earthmover

icechunk: git for single-cell and spatial data

SCVERSE 2026 | COPENHAGEN

Luiz Irber, PhD
Earthmover PBC

these slides
slides.earthmover.io/main/scverse-2026
docs: icechunk.io

versioning
commits, branches and tags for array data

zero-copy ingest
reference the archive, copy nothing

consistency
readers never see half a write

ICECHUNK DOES ALL THREE. TODAY: THE FIRST TWO.

THE PROBLEM

nobody knows which copy made the figure

The Braun brain atlas on CELLxGENE is one 12 GB `.h5ad`, 1.67 million cells. Every annotation pass copies it and edits `obs`. Each copy carries the full 12 GB matrix.



48 GB
four copies of one unchanged matrix

no author
no record of who changed cell_type, or when

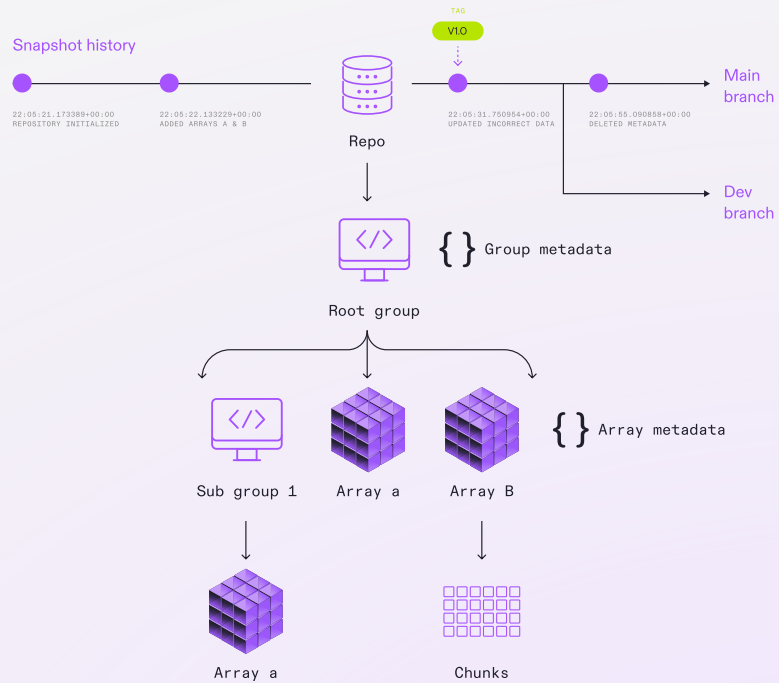
no source
no way to tell which copy the figure came from

ICECHUNK

git for array data

- **Commits, branches and tags** on a Zarr store. Every old version stays readable.
- **Transactions:** readers never see half a write
- **Virtual chunks:** point at the files you already have
- A library, not a server. Python, Julia and Java today, more languages in progress.
- From the maintainers of Zarr and Xarray. In production at NASA and NOAA.

icechunk.io

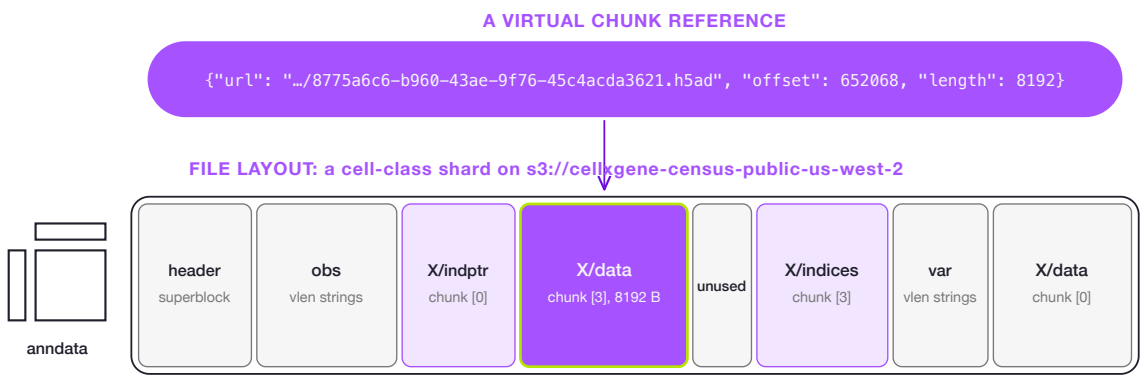


ICECHUNK · ZERO-COPY INGEST

reference the archive, copy nothing

A virtual chunk is a pointer `(url, offset, length)` into a file that already exists.

- The CSR arrays of an `.h5ad` are gzip HDF5 chunks. Zarr reads them in place.
- Read only the cells you query: 1,000 cells fetch 1.3 MB of a 13 MB shard
- Fix bad metadata or a corrupt chunk with one native chunk. The archive stays untouched.



X/data, X/indices and X/indptr each get one reference per chunk. The repo copies nothing.

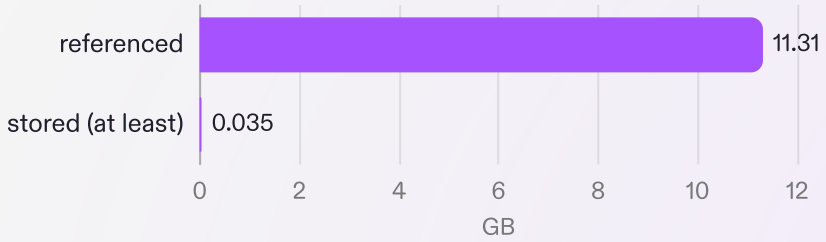
ICECHUNK · IN PRACTICE

an atlas with zero copied bytes

Braun et al. 2023, first-trimester human brain atlas. 11 `.h5ad` files, 1,654,197 cells, on CZI's public S3.

0.3%

stored in the repo, against the bytes it references.
Manifests 15 MB, native `obs` and `var` 20 MB.
Zero expression bytes copied.



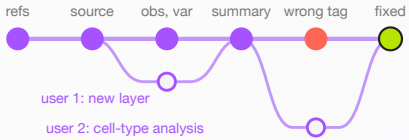

```python
import anndata as ad, scanpy as sc, zarr
import braun  # earthmover-support/2026-scverse

session = repo.writable_session("main")
for c in braun.CLASSES:
    # walks the HDF5 chunk index, reads no counts
    ds = braun.virtual_matrix(c.census_url, braun.registry())
    ds.vz.to_icechunk(session.store, group=f"{c.group}/X",
                      last_updated_at=checked_at)
session.commit("virtual refs: 11 cell classes")

root = zarr.open(repo.readonly_session("main").store, mode="r")
adata = ad.read_zarr(root["classes/erythrocyte"])
sc.pp.normalize_total(adata, target_sum=1e4)
sc.pp.log1p(adata)
sc.pp.highly_variable_genes(adata, n_top_genes=2000)
# the same 2,000 genes as from the .h5ad read directly
```

scanpy reads `X` as a normal sparse matrix. The bytes stay in CZI's bucket.

ICECHUNK · WHAT YOU GET

history, undo, spatial

history you can audit



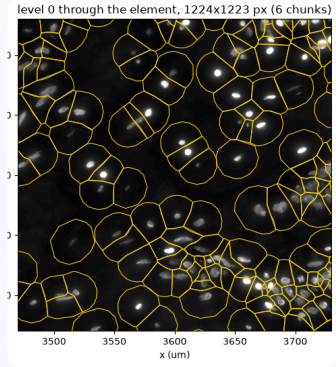
Six commits on one atlas. Every snapshot stays readable. Two users work on two branches, and nobody rewrites `X`.

undo a mistake

```
root.attrs["census_release"] = "2024-01-01"
session.commit("record census release")
# wrong tag. fix it with one more commit:
root.attrs["census_release"] = "2025-11-08"
session.commit("correct census release")
```

Both commits stay in the history. Open either snapshot and read what the data said then. No copy, no renamed file.

spatialdata too



Xenium, 1.33 GB, 1.05% copied. This window reads 6 of 910 chunks.

earthmover



slides.earthmover.io/main/scverse-2026

github.com/earthmover-support/2026-scverse

icechunk.io · virtualizarr.readthedocs.io · earthmover.io

BONUS SLIDES

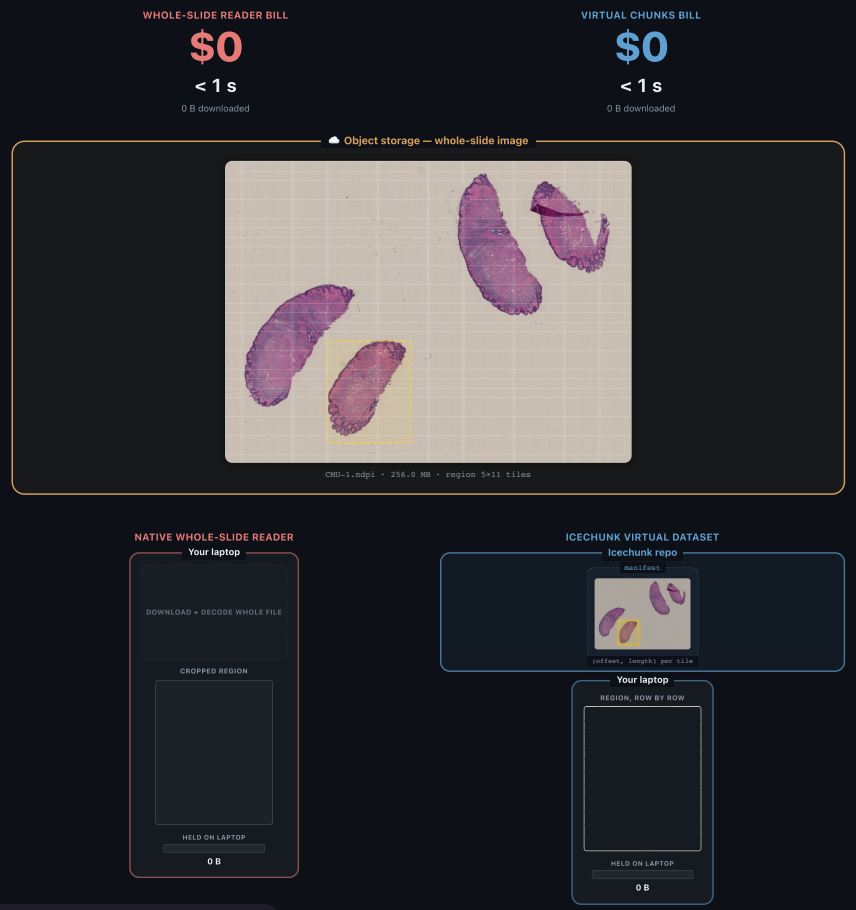
in case there is time, or questions



ICECHUNK ILLUSTRATED

interactive animations

Ian Hunt-Isaak, earth-mover.github.io/icechunk-illustrated



ICECHUNK ILLUSTRATED

query the archive in place: a whole-slide image



ICECHUNK ILLUSTRATED

only the chunks that changed

ICECHUNK ILLUSTRATED

schrödinger's chunks: consistent reads

ICECHUNK ILLUSTRATED

the race condition: atomic commits

ICECHUNK ILLUSTRATED

move without moving data

DATASCALES

single-cell on zarr and icechunk at the allen institute

Alexander Holly, Christian La France and Aishwarya Chander, Allen Institute for Immunology.

- github.com/aifimmunology/DataScales: tools, benchmarks and workflows
- github.com/A-Jolly-Holly/annizarr: h5ad to a versioned Icechunk store in one command
- Two users, two branches: one normalizes into a new layer, one commits a cell-type analysis. Nobody rewrites `X`.
- 30 million cells from 5 cohorts merged into one sorted store

```
pip install "annizarr[icechunk]"
annizarr convert sample.h5ad -o repo.icechunk --ic
annizarr add-expr repo.icechunk --branch dev
```

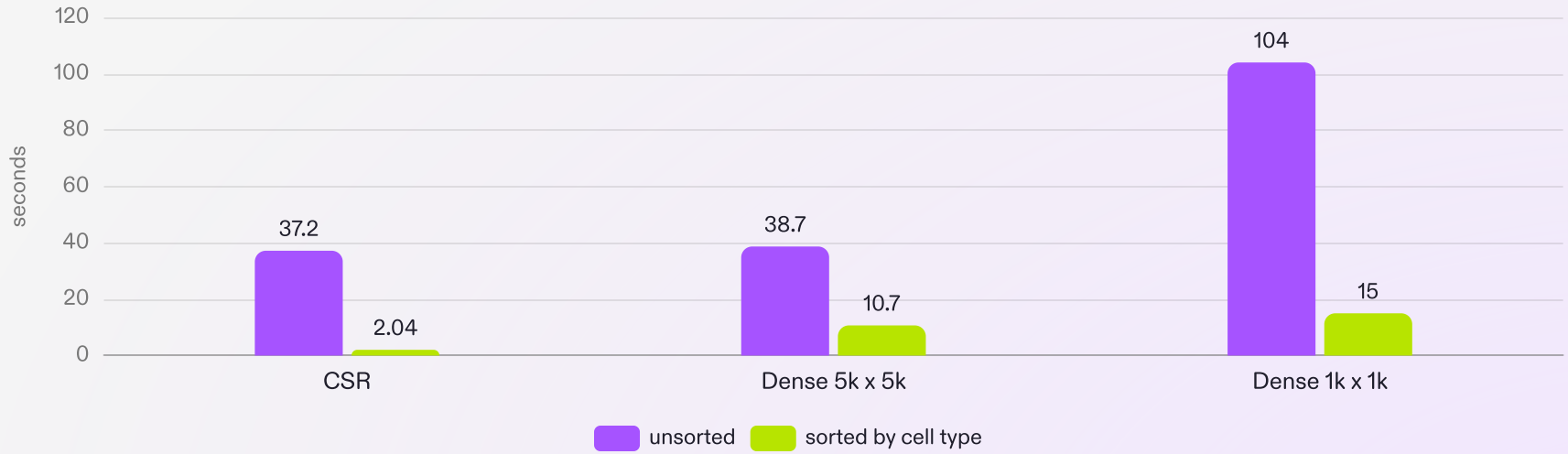
```
import annizarr as az

repo = az.Repo("repo.icechunk")
repo.checkout("experiment", create=True)
root = repo.open_zarr("w")
repo.commit("scaled X")
repo.log()
```

For data you own: native chunks, your layout, your sort order. Reference what you do not own.

DATASCALES

sort by what you query

Cell-type query of about 200,000 B cells out of 2 million, stores on GCS, same chunk layout per pair. Only the row order changes.



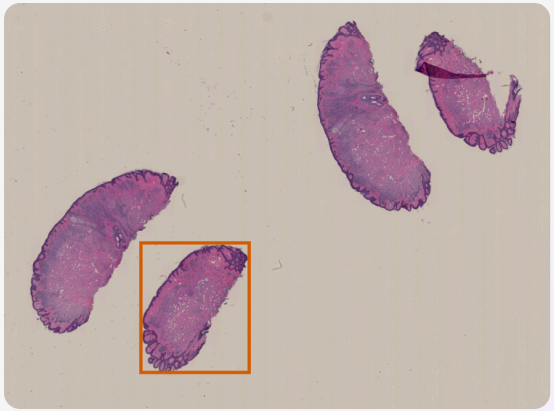
A virtual reference keeps the source's row order. Reference first. Rewrite the subset you query most, later.

SCIPY 2026

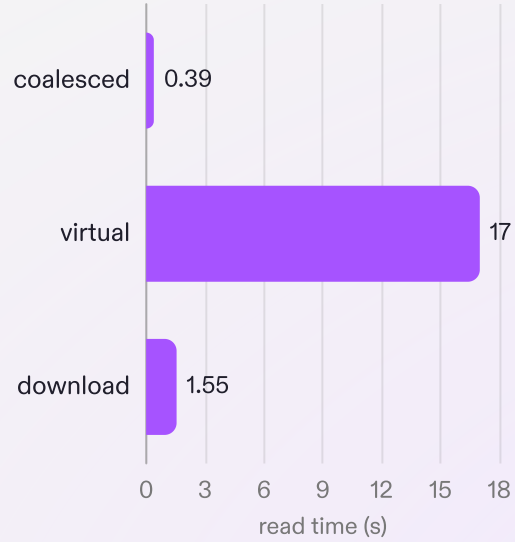
two slides from ian hunt-isaak's talk

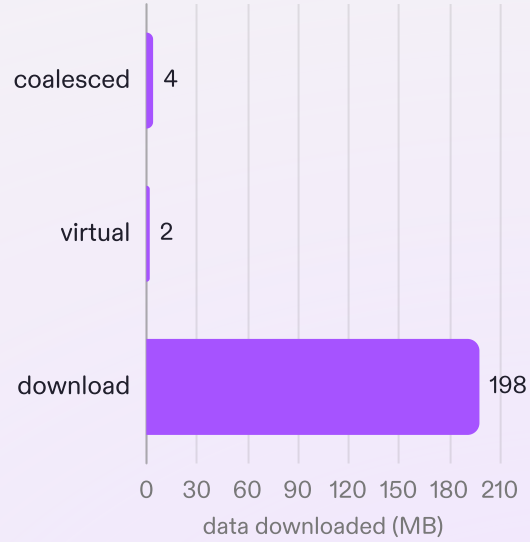
slides.earthmover.io/main/xarray-icechunk-bio-scipy-2026

one tissue region from a remote whole-slide image



the region you asked for, on a 6 GB slide



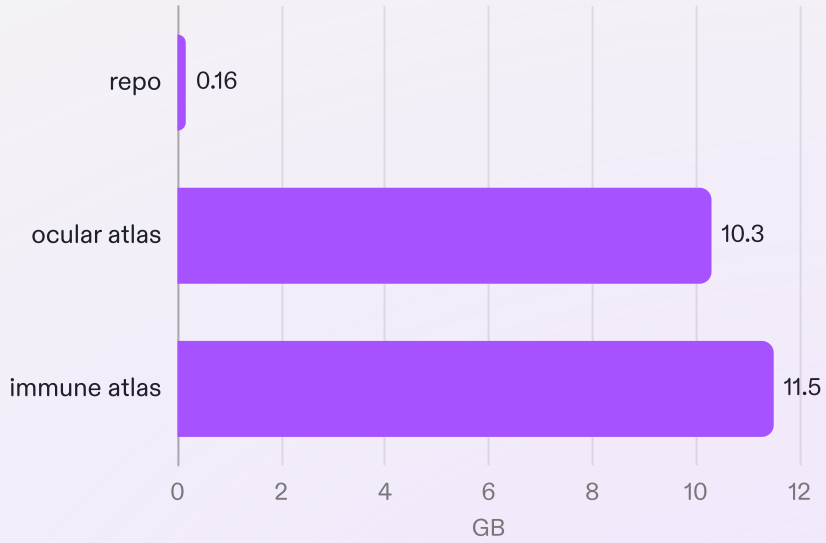


SCIPY 2026

select across two atlases without downloading either

```
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")

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

`immune` is 1,821,725 cells by 32,357 genes. Two CELLxGENE h5ad files, 11.5 GB and 10.3 GB. The repo is 158 MB of references, not counts.



LIMITS

what breaks, and the check that shows it

limit	why	check
`obs` and `var` do not virtualize	HDF5 variable-length strings. Zarr v3 rejects that dtype. The repo writes both natively.	`pixi run check`
a COMPACT dataset reads as zeros	An HDF5 dataset under about 64 kB has no offset. VirtualiZarr emits an empty manifest. **No error.**	`pixi run check-s3`
virtual matrices do not concatenate	The CSR concat axis is `nnz`. zarr-python has a rectilinear grid. VirtualiZarr requires uniform chunks.	`pixi run check-concat`
every reference needs a checksum	A changed source returns different numbers with no error. `last_updated_at` is whole seconds. Round it up.	`pixi run check-s3`
the source owns the chunk grid	A reference keeps the source's chunks and row order. A different query layout needs a rewrite.	DataScales slides

One check per limit. An upstream fix breaks a check, not a notebook.

LIMITS

spatialdata: shapes and points need a path

What a maintainer sees in ten seconds, with released spatialdata 0.8.0:

```
image committed to an Icechunk store and read back: ok
same shapes written to a local store: ok
same shapes to an Icechunk store raised AttributeError:
  'IcechunkStore' object has no attribute 'root'
```

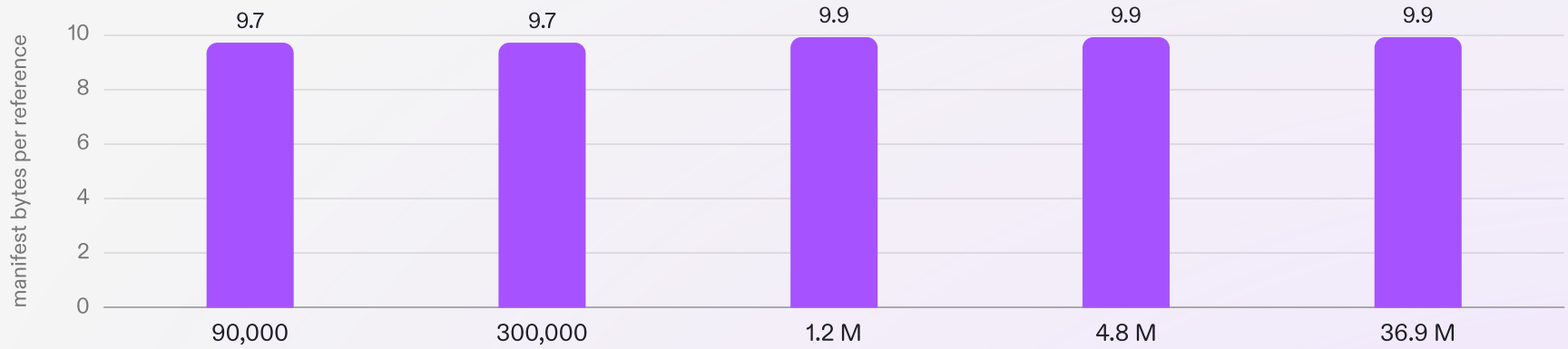
Same object, same group path. Only the store differs.

- Images, labels and tables write to Icechunk today.
- The shapes writer reads `store.root`, a filesystem path.
- PR #1107 fixes the store abstraction, then its `parquet_fs_and_path` refuses any store that is not local or fsspec.
- Five commits on top of #1107 fix that second stop.
- `pixi run check-store-path` prints the lines on the left.

LIMITS

does it scale to an atlas?

Manifest cost stays near 10 bytes per reference from 90,000 to 36.9 million references. `pixi run check-manifest`



- At 36.9 million references the manifest is 364 MB. A cold read takes 1.1 s, a warm read 0.6 ms.
- Opening the repo stays between 0.7 and 2.6 ms across the whole range.
- At 10 bytes per reference, a human atlas `X` manifest is about 12 MB.