



Published in final edited form as:

Cancer Res. 2017 November 01; 77(21): e39–e42. doi:10.1158/0008-5472.CAN-17-0344.

Software for the integration of multi-omics experiments in Bioconductor

Marcel Ramos^{1,2}, Lucas Schiffer^{1,2}, Angela Re³, Rimsha Azhar^{1,2}, Azfar Basunia⁴, Carmen Rodriguez^{1,2}, Tiffany Chan^{1,2}, Phil Chapman⁵, Sean R. Davis⁶, David Gomez-Cabrero⁷, Aedin C. Culhane^{4,8}, Benjamin Haibe-Kains^{9,10,11,12}, Kasper D. Hansen^{13,14}, Hanish Kodali^{1,2}, Marie S. Louis^{1,2}, Arvind S. Mer⁹, Markus Riester¹⁵, Martin Morgan¹⁶, Vince Carey^{4,17}, and Levi Waldron^{1,2,*}

¹CUNY School of Public Health, New York, New York, USA

²Institute for Implementation Science in Population Health, City University of New York, New York, USA

³Laboratory of Translational Genomics, Centre for Integrative Biology, University of Trento, Polo Scientifico e Tecnologico Fabio Ferrari, Trento, Italy

⁴Harvard TH Chan School of Public Health, Boston, Massachusetts, USA

⁵Computational Biology Support Team, Cancer Research UK Manchester Institute, The University of Manchester, Manchester, UK

⁶Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, MD

⁷Mucosal and Salivary Biology Division, King's College London Dental Institute, London, UK

⁸Dana-Farber Cancer Institute, 450 Brookline Ave, Boston, MA (was 16)

⁹Princess Margaret Cancer Center, University Health Network, Toronto, Ontario, Canada

¹⁰Department of Medical Biophysics, University of Toronto, Toronto, Ontario, Canada

¹¹Department of Computer Science, University of Toronto, Toronto, Ontario, Canada

¹²Ontario Institute of Cancer Research, Toronto, Ontario, Canada

¹³Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, Maryland, USA

¹⁴McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins School of Medicine, Baltimore, Maryland, USA

¹⁵Novartis Institutes for BioMedical Research, Cambridge, Massachusetts, USA

¹⁶Roswell Park Cancer Institute, University of Buffalo, Buffalo, New York, USA

*Correspondence to Levi Waldron <levi.waldron@sph.cuny.edu>.

The authors declare no potential conflicts of interest.

¹⁷Channing Division of Network Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, Massachusetts, USA

Abstract

Multi-omics experiments are increasingly commonplace in biomedical research, and add layers of complexity to experimental design, data integration, and analysis. R and Bioconductor provide a generic framework for statistical analysis and visualization, as well as specialized data classes for a variety of high-throughput data types, but methods are lacking for integrative analysis of multi-omics experiments. The MultiAssayExperiment software package, implemented in R and leveraging Bioconductor software and design principles, provides for the coordinated representation of, storage of, and operation on multiple diverse genomics data. We provide the unrestricted multiple 'omics data for each cancer tissue in The Cancer Genome Atlas (TCGA) as ready-to-analyze MultiAssayExperiment objects, and demonstrate in these and other datasets how the software simplifies data representation, statistical analysis, and visualization. The MultiAssayExperiment Bioconductor package reduces major obstacles to efficient, scalable and reproducible statistical analysis of multi-omics data and enhances data science applications of multiple omics datasets.

INTRODUCTION

Multi-assay experiments collect multiple, complementary data types for a set of specimens. Bioconductor (1) provides classes to ensure coherence between a single assay and patient data during data analysis, such as `eSet` and `SummarizedExperiment` (2). However, novel challenges arise in data representation, management, and analysis of multi-assay experiments (3), , that cannot be addressed by these or other single-assay data architectures. These include 1) coordination of different assays on, for example, genes, microRNAs, or genomic ranges, 2) coordination of missing or replicated assays, 3) sample identifiers that differ between assays, 4) re-shaping data to fit the variety of existing statistical and visualization packages, 5) doing the above in a concise and reproducible way that is amenable to new assay types and data classes.

The need for a unified data model for multi-omics experiments has been recognized in other projects, such as `MultiDataSet` (4) and `CNAMet` (5). Our developments are motivated by an interest in bridging effective single-assay Application Program Interface (API) elements, including endomorphic feature and sample subset operations, to multi-omic contexts of arbitrary complexity and volume (Supplemental Table 1). A main concern in our work is to allow data analysts and developers to simplify the management of both traditional in-memory assay stores for smaller datasets, and out-of-memory assay stores for very large data, in such formats as HDF5 (6), tabix-indexed VCF (7), or Google BigTable (8).

MultiAssayExperiment provides data structures and methods for representing, manipulating, and integrating multi-assay genomic experiments. It integrates an *open-ended* set of R and Bioconductor single-assay data classes, while abstracting the complexity of back-end data objects a sufficient set of data manipulation, extraction, and re-shaping methods to interface with most R/Bioconductor data analysis and visualization tools. We demonstrate its use by

representing unrestricted data from TCGA as a single `MultiAssayExperiment` object per cancer type, and demonstrating greatly simplified multi-assay analyses with these and other public multi-omics datasets.

MATERIALS AND METHODS

`MultiAssayExperiment` <https://bioconductor.org/packages/MultiAssayExperiment> introduces a Bioconductor object-oriented S4 class, defining a general data structure for representing multi-omics experiments. This data class has three key components: 1) *colData*, a “primary” dataset containing patient or cell line-level characteristics such as pathology and histology, 2) *ExperimentList*, a list of results from complementary experiments, and 3) *sampleMap*, a map that relates these elements (Figure 1a). *ExperimentList* data elements may be of any data class that has standard methods for basic subsetting (single square bracket ‘[’) and dimension names and sizes (`dimnames()` and `dim()`). Key methods available for manipulating the `MultiAssayExperiment` data class include:

A constructor function and associated validity checks that simplifies creating `MultiAssayExperiment` objects, while allowing for flexibility in representing complex experiments.

Subsetting operations allowing data selection by genomic identifiers or ranges, clinical/pathological variables, available complete data (subsets that include no missing values), and by specific assays.

Robust and intuitive extraction and replacement operations for components of the *MultiAssayExperiment*.

The *MultiAssayExperiment* API is based wherever possible on *SummarizedExperiment* while supporting heterogeneous multi-omics experiments. `MultiAssayExperiment` design, constructor, subsetting, extraction, and helper methods, as well as methods and code for the examples demonstrated here, are detailed in the Supplemental Methods.

RESULTS

The `MultiAssayExperiment` class and methods (Table 1) provide a flexible framework for integrating and analyzing complementary assays on an overlapping set of samples. It integrates any data class that supports basic subsetting and dimension names, so that many data classes are supported by default without additional accommodations. The `MultiAssayExperiment` class (Figure 1a) ensures correct alignment of assays and patients, provides coordinated subsetting of samples and features while maintaining correct alignment, and enables simple integration of data types to formats amenable to analysis by existing tools. Basic usage is outlined in Video 1 (<https://youtu.be/w6HWAHaDpyk>), and in the *QuickStartMultiAssay* vignette accompanying the package.

We coordinated over 300 assays from over 11,000 patients of 33 different cancer types from The Cancer Genome Atlas, as one `MultiAssayExperiment` per cancer type (Supplemental Table 2). These data objects link each assay to their patient of origin, allowing more straightforward selection of cases with complete data for assays of interest, and integration

of data across assays and between assays and clinical data. We demonstrate applications of MultiAssayExperiment for visualizing the overlap in assays performed for adrenocortical carcinoma patients (Figure 1b), confirming recently reported correlations between somatic mutation and copy number burden in colorectal cancer and breast cancer (Figure 1c), identifying a SNP/methylation quantitative trait locus using remotely stored tabix-indexed VCF files for the 1000 genomes project (Figure 1d), multi-assay gene set analysis for ovarian cancer (Supplemental Figures 1–2), and calculating correlations between copy number, gene expression, and protein expression in the NCI-60 cell lines (Supplemental Figure 3). Demonstrative code chunks and fully reproducible scripts are given to demonstrate the simple and powerful flexibility provided by MultiAssayExperiment.

DISCUSSION

MultiAssayExperiment enables coordinated management and extraction of complex multi-assay experiments and clinical data, with the same ease of user-level coding as for a single experiment. Its extensible design supports any assay data class meeting basic requirements, including out-of-memory representations for very large datasets. We have confirmed “out-of-the-box” compatibility with on-disk data representations, including the DelayedMatrix class via an HDF5 backend (6), and the VcfStack class based on the GenomicFiles infrastructure. Future work will focus on higher-level visualization, integration, and analysis tools using MultiAssayExperiment as a building block. This project will receive long-term support as a necessary element of multi-assay data representation and analysis in Bioconductor.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Financial Support

The authors' work was funded by the National Cancer Institute of the National Institutes of Health [U24CA180996 to M. Morgan].

This work was supported by the CUNY High Performance Computing Center, which is operated by the College of Staten Island and funded, in part, by grants from the City of New York, State of New York, CUNY Research Foundation, and National Science Foundation Grants CNS-0958379, CNS-0855217 and ACI 1126113.

References

1. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods* Nature Publishing Group. 2015; 12:115–21.
2. Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, Gentleman R, et al. Software for computing and annotating genomic ranges. *PLoS Comput Biol*. 2013; 9:e1003118. [PubMed: 23950696]
3. Kannan L, Ramos M, Re A, El-Hachem N, Safikhani Z, Gendoo DMA, et al. Public data and open source tools for multi-assay genomic investigation of disease. *Brief Bioinform*. 2016; 17:603–15. [PubMed: 26463000]

4. Hernandez-Ferrer C, Ruiz-Arenas C, Beltran-Gomila A, González JR. MultiDataSet: an R package for encapsulating multiple data sets with application to omic data integration. BMC Bioinformatics. 2017; 18:36. [PubMed: 28095799]
5. Louhimo R, Hautaniemi S. CNAmets: an R package for integrating copy number, methylation and expression data. Bioinformatics. 2011; 27:887–8. [PubMed: 21228048]
6. Folk, M., Heber, G., Koziol, Q., Pourmal, E., Robinson, D. An Overview of the HDF5 Technology Suite and Its Applications. Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases; New York, NY, USA: ACM; 2011. p. 36-47.
7. Obenchain V, Lawrence M, Carey V, Gogarten S, Shannon P, Morgan M. VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants. Bioinformatics. 2014; 30:2076–8. [PubMed: 24681907]
8. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, et al. Bigtable: A Distributed Storage System for Structured Data. ACM Trans Comput Syst New York, NY, USA: ACM. 2008; 26:4:1–4:26.
9. Conway, JR., Lex, A., Gehlenborg, N. bioRxiv [Internet]. Cold Spring Harbor Labs Journals; 2017. UpSetR: An R Package For The Visualization Of Intersecting Sets And Their Properties. Available from: <http://www.biorxiv.org/content/early/2017/03/25/120600>
10. Davoli, T., Uno, H., Wooten, EC., Elledge, SJ. Tumor aneuploidy correlates with markers of immune evasion and with reduced response to immunotherapy; Science [Internet]. 2017. p. 355 Available from: <http://dx.doi.org/10.1126/science.aaf8399>

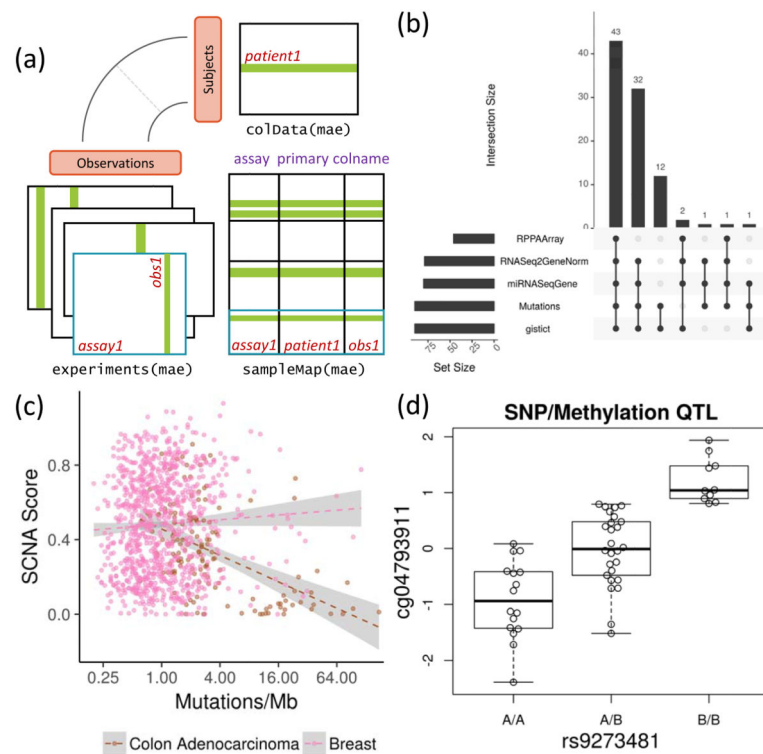


Figure 1. MultiAssayExperiment design and applications

(a) The MultiAssayExperiment object schematic shows the design of the infrastructure class, detailed in Supplemental Table 3. The `colData` provides data about the patients, cell lines, or other biological units, with one row per unit and one column per variable. The `experiments` are a list of assay datasets of arbitrary class, with one column per observation. The `sampleMap` links a single table of patient data (`colData`) to a list of experiments via a simple but powerful table of experiment:patient edges (relationships), that can be created automatically in simple cases or in a spreadsheet if assay-specific sample identifiers are used. `sampleMap` relates each column (observation) in the assays (`experiments`) to exactly one row (biological unit) in `colData`; however, one row of `colData` may map to zero, one, or more columns per assay, allowing for missing and replicate assays. Green stripes indicate a mapping of one subject to multiple observations across experiments. **(b) The UpSetR (9) graphic** represents a complex Venn diagram of assay availability for patients in a MultiAssayExperiment. This reduced adrenocortical carcinoma object is provided as an example dataset in the MultiAssayExperiment package. The barplot on the left shows sample size of each experiment; links to its right indicate combinations of 1 to 4 experiments, with bars above showing the number of patients having exactly those data types. **(c) Extent of copy number alteration vs. somatic mutation burden.** Cancer types with high levels of aneuploidy often show a positive correlation of mutation load and chromosomal instability (CIN) (10), perhaps due to a higher tolerance of deleterious mutations, as shown here in orange for breast cancer. Tumors with a hypermutator phenotype rarely display extensive CIN, resulting in a negative correlation of mutation load and CIN in cancer types where hypermutation is common (shown in grey for colon adenocarcinoma). **(d) Methylation Quantitative Trait Locus** identified from an on-disk

representation of Variant Call Format (VCF) files of the 1000 Genomes Project integrated with 450K methylation array data as a MultiAssayExperiment.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 1

Summary of the MultiAssayExperiment API.

Category and Function	Description	Returned class
Constructors		
MultiAssayExperiment	Create a MultiAssayExperiment object	MultiAssayExperiment
ExperimentList	Create an ExperimentList from a List or list	ExperimentList
Accessors		
colData	Get or set data that describe the samples	DataFrame
experiments	Get or set the list of experimental data objects as original classes	ExperimentList
assays	Get the list of experimental data numeric matrices	SimpleList
assay	Get the first experimental data numeric matrix	matrix, matrix-like
sampleMap	Get or set the map relating observations to subjects	DataFrame
metadata	Get or set additional data descriptions	list
rownames	Get row names for all experiments	CharacterList
colnames	Get column names for all experiments	CharacterList
Subsetting		
mae[i, j, k]	Get rows, columns, and/or experiments	MultiAssayExperiment
mae[i, ,]	GRanges, character, integer, logical, List, list	MultiAssayExperiment
mae[, j,]	character, integer, logical, List, list	MultiAssayExperiment
mae[, , k]	character, integer, logical	MultiAssayExperiment
mae[[i]]	Get or set object of arbitrary class from experiments	(varies)
mae[[i]]	character, integer, logical	
mae\$column	Get or set colData column	vector (varies)
Management		
complete.cases	Identify subjects with complete data in all experiments	vector (logical)
duplicated	Identify subjects with replicate observations per experiment	list of LogicalLists
mergeReplicates	Merge replicate observations within each experiment	MultiAssayExperiment
intersectRows	Return features that are present for all experiments	MultiAssayExperiment
intersectColumns	Return subjects with data available for all experiments	MultiAssayExperiment
prepMultiAssay	Troubleshoot common problems when constructing main class	list
Reshaping		
longFormat	Return a long and tidy DataFrame with optional colData columns	DataFrame
wideFormat	Create a wide DataFrame, 1 row per subject	DataFrame
Combining		
c	Concatenate an experiment	MultiAssayExperiment

Note. assay refers to a procedure for measuring the biochemical or immunological activity of a sample, e.g. RNA-seq, segmented copy number, and somatic mutation calls would be considered three different assays. experiment refers to the application of an assay to a set of samples. In general it is assumed that each experiment uses a different assay type, although an assay type may of course be repeated in different experiments. mae refers to a MultiAssayExperiment object. subject refers to patient, cell line, or other biological unit. observation refers to results of an assay, e.g. gene expression, somatic mutations, etc. features refer to measurements returned by the assays, labeled by row names or genomic ranges.