# anndataR improves interoperability between R and Python in single-cell transcriptomics

Louise Deconinck *[1,2], Luke Zappia *[3], Robrecht Cannoodt * †[1,2,3], Martin Morgan[4], scverse core[8], Isaac Virshup[5,6], Chananchida Sang-Aram[1,2], Danila Bredikhin[7,8], Brian Schilder[9], Ruth Seurinck[1,2], and Yvan Saeys[1,2]

[1]Data Mining and Modelling for Biomedicine, VIB Center for Inflammation Research
[2]Department of Mathematics, Computer Science and Statistics, Ghent University, Belgium
[3]Data Intuitive BV
[4]Department of Biostatistics and Bioinformatics, Roswell Park Comprehensive Cancer Center
[5]Institute of Computational Biology, Helmholtz, Center Munich, Munich, Germany
[6]Chan Zuckerberg Initiative, Redwood City, CA, USA
[7]Department of Genetics, Stanford University, Stanford, CA, USA
[8]scverse
[9]Department of Brain Sciences, Faculty of Medicine, Imperial College London, London, UK

August 7, 2025

## Abstract

**Summary:** Many single-cell transcriptomics datasets are stored in the HDF5-backed AnnData (H5AD) file format, as popularised by the Python scverse ecosystem. However, accessing these datasets from R, allowing users to take advantage of the strengths of each language, can be difficult. anndataR facilitates this access by allowing users to natively read and write H5AD files in R, convert them to and from SingleCellExperiment or Seurat objects, or even work with the resulting R AnnData object directly. We perform rigorous testing to ensure compatibility between Python-written and R-written H5AD files, guaranteeing long-term interoperability between languages.
**Availability:** anndataR's source code is available on GitHub at scverse/anndataR under the MIT license. It is compatible with R version 4.5, has been archived at 10.5281/zenodo.15689523 and is submitted to Bioconductor. Installation instructions and tutorials can be found in the online documentation. Issues can be reported at the GitHub repository.
**Contact:** robrecht@data-intuitive.com

## 1 Introduction

In the single-cell transcriptomics field, three main analysis ecosystems exist: scverse [1], Seurat [2] and Bioconductor [3], each defining its own in-memory data format. Two of these, the SingleCellExperiment object used by Bioconductor and the Seurat object, are implemented in R, while the scverse AnnData object [4] is Python based.

Each ecosystem has specific advantages for single-cell transcriptomics (scRNA-seq) data analysis and best practices and benchmarks recommend combining

*These authors contributed equally to this work.
†Corresponding author. robrecht@data-intuitive.com

tools from different ecosystems to obtain an optimal processing pipeline for scRNA-seq data [5]. For instance, Seurat easily accommodates multi-modal assays, while Bioconductor comes with easy access to extensive statistical tooling and scverse leverages scalability and access to machine learning ecosystems. On top of that, small implementation differences in basic functionality (e.g. PCA) between the ecosystems may produce significantly different results [6]. As a result, users may need to switch between data formats or even programming languages to perform different analysis steps or to replicate existing analyses. Unfortunately, this is not straightforward and a number of issues arise when trying to convert data between formats. These issues stem from a) structural differences between the data formats and b) the different programming languages.

## 1.1 Differences in structure

All three data formats were developed to work with single-cell transcriptomics data but they take different approaches and structure their objects differently. An overview of the structure of these objects can be found in Table 1.

For example, the AnnData object contains a `varm` slot which stores multidimensional variable annotations (such as PCA loadings) but SingleCellExperiment and Seurat instead store this information as metadata of the cell reduction objects. The Seurat object also has no corresponding slot for the AnnData `varp` slot, which is used for pairwise variable level annotations. Seurat and SingleCellExperiment can also store multimodal data, which is not possible in AnnData. Additionally, the frameworks differ in matrix orientation conventions. AnnData uses an observations by features layout, with observations (cells) in rows and features (genes) in columns. Conversely, SingleCellExperiment and Seurat implement these matrices with cells as columns and genes as rows. Because of these structural differences, in-depth knowledge of each object and the related storage conventions is required to correctly convert between formats, which presents a barrier to users moving between ecosystems.

| **Stored information** | **AnnData** | **SCE** | **Seurat v5** |
|---|---|---|---|
| Expression matrices | X, layers | assays | layers |
| Unfiltered expression matrices (different features) | raw | altExp | layers |
| Paired measurements (spike in, CITE-seq, …) | | altExp | assays |
| Multi-omics data (different cells) | | | assays |
| Cell-level metadata (cell type, …) | obs | colData | cell level metadata |
| Feature-level metadata (number of cells, …) | var | rowData | assay level metadata |
| Multidimensional cell-level metadata (dimensionality reductions) | obsm | reducedDims | reductions |
| Multidimensional feature-level metadata (loadings of dimensionality reductions) | varm | reducedDims | reductions |
| Cell by cell similarity matrices | obsp | colPairs | graphs |
| Feature by feature similarity matrices | varp | rowPairs | |
| Other unstructured metadata | uns | metadata | misc |

Table 1: Overview of the three different data formats and the slots where specific information is stored.

## 1.2 Different programming languages

The different programming languages also complicate interoperability. Seurat and Bioconductor are written in R while scverse is written in Python. As mentioned earlier, tools in both programming languages are required to optimally analyse a single dataset. To address this, two other core data structures within the scverse ecosystem have some cross-language in-

[1] Incomplete: only reads and converts X, obs, var, obsm
[2] Combines conversion functionality from other packages

| Tool | Prog. language | FFI | File format | Conversion |
|---|---|---|---|---|
| dynverse/ anndata [7] | R | ✓ | | |
| anndata2ri [7] | Python | ✓ | | AnnData↔SCE |
| Loom [8, 9] | R, Python | | Loom (h5) | |
| scDIOR [10] | R, Python | | scDIOR (h5) | AnnData↔SCE AnnData↔Seurat |
| BiocPy with rds2py [11] | Python | | | |
| alabaster dolomite [12, 13] | R, Python | | HDF5, csv | |
| SeuratDisk [14] | R | | h5Seurat | Seurat↔AnnData |
| zellkonverter [15] | R | ✓ | | AnnData↔SCE |
| sceasy [16] | R | ✓ | | Converts between AnnData, SCE, Seurat and Loom |
| schard [17] | R | | | Converts between AnnData, SCE and Seurat[1] |
| anndataR | R | | | Converts between AnnData, SCE and Seurat |
| SCUBA [18] | R | ✓ | provides API | |
| scKirby [19] | R | ✓ | | Converts between AnnData, SCE, Seurat and Loom[2] |

Table 2: Overview of packages aiming to provide some form of interoperability between AnnData, SingleCellExperiment and Seurat objects

teroperability: MuData [20], designed for multimodal data, provides R and Julia implementations, and SpatialData [21] maintains active collaboration with R developers [22]. For single-cell transcriptomics analysis, there are many software packages that facilitate the conversion between different objects (Table 2). They use a combination of approaches to transfer data between programming languages: they either use a so-called foreign function interface (FFI) to transfer information between languages during an interactive session or they write and read the data to and from disk.

FFIs like reticulate [23] or rpy2 [24] allow a program written in one language to call or use functions written in another language. This allows users to flexibly call foreign functions, transferring data from the native language to the foreign language as needed. However, such an approach faces several limitations. It is limited to built-in datatypes (such as vectors, lists or integers), requires managing both a Python and R environment, and naive use of these interfaces can lead to excessive memory usage. Software packages that use FFI to facilitate interaction with single-cell specific data formats include anndata2ri [7] and dynverse/anndata [25].

The second approach, working on an object that gets intermediately stored on disk, requires a standard on-disk format accessible from both languages. A shared standard, loom, was introduced but was not fully taken up by the community. The AnnData HDF5-based [26] H5AD format has become the de facto standard for Python users and in some public repositories while R objects are commonly saved as non-interoperable binary Rds files. To address this, packages such as rds2py [11], alabaster [12] and dolomite [13] provide ways to read and write SingleCellExperiment objects in Python. SeuratDisk [14] provides a way to save Seurat objects as interoperable HDF5 files.

Other packages, including sceasy [16] and zellkonverter [15], combines these approaches to make H5AD files useable in R. They allow reading and writing of H5AD files to and from Seurat or SingleCellExperiment object but do so by using a foreign function interface to read the H5AD file before converting formats. This approach only partially alleviates the burden of working with these FFIs. R users (or package maintainers) still need to maintain a Python envi-

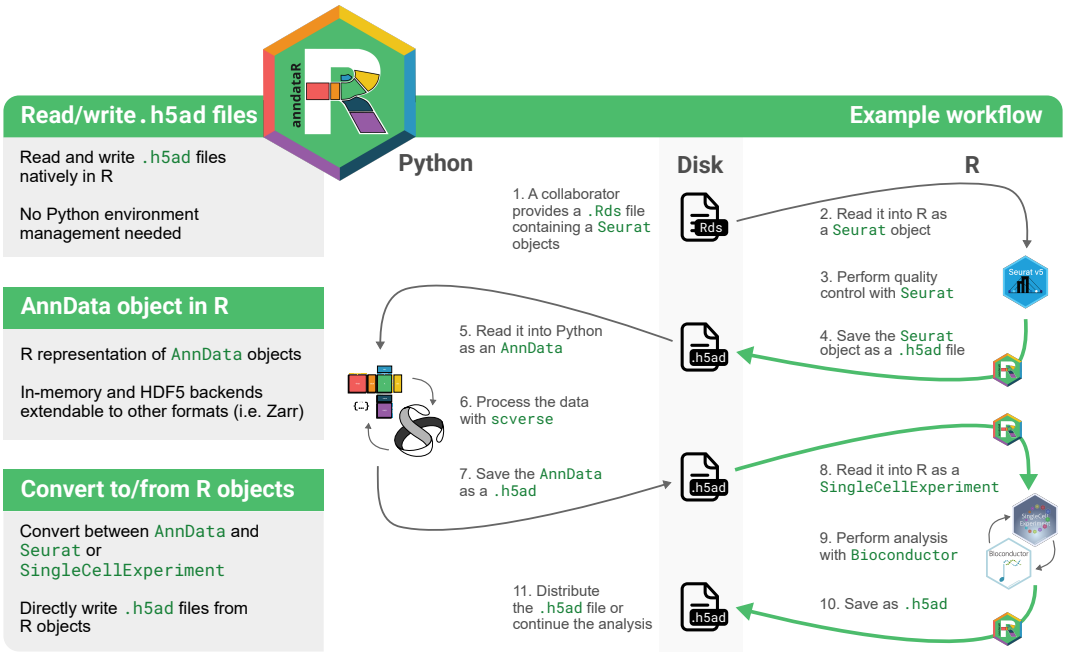


Figure 1: Feature overview and showcase of how to use anndataR in an analysis workflow.

ronment and be mindful of in-memory duplication of data, even when all their analysis is performed in R. The data type conversions from Python to R can also be more cumbersome than those needed when reading in the HDF5 file from disk.

## 2 Results

To overcome these limitations we introduce anndataR, a new package which allows native reading and writing of H5AD files in R, without requiring a Python environment, provides an R AnnData object and facilitates conversion to and from other data formats. anndataR was conceived and developed as a collaborative effort between members of both the scverse and Bioconductor communities, who together identified the gaps in the current interoperability climate.

anndataR is unique in its ability to read and write H5AD files natively in R and convert them to SingleCellExperiment or Seurat objects. By avoiding FFIs, anndataR avoids many of the associated challenges. The conversion functionality it offers bridges the gap between the ecosystems, with sensible defaults and fine-grained control for advanced use. anndataR allows users direct access to an R AnnData representation, making it possible to interact with the object directly, in order to solve conversion issues and to facilitate easy extraction of relevant parts of the data.

## 3 Software Design

Users primarily interact with anndataR by reading in an existing H5AD file and specifying whether to return an AnnData object in R (either in memory or HDF5-backed), or immediately perform a conversion to a SingleCellExperiment or Seurat object, as shown in Supplementary Figure 1. The resulting R object can then be used as normal and can be saved to a new H5AD file on completion of the analysis. An example of how anndataR can be used in an analysis workflow is shown in Figure 1.

To enable this workflow, we use rhdf5 [27], an R

interface to the HDF5 library which handles the low-level interactions with HDF5 files to natively read and write the H5AD format according to the AnnData on-disk specification. To take advantage of existing R analysis packages users also need to be able to convert between AnnData objects and SingleCellExperiment or Seurat objects. By default the anndataR conversion functions try to convert as much data as possible, but also allow the experienced user to completely specify the details of the conversion by providing a mapping for each slot in the resulting object. Full details of the conversion can be found in the package documentation.

We are aware that small changes in any of these data formats can quickly lead to failures in reading, writing or converting. To ensure the package functionality is robust against data format changes, we perform extensive testing. H5AD files written with anndataR are validated against the ones produced by the Python anndata package. This is necessary because there are differences in how Python and R handle data, such as row-major versus column-major matrices, and in how the Python and R HDF5 libraries translate data to H5 datatypes. We perform these checks using a number of round-trip tests for each AnnData slot, checking whether an R-written H5AD slot can be read in Python and vice versa. We also check for differences in the generated H5AD files using the `h5diff` utility. Additionally, we verify the conversions to and from SingleCellExperiment and Seurat by slot-by-slot verification of the objects. This approach to exhaustively testing functionality has been key for the development of anndataR, allowing us to document known limitations and incompatibilities.

Finally, in order to represent the complex AnnData structure and replicate the Python interface, make use of inheritance, keep memory usage low and use reference semantics, we chose to use the R6 object oriented class system.

# 4 Conclusion

anndataR meaningfully adds to the interoperability landscape between R and Python for single-cell transcriptomics by allowing users to work natively with H5AD files in R and providing conversion functionality to and from SingleCellExperiment and Seurat objects. This conversion functionality not only provides reasonable defaults but also allows fine-grained user control. Additionally, the package functionality, such as compatibility between R-written and Python-written H5AD files, is rigorously tested.

Finally, the modular design of the software makes it easy to extend the functionality of the package, enabling future support for additional file formats (e.g., Zarr) and other modalities, such as scATAC-seq or CITE-seq (via MuData [20]) and spatial data (via the SpatialData [21] framework).

# 5 Acknowledgments

This work was supported by the Research Foundation - Flanders (FWO) [1SF3822N to L.D.]; Chan Zuckerberg Initiative Essential Open Source Software for Science grant [EOSS6-0000000743 to L.Z. and R.C.]; Chan Zuckerberg Initiative Foundation (CZIF) - CZI Seed Networks [CZIF2019-002443 to M.M.]; Ghent University Special Research Fund [BOF21-DOC-105 to C.S.] and the Flanders AI Research (FAIR) Program [174B09119 to R.S. and Y.S.];

We acknowledge the contribution of the scverse core. Members are: Philipp Angerer, Danila Bredikhin, Can Ergen-Behr, Emma Dann, Severin Dicks, Jennifer Foltz, Ilan Gold, Lukas Heumos, Mikaela Koutrouli, Ori Kronfeld, Luca Marconato, Giovanni Palla, Anna Schaar, Roshan Sharma, Gregor Sturm, Tim Treis, Wouter-Michiel Vierdag, Isaac Virshup, Kai Zhang.

# References

[1] Isaac Virshup, Danila Bredikhin, Lukas Heumos, Giovanni Palla, Gregor Sturm, Adam Gayoso, Ilia Kats, Mikaela Koutrouli, Scverse Community, Bonnie Berger, Dana Pe'er, Aviv Regev, Sarah A Teichmann, Francesca Finotello, F Alexander Wolf, Nir Yosef, Oliver Stegle, and Fabian J Theis. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nat Biotechnol*, 41(5):604–606, May 2023.

[2] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol*, 33(5):495–502, May 2015.

[3] Robert A Amezquita, Aaron T L Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, Levi Waldron, Hervé Pagès, Mike L Smith, Wolfgang Huber, Martin Morgan, Raphael Gottardo, and Stephanie C Hicks. Orchestrating single-cell analysis with bioconductor. *Nat Methods*, 17(2):137–145, February 2020.

[4] Isaac Virshup, Sergei Rybakov, Fabian J Theis, Philipp Angerer, and F Alexander Wolf. anndata: Access and store annotated data matrices. *J. Open Source Softw.*, 9(101):4371, September 2024.

[5] Lukas Heumos, Anna C Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D Lücken, Daniel C Strobl, Juan Henao, Fabiola Curion, Single-cell Best Practices Consortium, Herbert B Schiller, and Fabian J Theis. Best practices for single-cell analysis across modalities. *Nat Rev Genet*, 24(8):550–572, August 2023.

[6] Joseph M Rich, Lambda Moses, Pétur Helgi Einarsson, Kayla Jackson, Laura Luebbert, A Sina Booeshaghi, Sindri Antonsson, Delaney K Sullivan, Nicolas Bray, Páll Melsted, et al. The impact of package selection and versioning on single-cell rna-seq analysis. *bioRxiv*, 2024.

[7] Philipp A., Gökçen Eraslan, Isaac Virshup, and Scott Gigante. *theislab/anndata2ri v1.0.4*, August 2020.

[8] Linnarsson Lab. loompy. `https://github.com/linnarsson-lab/loompy`. Accessed: 2025-6-5.

[9] Paul Hoffman and Rahul Satija. *loomR: An R interface for loom files*, 2018. R package version 0.2.0.

[10] Huijian Feng, Lihui Lin, and Jiekai Chen. scDIOR: single cell RNA-seq data IO software. *BMC Bioinformatics*, 23(1):16, January 2022.

[11] Jayaram Kancherla. *rds2py*. Accessed: 2025-6-5.

[12] Aaron Lun. *alabaster.base: Save Bioconductor Objects To File*, 2024.

[13] Aaron Lun and Jayaram Kancherla Kancherla. dolomite-base. `https://github.com/ArtifactDB/dolomite-base`.

[14] Paul Hoffman. *SeuratDisk: Interfaces for HDF5-Based Single Cell File Formats*, 2023. R package version 0.0.0.9021.

[15] Luke Zappia and Aaron Lun. *zellkonverter: Conversion Between scRNA-seq Objects*, 2024. R package version 1.16.0.

[16] Batuhan Cakir, Martin Prete, Ni Huang, Stijn van Dongen, Pinar Pir, and Vladimir Yu Kiselev. Comparison of visualization tools for single-cell RNAseq data. *NAR Genom Bioinform*, 2(3):lqaa052, September 2020.

[17] Pavel Mazin. *schard: package to load h5ad into R*, 2024.

[18] William M Showers, Jairav Desai, Krysta L Engel, Clayton Smith, Craig T Jordan, and Austin E Gillen. SCUBA implements a storage format-agnostic API for single-cell data access in R. *F1000Res.*, 13:1256, June 2025.

[19] Brian Schilder. sckirby. `https://github.com/neurogenomics/scKirby`, 2024.

[20] Danila Bredikhin, Ilia Kats, and Oliver Stegle. MUON: multimodal omics analysis framework. *Genome Biol*, 23(1):42, February 2022.

[21] Luca Marconato, Giovanni Palla, Kevin A Yamauchi, Isaac Virshup, Elyas Heidari, Tim Treis, Wouter-Michiel Vierdag, Marcella Toth, Sonja Stockhaus, Rahul B Shrestha, et al. Spatialdata: an open and universal data framework for spatial omics. *Nature Methods*, 22(1):58–62, 2025.

[22] Luca Marconato, Wouter-Michiel Vierdag, Kevin Yamauchi, Clarence Mah, Helena Crowell, Quentin Blampey, Benjamin Rombaut, Mark S Keller, Lotte Pollaris, Estella Y Dong, and et al. 1st spatialdata developer workshop, Dec 2024.

[23] Kevin Ushey, JJ Allaire, and Yuan Tang. *reticulate: Interface to 'Python'*, 2025.

[24] rpy2: Python-R bridge. `https://rpy2.github.io/`, 2025.

[25] Robrecht Cannoodt. *anndata: 'anndata' for R*, 2023.

[26] The HDF Group. *Hierarchical Data Format, version 5*.

[27] Bernd Fischer, Mike Smith, and Gregoire Pau. *rhdf5: R Interface to HDF5*, 2025. R package version 2.50.2.