# Just Over the Horizon

## Big Data in AnnData

ilan.gold@scverse.org

Ilan Gold

github.com/ilan-gold

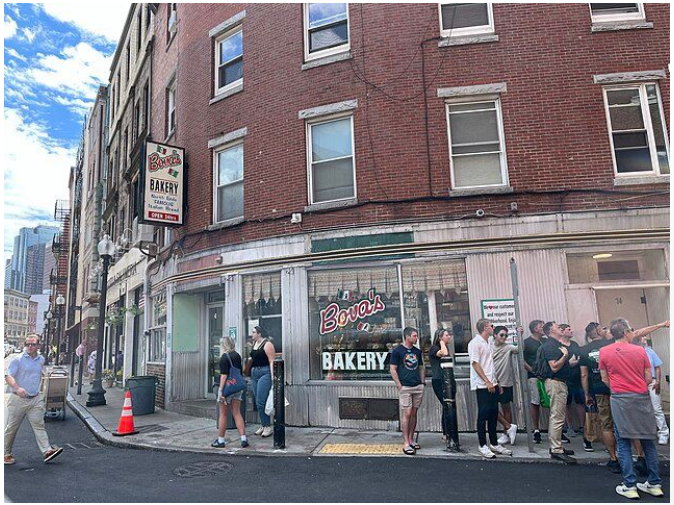
NUMFOCUS
OPEN CODE = BETTER SCIENCE

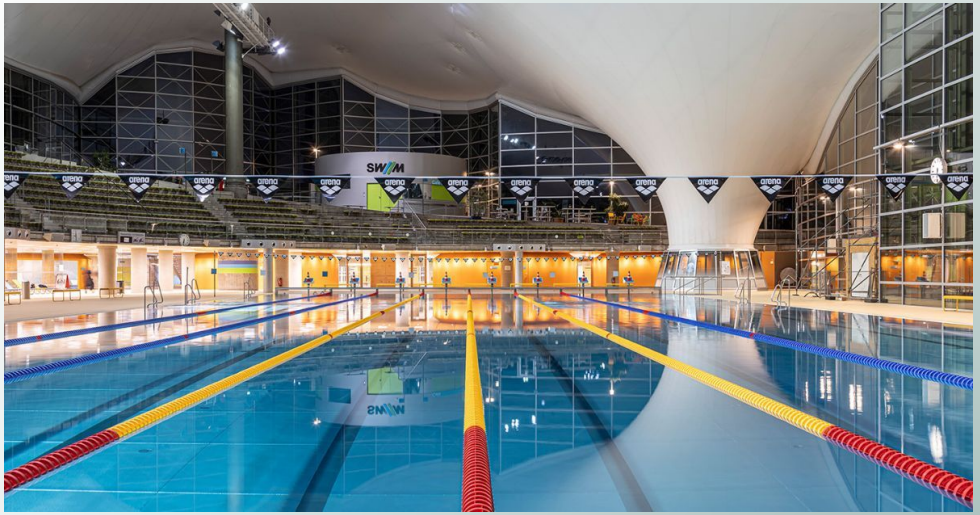
# Who am I?

Beginnings...

# Who am I?

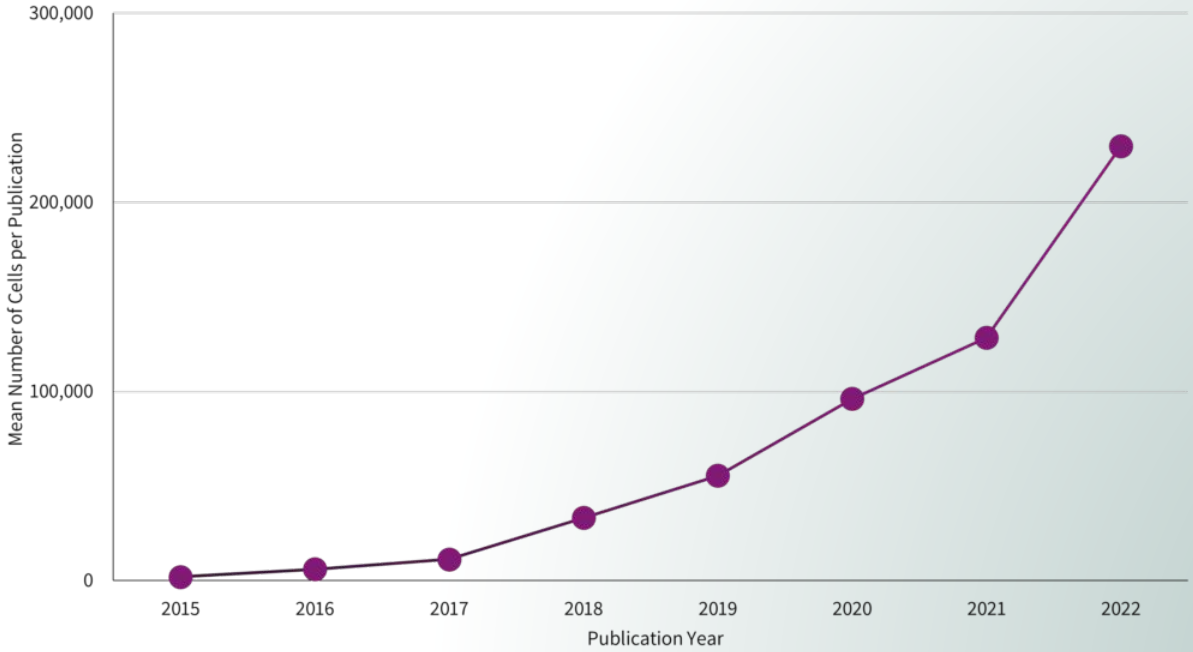
More relevantly...



# Growing Throughput

Millions of cells, spots, nuclei...




https://docs.google.com/spreadsheets/d/1En7-UV0k0laDiIfjFkdn7dggyR7jIk3WH8QgXaMOZF0/edit?gid=0#gid=0

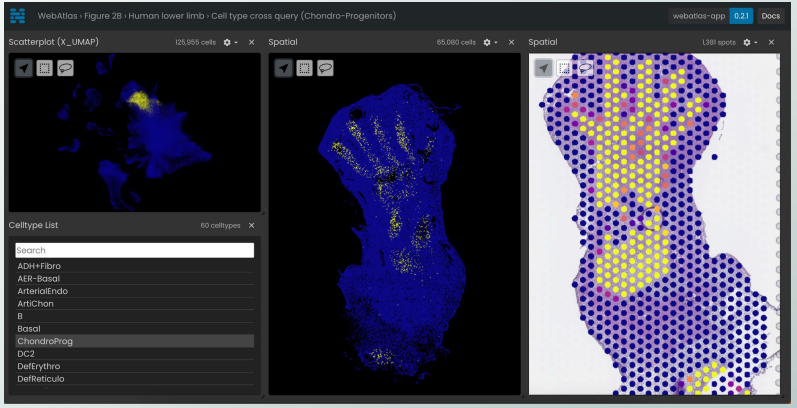
https://www.parsebiosciences.com/blog/publication-trends-of-scrna-seq-research/

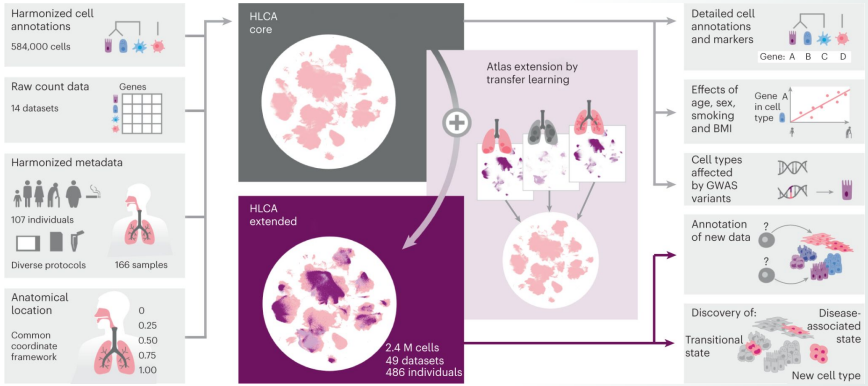
# Dataset + Dataset + … = New (Bigger) Dataset

Atlases, Consortia, and More

HuBMAP

Human BioMolecular Atlas Program




Census

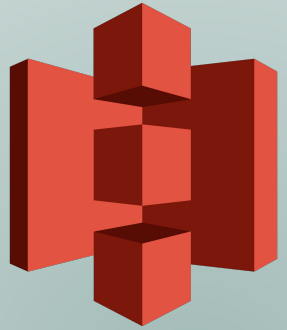
Powered by [tile]DB

# Different Environments

Data lives everywhere!

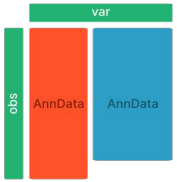
+


+



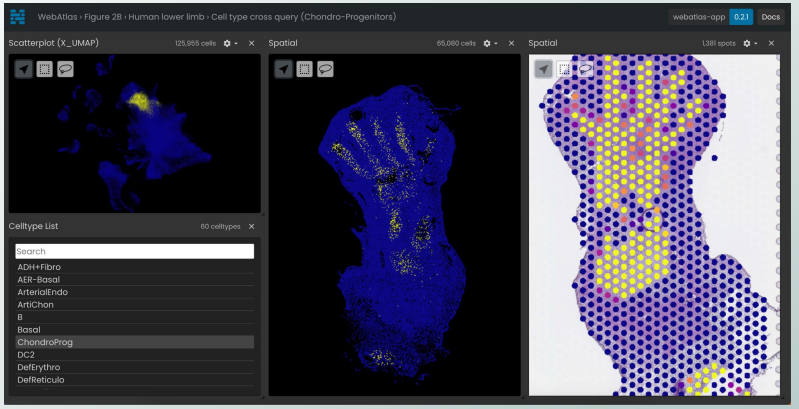
# Different Platforms

## Across Languages and Tools




mudata

multimodal data

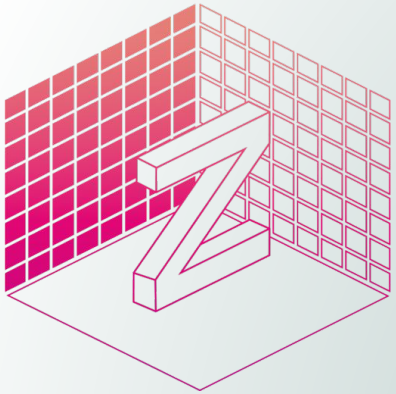



# On disk storage

What works best for different environments?

## Cloud Ready

- Language agnosticism
- Fast access
- Modular API (on-premise s3, GCP)
- Scalable

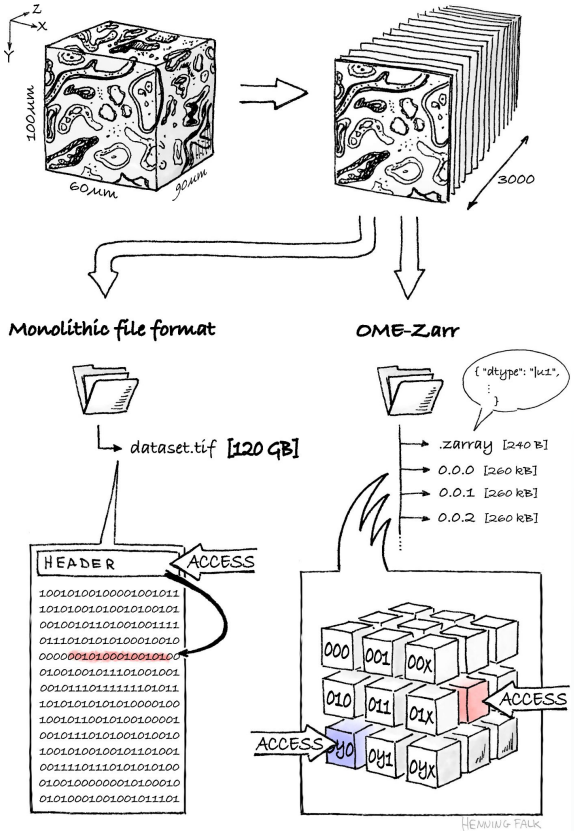


[tile]DB

HDF

# What are the properties we want?

Flexibility, Flexibility, Flexibility




## Cloud Ready

- Take what you want, when you want
- Quick, parallelizable access
- Modifiable

https://github.com/zarr-developers/zarr-illustrations-falk-2022/blob/main/300dpi/monolithic-vs-chunked-300dpi.png

# Dask for Python

Distributed compute, parallelism, out-of-RAM with one tool

## What is Dask?

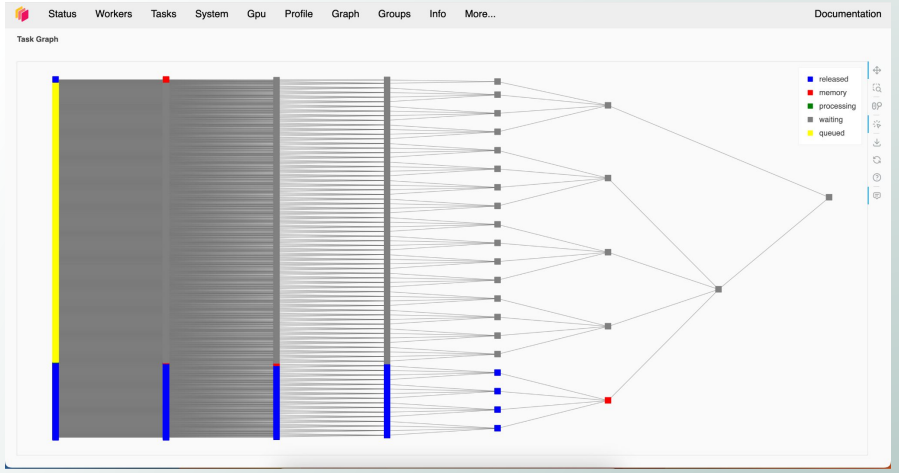
- “Easy Parallel Python that does what you need”
  - Lazy
  - Out-of-core
  - Compute agnostic
  - Array-focused

# Dask for Python

## Memory + Compute

What does lazy compute look like?

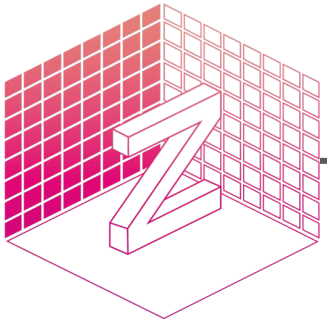
- Scheduling
- Task graphs
- i/o

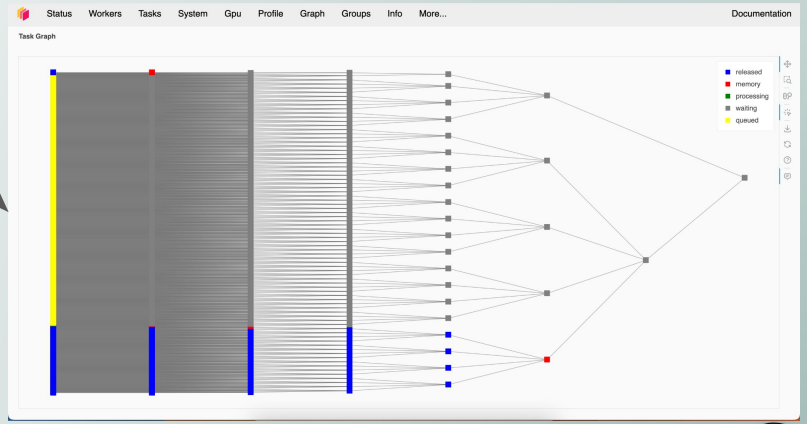




# Dask for Python

Distributed compute, parallelism, and out-of-RAM with one tool




https://data-apis.org/array-api/latest/

# APIs (for now)

## Getting Your Data into `AnnData`

### APIs

- `anndata.experimental.read_elem_as_dask`
- `dask.array.map_blocks`
- `anndata.{experimental.}read_elem`

Alacritty

```
In [1]: import anndata as ad

In [2]: from anndata.experimental import read_elem_as_dask

In [3]: import fsspec, zarr

In [4]: mapper = fsspec.get_mapper(
   ...:     "https://vitessce-demo-data.storage.googleapis.com/anndata-demos/BALF_VIB-UGent_processed_cleaned.zarr/"
   ...: )

In [5]: store = zarr.LRUStoreCache(mapper, max_size=2**28)

In [6]: group = zarr.open_consolidated(store)
```

Prepare a `zarr.Group`

```
In [7]: adata = ad.AnnData(
   ...:     X = read_elem_as_dask(group["X"]),
   ...:     obs = ad.read_elem(group["obs"]),
   ...:     var = ad.read_elem(group["var"])
   ...: )

In [8]: adata
Out[8]:
AnnData object with n_obs × n_vars = 275056 × 24740
    obs: 'orig.ident', 'Age', 'Sex', 'Race', 'Ethnicity', 'BMI', 'Pre-existing heart disease', 'Pre-existing lung disease', 'Pre-existing kidney disease',
 'Pre-existing diabetes', 'Pre-existing hypertension', 'Pre-existing immunocompromised condition', 'Smoking', 'SARS-CoV-2 PCR', 'SARS-CoV-2 Ab', 'Symptoma
tic', 'Admitted to hospital', 'Highest level of respiratory support', 'Vasoactive agents required during hospitalization', '28-day death', '28-day outcome
', 'Disease classification', 'Organ System', 'Source', 'Days since hospital admission', 'SOFA', 'Technology', 'Method', 'CITE-Seq panel', 'Reference', 'In
stitute', 'Creation date', 'Annotation'
    var: 'feature_type', 'gene_id'
```

Use APIs to create the `AnnData` object

“In memory”
`pandas.DataFrame`

```
In [9]: adata.obs
Out[9]:
                        orig.ident    Age Sex     Race  ...         Reference      Institute  Creation date   Annotation
COV002_AAACCCAAGAGTCTTC-1     COV002  40-50   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil
COV002_AAACCCAAGATAGGGA-1     COV002  40-50   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0      Doublet
COV002_AAACCCAAGCATGCGA-1     COV002  40-50   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0  CD8+ T-cell
COV002_AAACCCAAGGGTACAC-1     COV002  40-50   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Macrophage
COV002_AAACCCAAGTACGAGC-1     COV002  40-50   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0  CD8+ T-cell
...                              ...    ...  ..      ...  ...               ...            ...            ...          ...
COV037_TTTGTTGTCAGAGTTC-1     COV037  18-30   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil
COV037_TTTGTTGTCAGCACCG-1     COV037  18-30   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil
COV037_TTTGTTGTCCTCAGAA-1     COV037  18-30   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil
COV037_TTTGTTGTCCTTGGAA-1     COV037  18-30   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil
COV037_TTTGTTGTCTCCTGCA-1     COV037  18-30   F  Unknown  ...  GRCh38 + SARSCoV  VIB/IRC-UGent     20200908.0   Neutrophil

[275056 rows x 33 columns]

In [10]: adata.X
Out[10]: dask.array<make_dask_chunk, shape=(275056, 24740), dtype=float64, chunksize=(1000, 24740), chunktype=scipy.csr_matrix>
```

“Out-of-core”
`dask.array.Array`

```
In [11]: adata[adata.obs["orig.ident"] == "COV002", :]
Out[11]:
View of AnnData object with n_obs × n_vars = 20172 × 24740
    obs: 'orig.ident', 'Age', 'Sex', 'Race', 'Ethnicity', 'BMI', 'Pre-existing heart disease', 'Pre-existing lung disease', 'Pre-existing kidney disease',
 'Pre-existing diabetes', 'Pre-existing hypertension', 'Pre-existing immunocompromised condition', 'Smoking', 'SARS-CoV-2 PCR', 'SARS-CoV-2 Ab', 'Symptoma
tic', 'Admitted to hospital', 'Highest level of respiratory support', 'Vasoactive agents required during hospitalization', '28-day death', '28-day outcome
', 'Disease classification', 'Organ System', 'Source', 'Days since hospital admission', 'SOFA', 'Technology', 'Method', 'CITE-Seq panel', 'Reference', 'In
stitute', 'Creation date', 'Annotation'
    var: 'feature_type', 'gene_id'

In [12]: adata[adata.obs["orig.ident"] == "COV002", :].X.compute()
Out[12]:
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 41390886 stored elements and shape (20172, 24740)>
```

Lazy ops! Only get data with `compute`

# The Future: xarray

## Unifying `AnnData`, `Dask`, and `xarray`

### What is xarray?

- Lazy, annotated, data
- Backend/compute agnostic
- Integrated with python scientific ecosystem
- *very similar to* `AnnData`

# The Future: xarray

Unifying `AnnData`, `Dask`, and `xarray`

### APIs (PR #1247)

- At first, just replacing `pandas.DataFrame`
- `anndata.experimental.read_{elem_lazy,backed}`
- Bring `xarray` to dataframes → lazy `groupby`?

Alacritty

```
In [1]: from anndata.experimental import read_backed, read_elem_lazy

In [2]: import fsspec, zarr

In [3]: mapper = fsspec.get_mapper(
   ...:     "https://vitessce-demo-data.storage.googleapis.com/anndata-demos/BALF_VIB-UGent_processed_cleaned.zarr/"
   ...: )
   ...:

In [4]: store = zarr.LRUStoreCache(mapper, max_size=2**28)

In [5]: adata = read_backed(store)
<frozen abc>:106: FutureWarning: xarray subclass Dataset2D should explicitly define __slots__

In [6]: adata
Out[6]:
AnnData object with n_obs × n_vars = 275056 × 24740
    obs: 'orig.ident', 'Age', 'Sex', 'Race', 'Ethnicity', 'BMI', 'Pre-existing heart disease', 'Pre-existing lung disease', 'Pre-
existing kidney disease', 'Pre-existing diabetes', 'Pre-existing hypertension', 'Pre-existing immunocompromised condition', 'Smok
ing', 'SARS-CoV-2 PCR', 'SARS-CoV-2 Ab', 'Symptomatic', 'Admitted to hospital', 'Highest level of respiratory support', 'Vasoacti
ve agents required during hospitalization', '28-day death', '28-day outcome', 'Disease classification', 'Organ System', 'Source',
 'Days since hospital admission', 'SOFA', 'Technology', 'Method', 'CITE-Seq panel', 'Reference', 'Institute', 'Creation date', 'A
nnotation'
    var: 'feature_type', 'gene_id'
    obsm: 'X_umap'
    layers: 'X_csc'
```

Prepare a `zarr.Group`

Fully lazy reading!

```
In [7]: adata.obs
Out[7]:
<xarray.Dataset2D> Size: 75MB
Dimensions:                                 (obs_names: 275056)
Coordinates:
  * obs_names                               (obs_names) object 2MB ...
Data variables: (12/33)
    orig.ident                              (obs_names) category 2MB ...
    Age                                     (obs_names) category 2MB ...
    Sex                                     (obs_names) category 2MB ...
    Race                                    (obs_names) category 2MB ...
    Ethnicity                               (obs_names) category 2MB ...
    BMI                                     (obs_names) category 2MB ...
    ...                                      ...
    Method                                  (obs_names) category 2MB ...
    CITE-Seq panel                          (obs_names) category 2MB ...
    Reference                               (obs_names) category 2MB ...
    Institute                               (obs_names) category 2MB ...
    Creation date                           (obs_names) float64 2MB dask.array<chunksize=(34382,), meta=np.ndarray>
    Annotation                              (obs_names) category 2MB ...
```

“Out-of-core”
`xarray.Dataset`

```
In [8]: adata.X
Out[8]: dask.array<make_dask_chunk, shape=(275056, 24740), dtype=float64, chunksize=(1000, 24740), chunktype=scipy.csr_matrix>

In [9]: adata.obs["SARS-CoV-2 PCR"].astype("bool")
Out[9]:
<xarray.DataArray 'SARS-CoV-2 PCR' (obs_names: 275056)> Size: 275kB
dask.array<astype, shape=(275056,), dtype=bool, chunksize=(34382,), chunktype=numpy.ndarray>
Coordinates:
  * obs_names  (obs_names) object 2MB 'COV002_AAACCCAAGAGTCTTC-1' ... 'COV037...

In [10]: adata.obs["Annotation"]
Out[10]:
<xarray.DataArray 'Annotation' (obs_names: 275056)> Size: 2MB
[275056 values with dtype=category]
Coordinates:
  * obs_names  (obs_names) object 2MB 'COV002_AAACCCAAGAGTCTTC-1' ... 'COV037...
```

`dask.array.Array` /
Categorical Data