The Journal of Open Source Software

OpenOmics: A bioinformatics API to integrate multi-omics datasets and interface with public databases.

Nhat C. Tran[*1] **and Jean X. Gao**[1]

1 Department of Computer Science and Engineering, The University of Texas at Arlington

DOI: 10.21105/joss.03249

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @arfon

Submitted: 30 April 2021
Published: 09 May 2021

Summary

Leveraging large-scale multi-omics data is emerging as the primary approach for systemic research of human diseases and general biological processes. As data integration and feature engineering are the vital steps in these bioinformatics projects, there currently lacks a tool for standardized preprocessing of heterogeneous multi-omics and annotation data within the context of a clinical cohort. OpenOmics is a Python library for integrating heterogeneous multi-omics data and interfacing with popular public annotation databases, e.g., GENCODE, Ensembl, BioGRID. The library is designed to be highly flexible to allow the user to parameterize the construction of integrated datasets, interactive to assist complex data exploratory analyses, and scalable to facilitate working with large datasets on standard machines. In this paper, we demonstrate the software design choices to support the wide-ranging use cases of OpenOmics with the goal of maximizing usability and reproducibility of the data integration framework.

Statement of need

Recent advances in sequencing technology and computational methods have enabled the means to generate large-scale, high-throughput multi-omics data (Lappalainen et al., 2013), providing unprecedented research opportunities for cancer and other diseases. These methods have already been applied to a number of problems within bioinformatics, and indeed several integrative disease studies (Hassan et al., 2020; Network & others, 2014; Ren et al., 2016; Zhang et al., 2014). In addition to the genome-wide measurements of different genetic characterizations, the growing public knowledge-base of functional annotations (Consortium, 2016; Derrien et al., 2012), experimentally-verified interactions (Chou et al., 2015, 2017; Oughtred et al., 2018; Yuan et al., 2013), and gene-disease associations (Chen et al., 2012; Huang et al., 2018; Piñero et al., 2016) also provides the prior-knowledge essential for system-level analyses. Leveraging these data sources allow for a systematic investigation of disease mechanisms at multiple molecular and regulatory layers; however, such task remains nontrivial due to the complexity of multi-omics data.

While researchers have developed several mature tools to access or analyze a particular single omic data type (Stuart & Satija, 2019; Wolf et al., 2018), the current state of integrative data platforms for multi-omics data is lacking due to three reasons. First, pipelines for data integration carry out a sequential tasks that does not process multi-omics datasets holistically. Second, the vast size and heterogeneity of the data poses a challenge on the necessary data storage and computational processing. And third, implementations of data pipelines are close-ended for down-stream analysis or not conductive to data exploration use-cases. Additionally, there is currently a need for increased transparency in the process of multi-omics data

*corresponding author

integration, and a standardized data preprocessing strategy is important for the interpretation and exchange of bioinformatic projects. Currently, there exist very few systems that, on the one hand, supports standardized handling of multi-omics datasets but also allows to query the integrated dataset within the context of a clinical cohort.

Related works

There are several existing platforms that aids in the integration of multi-omics data, such as Galaxy, Anduril, MixOmics and O-Miner. First, Galaxy (Boekel et al., 2015) and Anduril (Cervera et al., 2019) are mature platforms and has an established workflow framework for genomic and transcriptomic data analysis. Galaxy contains hundreds of state-of-the-art tools of these core domains for processing and assembling high-throughput sequencing data. Second, MixOmics (Rohart et al., 2017) is an R library dedicated to the multivariate analysis of biological data sets with a specific focus on data exploration, dimension reduction and visualisation. Third, O-Miner (Sangaralingam et al., 2019) is web tool that provides a pipeline for analysis of both transcriptomic and genomic data starting from raw image files through in-depth bioinformatics analysis. However, as large-scale multi-omic data analysis demands continue to grow, the technologies and data analysis needs continually change to adapt with `big data`. For instance, the data manipulation required for multi-omics integration requires a multitude of complex operations, but the point and click interface given in existing Galaxy tools can be limiting or not computationally efficient. Although the MixOmics toolkit provides an R programming interface, it doesn't yet leverage high-performance distributed storage or computing resources. Finally, while O-Miner can perform end-to-end analysis in an integrated platform, its interim analysis results cannot be exported elsewhere for down-stream analysis.

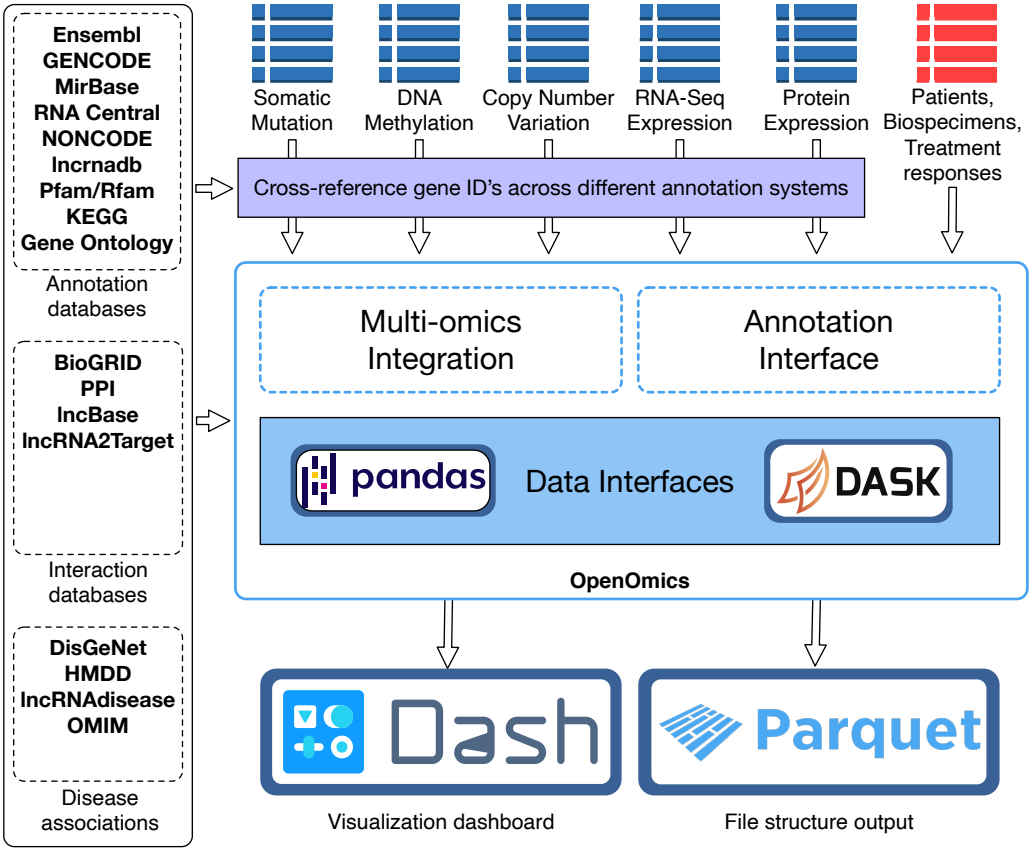

Figure 1: Overall OpenOmics System Architecture, Data Flow, and Use Cases.

The OpenOmics library

OpenOmics consists of two core modules: multi-omics integration and annotation interface. An overview visualization of the OpenOmics system architecture is provided in Figure 1.

Multi-omics integration

Tabular data are everywhere in bioinformatics. To record expression quantifications, annotations, or variant calls, data are typically stored in various tabular-like formats, such as BED, GTF, MAF, and VCF, which can be preprocessed and normalized to row indexed formats. Given any processed single-omic dataset, the library generalizes the data as a tabular structure where rows correspond to observation samples and columns correspond to measurements of different biomolecules. The core functionality of the Multi-omics Integration module is to integrate the multiple single-omic datasets for the overlapping samples. By generating multi-omics data for the same set of samples, our tool can provide the necessary data structure to develop insights into the flow of biological information across multiple genome, epigenome, transcriptome, proteome, metabolome and phenome levels. The user can import and integrate the following supported omic types:

- Genomics: single nucleotide variants (SNV), copy number variation (CNV)
- Epigenomics: DNA methylation
- Transcriptomics: RNA-Seq, miRNA expression, lncRNA expression, microarrays
- Proteomics: reverse phase protein array (RPPA), iTRAQ

After importing each single omics data, OpenOmics stores a Pandas Dataframe (McKinney, 2010) that is flexible for a wide range of tabular operations. For instance, the user is presented with several functions for preprocessing of the expression quantifications to normalize, filter outliers, or reduce noise.

Within a study cohort, the clinical characteristics are crucial for the study of a disease or biological phenomenon. The user can characterize the set of samples using the Clinical Data structure, which is comprised of two levels: Patient and Biospecimen. A Patient can have attribute fields on demographics, clinical diagnosis, disease progression, treatment responses, and survival outcomes. Typically, multi-omics data observations are captured at the Biospecimen level and each Patient can have multiple Biospecimens. OpenOmics tracks the ID's of biospecimens and the patient it belongs to, so the multi-omics data are organized in a hierarchical order to enable aggregated operations.

Annotation interface

After importing and integrating the multi-omic data, the user can supplement their dataset with various annotation attributes from public data repositories such as GENCODE, Ensembl, and RNA Central. With just a few operations, the user can easily download a data repository of choice, select relevant attributes, and efficiently join a variable number of annotation columns to their genomics, transcriptomics, and proteomics data. The full list of databases and the availability of annotation attributes is listed in Table 1.

For each public database, the Annotation Interface module provides a series of interfaces to perform specific importing, preprocessing, and annotation tasks. At the import step, the module can either fetch the database files via a file-transfer-protocol (ftp) URL or load a locally downloaded file. At this step, the user can specify the species, genome build, and version of the database by providing a ftp URL of choice. To streamline this process, the module

automatically caches downloaded file to disk, uncompress them, and handle different file extensions, including FASTA, GTF, VCF, and other tabular formats. Then, at the preprocessing step, the module selects only the relevant attribute fields specified by the user and perform necessary data cleanings. Finally, the annotation data can be annotated to an omics dataset by performing a SQL-like join operation on a user-specified index of the biomolecule name or ID. If the user wishes to import an annotation database not yet included in OpenOmics, they can extend the Annotation Dataset API to specify their own importing, preprocessing, and annotation tasks in an object-oriented manner.

An innovative feature of our integration module is the ability to cross-reference the gene ID's between different annotation systems or data sources. When importing a dataset, the user can specify the level of genomic index, such as at the gene, transcript, protein, or peptide level, and whether it is a gene name or gene ID. Since multiple single-omics datasets can use different gene nomenclatures, the user is able to convert between the different gene indexing methods by reindexing the annotation dataframe with a index column of choice. This not only allows the Annotation Interface to select and join the annotation data to the correct index level, but also allow the user to customize the selection and aggregation of biological measurements at different levels.

Data Repository	Annotation Data Available	Index	# entries
GENCODE	Genomic annotations, primary sequence	RNAs	60,660
Ensembl	Genomic annotations	Genes	232,186
MiRBase	MicroRNA sequences and annotatinos	MicroRNAs	38,589
RNA Central	ncRNA sequence and annotation collection	ncRNAs	14,784,981
NONCODE	lncRNA sequences and annotations	LncRNAs	173,112
lncrnadb	lncRNA functional annotations	LncRNAs	100
Pfam	Protein family annotation	Proteins	18,259
Rfam	RNA family annotations	ncRNAs	2,600
Gene Ontology	Functional, cellular, and molecular annotations	Genes	44,117
KEGG	High-level functional pathways	Genes	22,409
DisGeNet	gene-disease associations	Genes	1,134,942
HMDD	microRNA-disease associations	MicroRNAs	35,547
lncRNAdisease	lncRNA-disease associations	LncRNAs	3,000
OMIM	Ontology of human diseases	Diseases	25,670

Table 1: Public annotation databases and availability of data in the Human genome.

System design

This section describes the various implementation details behind the scalable processing and efficient data storage, and the design choices in the development operations.

While the in-memory Pandas dataframes utilized in our data structures are fast, they have size and speed limitations when the dataset size approaches the system memory limit. When this is an issue, the user can enable out-of-memory distributed data processing on all OpenOmics operations, implemented by the Dask framework (Rocklin, 2015). When memory resources is limited, data in a Dask dataframe can be read directly from disk and is only brought into memory when needed during computations (also called lazy evaluations). When performing data query operations on Dask dataframes, a task graph containing each operation is built and is only evaluated on command, in a process called lazy loading.

Operations on Dask dataframes are the same as Pandas dataframes, but can utilize multiple workers and can scale up to clusters by connecting to a cluster client with minimal configuration. To enable this feature in OpenOmics, the user simply needs to explicitly enable an

option when importing an omics dataset, importing an annotation/interaction database, or importing a MultiOmics file structure on disk.

Software requirements

OpenOmics is distributed as a readily installable Python package from the Python Package Index (PyPI) repository. For users to install OpenOmics in their own Python environment, several software dependencies are automatically downloaded to reproduce the computing environment.

OpenOmics is compatible with Python 3.6 or higher, and is operational on both Linux and Windows operating systems. The software requires as little as 4 GB of RAM and 2 CPU cores, and can computationally scale up to large-memory multi-worker distributed systems such as a compute cluster. To take advantage of increased computational resource, OpenOmics simply requires one line of code to activate parallel computing functionalities.

Development operations

We developed OpenOmics following modern software best-practices and package publishing standards. For the version control of our source-code, we utilized a public GitHub repository which contains two branches, master and develop. The master branch contains stable and well-tested releases of the package, while the develop branch is used for building new features or software refactoring. Before each version is released, we utilize Github Actions for continuous integration, building, and testing for version and dependency compatibility. Our automated test suite covers essential functions of the package and a reasonable range of inputs and conditions.

Conclusion

A standardized data preprocessing strategy is essential for the interpretation and exchange of bioinformatics research. OpenOmics provides researchers with the means to consistently describe the processing and analysis of their experimental datasets. It equips the user, a bioinformatician, with the ability to preprocess, query, and analyze data with modern and scalable software technology. As the wide array of tools and methods available in the public domain are largely isolated, OpenOmics aims toward a uniform framework that can effectively process and analyze multi-omics data in an end-to-end manner along with biologist-friendly visualization and interpretation.

Acknowledgements

N/A.

References

Boekel, J., Chilton, J. M., Cooke, I. R., Horvatovich, P. L., Jagtap, P. D., Käll, L., Lehtiö, J., Lukasse, P., Moerland, P. D., & Griffin, T. J. (2015). Multi-omic data analysis using galaxy. *Nature Biotechnology*, *33*(2), 137–139. https://doi.org/10.1038/nbt.3134

Cervera, A., Rantanen, V., Ovaska, K., Laakso, M., Nuñez-Fontarnau, J., Alkodsi, A., Casado, J., Facciotto, C., Häkkinen, A., Louhimo, R., & others. (2019). Anduril 2: Upgraded large-scale data integration framework. *Bioinformatics*, *35*(19), 3815–3817. https://doi.org/10.1093/bioinformatics/btz133

Chen, G., Wang, Z., Wang, D., Qiu, C., Liu, M., Chen, X., Zhang, Q., Yan, G., & Cui, Q. (2012). LncRNADisease: A database for long-non-coding RNA-associated diseases. *Nucleic Acids Research*, *41*(D1), D983–D986. https://doi.org/10.1093/nar/gks1099

Chou, C.-H., Chang, N.-W., Shrestha, S., Hsu, S.-D., Lin, Y.-L., Lee, W.-H., Yang, C.-D., Hong, H.-C., Wei, T.-Y., Tu, S.-J., & others. (2015). miRTarBase 2016: Updates to the experimentally validated miRNA-target interactions database. *Nucleic Acids Research*, *44*(D1), D239–D247.

Chou, C.-H., Shrestha, S., Yang, C.-D., Chang, N.-W., Lin, Y.-L., Liao, K.-W., Huang, W.-C., Sun, T.-H., Tu, S.-J., Lee, W.-H., & others. (2017). miRTarBase update 2018: A resource for experimentally validated microRNA-target interactions. *Nucleic Acids Research*, *46*(D1), D296–D302.

Consortium, R. (2016). RNAcentral: A comprehensive database of non-coding RNA sequences. *Nucleic Acids Research*, gkw1008.

Derrien, T., Johnson, R., Bussotti, G., Tanzer, A., Djebali, S., Tilgner, H., Guernec, G., Martin, D., Merkel, A., Knowles, D. G., & others. (2012). The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Research*, *22*(9), 1775–1789. https://doi.org/10.1101/gr.132159.111

Hassan, M. A., Al-Sakkaf, K., Shait Mohammed, M. R., Dallol, A., Al-Maghrabi, J., Aldahlawi, A., Ashoor, S., Maamra, M., Ragoussis, J., Wu, W., & others. (2020). Integration of transcriptome and metabolome provides unique insights to pathways associated with obese breast cancer patients. *Frontiers in Oncology*, *10*, 804. https://doi.org/10.3389/fonc.2020.00804

Huang, Z., Shi, J., Gao, Y., Cui, C., Zhang, S., Li, J., Zhou, Y., & Cui, Q. (2018). HMDD v3. 0: A database for experimentally supported human microRNA–disease associations. *Nucleic Acids Research*.

Lappalainen, T., Sammeth, M., Friedländer, M. R., Ac't Hoen, P., Monlong, J., Rivas, M. A., Gonzalez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P. G., & others. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, *501*(7468), 506–511.

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Network, C. G. A. R., & others. (2014). Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, *511*(7511), 543–550.

Oughtred, R., Stark, C., Breitkreutz, B.-J., Rust, J., Boucher, L., Chang, C., Kolas, N., O'Donnell, L., Leung, G., McAdam, R., Zhang, F., Dolma, S., Willems, A., Coulombe-Huntington, J., Chatr-aryamontri, A., Dolinski, K., & Tyers, M. (2018). The BioGRID interaction database: 2019 update. *Nucleic Acids Research*, *47*(D1), D529–D541. https://doi.org/10.1093/nar/gky1079

Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., García-García, J., Sanz, F., & Furlong, L. I. (2016). DisGeNET: A comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, gkw943. https://doi.org/10.1093/nar/gkw943

Ren, S., Shao, Y., Zhao, X., Hong, C. S., Wang, F., Lu, X., Li, J., Ye, G., Yan, M., Zhuang, Z., & others. (2016). Integration of metabolomics and transcriptomics reveals major

metabolic pathways and potential biomarker involved in prostate cancer. *Molecular & Cellular Proteomics*, *15*(1), 154–163. https://doi.org/10.1074/mcp.M115.052381

Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. In K. Huff & J. Bergstra (Eds.), *Proceedings of the 14th python in science conference* (pp. 130–136). https://doi.org/10.25080/Majora-7b98e3ed-013

Rohart, F., Gautier, B., Singh, A., & Le Cao, K.-A. (2017). mixOmics: An r package for 'omics feature selection and multiple data integration. *PLoS Computational Biology*, *13*(11), e1005752. https://doi.org/10.1371/journal.pcbi.1005752

Rohart, F., Gautier, B., Singh, A., & Le Cao, K.-A. (2017). mixOmics: An r package for 'omics feature selection and multiple data integration. *PLoS Computational Biology*, *13*(11), e1005752. https://doi.org/10.1371/journal.pcbi.1005752

Sangaralingam, A., Dayem Ullah, A. Z., Marzec, J., Gadaleta, E., Nagano, A., Ross-Adams, H., Wang, J., Lemoine, N. R., & Chelala, C. (2019). 'multi-omic'data analysis using o-miner. *Briefings in Bioinformatics*, *20*(1), 130–143. https://doi.org/10.1093/bib/bbx080

Stuart, T., & Satija, R. (2019). Integrative single-cell analysis. *Nature Reviews Genetics*, *20*(5), 257–272.

Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, *19*(1), 1–5. https://doi.org/10.1186/s13059-017-1382-0

Yuan, J., Wu, W., Xie, C., Zhao, G., Zhao, Y., & Chen, R. (2013). NPInter v2. 0: An updated database of ncRNA interactions. *Nucleic Acids Research*, *42*(D1), D104–D108. https://doi.org/10.1093/nar/gkt1057

Zhang, B., Wang, J., Wang, X., Zhu, J., Liu, Q., Shi, Z., Chambers, M. C., Zimmerman, L. J., Shaddox, K. F., Kim, S., & others. (2014). Proteogenomic characterization of human colon and rectal cancer. *Nature*, *513*(7518), 382–387.