Legacy Data Formats

- h5ad
- ndpi
- many others...

read them without converting

Converting?

- Compliance
- Storage
- Drift

Virtualization

the prerequisite for loading
data into Xarray

What is it?

read the file where it lives — no copy

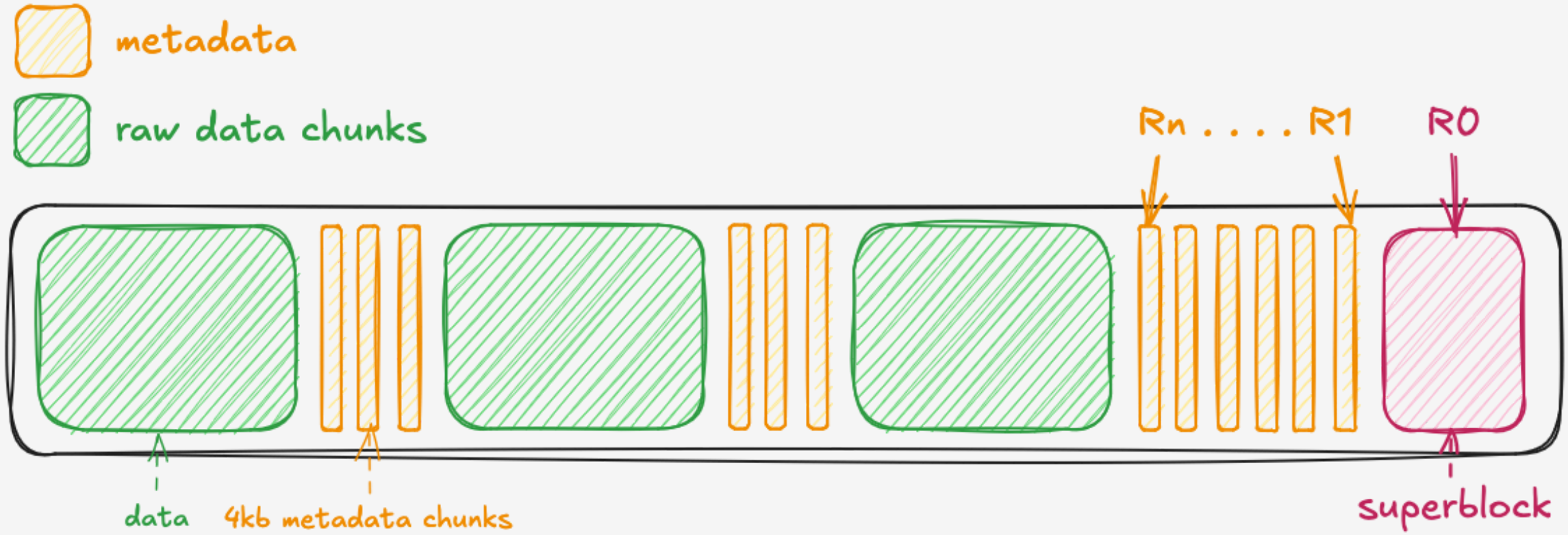
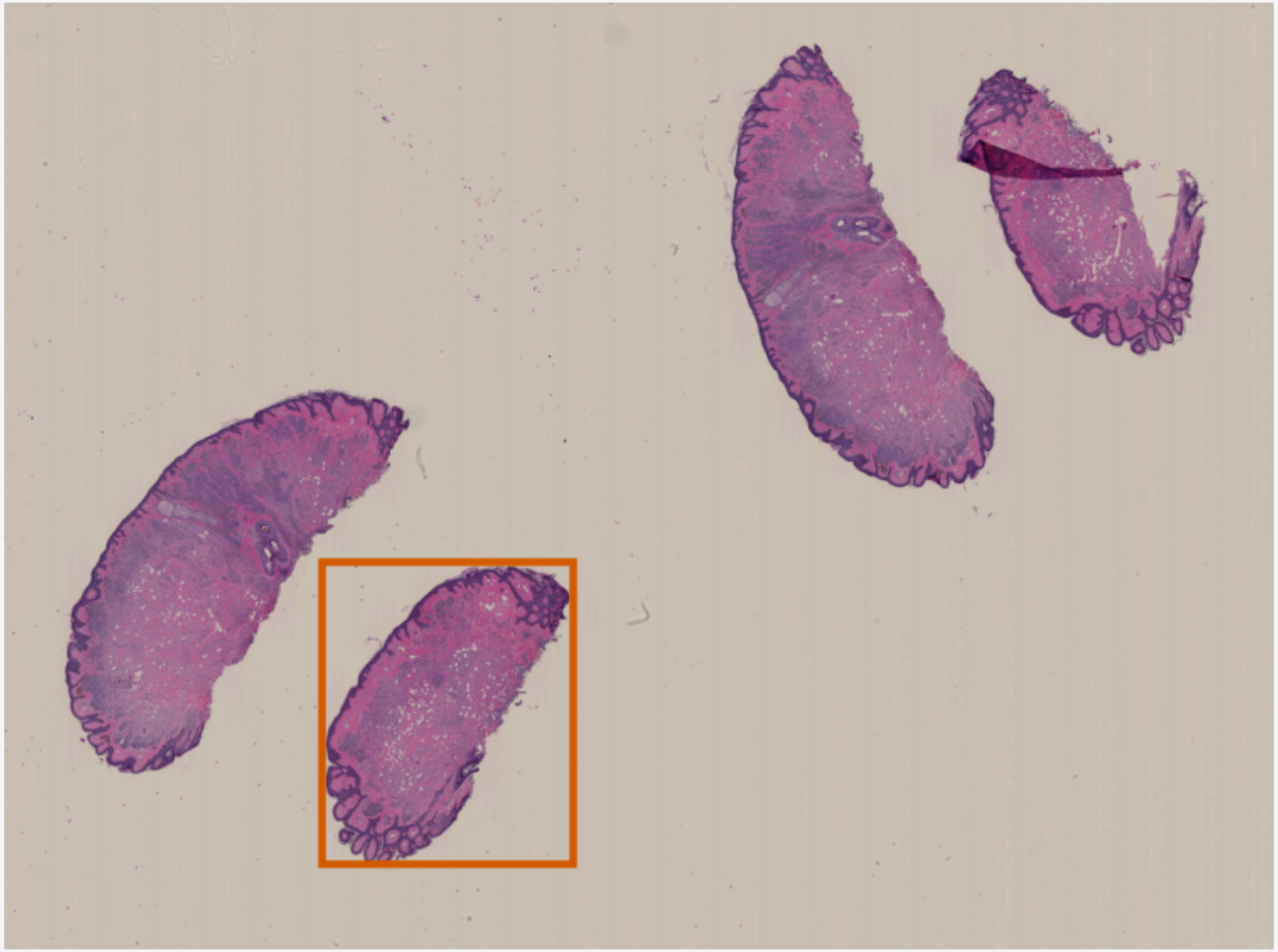


IMAGE: TOM NICHOLAS · EARTHMOVER.IO/BLOG

Go to visual: https://earth-mover.github.io/icechunk-illustrated/virtual/slide/

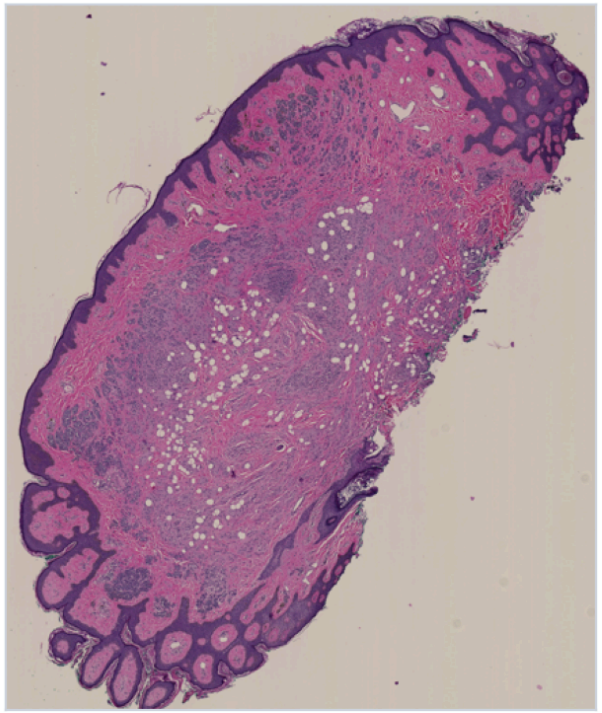
ndpi

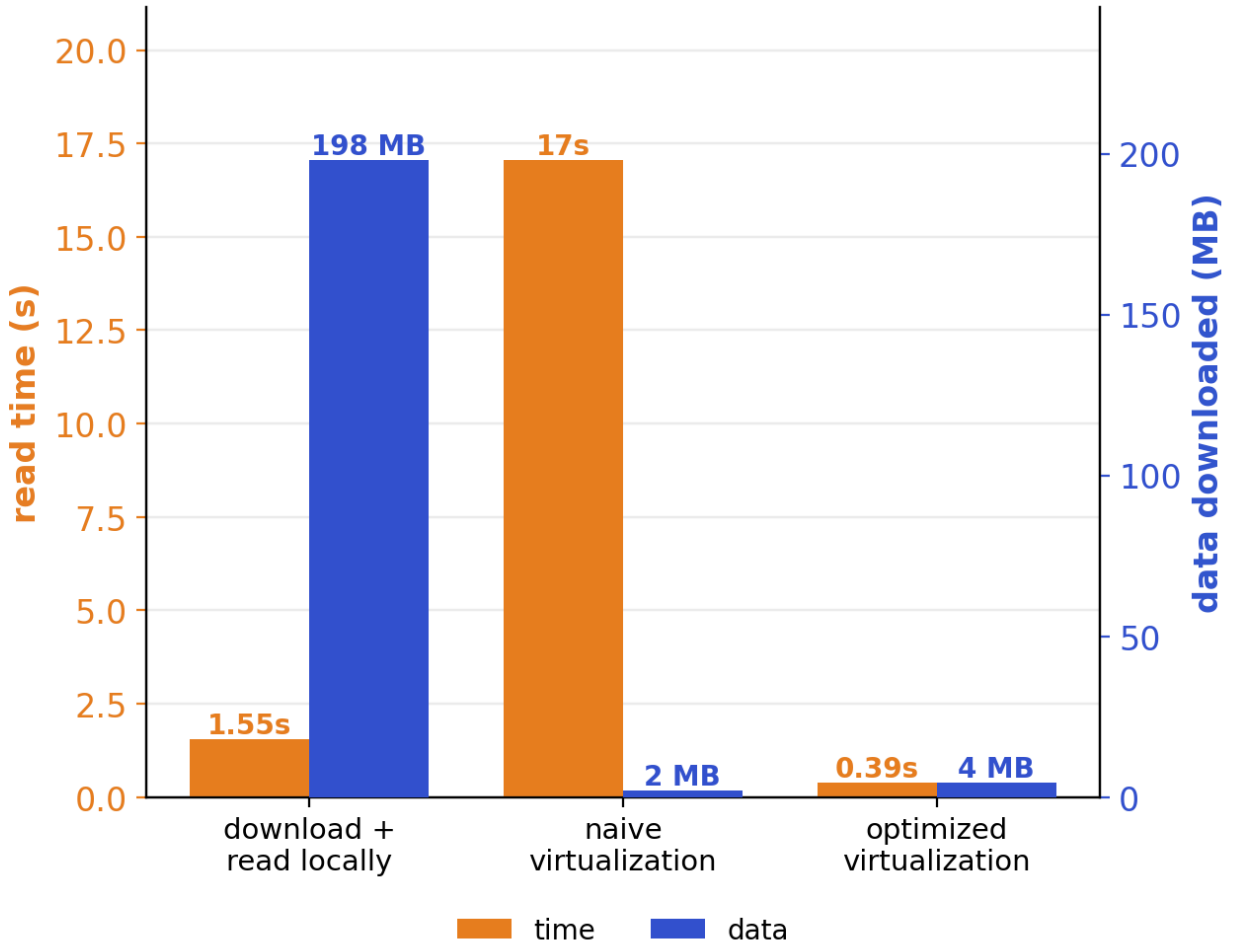


ndpi

Reading one tissue region from a remote whole-slide image (~80 ms latency)

the region you asked for





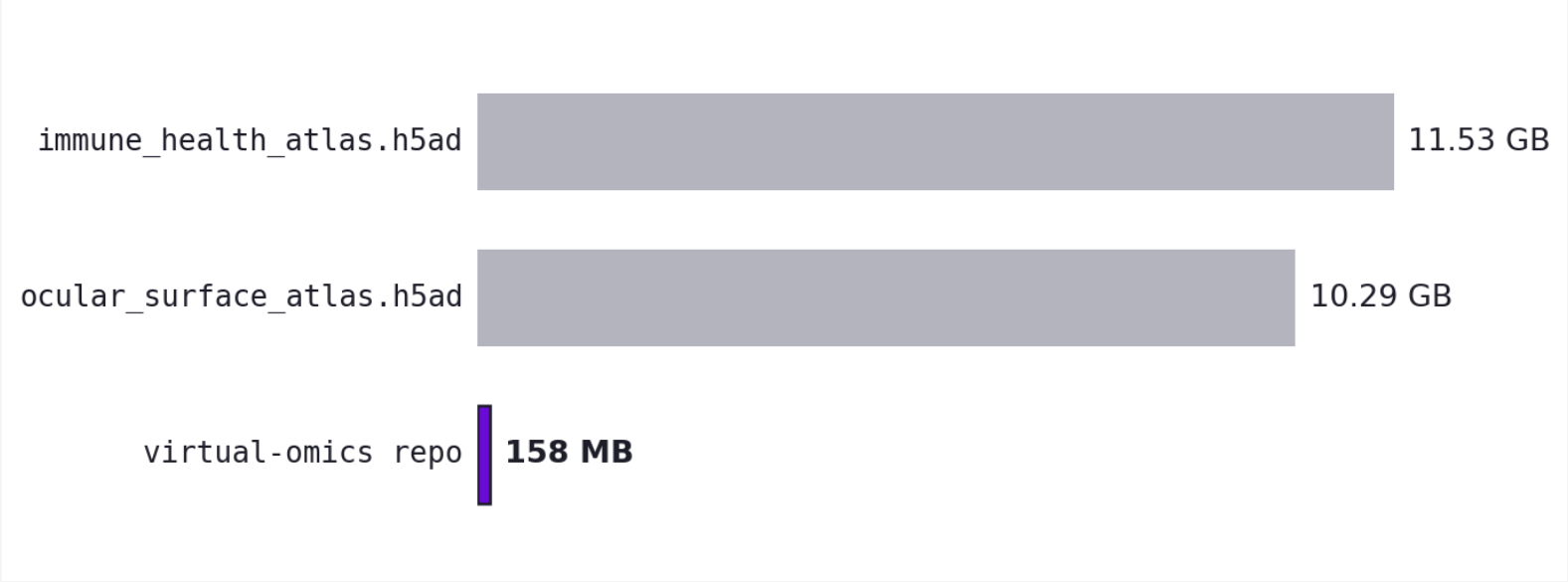
h5ad

- I can select across multiple files without downloading both!

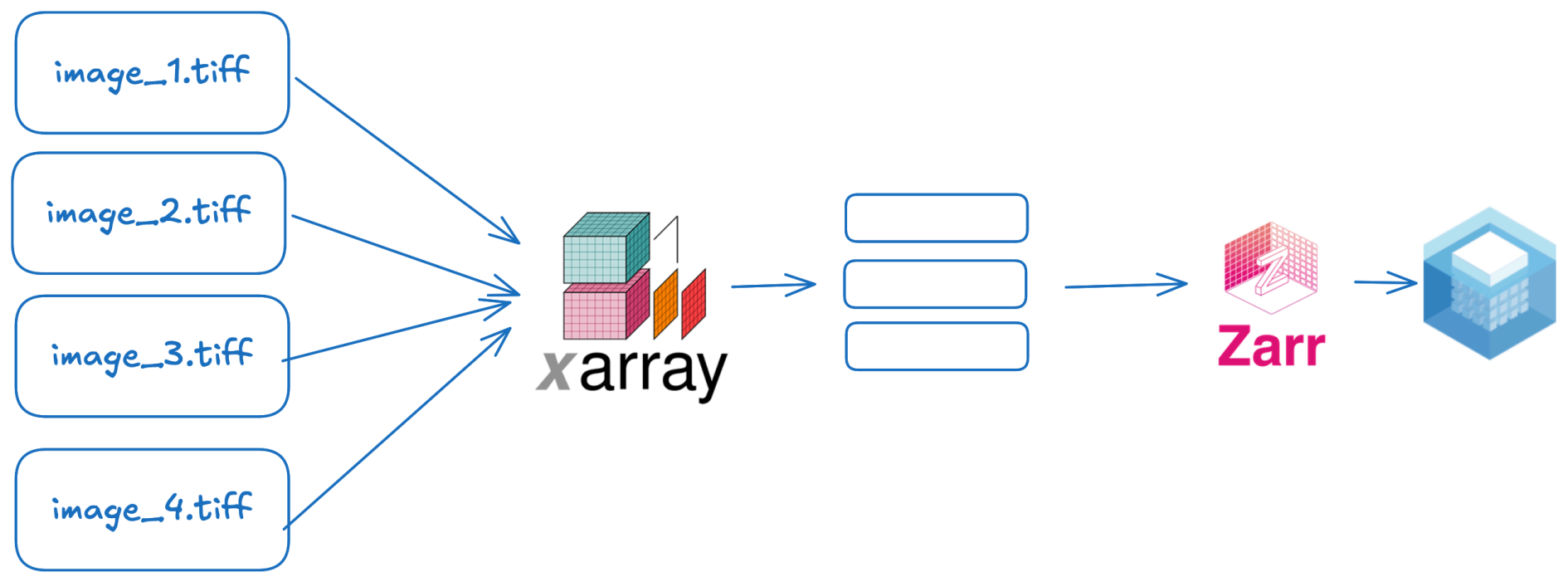
```
repo   = Client().get_repo("earthmover-public/virtual-omics")
immune = open_anndata(repo, group="allen/immune_health_atlas")
ocular = open_anndata(repo, group="ocular_surface_atlas")
immune
```

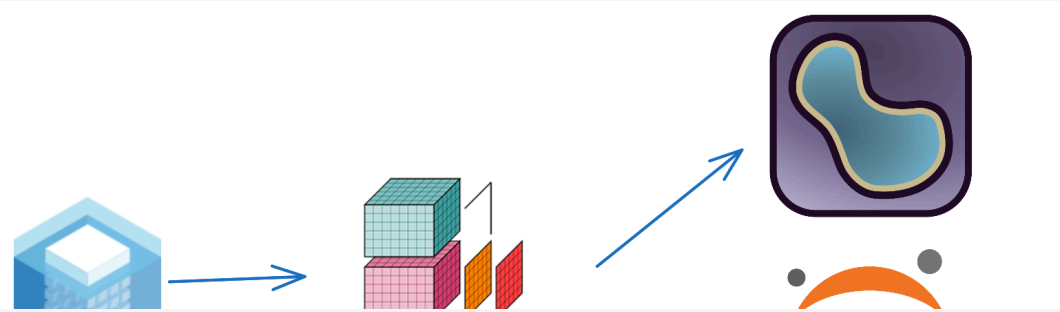
```
AnnData object with n_obs × n_vars = 1821725 × 32357
    obs: 'AIFI_L1', 'AIFI_L3', 'subject.cmv', 'subject.ageGroup',
         'sex', 'donor_id', 'cohort.cohortGuid', 'sample.visitName'
    var: 'mito', 'highly_variable', 'means', …
```

```
one_immune = immune[immune.obs.donor_id == d1].to_memory()
one_ocular = ocular[ocular.obs.donor_id == d2].to_memory()
```



xarray and virtualization





conclusion

OME — try out the prototype · collaborate

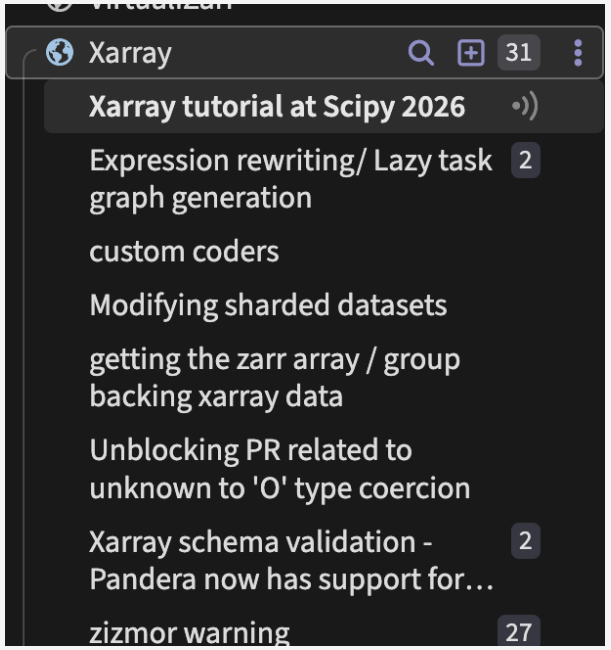
Indexes — let's build them together

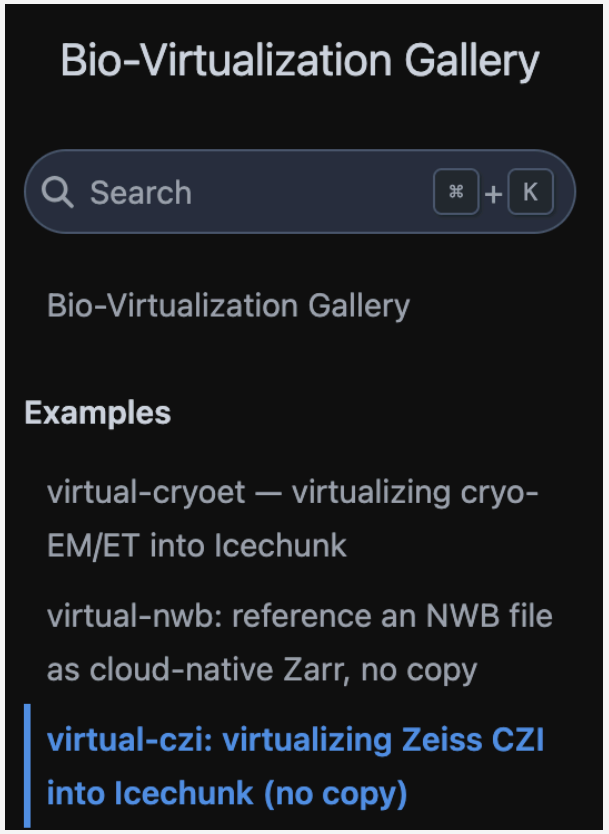
Virtualization — what would you do with this?

Collaborate

OSSCI.ZULIPCHAT.COM
NAPARI.ZULIPCHAT.COM
IMAGESC.ZULIPCHAT.COM
SCVERSE.ZULIPCHAT.COM

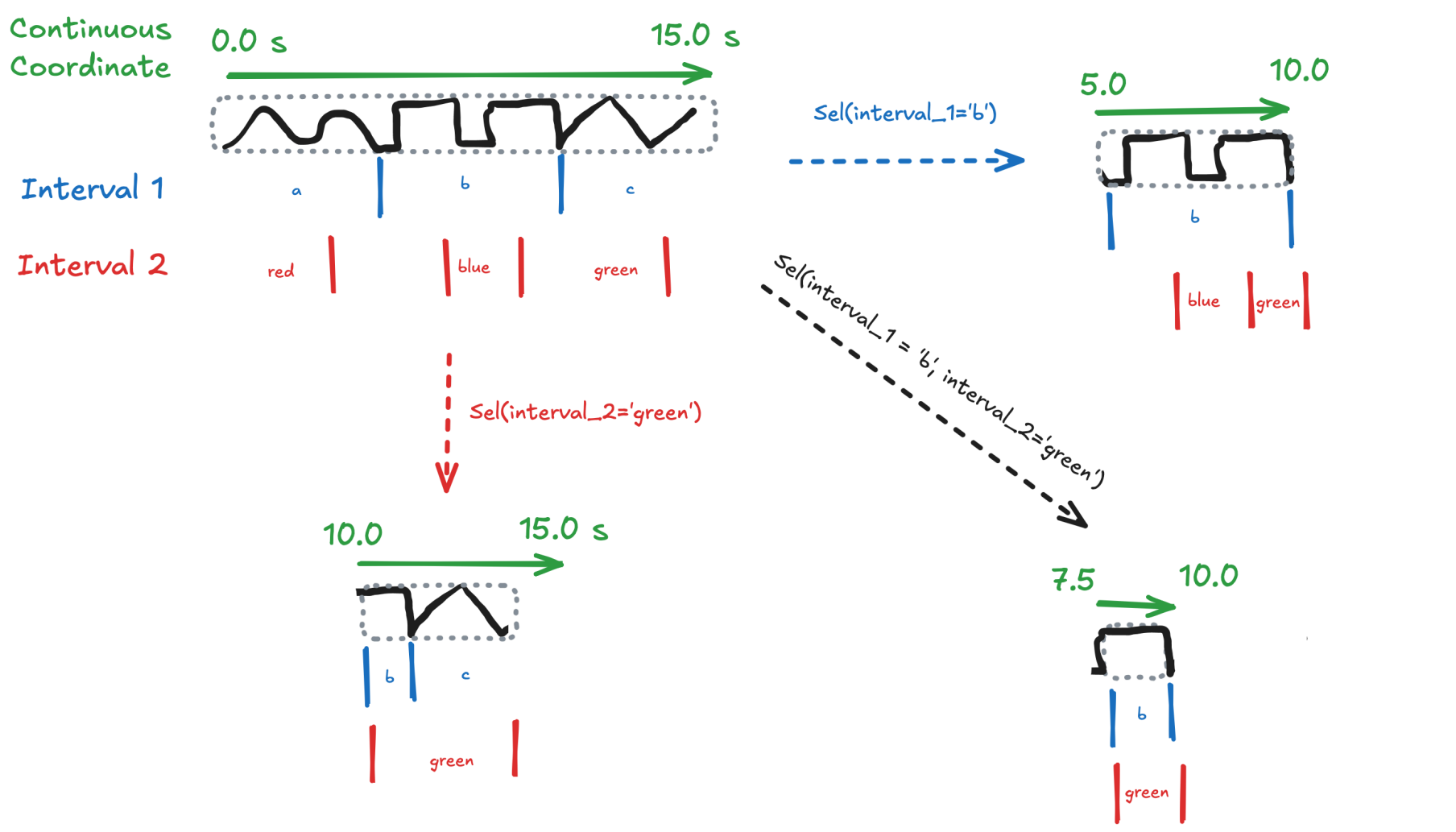
IAN@EARTHMOVER.IO · IANHUNTISAAK@GMAIL.COM





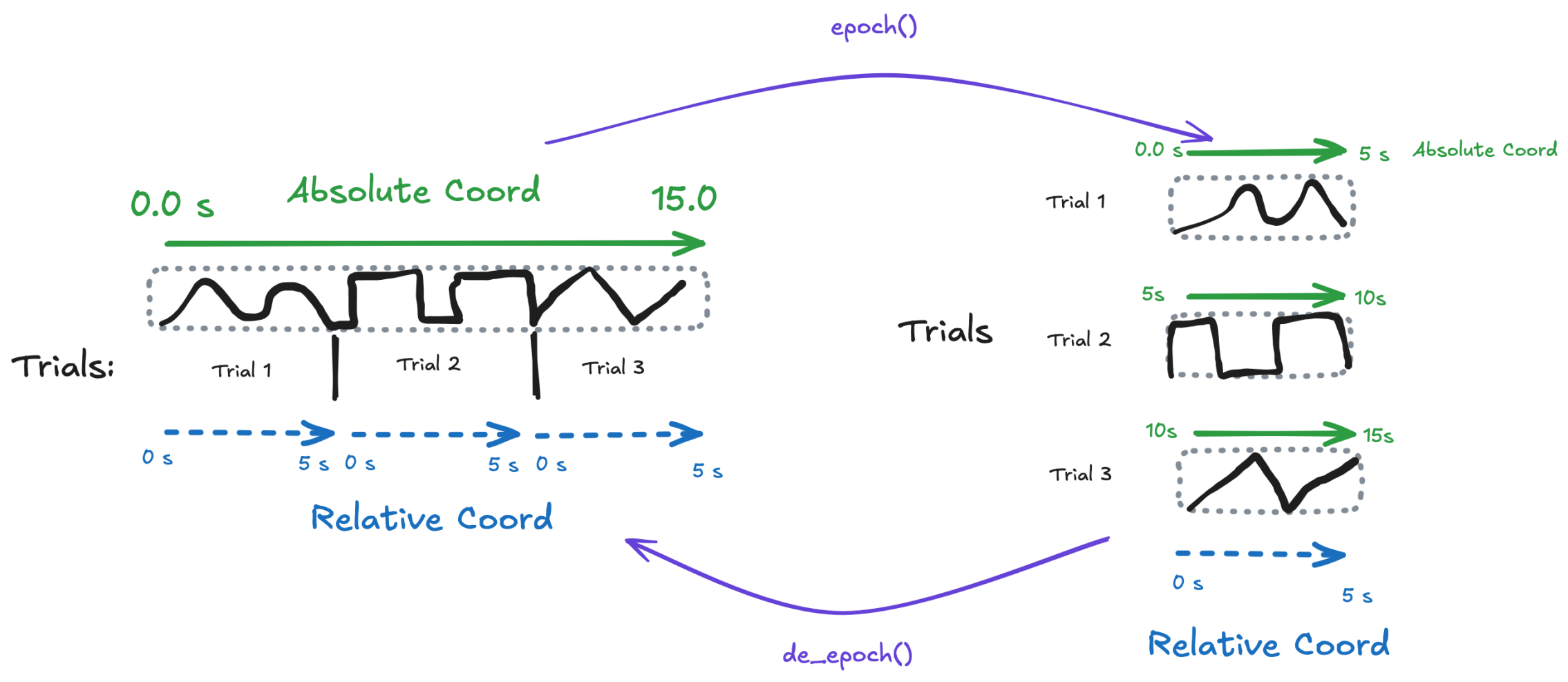
linked indexes

one selection across two linked coordinates



absolute ⇄ relative

`epoch()` / `de_epoch()` — same data, per-trial or wall-clock



IANHUNTISAAK.COM/XARRAY-LINKED-INDEXES