Bredikhin et al. Genome Biology (2022) 23:42
https://doi.org/10.1186/s13059-021-02577-8

Genome Biology

SOFTWARE Open Access

MUON: multimodal omics analysis framework

Danila Bredikhin[1,3,2*], Ilia Kats[3] and Oliver Stegle[1,3,4,5*]

* Correspondence: danila.bredikhin@embl.de; o.stegle@dkfz-heidelberg.de
[1] European Molecular Biology Laboratory (EMBL), Genome Biology Unit, Heidelberg, Germany
Full list of author information is available at the end of the article

Abstract

Advances in multi-omics have led to an explosion of multimodal datasets to address questions from basic biology to translation. While these data provide novel opportunities for discovery, they also pose management and analysis challenges, thus motivating the development of tailored computational solutions. Here, we present a data standard and an analysis framework for multi-omics, MUON, designed to organise, analyse, visualise, and exchange multimodal data. MUON stores multimodal data in an efficient yet flexible and interoperable data structure. MUON enables a versatile range of analyses, from data preprocessing to flexible multi-omics alignment.

Background

Multi-omics designs, that is the simultaneous profiling of multiple omics or other modalities for the same sample or cells, have recently gained traction across different biological domains. Multi-omics approaches have been applied to enable new insights in basic biology and translational research [1, 2].

On the one hand, the emerging multi-omics datasets result in novel opportunities for advanced analysis and biological discovery [3]. Critically, however, multi-omics experiments and assays pose considerable computational challenges, both concerning the management and processing as well as the integration of such data [4, 5]. Major challenges include efficient storage, indexing and seamless access of high-volume datasets from disk, the ability to keep track and link biological and technical metadata, and dealing with the dependencies between omics layers or individual features. Additionally, multi-omics datasets need to be converted into specific file formats to satisfy input requirements for different analysis and visualisation tools.

While specialised frameworks for the analysis of different omics data types have been proposed, including for bulk and single-cell RNA-seq [6–9] or epigenetic variation data [10–13], there is a lack of comprehensive solutions that specifically address multi-omics designs. Additionally, there currently exists no open exchange format for sharing multi-omics datasets that is accessible from different

programming languages. The currently existing solutions for multi-omics data (Seurat [8], MultiAssayExperiment [14]) are confined to the R programming language ecosystem, and typically require loading the full dataset, a limitation that prohibits dealing with larger datasets and can only be partially mitigated by using additional third-party software [15, 16].

To address this, we here present MUON (*mu*ltimodal *o*mics a*n*alysis), an analysis framework that is designed from the ground-up to organise, analyse, visualise, and exchange multimodal data. MUON is implemented in Python and comes with an extensive toolbox to flexibly construct, manipulate and analyse multi-omics datasets. At the core of the framework is MuData, an open data structure standard, which is compatible with and extends previous data formats for single omics [9, 17]. MuData files can be seamlessly accessed from different programming languages, including Python [18], R [19], and Julia [20]. We illustrate MUON in the context of different vignettes of its application with a major focus on single-cell data, including analysis of combined gene expression and chromatin accessibility assays as well as gene expression and epitope profiling.

Results

MuData: a cross-platform multimodal omics data container

At the core of MUON is MuData (*mu*ltimodal *data*)—an open data structure for multimodal datasets. MuData handles multimodal datasets as containers of unimodal data. This hierarchical data model generalises existing matrix-based data formats for single omics, whereby data from each individual omics layer are stored as an AnnData [17] object (Fig. 1a, c). MuData also provides a coherent structure for storing associated metadata and other side information, both at the level of samples (e.g. cells or individuals) and features (e.g. genes or genomics locations).

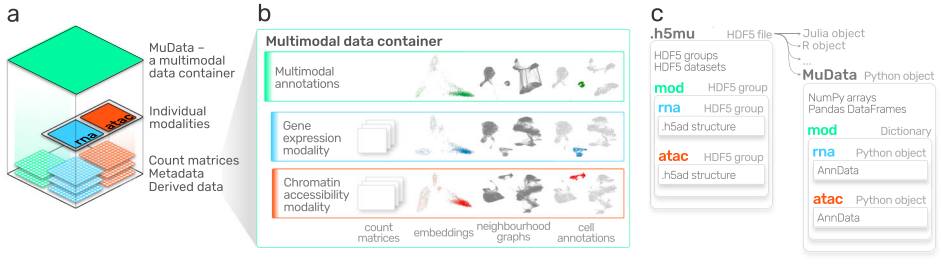

Fig. 1 Architecture and content of a multimodal data container (MuData). **a** Schematic representation of the hierarchical structure of a MuData container. Raw data matrices from multiple modalities together with associated metadata are encapsulated in an array structure. For illustration, blue and red denote RNA-seq and ATAC-seq data modalities; green denotes multimodal annotation or derived data. **b** Example content of the structure in **a**. Shown are example content of a MuData container, consisting of count matrices, embeddings, neighbourhood graphs and cell annotations for individual modalities (blue, red), as well as derived data from multi-omics analyses (green). **c** Schematic representation of MUON storage model and its serialisation scheme using the HDF5 file format on disk. Left: Hierarchy of the storage model, with plates denoting different levels of hierarchy. Arrows signify access schemes of the HDF5 file using various programming languages. Right: Representation of the MuData object in Python, with metadata and derived annotations represented as NumPy arrays or Pandas DataFrames, and with individual modalities as AnnData objects

Metadata tables can either be specific to a single stored data modality, or they can represent joint sample annotations that apply to all modalities stored in a MuData container. In a similar manner, MuData containers can be used to store derived data and analysis outputs, such as cluster labels or an inferred sample embedding (Fig. 1b).

MuData objects are serialised to HDF5 [21] files by default—the industry standard for storing hierarchical data. Individual omics layers are serialised using the existing AnnData serialisation format, thus permitting direct access to single omics using existing toolchains that build on this data standard (Fig. 1c). Basic access to MuData files is possible from all major programming environments that support access to HDF5 array objects. Additionally, MUON comes with dedicated libraries to create, read and write MuData files from Python, R, and Julia. These tools facilitate the exchange of multi-omics data across platforms and ensure consistent file format definitions.

MUON: a framework for multimodal omics data

The MUON framework allows for managing, processing, and visualising multi-omics data using the MuData containers. Existing workflows developed for single-omics can be reused and applied to the contents of a multi-omics container. For example, individual modalities of the simultaneous gene expression and chromatin accessibility profiling [22] can be processed using existing RNA [23] and ATAC [24] workflows. In this manner, canonical processing steps, including quality control, sample filtering, data normalisation and the selection of features for analysis can be transferred from single-omics analysis (Fig. 2a).

The integration of multiple modalities within a MuData container facilitates the definition of multi-omics analysis workflows, allowing to flexibly combine alternative processing steps (from left to right in Fig. 2b). For example, single-omics dimensionality reduction methods such as principal component analysis or factor analysis [25–28] can be used to separately process RNA-seq and ATAC-seq count matrices. Additionally, MUON comes with interfaces to multi-omics analysis methods that jointly process multiple modalities, including multi-omics factor analysis [29, 30] (MOFA) to obtain lower-dimensional representations, and weighted nearest neighbours [31] (WNN) to calculate multimodal neighbours. Once the results from either dimensionality reduction strategy are stored in a MUON container, they can be used as input for defining cell neighbour graphs. This graph can be either estimated from individual omics modalities, from a multi-omics representation (e.g. as obtained from MOFA), or by fusing two single-omics neighbour representations (e.g. using methods such as similarity network fusion, SNF [32], or WNN [31]).

Finally, the latent or neighbourhood representations can serve as a starting point for downstream analysis and interpretation. For example, uniform manifold approximation and projection (UMAP) [33] can be directly applied to cell neighbourhood graphs to generate nonlinear embeddings of cells. Similarly, the alternative cell neighbourhood graphs can be used as input for identifying connected components and thereby putative cell types (e.g. using multiplex community detection techniques [34]).

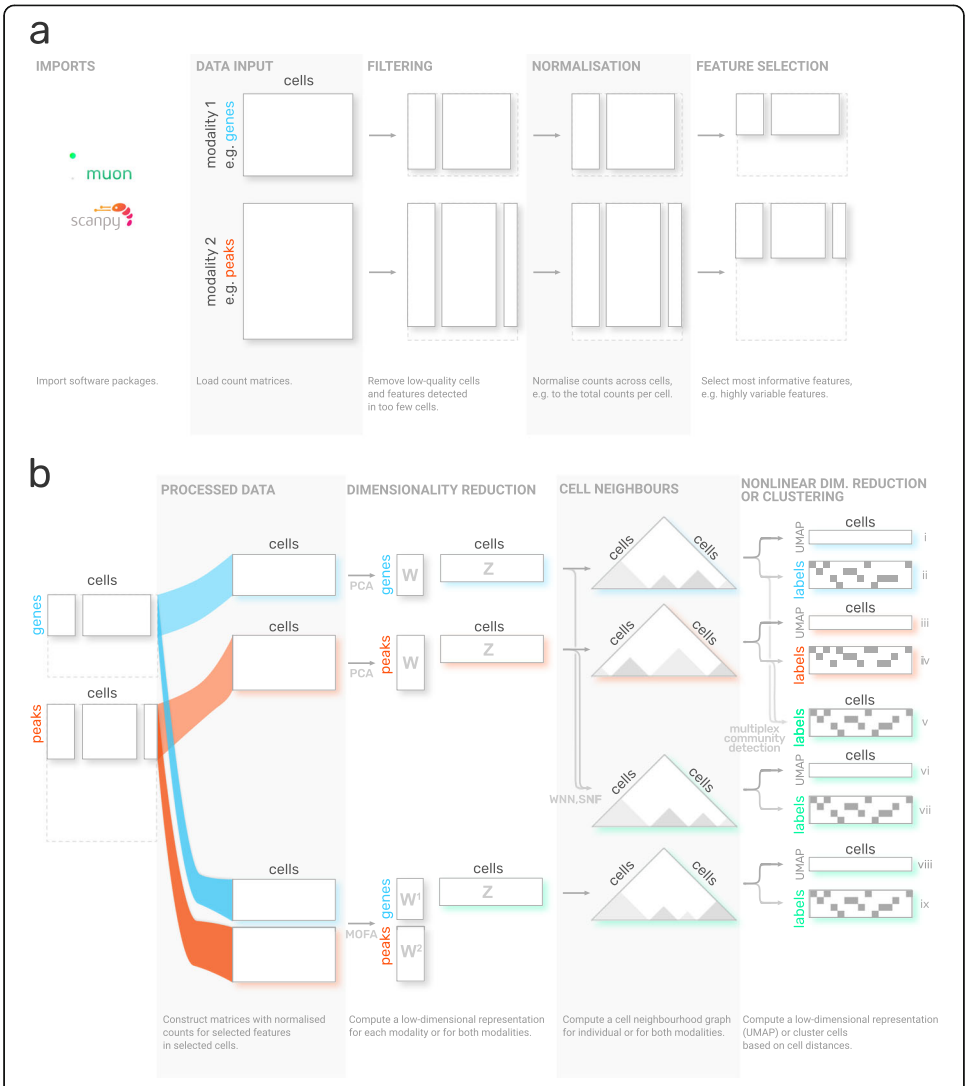

Fig. 2 Example multi-omics analysis workflows implemented using MUON. **a** Construction and processing of individual modalities of a multi-omics scRNA-seq and scATAC-seq dataset. Processing steps for individual omics from left to right. Rectangles denote count matrices following each processing step, which are stored in a shared MUON data container. MUON provides processing functionalities for a wide range of single-omics, including RNA-seq, ATAC-seq, CITE-seq. Existing workflows and methods can be utilised, including those implemented in scanpy. Respective analysis steps are described below each step. **b** Alternative workflows for integrating multiple omics for latent space inference and clustering. MUON enables combining alternative analysis steps to define tailored multi-omics data integrations. Shown are canonical workflows from left to right: dimensionality reduction, definition of cell neighbourhood graphs, followed by either nonlinear estimation of cell embeddings or clustering. Letters W and Z denote matrices with feature weights (loadings) and factors (components), respectively. Triangles represent cell-cell distance matrices, with shading corresponding to cell similarity. Green colour signifies steps that combine information from multiple modalities; steps based on individual modalities only are marked with blue (RNA) or red (ATAC) respectively. The outputs of the respective workflows (right) are from top to bottom: UMAP space (i) and cell labels (ii) based on RNA or alternatively based on ATAC modality (iii, iv), cell labels based on two cell neighbour graphs from individual modalities (v), UMAP space and cell labels based on WNN output (vi, vii), UMAP space and cell labels based on MOFA output (viii, ix)

The flexibility to choose and control individual processing steps in MUON makes it possible to compose tailored workflows for a particular dataset.

Application of MUON to single-cell multi-omics data

To illustrate MUON, we considered data from simultaneous scRNA-seq and scATAC-seq profiling of peripheral blood mononuclear cells (PBMCs), which were generated using the Chromium Single Cell Multiome ATAC + Gene Expression protocol by 10x Genomics [22]. Features in the RNA modality correspond to the expression level of genes, whereas the ATAC modality encodes accessible genomic loci as peaks. MUON supports the application of alternative dimensionality reduction strategies (Fig. 2). For example, multi-omics factors analysis [30]—an approach for integrating different omics modalities based on matrix factorization—yields a lower dimensional representation, including factors that capture variation of individual omics or shared variability (Additional file 1: Fig S1a), which in turn can be interpreted on the level of individual features (Fig. 3a, Additional file 1:Fig S1b). Here, the factors that explain the largest fraction of variance in PBMCs capture canonical biological differences, such as the myeloid—lymphoid axis and cytotoxicity (Fig. 3a, left). These factors capture both variation in mRNA abundance and chromatin accessibility, e.g. as CD3E expression and BCL11B promoter accessibility, which are characteristic for T cells [35, 36] (Fig. 3a, right).

A two-dimensional latent space recapitulating the structure of the data is commonly used for visualising cell type composition, cell-level covariates, or feature counts. For this, it is important that MUON allows to generate, store, and operate with multiple different embeddings constructed for individual modalities (Fig. 3b, left) or jointly for both modalities based, for instance, on the MOFA factors or the WNN graph (Fig. 3b, right). Such visualisations can be generated from MuData objects without loading all the data into memory.

As a second example, we considered CITE-seq [37] data, which comprise gene expression and epitope abundance information in the same cells. To process the latter, specialised normalisation strategies for denoising and scaling [38] are available. Normalised protein counts can then be used to define cell types, akin to gating in flow cytometry [39] (Fig. 3c). Once the count matrices are processed, these can be integrated using alternative multimodal options (Fig. 2). For instance, using both modalities for cell-type annotation as well as for dimensionality reduction allows to attribute the distinction between naïve and memory T cells to the abundance of CD45 isoforms RA and RO at the protein level (Fig. 3d).

Discussion

Multimodal omics designs are increasingly accessible, allowing for characterising and integrating different dimensions of cellular variation, including gene expression, DNA methylation, chromatin accessibility, and protein abundance [3, 40, 41]. MUON directly addresses the computational needs posed by such multi-omics designs, including data processing, analysis, interpretation, and sharing (Fig. 1). Designed for the Python ecosystem, MUON operates on MuData objects that build on community standards for

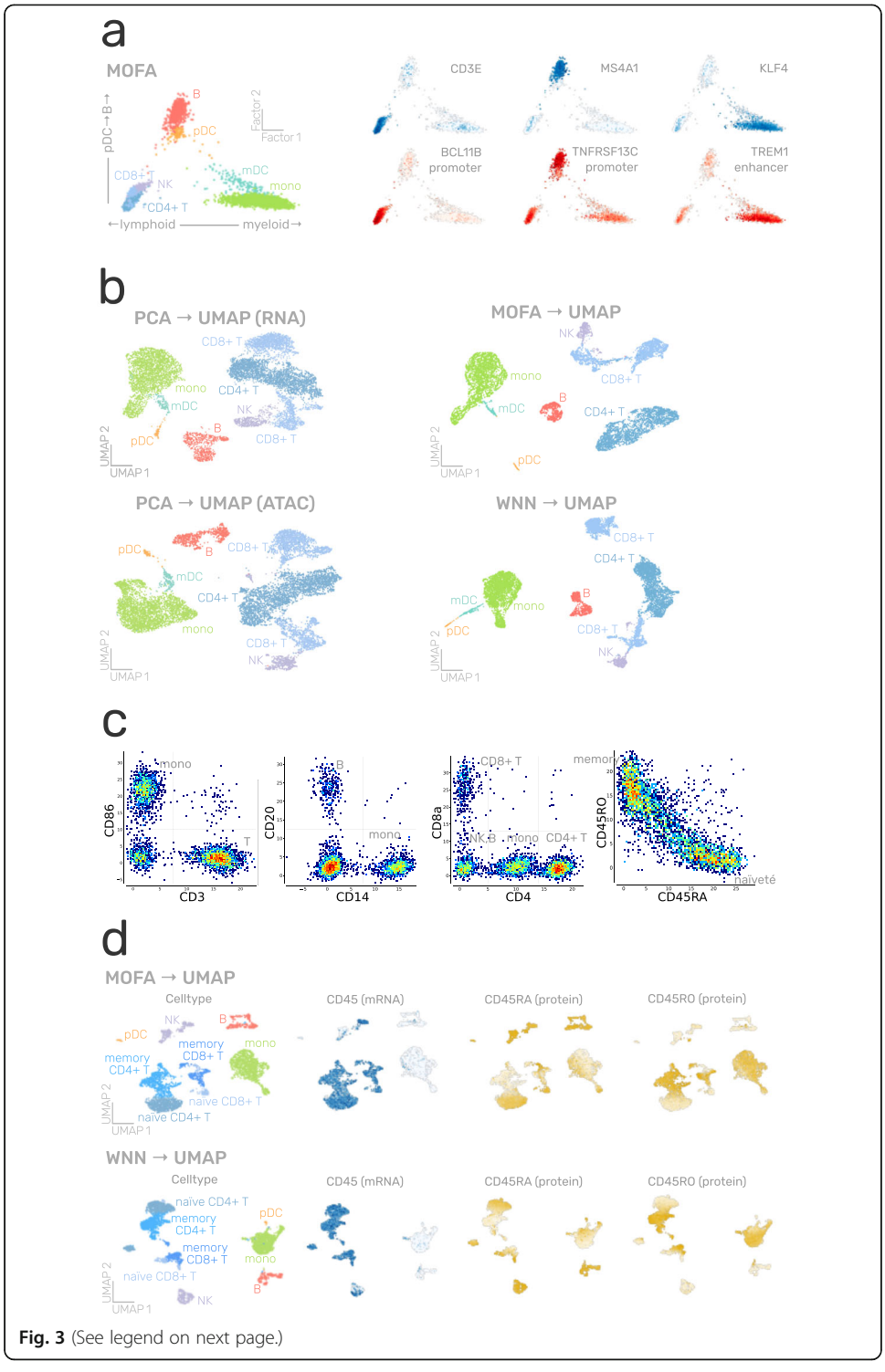

Fig. 3 (See legend on next page.)

(See figure on previous page.)
Fig. 3 Single-cell multi-omics datasets processed and visualised using MUON. **a** MOFA factors estimated from simultaneous scRNA-seq and scATAC-seq profiling of PBMCs, with cells coloured by either left: coarse-grained cell type; or right: gene expression (in blue) and peak accessibility (in red). Displayed genes and peaks are selected to represent cell-type-specific variability along factor axes. **b** UMAP latent space for the same dataset as in **a**, constructed from left: principal components for individual modalities; or right: MOFA factors and WNN cell neighbourhood graph. Cells are coloured by coarse-grained cell type. **c.** Examples of individual feature values of protein abundance in the CITE-seq profiling of PBMCs after applying dsb normalisation. Colours correspond to the relative local density of cells with red for high density and blue for low density. **d** UMAP latent space for the same dataset as in **c**, constructed from MOFA factors (top) or WNN cell neighbourhood graph (bottom). Cells are coloured by their coarse-grained cell type or feature values (blue for gene expression, yellow for protein abundance

single-omics analysis [9]. Serialisation to HDF5 makes MuData objects accessible to other programming languages, including R and Julia.

MUON is designed in a modular fashion, which means that existing methods and tools for processing individual omics can be reused to design more complex analysis workflows (Figs. 2 and 3). At the same time, the software facilitates combining single-omics analysis methods with a growing spectrum of multi-omics integration strategies [42, 43] to define novel multi-omics workflows.

Looking ahead, MUON will be a robust platform to build upon and support future developments. On the one hand, handling novel assays for multi-omics that are emerging can be integrated. For example, mRNA and proteins can be assayed together not only with CITE-seq [37] but also with QuRIE-seq [44] or INs-seq [45]. Other examples include explicit support for genomic-coordinate based assays [46] or assays with spatial coordinates [47]. Moreover, trimodal assays such as scNMT-seq [48] or TEA-seq [49] allow to generate data beyond just two modalities and can be handled with MUON, which is designed to manage an arbitrary number of modalities. On the other hand, the complexity of experimental designs is rapidly increasing [50, 51]. Already, MUON can take additional covariates into account during multimodal integration, for example, to perform temporally aware factor analysis [52]. Future development of MUON will include incorporating additional relationships in MuData, for example, to explicitly model the dependencies between feature sets across omics, or to account for dependencies between multiple sets of multi-omics experiments.

Conclusions

With MuData proposing a standardised and language-agnostic approach to manage, store, and share multimodal omics data, it is now possible to build methods and tools that can be applied to an increasingly large number of multi-omics datasets. As a multimodal framework, MUON addresses the need for multi-omics analysis workflows that are well integrated into the existing Python ecosystem, in particular with tools for omics analysis such as Scanpy [9]. At the same time, MuData facilitates the compatibility and data exchange with R and Julia.

Methods

Implementation of MuData

The reference MuData implementation is written in the Python programming language and builds on AnnData [17]. A MuData object can be cast as a collection of single-omics modalities, each of which is represented as an AnnData object. Additionally, the MuData object provides basic selector operations, including access to individual modalities, subsetting of samples and/or features. When subsetting samples, these are selected in each modality as well as in multimodal annotations; features from different modalities can be used to obtain a MuData object with desired features. As with AnnData, unstructured data can be stored in a MuData object, which can be used for recording assay-specific information. Feature relations across modalities can be stored in the MuData object as a sparse multimodal graph.

MuData objects are serialised to .h5mu files, which are based on HDF5—industry standard for hierarchical storage of numerical data supported by many programming languages [21]. Individual modalities are stored in the file hierarchy in a way compliant with AnnData serialisation, enabling access to individual modalities from disk. Disk backing is implemented for MuData objects so that MuData files can be read without loading count matrices of individual modalities.

Cross-language capabilities of MuData files are demonstrated with Julia and R libraries. Julia library implements native AnnData and MuData objects whereas R libraries create MultiAssayExperiment [14] or Seurat [8] objects with information from MuData files. As .h5ad and .h5mu are not the native formats for R frameworks, standards are still to be developed for how to serialise auxiliary information stored in the R object—and, conversely, deserialize this information back from the files.

Implementation of MUON

MUON has been implemented in the Python programming language and builds on a number of existing numerical and scientific open-source libraries, in particular, NumPy [53], Scipy [54], Sklearn [55], Pandas [56], h5py [57], AnnData [17], and Scanpy [9] for omics data handling, MOFA+ [30] for multimodal data integration and matplotlib [58] and seaborn [59] for data visualisation. The weighted nearest neighbours (WNN) method has been implemented following [31] describing the original method and [49] describing its generalisation to an arbitrary number of modalities.

Comparison of MuData with alternative data formats

MuData and MUON take inspiration and build on concepts from AnnData [17] and Scanpy [9]. In fact, the software incorporates ideas and extends it in a modular fashion, similar to the existing practice in the Bioconductor community [60].

	MuData	AnnData [9, 17]	Seurat [8]	MultiAssay Experiment [14]
Main programming environment	Python	Python	R	R
Objects can contain data out of memory (on disk)	Yes	Yes	No†	Yes‡
Native serialisation accessible from multiple languages	Yes (.h5mu)	Yes (.h5ad)	No (.rds)††	No (.rds)‡‡
Native support for I/O operations	Python, Julia,	Python	R	R

Comparison of MuData with alternative data formats *(Continued)*

	MuData	AnnData [9, 17]	Seurat [8]	MultiAssay Experiment [14]
	R*			
Support for multiple modalities	Yes	No	Yes	Yes
Support for data missing in some modalities	Yes	NA	No	Yes
Support for multimodal embeddings	Yes	NA	No	No

*Deserialized to MAE or Seurat objects
†With SeuratDisk library, in-memory Seurat objects can be constructed from parts of the data stored in HDF5 files
‡Only possible with HDF5Array library for matrices stored in external HDF5 files
††With SeuratDisk library, in-memory Seurat objects can be exported to HDF5 files
‡‡Only matrices stored in external HDF5 files, exported with HDF5Array library, can be accessed

Processing gene expression and chromatin accessibility data

Single-cell multiome ATAC + gene expression demonstration data for peripheral blood mononuclear cells (PBMCs) from a healthy donor with granulocytes removed through cell sorting processed with ARC 1.0.0 pipeline were provided by 10X Genomics (https://support.10xgenomics.com/single-cell-multiome-atac-gex/datasets). Log-normalisation was used for both gene and peak counts, and respective values for highly variable features scaled and centred to zero mean and unit variance were then used as input to discussed algorithms such as PCA, as implemented in scikit-learn [55] and scanpy [9], or MOFA+ [30]. Differentially expressed genes and differentially accessible peaks were identified with respective functionality in scanpy and were used to compile gene lists for cell type identification.

The respective vignettes are available at https://muon-tutorials.readthedocs.io/en/latest/single-cell-rna-atac.

Processing CITE-seq data

CITE-seq data for PBMCs from a healthy donor were provided by 10X Genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.2/5k_pbmc_protein_v3). Log-normalisation was used for gene counts, and dsb [38] was used to denoise and scale protein counts. Respective values for highly variable features scaled and centred to zero mean and unit variance were then used as input to discussed algorithms. The respective vignettes are available at https://muon-tutorials.readthedocs.io/en/latest/cite-seq.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-021-02577-8.

Additional file 1. Figure S1
Additional file 2. Review history

Acknowledgements

We are grateful to the members of the Stegle and Theis labs for the discussions on the MuData and MUON design.

Peer review information

Barbara Cheifet was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Review history

The review history is available as Additional file 2.

Authors' contributions

DB conceived the idea of and designed the data format and the framework. DB and IK implemented the corresponding Python packages, the Julia library and the R packages. IK implemented the WNN method for the Python package and led the development of the Julia library. DB performed multimodal datasets analyses and wrote the tutorials and vignettes. DB wrote the initial draft of the manuscript. DB and OS led the writing of the manuscript. All authors read, edited, and approved the final manuscript.

Authors' information

Twitter handles: @gtcaa (Danila Bredikhin); @OliverStegle (Oliver Stegle)

Funding

D.B. is supported by the EMBL International PhD Programme and a Darwin Trust fellowship. Research in the Stegle research group was further supported by the European Commission (grant agreements 810296, 874769) and the BMBF. Open Access funding enabled and organized by Projekt DEAL.

Availability of Data and Materials

Data on simultaneous scRNA-seq & scATAC-seq profiling [22] of PBMCs from a healthy donor is available from the 10X Genomics website (10k cells with granulocytes removed through cell sorting, Cell Ranger ARC 1.0.0) [61].
CITE-seq [37] data on PBMCs from a healthy donor is available from the 10X Genomics website (5k cells with a panel of TotalSeq^{TM}-B antibodies, v3 Chemistry, Cell Ranger 3.0.2) [62].
MUON source code is available at https://github.com/scverse/muon [63] under the BSD3 license. Documentation and tutorials for MUON can be accessed at https://muon.readthedocs.io/ and at https://muon-tutorials.readthedocs.io, respectively.
MuData implementation is available in a standalone Python library at https://github.com/scverse/mudata [64] under the BSD3 license. Its documentation is accessible at https://mudata.readthdocs.io.
The source code for Julia can be accessed at [65], for the bioconductor R library—at [66], for the Seurat R library—at [67].
The version of the code used in the manuscript is deposited on Zenodo [68].

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] European Molecular Biology Laboratory (EMBL), Genome Biology Unit, Heidelberg, Germany. [2]Collaboration for joint PhD degree between EMBL and Heidelberg University, Faculty of Biosciences, Heidelberg, Germany. [3]Division of Computational Genomics and Systems Genetics, German Cancer Research Center (DKFZ), Heidelberg, Germany. [4]Faculty of Biosciences, Heidelberg University, Heidelberg, Germany. [5]Wellcome Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridgeshire CB10 1SA, UK.

Received: 14 June 2021 Accepted: 14 December 2021
Published online: 01 February 2022

References

1. Hasin Y, Seldin M, Lusis A. Multi-omics approaches to disease. Genome Biol. 2017;18(1):83. https://doi.org/10.1186/s13059-017-1215-1.
2. Zhu C, Preissl S, Ren B. Single-cell multimodal omics: the power of many. Nat Methods. 2020;17(1):11–4. https://doi.org/10.1038/s41592-019-0691-5.
3. Argelaguet R, Cuomo ASE, Stegle O. Marioni JC. Computational principles and challenges in single-cell data integration. Nat Biotechnol. 2021. https://doi.org/10.1038/s41587-021-00895-7.
4. Conesa A, Beck S. Making multi-omics data accessible to researchers. Sci Data. 2019;6(1):251. https://doi.org/10.1038/s41597-019-0258-4.
5. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016;3(1):160018. https://doi.org/10.1038/sdata.2016.18.
6. ATL L, DJ MC, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. F1000Res. 2016;2122. https://doi.org/10.12688/f1000research.9501.2.
7. DJ MC, Campbell KR, ATL L, Wills QF. scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R. Bioinformatics. 33(8):1179–86. https://doi.org/10.1093/bioinformatics/btw777.
8. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33(5):495–502. https://doi.org/10.1038/nbt.3192.
9. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1):15. https://doi.org/10.1186/s13059-017-1382-0.
10. Danese A, Richter ML, Chaichoompu K, Fischer DS, Theis FJ, Colomé-Tatché M. EpiScanpy: integrated single-cell epigenomic analysis. EpiScanpy: integrated single-cell epigenomic analysis. Nat Commun. 2021;12(1):1-8. https://doi.org/10.1038/s41467-021-25131-3.
11. Stuart T, Srivastava A, Lareau C, Satija R. Multimodal single-cell chromatin analysis with Signac. bioRxiv. 2020.11.09.373613. https://doi.org/10.1101/2020.11.09.373613.

12. Granja JM, Corces MR, Pierce SE, Bagdatli ST, Choudhry H, Chang HY, Greenleaf WJ. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. Nat Genet. 2021;53(3):403-11. https://doi.org/10.1038/s41588-021-00790-6.
13. Fang R, Preissl S, Li Y, Hou X, Lucero J, Wang X, et al. Comprehensive analysis of single cell ATAC-seq data with SnapATAC. Nat Commun. 2021;12(1):1337. https://doi.org/10.1038/s41467-021-21583-9.
14. Ramos M, Schiffer L, Re A, Azhar R, Basunia A, Rodriguez C, et al. Software for the Integration of Multiomics Experiments in Bioconductor. Cancer Res. 2017;77(21):e39–42. https://doi.org/10.1158/0008-5472.CAN-17-0344.
15. Hoffman P, Satija R. SeuratDisk: Interfaces for HDF5-Based Single Cell File Formats. GitHub. https://github.com/mojaveazure/seurat-disk.
16. Pagès H. HDF5Array: HDF5 backend for DelayedArray objects, 2018. URL https://bioconductor.org/packages/HDF5Array R package version. 1.
17. Virshup I, Rybakov S, Theis FJ, Angerer P, Wolf FA. bioRxiv. 2021.12.16.473007. https://doi.org/10.1101/2021.12.16.473007.
18. Van Rossum G, Drake FL Jr. Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam; 1995.
19. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2021. URL https://www.R-project.org/.
20. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A Fresh Approach to Numerical Computing. SIAM Rev. 2017;59(1):65–98. https://doi.org/10.1137/141000671.
21. The HDF5® Library & File Format. http://www.hdfgroup.org/HDF5. Accessed 14 May 2021.
22. Single Cell Multiome ATAC + Gene Expression - 10x Genomics. https://www.10xgenomics.com/products/single-cell-multiome-atac-plus-gene-expression. Accessed 14 May 2021.
23. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019;15:e8746. https://doi.org/10.15252/msb.20188746.
24. Yan F, Powell DR, Curtis DJ, Wong NC. From reads to insight: a hitchhiker's guide to ATAC-seq data analysis. Genome Biol. 2020;21(1):22. https://doi.org/10.1186/s13059-020-1929-3.
25. Leek JT, Storey JD. Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genet. 2007;3(9):1724–35. https://doi.org/10.1371/journal.pgen.0030161.
26. Lee D, Cheng A, Lawlor N, Bolisetty M, Ucar D. Detection of correlated hidden factors from single cell transcriptomes using Iteratively Adjusted-SVA (IA-SVA). Sci Rep. 2018;8(1):17040. https://doi.org/10.1038/s41598-018-35365-9.
27. Buettner F, Pratanwanich N, McCarthy DJ, Marioni JC, Stegle O. f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. Genome Biol. 2017;18(1):212. https://doi.org/10.1186/s13059-017-1334-8.
28. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert J-P. A general and flexible method for signal extraction from single-cell RNA-seq data. Nat Commun. 2018;9(1):284. https://doi.org/10.1038/s41467-017-02554-5.
29. Argelaguet R, Velten B, Arnol D. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. Mol Syst Biol. 2018;14(6):e8124. https://doi.org/10.15252/msb.20178124.
30. Argelaguet R, Arnol D, Bredikhin D, Deloro Y, Velten B, Marioni JC, et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. 2020;21(1):111. https://doi.org/10.1186/s13059-020-02015-1.
31. Hao Y, Hao S, Andersen-Nissen E, Mauck III WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184(13):3573-3587. https://doi.org/10.1016/j.cell.2021.04.048.
32. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, et al. Similarity network fusion for aggregating data types on a genomic scale. Nat Methods. 2014;11(3):333–7. https://doi.org/10.1038/nmeth.2810.
33. Becht E, McInnes L, Healy J, Dutertre CA, Kwok IW, Ng LG, Ginhoux F, Newell EW. Dimensionality reduction for visualizing single-cell data using UMAP. Nat Biotechnol. 2019;37(1):38-44. https://doi.org/10.1038/nbt.4314.
34. Mucha PJ, Richardson T, Macon K, Porter MA, Onnela J-P. Community structure in time-dependent, multiscale, and multiplex networks. Science. 2010;328(5980):876–8. https://doi.org/10.1126/science.1184819.
35. Clevers H, Alarcon B, Wileman T, Terhorst C. The T cell receptor/CD3 complex: a dynamic protein ensemble. Annu Rev Immunol. 1988;6(1):629–62. https://doi.org/10.1146/annurev.iy.06.040188.003213.
36. Liu P, Li P, Burke S. Critical roles of Bcl11b in T-cell development and maintenance of T-cell identity. Immunol Rev. 2010;238(1):138–49. https://doi.org/10.1111/j.1600-065X.2010.00953.x.
37. Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, et al. Simultaneous epitope and transcriptome measurement in single cells. Nat Methods. 2017;14(9):865–8. https://doi.org/10.1038/nmeth.4380.
38. Mulè MP, Martins AJ, Tsang JS. Normalizing and denoising protein expression data from droplet-based single cell profiling. bioRxiv. 2020:2020.02.24.963603. https://doi.org/10.1101/2020.02.24.963603.
39. Mattanovich D, Borth N. Applications of cell sorting in biotechnology. Microb Cell Fact. 2006;5(1):12. https://doi.org/10.1186/1475-2859-5-12.
40. Kelsey G, Stegle O, Reik W. Single-cell epigenomics: Recording the past and predicting the future. Science. 2017;358(6359):69–75. https://doi.org/10.1126/science.aan6826.
41. Efremova M, Teichmann SA. Computational methods for single-cell omics across modalities. Nat Methods. 2020;17(1):14–7. https://doi.org/10.1038/s41592-019-0692-4.
42. Lee J, Hyeon DY, Hwang D. Single-cell multiomics: technologies and data analysis methods. Exp Mol Med. 2020;52(9):1428–42. https://doi.org/10.1038/s12276-020-0420-2.
43. Miao Z, Humphreys BD, McMahon AP, Kim J. Multi-omics integration in the age of million single-cell data. Nat Rev Nephrol. 2021;17(11):1–15. https://doi.org/10.1038/s41581-021-00463-x.
44. Rivello F, van Buijtenen E, Matuła K, van Buggenum JA, Vink P, van Eenennaam H, Mulder KW, Huck WT. Single-cell intracellular epitope and transcript detection reveals signal transduction dynamics. Cell Rep Methods. 2021;1(5):100070. https://doi.org/10.1016/j.crmeth.2021.100070.
45. Katzenelenbogen Y, Sheban F, Yalin A, Yofe I, Svetlichnyy D, Jaitin DA, et al. Coupled scRNA-Seq and Intracellular Protein Activity Reveal an Immunosuppressive Role of TREM2 in Cancer. Cell. 2020;182:872–85 e19. https://doi.org/10.1016/j.cell.2020.06.032.

46. Stovner EB, Sætrom P. PyRanges: efficient comparison of genomic intervals in Python. Bioinformatics. 2020;36(3):918–9. https://doi.org/10.1093/bioinformatics/btz615.
47. Palla G, Spitzer H, Klein M, Fischer D, Schaar AC, Kuemmerle LB, et al. Squidpy: a scalable framework for spatial single cell analysis. bioRxiv. 2021:2021.02.19.431994. https://doi.org/10.1101/2021.02.19.431994.
48. Clark SJ, Argelaguet R, Kapourani C-A, Stubbs TM, Lee HJ, Alda-Catalinas C, et al. scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. Nat Commun. 2018;9:781. https://doi.org/10.1038/s41467-018-03149-4.
49. Swanson E, Lord C, Reading J, Heubeck AT, Genge PC, Thomson Z, et al. Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using TEA-seq. Elife. 2021;10. https://doi.org/10.7554/eLife.63632.
50. Rood JE, Stuart T, Ghazanfar S, Biancalani T, Fisher E, Butler A, et al. Toward a Common Coordinate Framework for the Human Body. Cell. 2019:1455–67. https://doi.org/10.1016/j.cell.2019.11.019.
51. Rozenblatt-Rosen O, Shin JW, Rood JE, Hupalowska A, Human Cell Atlas Standards and Technology Working Group, Regev A, et al. Building a high-quality Human Cell Atlas. Nat Biotechnol. 2021;39(2):149–53. https://doi.org/10.1038/s41587-020-00812-4.
52. Velten B, Braunger JM, Arnol D, Argelaguet R, Stegle O. Identifying temporal and spatial patterns of variation from multi-modal data using MEFISTO. bioRxiv. 2020.11.03.366674. https://doi.org/10.1101/2020.11.03.366674.
53. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020;585(7825):357–62. https://doi.org/10.1038/s41586-020-2649-2.
54. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17(3):261–72. https://doi.org/10.1038/s41592-019-0686-2.
55. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
56. McKinney W. Others. pandas: a foundational Python library for data analysis and statistics. Python High Perform Sci Comput. 2011;14:1–9.
57. Collette A. Python and HDF5: unlocking scientific data. "O'Reilly Media, Inc."; 2013.
58. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Engi. 2007;9(03):90–5. https://doi.org/10.1109/MCSE.2007.55.
59. Waskom M. seaborn: statistical data visualization. J Open Source Softw. 2021;6:3021.
60. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. Nat Methods. 2015;12(2):115–21. https://doi.org/10.1038/nmeth.3252.
61. Genomics 10x. PBMC from a healthy donor - granulocytes removed through cell sorting (10k). [accessed 10 Dec 2021]. https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-granulocytes-removed-through-cell-sorting-10-k-1-standard-1-0-0
62. Genomics 10x. 5k Peripheral blood mononuclear cells from a healthy donor (v3 chemistry). [accessed 10 Dec 2021]. https://www.10xgenomics.com/resources/datasets/5-k-peripheral-blood-mononuclear-cells-pbm-cs-from-a-healthy-donor-v-3-chemistry-3-1-standard-3-0-2
63. Bredikhin D, Kats I, Stegle O. muon: multimodal omics Python framework. Github. https://github.com/scverse/muon.
64. Bredikhin D, Kats I, Stegle O. mudata: multimodal data. Github. https://github.com/scverse/mudata.
65. Bredikhin D, Kats I, Stegle O. Muon.jl. Github. https://github.com/scverse/Muon.jl.
66. Bredikhin D, Kats I, Stegle O. MuData. Github. https://github.com/PMBio/MuDataMAE.
67. Bredikhin D, Kats I, Stegle O. MuDataSeurat. Github. https://github.com/PMBio/MuDataSeurat.
68. Bredikhin D, Kats I, Stegle O. Muon: multimodal omics analysis framework. Zenodo. 2021. https://doi.org/10.5281/ZENODO.5557542.

Publisher's Note