nature methods

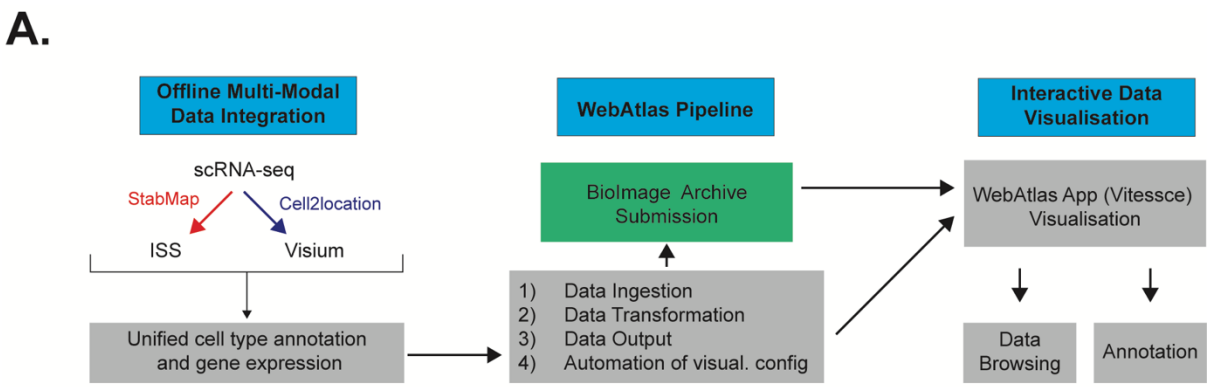
Supplementary information

https://doi.org/10.1038/s41592-024-02371-x

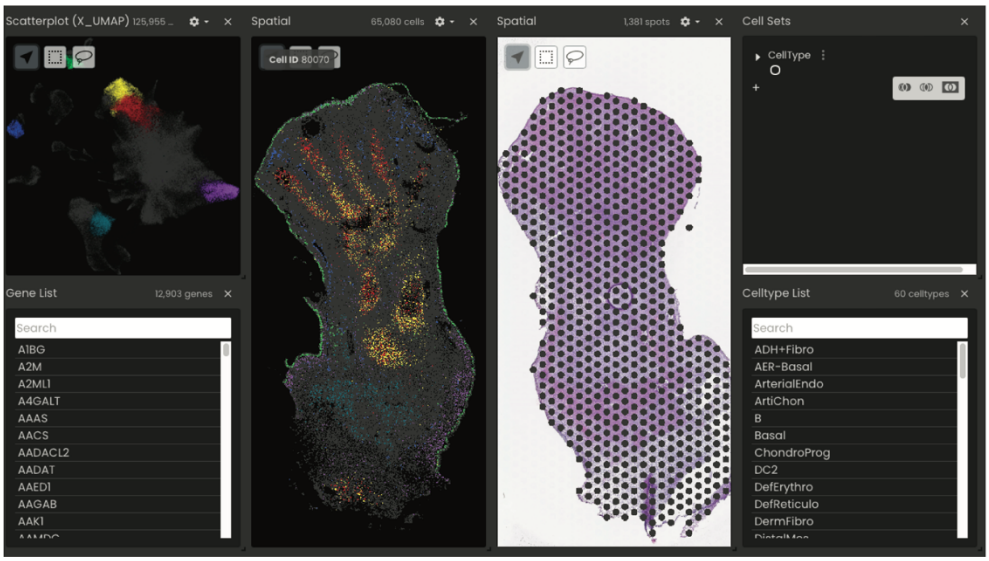
WebAtlas pipeline for integrated single-cell and spatial transcriptomic data

In the format provided by the authors and unedited

Supplementary Figure 1



B.

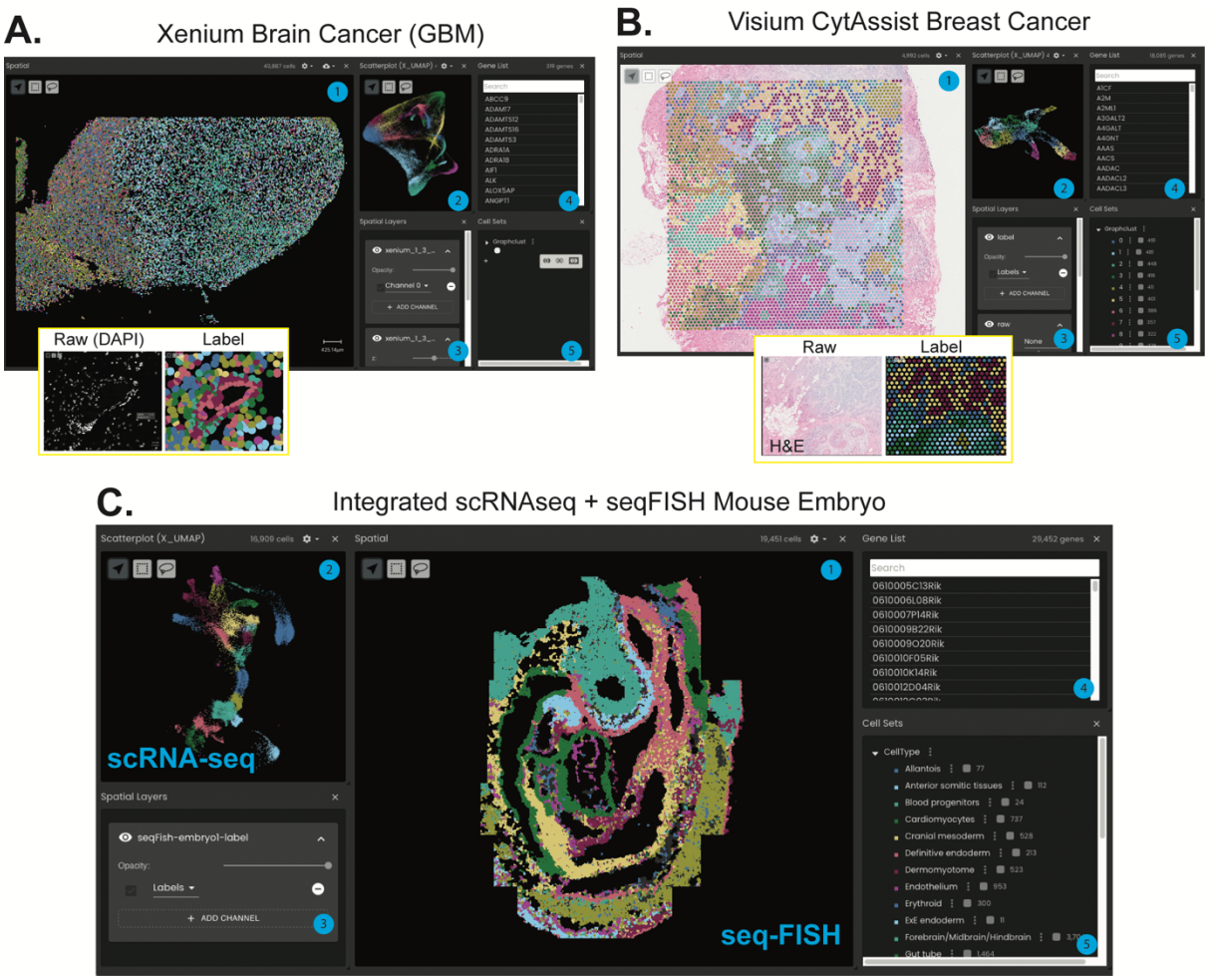


Direct visualisation of lower limb atlas datasets on the BioImage Archive via WebAtlas.

A) Workflow for WebAtlas ingestion, BioImage Archive upload and direct visualisation of multi-modal limb datasets.

B) A snapshot of the WebAtlas app visualising integrated scRNA-seq, ISS and Visium datasets of the human lower limb loaded from the BioImage Archive.

Supplementary Figure 2



Different ST technologies on WebAtlas.

WebAtlas app snapshots of (A) Xenium Human Brain Cancer (Glioblastoma), (B) Visium CytAssist Human Breast Cancer, and (C) integrated scRNA-seq and seqFISH mouse embryo datasets. For each panel, the Vitessce component windows show the following as numbered. (1) Spatial map of segmented single cells or Visium spots coloured according to annotated cell types or cell/spot transcriptomic clusters. (2) UMAP representation of cell types or cell/spot transcriptomes. (3) Spatial layer console to toggle and adjust raster images, label masks and molecules. (4) Gene search console. (5) Cell type or cell/spot cluster search console. Inset panels show raw fluorescent or brightfield microscopy images, including DAPI for Xenium and H&E images for Visium CytAssist, as well as cell segmentation label images. The Xenium inset also shows RNA molecules.

Supplementary Note 1: The landscape of platforms for dissemination and navigation of tissue atlas data

The integration of single cell and spatial transcriptomics has become a standard approach for building comprehensive tissue atlases. However, the diversity and scale of data types poses significant challenges to usable and interpretable access of multi-modal tissue atlases. Here, we provide a comparison of the WebAtlas pipeline to existing alternative platforms (Sup Table 2) and elaborate our key advances that democratise access to complex tissue atlases.

		Desktop		**Web**				
Platform Features		Loupe Browser	MoBIE[11]	CellxGene Data Portal	Visinity[12]	TissUUmaps 3[13]	SODB[14]	WebAtlas via Vitessce
Visualisation	Tabular data	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Multiscale raster images	Yes	Yes	No	Yes	Yes	No	Yes
	Cell and RNA segmentation	No	Yes	No	Yes	Yes	No	Yes
Remote data navigation		No	Yes	Yes	Yes	Yes	Yes	Yes
Cloud-optimised data storage		No	Yes	No	No	No	No	Yes
Simultaneous browsing of scRNA-seq and ST modalities		No	No	No	No	No	No	Yes
Cross-query of multiple modalities		No	No	No	No	No	No	Yes

Supplementary Table 1: Comparison of WebAtlas with alternative platforms.

Simultaneous browsing and cross-query of multiple modalities: The coordinated navigation of scRNA-seq and ST datasets greatly facilitates biological insights from tissue atlases. Amongst existing platforms, WebAtlas uniquely enables the browsing and cross-query of integrated scRNA-seq and ST modalities. While the MoBIE plugin supports browsing individual imaging-based ST datasets or overlaying images from correlative microscopy, WebAtlas allows simultaneous browsing of integrated single cell and spatial transcriptomic datasets as well as their cell type and gene expression cross-query. This is enabled by 1) the WebAtlas data ingestion pipeline that can load datasets from most common scRNA-seq and ST technologies, and configure them for integrated visualisation and 2) the Vitessce framework that supports visualisation of multimodal single cell and spatial datasets.

Cloud-optimised data storage: WebAtlas adopts the cloud-ready Zarr format, which provides an agnostic approach to the underlying storage system, and employs an array chunking strategy. The data is stored as a collection of text files (i.e. with metadata) along with data divided into compressed chunks which can be read individually or in parallel, allowing very large datasets to scale more efficiently in the cloud. While TissUUmaps 3 and SODB use native AnnData objects for tabular data and TissUUmaps 3 uses OME-TIFF file format for imaging data, WebAtlas converts all scRNA-seq and ST data objects to Zarr. Our data standardisation is strongly aligned with the community-defined next-generation file format (NGFF)[15] – OME-Zarr.

Web application: WebAtlas does not require users to install specialised local software, unlike the Loupe Browser (10X Genomics) and the Fiji plugin MoBIE, and is easily accessible on a web browser. This is enabled through the adoption of the cloud-ready Zarr format and the Vitessce serverless web framework in the WebAtlas pipeline.

Multiscale raster image visualisation: ST technologies generate complex tissue image data, ranging from brightfield H&E images in Visium to multi-cyclic and multi-channel fluorescent microscopy images in imaging-based ST methods, such as ISS. The diversity and scale (*i.e.* high resolution imaging of large tissue areas) of these image data types poses challenges to their online dissemination. While CellxGene and SODB platforms can serve tabular gene expression files, they do not support online browsing of full resolution tissue images. WebAtlas provides efficient multi-scale visualisation of diverse ST tissue image types via the OME-Zarr format and the Vitessce tool.

Cell and RNA segmentation visualisation: The analysis of imaging-based ST data can segment single cells and RNA molecules/spots in tissues. While SODB is limited to visualising segmented single cells as points, WebAtlas uses OME-Zarr and Vitessce for multi-scale visualisation of cell segmentation label masks that can depict complex cell shapes and cell-cell contacts at high resolution *in situ*. WebAtlas also visualises RNA molecules as points embedded in AnnData.

Limitations: We observed that while many millions of RNA molecules can be ingested into the WebAtlas Pipeline and that the WebAtlas App can visualise 1.1 million RNA molecules on the lower limb ISS dataset and 6.5 million molecules downsampled from a Xenium dataset (see examples on the WebAtlas portal), the responsiveness of the WebAtlas app is dramatically reduced when we render over 10 million molecules on Xenium and MERSCOPE datasets (not shown). Vitessce utilises a visualisation framework called deck.gl (https://deck.gl/) that leverages the WebGL 2.0 standard (https://registry.khronos.org/webgl/specs/latest/2.0/) to access GPUs on the client device. This significantly improves the performance of visual exploration of large datasets in the web browser. However, despite the GPU-accelerated rendering in Vitessce, there is still eventually a limit to the number of points that can be rendered in the web browser, and the user experience will vary depending on the specification of the client device.

To overcome this scalability limitation, it is necessary to change the architecture of how point data is visualised. Possible future scalability improvements include the rasterization of point data so that point data can be rendered efficiently as multi-scale NGFF images. Whilst offering a visual representation of the data, this solution would be at the expense of the user being able to interact with the molecule data through the web interface. Alternatively, multi-scale point clouds would provide a sequence of progressively more downsampled copies of the point data. Using a similar concept to the multi-scale pyramidal image formats, the app would then only load the appropriate point resolution and region that corresponded to the area of interest in the sample that the user requested to view.

Supplementary Note 2

WebAtlas data ingestion

The WebAtlas data ingestion pipeline, written in Nextflow[16], requires the user to provide a YAML file that defines input datasets. Each dataset can be composed of tabular data and/or images. Currently supported dataset types are AnnData object (*e.g.* HDF5 files of tabular gene expression from scRNA-seq and ST), Visium SpaceRanger output (up to version 1.2.0), Xenium output (up to version 1.3), MERSCOPE output (version 2022.5.26), molecules CSV/TSV file (*e.g.* RNA spots from customised ISS), and any raster images including raw microscopy and cell/spot label masks supported by bioformats2raw (https://github.com/glencoesoftware/bioformats2raw). The ingestion of different scRNA-seq and ST modalities is detailed below.

The user must specify the paths or URLs for each dataset and their corresponding types, along with visualisation options for Vitessce, including the final URL hosting the data. We provide various template YAML files for different modalities on our Github repository (see Code availability section).

The pipeline converts tabular AnnData to Zarr format via the canonical write_zarr function within the ScanPy package[17]. Raster images are converted to OME-Zarr using the bioformats2raw tool using default parameters.

The pipeline outputs data objects converted to Zarr and a View Config JSON file, which configures the WebAtlas web application (see below). To visualise datasets on the web app, the user needs to ensure that the Zarr data objects and the View Config file are accessible, which can be accomplished through local hosting or by placing the files onto cloud-based services such as AWS S3 bucket or Google cloud (see Vitessce guidance at http://vitessce.io/docs/data-hosting/).

1. scRNA-seq data

We ingest scRNA-seq datasets, including tabular gene expression data and cell type annotation tables, in the AnnData format and convert them to AnnData-Zarr.

2. Visium data

We can ingest raw Visium and Visium CytAssist datasets from the SpaceRanger output directories. We convert tabular spot by gene expression files to AnnData-Zarr and H&E raster images to OME-Zarr. To visualise Visium spots to be overlaid on the H&E images, we generate label images of Visium spots based on the spatial information included in the SpaceRanger output files defining each spot's centre coordinates and diameter. For Visium datasets that have been integrated with scRNA-seq by Cell2location, we can ingest Cell2location output AnnData objects that list the deconvolved cell type abundances per Visium spot. To visualise deconvolved cell types on Visium data, which are formatted as continuous cell abundance numbers per Visium spot, further preprocessing is required and is described in the integrated modality visualisation section below.

3. ISS data

The fluorescence images of limb ISS data[8] along and the single-cell segmentation label image were formatted as raw tiff files and the corresponding tabular data was saved as an AnnData object. We loaded tabular gene expression data and raster images as OME-Zarr. Additionally, segmented RNA spots/molecules were loaded as embeddings in the AnnData format.

4. Xenium data

We used 4 Xenium datasets provided by 10X Genomics, including a human breast tumour[18] (Xenium file format version 1.0.1) and human brain tissue including glioblastoma tumours (Xenium file format version 1.3.0) (provided in https://www.10xgenomics.com/resources/datasets/xenium-human-brain-preview-data-1-standard). The tabular cell by gene expression input files, available as 10x-Genomics-formatted HDF5 files, are ingested and converted to AnnData-Zarr using a dedicated loading function in ScanPy. Raster microscopy images are ingested to OME-Zarr. The cell segmentation masks are formatted as polygons in the Xenium file format, and are converted to label images in OME-Zarr format. Additional information provided by Xenium technology, such as cell centroid coordinates and default clustering labels, are loaded using additional scripts in the pipeline.

5. MERSCOPE data

We used a human breast cancer MERSCOPE FFPE dataset released by Vizgen (Human Immuno-oncology Data Release from https://vizgen.com/data-release-program/). The MERSCOPE data format stores metadata in separate files, necessitating the development of specialised loading functions to construct the AnnData object as well as to generate labelled images. We use the pandas package to load the multiple CSV files from the MERSCOPE output. We load the cell by gene matrix filtering out blank control barcodes which we identify by being prefixed with "Blank". We load cell metadata and along with the expression matrix we build an AnnData object. To be able to map cells to labelled images we transform the cell centroid micron coordinates included in each cell metadata with a transformation matrix provided by MERSCOPE to obtain pixel coordinates. In a similar manner, we obtain segmentation pixel coordinates from a cell boundaries HDF5 file and the micron to pixel transformation matrix. We use these segmentations to generate labelled images in tiff format, where each segmentation is assigned the corresponding cell ID. As for the raw image, we aim to obtain a single multipage tiff image that will then get converted to OME Zarr. MERSCOPE outputs each channel as a different tiff file and thus we first concatenate them into a single file through the pyvips[19] package and set all necessary OME metadata that is then used by bioformats2raw when performing the conversion to Zarr.

WebAtlas visualisation via Vitessce

The WebAtlas Data ingestion pipeline creates a Vitessce View Config JSON file to facilitate data visualisation, outlining pertinent information such as input datasets, specifications of each data type, embeddings to be represented, component layout within the app, and the transformed dataset's behaviour in Vitessce. The View Config file conforms to the guidelines outlined in the Vitessce documentation (http://vitessce.io/docs/view-config-json/).

To interactively annotate cells/spots in cell atlassing datasets, we use the Vitessce lasso tool (Fig 2G). Subsequently, we download these cell/spot annotations through Vitessce and transfer them to the associated AnnData object through the use of a native Python script within the Jupyter environment (Fig 2F).

WebAtlas visualisation of integrated modalities

To facilitate the integrated visualisation of gene expression and cell types in the scRNA-seq/ISS/Visium datasets, it is necessary to preprocess all the data.

Firstly, we manipulate the expression matrices of all data modalities to facilitate the visualisation of deconvolved cell type abundances in Visium. In the Visium data, we concatenate the cell type abundance predictions from Cell2location into the spot by gene expression matrix and identify which features are cell type predictions using a boolean column labelled "is_celltype". The genes from the original spot by gene part of the matrix are then labelled as "is_gene." This manipulation allows for the display of continuous cell type predictions generated by Cell2location, rather than showing only a single cell type prediction per Visium spot. Along with this, we expand the expression matrices of the corresponding scRNA-seq and ISS data to accommodate the “is_celltype” and "is_gene" columns and enable simultaneous searching across all modalities through the “featureList" component in Vitessce. This expansion involves translating the original categorical cell type values in the scRNA-seq and ISS label encoding into a one-hot encoding matrix. This is done by representing each label as a binary vector with a value of 1 in the corresponding category and 0s elsewhere. The axes between all the modalities are intersected, and the objects are sliced to contain only these values. We also ensure that each observation (cell/spot) of each modality has a unique identifier - as an integer - across all modalities by adding offsets to different datasets cell/spot IDs. This is done because an overlap can cause incorrect visualisation in Vitessce. The intersected AnnData objects then get written into AnnData-Zarr.

Secondly, in the View Config file, an appropriate coordination value is assigned for the type of observation for all three modalities, which may be cells or spots depending on the sequencing method used. The feature type and their corresponding values, which might be gene expression or cell type abundance, is also defined. These coordination values are used by Vitessce when the web application is rendered, and are leveraged to achieve the integrated visualisation. Users can utilise the cell type or gene lists to search across the chosen ontology and visualise the expression of selected features across all modalities, allowing visualisation of gene expression or cell type abundance across both spatial profiles and embedding spaces such as UMAP or t-SNE. This is also true when visualising hierarchical observations such as cell type or any other shared ontology. Additionally, we coordinate other properties such as the colour map range and zoom values, that can be controlled separately by dataset or in a unified manner, to emphasise or compare between modalities.

Our multimodal integration sub-pipeline first generates integrated datasets for specific gene and cell type subsets with shared coordination values. Then, it produces a Vitessce View Config file, consolidating each modality into a single dataset object for seamless interpretation by Vitessce. For each modality we define three observation-by-feature matrices. The first observation-by-feature matrix points to the concatenated genes and cell types matrix within the AnnData-Zarr. We set the "featureType" coordination value of this matrix as "combined".

This first matrix must be included so the software can access the full matrix. The second and third observation-by-feature matrices point to the same concatenated matrix but are filtered through the "featureFilterPath" option. We filter the matrices by pointing this "featureFilterPath" to the column within the AnnData object's feature axis that contains the boolean values that indicates whether that feature is a gene or a cell type. We respectively specify the "featureType" coordination value to "gene" in one observation-by-feature matrix and "celltype" in the other. These filtered matrices allow us to then set controls that load only one subset of the concatenated matrix at a time. For the datasets' image data we set the "featureType" coordination value as "combined". We then use the three "featureType" values, "combined", "gene" and "celltype", in the coordination space of the View Config. We refer to the "combined" value within the layout definition for the scatterplot and spatial components. Distinctly, we define two feature list components and refer one to the "gene" "featureType" and the other to "celltype". Thus, each list displays only the values that correspond to each feature type, and selecting a feature loads the respective column from the concatenated matrices which are visualised on the scatterplot and spatial components. The feature list components can load the complete set of features from any of the datasets as they contain the same data from the intersection step. For future visualisations that are similar, such View Config files can be used as a template to generate relevant configurations and examples of these are provided in the WebAtlas Github repo.

Supplementary Table 2: Datasets visualised on WebAtlas

Dataset and Reference	Technology	Cells/Visium spots	Image Size (pixels)	Genes	Molecules
Human Lower Limb (BIA accession number S-BIAD887[7])	scRNA-seq	125,955	N/A	26,509 12,903 after multimodal intersection	N/A
Human Lower Limb (BIA accession number S-BIAD822[7]))	ISS	65,080	26,525 x 47,831	90 26,522 after imputation 12,903 after multimodal intersection	1,164,802 (loaded on WebAtlas for the individual ISS dataset)
Human Lower Limb (BIA accession number S-BIAD887[8])	Visium*	1,279	15,040 x 26,680	13,730 12,903 after multimodal intersection	N/A
Human Breast Cancer	Visium CytAssist	4,992	19,505 x 21,571	18,085	N/A
Human Breast Cancer[18]	Xenium	167,782	35,416 x 25,779	313	43,664,540 (not loaded)
Human Brain Tumour	Xenium**	40,887	39,794 x 23,900	319	13,324,742 (not loaded)
Human Breast Cancer	MERSCOPE	710,073	110,485 x 94,805	500	490,398,542 (not loaded)
Foetal Immune[20]	scRNA-seq	911,873	N/A	33,538	N/A
SeqFISH Mouse embryo[10]	SeqFISH	19,451	N/A	351 29,452 after imputation	4,396,076 (not loaded)
SeqFISH Mouse embryo[10]	scRNA-seq	16909	N/A	29,452	N/A

Supplementary Table 2. Datasets visualised on WebAtlas in this study and their specifications.
All datasets can be publicly accessed on the WebAtlas portal.
**The lower limb study profiled multiple donors and anatomical regions across 8 Visium chips (i.e. capture areas). One Visium chip with a PCW5.5 section was used for the integrated lower limb WebAtlas. The rest of the samples can be accessed as individual datasets in our portal.*
***Two additional human brain Xenium datasets can be accessed in our portal including healthy brain and Alzheimer's disease tissue sections.*

Supplementary References

11. Pape, C. *et al.* MoBIE: a Fiji plugin for sharing and exploration of multi-modal cloud-hosted big image data. *Nat. Methods* **20**, 475–476 (2023).

12. Warchol, S. *et al.* Visinity: Visual Spatial Neighborhood Analysis for Multiplexed Tissue Imaging Data. *IEEE Trans. Vis. Comput. Graph.* **PP**, (2022).

13. Solorzano, L., Partel, G. & Wählby, C. TissUUmaps: interactive visualization of large-scale spatial gene expression and tissue morphology data. *Bioinformatics* **36**, 4363–4365 (2020).

14. Yuan, Z. *et al.* SODB facilitates comprehensive exploration of spatial omics data. *Nat. Methods* **20**, 387–399 (2023).

15. Moore, J. *et al.* OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. *Nat. Methods* **18**, 1496–1498 (2021).

16. Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316–319 (2017).

17. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).

18. Janesick, A. *et al.* High resolution mapping of the breast cancer tumor microenvironment using integrated single cell, spatial and in situ analysis of FFPE tissue. *bioRxiv* 2022.10.06.510405 (2022) doi:10.1101/2022.10.06.510405.

19. Cupitt, J. GitHub - libvips/pyvips: python binding for libvips using cffi. *GitHub* https://github.com/libvips/pyvips (2022).

20. Suo, C. *et al.* Mapping the developing human immune system across organs. *Science* **376**, eabo0510 (2022).