

2023

Multimodal, longitudinal, and mega-analysis of biomedical data

<https://hdl.handle.net/2144/47473>

"Downloaded from OpenBU. Boston University's institutional repository."

BOSTON UNIVERSITY
GRADUATE SCHOOL OF ARTS AND SCIENCES
AND
COLLEGE OF ENGINEERING

Dissertation

**MULTIMODAL, LONGITUDINAL, AND MEGA-ANALYSIS
OF BIOMEDICAL DATA**

by

LUCAS SCHIFFER

B.B.A., University of New Mexico, 2012
M.P.H., City University of New York, 2017

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2023

Approved by

First Reader

W. Evan Johnson, Ph.D.
Associate Professor of Medicine and Biostatistics

Second Reader

Jessica Leibler, Dr.P.H., Sc.M.
Assistant Professor of Environmental Health

MULTIMODAL, LONGITUDINAL, AND MEGA-ANALYSIS

OF BIOMEDICAL DATA

LUCAS SCHIFFER

Boston University Graduate School of Arts and Sciences

and College of Engineering, 2023

Major Professor: W. Evan Johnson, Associate Professor of Medicine and Biostatistics

ABSTRACT

Biomedical data science is a multi-disciplinary field concerned with the collection, storage, and interpretation of biomedical data that uses annotation, algorithms, and analysis to extract knowledge and insights from structured and unstructured data to be used in the development and evaluation of diagnostic tests, prognostic predictions, and therapeutic interventions. Biomedical data scientists perform this work using biomedical data that arises when samples are subjected to biochemical assays to quantitatively or qualitatively investigate their pathophysiological characteristics. Increasingly, biomedical data are generated at single-cell resolution and have consequently become far more hierarchical and multimodal in nature – that is, levels of organization encapsulate one another (e.g., samples belonging to subjects are made up of cells) and multiple biological modalities are profiled simultaneously. The paradigm shift adds significant complexity to the collection, storage, management, and analysis of biomedical data, but brings with it the promise of unprecedented insights to be gained from integrative analyses. These analyses are the focus of this dissertation, where the challenges of integrating biomedical data across multiple modalities, timepoints, and studies are examined through three research projects.

Challenges related to multimodal analysis of biomedical data will be explored through the development of MultimodalExperiment, a data structure that appropriately and efficiently represents multiomics data that is hierarchical, multimodal, and/or longitudinal in nature. A schematic of and methods for the data structure will be presented along with example usage to demonstrate how current challenges of alternative data structures are overcome, ease of data management is improved, and computational/storage efficiency is optimized.

Challenges related to longitudinal analysis of biomedical data will be explored in the context of a cohort study of cancer patients being treated with anti-programmed cell death protein 1/programmed cell death ligand 1 immunotherapies at Boston Medical Center. The progression-free survival status of study participants will be analyzed using linear mixed-effects models which incorporate longitudinal high-dimensional metabolomics data. Maps of metabolic pathways and a hypothesis will be presented to explain serum metabolites that are associated with progress-free survival status and possibly therapeutic efficacy.

Challenges related to mega-analysis of biomedical data will be explored through the creation of a pipeline to preprocess transcriptomics data from human host infected with tuberculosis to support machine learning and other tasks. The details of original software developed to provide more than 10,000 samples of clean high-quality machine learning-ready data from all related and eligible studies in the Gene Expression Omnibus repository will be illustrated. The importance improving diagnostic testing and therapeutic interventions for tuberculosis disease will be highlighted in the context of these data, and the specifics of why they represent a key ingredient for machine learning that helps overcome current challenges in the field will be explained.

TABLE OF CONTENTS

ABSTRACT	iv
TABLE OF CONTENTS	vi
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER ONE.....	1
Introduction	1
Multiomics Data	1
Multimodal Data.....	3
Longitudinal Data.....	4
Mega-Analysis.....	4
Dissertation Aims	5
Aim One	5
Aim Two.....	5
Aim Three.....	6
CHAPTER TWO.....	7
MultimodalExperiment.....	7
Introduction	7
Methods	10
Results	13
Discussion.....	25

CHAPTER THREE	27
Metabolomics of IO Therapy	27
Introduction	27
Methods	28
Results	30
Discussion.....	42
CHAPTER FOUR	44
tuberculosis.....	44
Introduction	44
Methods	47
Results	50
Discussion.....	53
CHAPTER FIVE	55
Conclusions	55
LIST OF JOURNAL ABBREVIATIONS	58
BIBLIOGRAPHY	61
CURRICULUM VITAE	75

LIST OF TABLES

Table 2.1. MultimodalExperiment API Reference.....	11
Table 3.1. Demographic and Clinical Characteristics.....	31
Table 3.2. Primary and Secondary Outcome Measures.	32

LIST OF FIGURES

Figure 1.1. Multiomics Data Represented as Matrices.....	1
Figure 1.2. Interrelationships Between Types of Multiomics Data.....	2
Figure 2.1. MultimodalExperiment Schematic.	9
Figure 2.2. MultimodalExperiment Constructor.	12
Figure 2.3. Rows/Columns of Vignette Data.	13
Figure 2.4. Construction of a MultimodalExperiment.	14
Figure 2.5. Maps After Initial Propagation.	15
Figure 2.6. Maps After Type Specification.	16
Figure 2.7. The cellMap After Cell to Sample Specification.	17
Figure 2.8. Maps After Sample to Subject Specification.	18
Figure 2.9. Maps After Harmonization.	20
Figure 2.10. Outer Join of All Maps.....	21
Figure 2.11. The ME Object Without Annotations.	22
Figure 2.12. Creations of ME Annotations.	23
Figure 2.13. The ME Object with Annotations.	24
Figure 2.14. Potential RNA-seq Applications.	26
Figure 3.1. Metabolites Associated with Progression-Free Survival Status.	36
Figure 3.2. Tryptophan, Folic Acid, and Lysine Metabolism.	39
Figure 3.3. ATP Synthesis by Purine Synthesis and Salvage.....	41
Figure 4.1. Data Provenance in the tuberculosis Package.....	46
Figure 4.2. GEO Records Initially Included in tuberculosis.	51
Figure 4.3. Example Usage of the tuberculosis Package.....	52

LIST OF ABBREVIATIONS

5,10-CH ₂ -THF	5,10-methylene tetrahydrofolic acid
ADP	adenosine-5'-diphosphate
AhR	aryl hydrocarbon receptor
AICAR	aminoimidazole-4-carboxamide ribonucleotide
AMP	adenosine monophosphate
API	application programming interface
ATP	adenosine-5'-triphosphate
CITE-seq	cellular indexing of transcriptomes and epitopes by sequencing
CMP	cytidine-5'-monophosphate
dATP	2'-deoxyadenosine 5'-triphosphate
dCMP	deoxycytidine-5'-monophosphate
dcRNA-seq	deconvoluted RNA sequencing
dGTP	2'-deoxyguanosine-5'-triphosphate
DNA	deoxyribonucleic acid
DHF	dihydrofolic acid
dTMP	thymidine-5'-phosphate
dUMP	2'-deoxyuridine 5'-monophosphate
FACS	fluorescence-activated cell sorting
FAD	flavin adenine dinucleotide
FAICAR	5-formamidoimidazole-4-carboxamide ribonucleotide
GDP	guanosine-5'-diphosphate

GEO	Gene Expression Omnibus
GMP	guanosine-5'-monophosphate
GTP	guanosine-5'-triphosphate
IDO	indoleamine 2,3-dioxygenase
IMP	inosine 5'-monophosphate
IO	immuno-oncology
LTBI	latent tuberculosis infection
NADP	nicotinamide adenine dinucleotide phosphate
NEAT-seq	sequencing of nuclear protein epitope abundance, chromatin accessibility, and the transcriptome in single cells
PBMCs	peripheral blood mononuclear cells
pbRNA-seq	pseudo-bulk RNA sequencing
pbRNAseq	pseudo-bulk RNA sequencing
PD-1/PD-L1	programmed cell death protein 1/programmed cell death ligand 1
PRPP	phosphoribosyl pyrophosphate
REAP-seq	RNA expression and protein sequencing assay
RMA	robust multichip average
RNA	ribonucleic acid
RNA-seq	RNA sequencing
SAICAR	succinylaminoimidazolecarboxamide ribotide
S-AMP	succinyl AMP

SAM	S-adenosylmethionine
SAH	S-adenosylhomocysteine
scADT-seq	single-cell antibody-derived tag sequencing
scADTseq	single-cell antibody-derived tag sequencing
scRNA-seq	single-cell RNA sequencing
scRNAseq	single-cell RNA sequencing
SRA	Sequence Read Archive
TCA	tricarboxylic acid
TCGA	The Cancer Genome Atlas
TDO	tryptophan 2, 3-dioxygenase
THF	tetrahydrofolic acid
tRNA	transfer RNA
XMP	xanthosine-5'-monophosphate

CHAPTER ONE

Introduction

Multiomics Data

Biomedical data arises when subjects provide samples that are assessed by biochemical assays to quantitatively or qualitatively investigate their pathophysiological characteristics. In high-throughput biomedical research, assay measurements are often presented as rectangular matrices of features by observations at bulk (i.e., tissue) or single-cellular resolution, encompassing the epigenotype and genotype of subjects being studied.^{1,2} A subjects by features matrix of phenotypes often accompanies assay matrices to provide relevant clinical and biological context. As is shown below in Figure 1.1, together these matrices represent the seven types of multiomics data: epigenomics, genomics, transcriptomics, proteomics, metabolomics, metagenomics, and phenomics.³

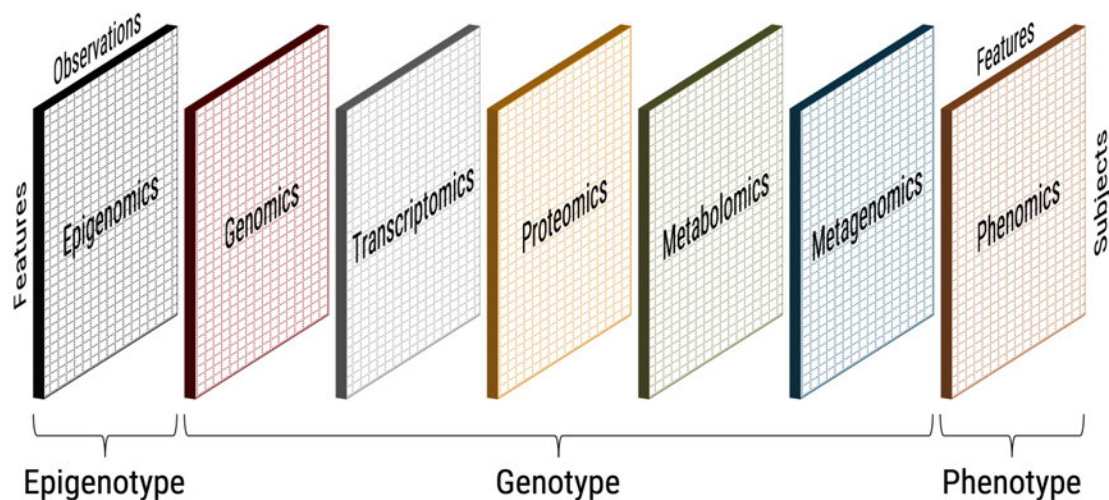


Figure 1.1. Multiomics Data Represented as Matrices. The epigenotype is represented by the epigenomics matrix, the genotype is represented by the genomics, transcriptomics, proteomics, metabolomics and metagenomics matrices, and the phenotype is represented by the phenomics matrix.

Biomedical data science seeks to ascribe meaning to these data through the lens of molecular biology, whose central dogma, whereby DNA (deoxyribonucleic acid) creates RNA (ribonucleic acid) which in turn creates proteins, was first outlined by Crick. It establishes the flow of genetic information in cells and has since been extended to include epigenetics – that is, RNA interacts with histones and DNA, both of which interact with one another, to create epigenetic silencing that is heritable.^{4,5} As of this writing, there is also evidence to suggest RNA itself may be inherited in organisms beyond yeast and plants to create epigenetic silencing.⁶⁻⁸ The diagrams of Crick and Egger et al. detailing these processes have been redrawn in Figure 1.2, along with a third diagram detailing the exchanges between proteins, metabolites, and microbes.

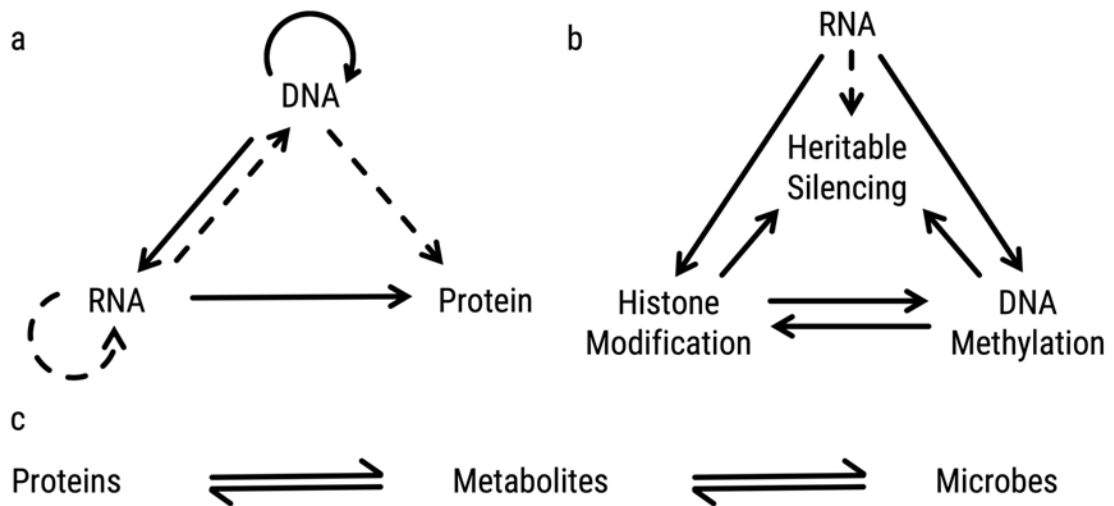


Figure 1.2. Interrelationships Between Types of Multiomics Data. (a) The central dogma of molecular biology states genetic information flows from DNA to RNA to protein – these general transfers occur in all cells and are denoted by solid lines; special transfers are denoted with dashed lines and only occur under certain circumstances. (b) Epigenetic changes confer heritable silencing when RNA, histones, and DNA interact with one another to create histone modifications and DNA methylation. There is also evidence to suggest that RNA itself may be inherited in organisms beyond yeast and plants to create epigenetic silencing; this relationship is shown with a dashed line. (c) Proteins produce and consume metabolites, metabolites are the environmental substrate for microbes, and microbes themselves produce and consume metabolites.

Considered as a whole, the hierarchy of multimodal data and its interrelationships establish the epigenotype and the genotype as the primary constituents of the phenotype. When a group of similar phenotypes or symptoms are recognized as a disease, this framework can be used to study etiology and develop diagnostic tests or therapeutic interventions.

Multimodal Data

Biomedical data are considered multimodal where either bulk tissue samples or single cells have been simultaneously profiled in more than one regard. For example, the CITE-seq (cellular indexing of transcriptomes and epitopes by sequencing) technique uses oligonucleotide labeling to profile both cell surface proteins and intracellular transcripts within the same single cells simultaneously using microfluidics.⁹ Such multimodal techniques add significant complexity to the collection, storage, management, and analysis of data, but confer advantages sufficient to merit their use. In the case of CITE-seq, the technique allows for far more in-depth analysis of gene expression specific to immune phenotypes than would otherwise be possible by FACS (fluorescence-activated cell sorting) and sequencing done on independent cell populations. Additional techniques to generate multimodal data include REAP-seq (RNA expression and protein sequencing assay), Methyl-HiC, and NEAT-seq (sequencing of nuclear protein epitope abundance, chromatin accessibility, and the transcriptome in single cells), among others – two commonalities emerge from a survey of multimodal techniques.^{10–16} One, most techniques in this domain are by necessity based on sequencing; and two, require an appropriate data structure to store, manage, and analyze data generated through their application.

Longitudinal Data

Biomedical data become longitudinal when measurements related to the same subjects are taken at multiple time points – these data can arise from either cohort studies or randomized controlled trials. The presence of multiple measurements in time increases both the volume of data generated and the complexity of analysis required to extract insights. Where cross-sectional and case-control studies can be analyzed by simple frequentist statistical tests, longitudinal study designs require marginal regression, mixed models, or other methods that properly account for the time-dependent autocorrelated nature of measurements. Yet, longitudinal data offer the greatest insights into etiology and have been instrumental in establishing modern evidence-based medicine and pharmacology; they are the cornerstone of cohort studies upon which randomized controlled trials are often based.

Mega-Analysis

Mega-analysis provides a means to combine disparate results across studies at the level of observations and enables the reuse of biomedical data for the derivation of further insights. It seeks to capture heterogeneity within the reference population to the greatest extent possible by pooling observations across studies and allows for adjustment of endogenous confounding. Joint analysis of pooled observations from multiple studies is done as if the observations came from a single study; however, care should be taken to limit technical variation across studies wherever possible – in biomedical research, this can often be accomplished through consistent *ab initio* processing of raw data and by correcting for batch effects. Finally, while greater statistical power is gained through increased sample size, mega-analysis often requires labor-intensive harmonization of clinical annotations.

Dissertation Aims

Aim One

This dissertation seeks to evaluate data structures written in the R language capable of storing and managing hierarchical multimodal multiomics data.¹⁷ It seeks to establish a novel data structure, MultimodalExperiment, as the most appropriate and efficient means to represent these data and demonstrate how challenges of alternative data structures are overcome, ease of data management is improved, and computational/storage efficiency is optimized. It does so by reviewing existing data structures, providing a schematic of and methods for the MultimodalExperiment, by demonstrating the utility of the novel data structure through practical examples, and by discussing its architecture and applications.

Aim Two

This dissertation seeks to present longitudinal metabolomics analysis from an immunology (IO) cohort study and detail serum metabolites associated with progression-free survival status among cancer patients being treated with anti PD-1/PD-L1 (programmed cell death protein 1/programmed cell death ligand 1) immunotherapies at Boston Medical Center. It seeks to draw connections between tryptophan, folic acid, lysine, and purine metabolism associated with progression-free survival status and possibly therapeutic efficacy in metabolic pathway diagrams. To thoroughly explain metabolites significantly associated with progression-free survival status, it presents a hypothesis about cytosolic calcium and the electron transport chain related to ATP production. Finally, it seeks to provide rationale as to why indoleamine 2,3-dioxygenase 1 (IDO1) inhibitors work poorly.

Aim Three

This dissertation seeks to present original software for the mega-analysis of transcriptomics of human host infected with tuberculosis. Specifically, it explains how the tuberculosis R/Bioconductor package and its data preprocessing pipeline, `tuberculosis.pipeline`, were implemented.^{18,19} It details how the tuberculosis package provides all related and eligible human samples from GEO (Gene Expression Omnibus) that did not come from cell lines, were not taken postmortem, and did not feature recombination.^{20,21} It discusses the importance of these high-quality machine learning-ready data – more than 10,000 samples in total – to improving diagnostic testing and therapeutic interventions for tuberculosis disease. Finally, this dissertation provides a summary of software development plans for ontology-driven curation of sample metadata for the tuberculosis package, and outlines challenges to machine learning mega-analysis of tuberculosis transcriptomics.

CHAPTER TWO

MultimodalExperiment

Introduction

Data structures for the storage, management, and analysis of multimodal multiomics data in the R language are limited to `MultiAssayExperiment` and `MultiDataSet`, both of which were developed just prior to the rapid growth of single-cell sequencing.^{22,23} At the time of their releases, these pieces of software fulfilled the requirements of the difficult task assigned to them and made trivial work of integrating bulk multiomics data like the TCGA (The Cancer Genome Atlas) compendium.²⁴ Yet, with the growth of microfluidics technologies integrated with sequencing at single-cell resolution, these existing data structures were stretched to their limits. Principally, the `MultiAssayExperiment` and `MultiDataSet` data structures lacked normalization of metadata consistent with biological hierarchy whereby experiments were related to a group of subjects who provided samples which were profiled by biochemical assays at bulk or single-cellular resolution.

This limitation produced a scenario that required complete duplication of metadata where annotations only changed slightly for a sample or cell. For example, if the same subject provided multiple tissue samples throughout a longitudinal study, the annotation of time point information would require complete repetition of subject metadata (age, sex, etc.) simply to denote the varying sample metadata. Without normalization, or separation of metadata into hierarchical relational tables pertaining to experiments, subjects, samples, and cells, the `MultiAssayExperiment` and `MultiDataSet` data structures were storage-inefficient in certain cases, and metadata updates required changing all duplicated values.

Through the development of the MultimodalExperiment data structure, the process of storing, managing, and analyzing multimodal multiomics data was recast in the framework of database architecture to provide a more comprehensive and efficient solution. Specifically, previous shortcomings were addressed by separating experiment, subject, sample, and cell annotations into distinct tables related to underlying biological data through maps. Maps being a series of two column tables of indices related to levels of biological hierarchy (i.e., experiments > subjects > samples > cells). Of the four maps shown in Figure 2.1, the experimentMap is unique in that it only contains a single index, experiment, as the second column, with the first column, type, providing the resolution (bulk or single-cell) of the underlying biological data. This deviation from the other maps is necessary because the experiments (rectangular feature by observation matrices of biological measurements) at the bottom of Figure 2.1 are stored as a single list and the experimentMap is used to distinguish between the two resolutions.

The purview of MultimodalExperiment, that biological data arises from experiments in which subjects provide samples that are either measured by bulk or single-cell techniques, confers computational and storage efficiency by allowing database-style join operations to form the basis of subsetting operations and by preventing the duplication of metadata at all levels of hierarchy. The propagation and harmonization of indices across annotations, maps, and experiments, also becomes trivial, and the need to manually manage maps is kept minimal. Finally, because these operations (propagate and harmonize) become computationally expensive when coordinating multiple experiments with numerous features, they have been made opt-in only for efficiency.

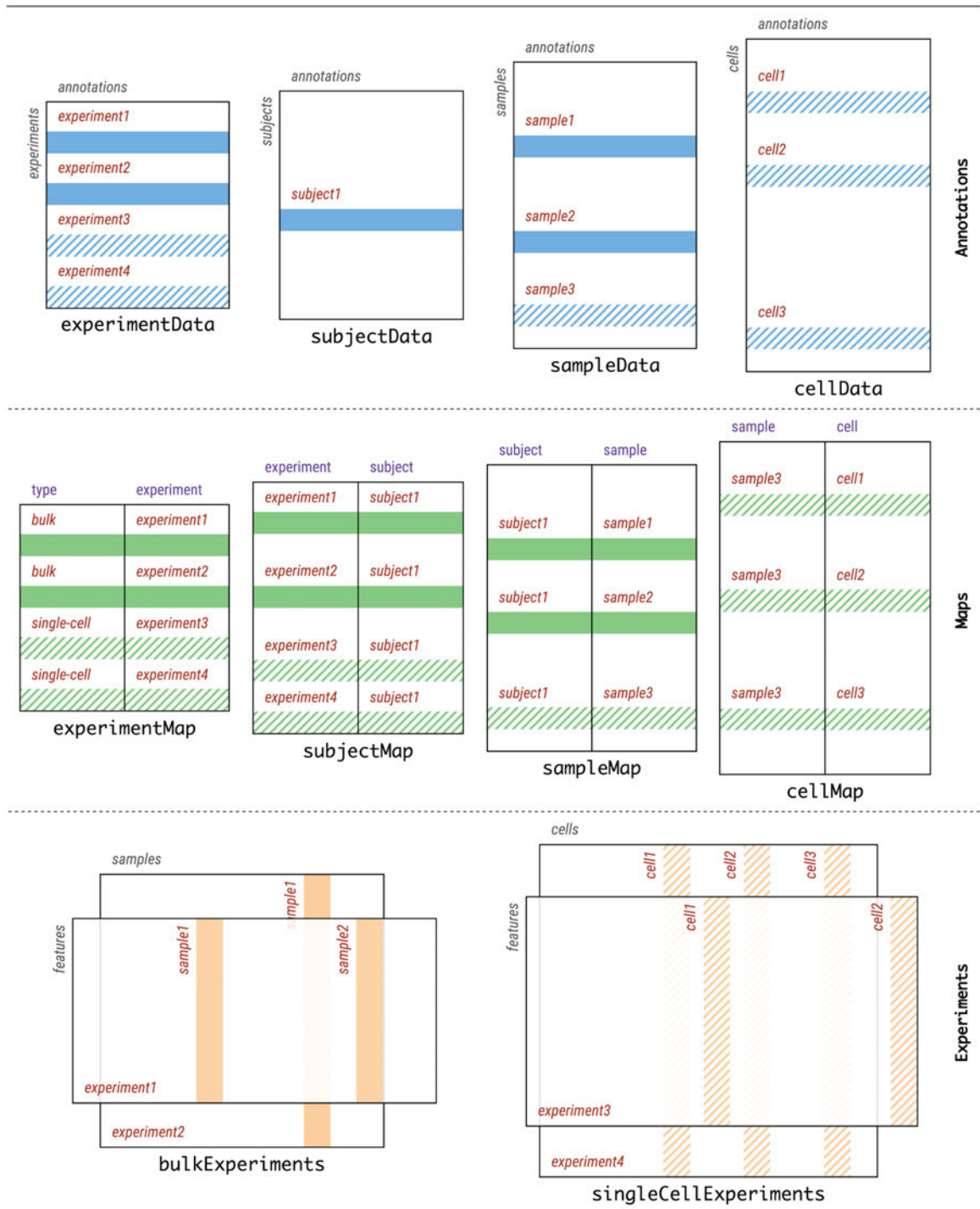


Figure 2.1. MultimodalExperiment Schematic. MultimodalExperiment features experiment, subject, sample, and cell annotations which are connected to underlying biological data (experiments) through maps. Slots and bulk or single-cell experiments can be accessed by their respectively named methods.

Methods

MultimodalExperiment is an R package that provides the MultimodalExperiment data structure as an S4 class; it has been released as open-source software through the Bioconductor project.^{1,25,26} The package also provides an application programming interface (API) of functions and methods to construct MultimodalExperiment objects and interact with them, as outlined in Table 2.1. To create a MultimodalExperiment object, the constructor function, MultimodalExperiment, is used to specify ten slots (experimentData, subjectData, sampleData, cellData, experimentMap, subjectMap, sampleMap, cellMap, experiments, and metadata) as shown in Figure 2.2. All slots except experiments and metadata are DataFrame objects – slots ending in Data represent annotations, and those ending in Map represent maps which connect annotation identifiers to one another and to underlying biological data.²⁷ The experiments slot is an ExperimentList object (the class and constructor are reexported from MultiAssayExperiment), and the metadata slot is a list as required by the Annotated S4 class from which MultimodalExperiment inherits.^{22,27}

The names of slots are also the names of methods to access them and can be used to get or set the components of a MultimodalExperiment object. For convenience, two methods, joinAnnotations and joinMaps, to perform outer joins of annotations and maps have been provided. Elements of the ExperimentList, experiments, can be accessed by index, name, or resolution (bulk or single-cell) using the suite of experiments methods; the experimentNames method can be used to get or set the names of experiments. Interaction with row/column names and square bracket subsetting work as they do generally in the R language, but rather use list inputs/outputs to manage multiple experiments simultaneously.

Constructors	
MultimodalExperiment	create a MultimodalExperiment object
ExperimentList	create an ExperimentList object
Slots	
experimentData	get or set experimentData (experiment annotations)
subjectData	get or set subjectData (subject annotations)
sampleData	get or set sampleData (sample annotations)
cellData	get or set cellData (cell annotations)
experimentMap	get or set experimentMap (experiment -> type map)
subjectMap	get or set subjectMap (subject -> experiment map)
sampleMap	get or set sampleMap (sample -> subject map)
cellMap	get or set cellMap (cell -> sample map)
experiments	get or set experiments
metadata	get or set metadata
Annotations	
joinAnnotations	join experimentData, subjectData, sampleData, and cellData
Maps	
joinMaps	join experimentMap, subjectMap, sampleMap, and cellMap
Experiments	
experiment(ME, i)	get or set experiments element by index
experiment(ME, "name")	get or set experiments element by name
bulkExperiments	get or set experiments element(s) where type == "bulk"
singleCellExperiments	get or set experiments element(s) where type == "single-cell"
Names	
rownames	get or set rownames of experiments element(s)
colnames	get or set colnames of experiments element(s)
experimentNames	get or set names of experiments
Subsetting	
ME[i, j]	subset rows and/or columns of experiments
ME[i,]	i: list, List, LogicalList, IntegerList, CharacterList
ME[, j]	j: list, List, LogicalList, IntegerList, CharacterList
Coordination	
propagate	propagate experiment, subject, sample, and cell indices across all tables
harmonize	harmonize experiment, subject, sample, and cell indices across all tables

Table 2.1. MultimodalExperiment API Reference. The MultimodalExperiment API features two constructor functions and methods for slots, annotations, maps, experiments, names, subsetting and coordination. The ExperimentList constructor is reexported from MultiAssayExperiment.

```

MultimodalExperiment(
  experimentData = DataFrame(),
  subjectData = DataFrame(),
  sampleData = DataFrame(),
  cellData = DataFrame(),
  experimentMap = DataFrame(
    type = character(),
    experiment = character()
  ),
  subjectMap = DataFrame(
    experiment = character(),
    subject = character()
  ),
  sampleMap = DataFrame(
    subject = character(),
    sample = character()
  ),
  cellMap = DataFrame(
    sample = character(),
    cell = character()
  ),
  experiments = ExperimentList(),
  metadata = list()
)

```

Figure 2.2. MultimodalExperiment Constructor. A MultimodalExperiment object is constructed by specifying its ten slots: experimentData, subjectData, sampleData, cellData, experimentMap, subjectMap, sampleMap, cellMap, experiments, and metadata. Slots ending in Map are maps of identifiers that connect experiment, subject, sample, and cell annotations (those ending in Data) to underlying biological data (feature by observation matrices) stored in an ExperimentList. The metadata slot can be used to store any metadata that is beyond the scope of MultimodalExperiment.

Two methods, propagate and harmonize, for the coordination of experiment, subject, sample, and cell indices across all tables are the backbone of the MultimodalExperiment API. These verbs are aware of the hierarchical structure of data defined by MultimodalExperiment and use database-style joins to efficiently add the set union of indices to annotations and maps (propagate), or ensure annotations, maps, and experiments only include the set intersection of indices (harmonize). As opt-in only verbs, they allow MultimodalExperiment components to be modified with greater computational efficiency.

Results

To demonstrate the utility of MultimodalExperiment, a vignette of object construction and manipulation using the *PBMCs of a Healthy Donor - 5' Gene Expression with a Panel of TotalSeq™-C Antibodies* dataset from 10x Genomics is provided here.²⁸ The dataset features single-cell RNA sequencing (scRNA-seq) and single-cell antibody-derived tag sequencing (scADT-seq) data that were generated simultaneously (i.e., CITE-seq data); the scRNA-seq data has been aggregated to bulk resolution using scuttle and is presented as pseudo-bulk RNA sequencing (pbRNA-seq) data.²⁹ A few rows/columns of each matrix are shown in Figure 2.3. The dataset does not have any subject metadata, but the PBMCs (peripheral blood mononuclear cells) are known to come from a single healthy donor.

```
pbRNAseq[1:4, 1:1, drop = FALSE]

##      SAMPLE-1
## A1BG      1020
## A1CF         0
## AAAS       413
## AACS       117

scADTseq[1:4, 1:4, drop = FALSE]

##      AAACCTGAGAGCAATT AAACCTGAGGCTCTTA AAACCTGAGTGAACGC AAACCTGCAAACGCGA
## CD3                225                1064                1833                18
## CD4                  0                  0                  0                  1
## CD14                 0                  0                  0                  3
## CD15                6890                29                  42                 47

scRNAseq[1:4, 1:4, drop = FALSE]

##      AAACCTGAGAGCAATT AAACCTGAGGCTCTTA AAACCTGAGTGAACGC AAACCTGCAAACGCGA
## A1BG                   1                   0                   0                   0
## A1CF                   0                   0                   0                   0
## AAAS                   0                   0                   0                   0
## AACS                   0                   0                   0                   0
```

Figure 2.3. Rows/Columns of Vignette Data. Single-cell RNA sequencing (scRNAseq) data have been made into pseudo-bulk RNA sequencing (pbRNAseq) data – the pbRNAseq features (genes) by observations (samples) matrix has only a single column because the scRNAseq data are known to come from a single subject. The column names of single-cell antibody-derived tag sequencing (scADTseq) and scRNAseq data match, as both modalities were generated simultaneously.

It is possible to construct a `MultimodalExperiment` by specifying each of its ten slots, but this practice is generally discouraged because it is difficult; it is far easier to call the constructor function without arguments and then manipulate the resulting empty object, as shown in figure 2.4. The insertion of an `ExperimentList` of named matrix or matrix-like (e.g., `SummarizedExperiment`, `SingleCellExperiment`, etc.) elements into the `experiments` slot will follow – features are the rows of matrices are and observations (samples or cells) are the columns.^{30,31} Then, to review which indices are present in the four maps (i.e., the `experimentMap`, `subjectMap`, `sampleMap`, and `cellMap`) their get methods can be called.

```
ME <-
  MultimodalExperiment()

experiments(ME) <-
  ExperimentList(
    pbRNAseq = pbRNAseq,
    scADTseq = scADTseq,
    scRNAseq = scRNAseq
  )

experimentMap(ME)

## DataFrame with 0 rows and 2 columns

subjectMap(ME)

## DataFrame with 0 rows and 2 columns

sampleMap(ME)

## DataFrame with 0 rows and 2 columns

cellMap(ME)

## DataFrame with 0 rows and 2 columns
```

Figure 2.4. Construction of a `MultimodalExperiment`. An empty `MultimodalExperiment` is created by calling the constructor without arguments and assigned to the variable `ME`. The `experiments` slot of `ME` is set as an `ExperimentList` of three named matrices. Then, the four maps of `ME` are reviewed by calling their get methods to inspect which indices are present.

Maps will initially have zero rows because indices have not been propagated from experiments, calling the propagate method and reviewing the maps again will reveal experiment names are present in the experimentMap and the subjectMap, as is shown in Figure 2.5. The sampleMap and cellMap still have zero rows because the type (bulk or single-cell) of experiments has not been specified in the experimentMap.

```
ME <-
  propagate(ME)

experimentMap(ME)

## DataFrame with 3 rows and 2 columns
##      type  experiment
## <character> <character>
## 1      NA    pbRNAseq
## 2      NA    scADTseq
## 3      NA    scRNAseq

subjectMap(ME)

## DataFrame with 3 rows and 2 columns
##  experiment  subject
## <character> <character>
## 1  pbRNAseq      NA
## 2  scADTseq      NA
## 3  scRNAseq      NA

sampleMap(ME)

## DataFrame with 0 rows and 2 columns

cellMap(ME)

## DataFrame with 0 rows and 2 columns
```

Figure 2.5. Maps After Initial Propagation. The ME object is updated by calling the propagate method to add the union of relevant indices to each table. The method is aware of the hierarchical nature of data and only inserts indices that can be determined to belong to a specific level of hierarchy – here experiment names were added to the experimentMap and the subjectMap.

Once experiment names are present in the experimentMap, the type (bulk or single-cell) of experiments can be specified, as shown in Figure 2.6. When propagate is called again, the sampleMap and cellMap will contain the indices of observations from experiments.

```

experimentMap(ME)[["type"]] <-
  c("bulk", "single-cell", "single-cell")

ME <-
  propagate(ME)

experimentMap(ME)

## DataFrame with 3 rows and 2 columns
##      type  experiment
## <character> <character>
## 1      bulk    pbRNAseq
## 2 single-cell  scADTseq
## 3 single-cell  scRNAseq

subjectMap(ME)

## DataFrame with 3 rows and 2 columns
##      experiment  subject
## <character> <character>
## 1      pbRNAseq      NA
## 2      scADTseq      NA
## 3      scRNAseq      NA

sampleMap(ME)

## DataFrame with 1 row and 2 columns
##      subject  sample
## <character> <character>
## 1      NA     SAMPLE-1

cellMap(ME)

## DataFrame with 5000 rows and 2 columns
##      sample      cell
## <character> <character>
## 1      NA AAACCTGAGAGCAATT
## 2      NA AAACCTGAGGCTCTTA
## 3      NA AAACCTGAGTGAACGC
## 4      NA AAACCTGCAAACGCGA
## 5      NA AAACCTGCAGCGTTTCG
## ...      ...      ...
## 4996     NA TTTGTCAGTTGGACCC
## 4997     NA TTTGTCAGTTGGAGGT
## 4998     NA TTTGTCAGTTTAGCTG
## 4999     NA TTTGTCATCATGGTCA
## 5000     NA TTTGTCATCTCGTTTA

```

Figure 2.6. Maps After Type Specification. The type column of the experimentMap is specified and the propagate method is call on the ME object. Review of the four maps reveals that indices of samples are now present in the sampleMap, and indices of cells are now present in the cellMap.

Next, because MultimodalExperiment considers single cells to have come from a sample, the cell to sample relationship should be specified in the cellMap. In the example being presented here, the sample index of the only pbRNAseq sample, SAMPLE-1, will be reused as the sample index for cells in the cellMap – this is appropriate because the pbRNAseq data have been constructed from the scRNAseq data. When the cellMap is inspected, the cell to sample relationship can be seen as shown in Figure 2.7.

```
cellMap(ME)[["sample"]] <-  
  "SAMPLE-1"  
  
cellMap(ME)  
  
## DataFrame with 5000 rows and 2 columns  
##      sample      cell  
##    <character> <character>  
## 1     SAMPLE-1 AAACCTGAGAGCAATT  
## 2     SAMPLE-1 AAACCTGAGGCTCTTA  
## 3     SAMPLE-1 AAACCTGAGTGAACGC  
## 4     SAMPLE-1 AAACCTGCAAACGCGA  
## 5     SAMPLE-1 AAACCTGCAGCGTTCCG  
## ...      ...      ...  
## 4996    SAMPLE-1 TTTGTCAGTTGGACCC  
## 4997    SAMPLE-1 TTTGTCAGTTGGAGGT  
## 4998    SAMPLE-1 TTTGTCAGTTTAGCTG  
## 4999    SAMPLE-1 TTTGTCATCATGGTCA  
## 5000    SAMPLE-1 TTTGTCATCTCGTTTA
```

Figure 2.7. The cellMap After Cell to Sample Specification. The sample column of the cellMap is specified and inspection of the cellMap shows cells are now considered to come from a sample.

Finally, SAMPLE-1 is specified as belonging to SUBJECT-1 in the sampleMap; this completes the structure of hierarchy defined by MultimodalExperiment (i.e., experiments > subjects > samples > cells) and all relationships can now be established by calling propagate again as is shown in Figure 2.8.

```

sampleMap(ME)[["subject"]] <-
  "SUBJECT-1"

ME <-
  propagate(ME)

experimentMap(ME)

## DataFrame with 3 rows and 2 columns
##      type  experiment
## <character> <character>
## 1      bulk    pbRNAseq
## 2 single-cell  scADTseq
## 3 single-cell  scRNAseq

subjectMap(ME)

## DataFrame with 6 rows and 2 columns
##      experiment  subject
## <character> <character>
## 1      pbRNAseq  SUBJECT-1
## 2      scADTseq  SUBJECT-1
## 3      scRNAseq  SUBJECT-1
## 4      pbRNAseq           NA
## 5      scADTseq           NA
## 6      scRNAseq           NA

sampleMap(ME)

## DataFrame with 1 row and 2 columns
##      subject  sample
## <character> <character>
## 1  SUBJECT-1  SAMPLE-1

cellMap(ME)

## DataFrame with 5000 rows and 2 columns
##      sample      cell
## <character> <character>
## 1      SAMPLE-1 AAACCTGAGAGCAATT
## 2      SAMPLE-1 AAACCTGAGGCTCTTA
## 3      SAMPLE-1 AAACCTGAGTGAACGC
## 4      SAMPLE-1 AAACCTGCAAACGCGA
## 5      SAMPLE-1 AAACCTGCAGCGTTTCG
## ...      ...      ...
## 4996      SAMPLE-1 TTTGTCAGTTGGACCC
## 4997      SAMPLE-1 TTTGTCAGTTGGAGGT
## 4998      SAMPLE-1 TTTGTCAGTTTAGCTG
## 4999      SAMPLE-1 TTTGTCATCATGGTCA
## 5000      SAMPLE-1 TTTGTCATCTCGTTTA

```

Figure 2.8. Maps After Sample to Subject Specification. The sampleMap is updated to establish a relationship between SAMPLE-1 and SUBJECT-1, propagate is called, and maps are inspected.

At first glance, the results of calling `propagate` in Figure 2.8 might appear incorrect because the `subjectMap` has values in the `experiment` column which have NA in the `subject` column, even where all relationships are known and have been specified. This is not the case but is rather the intended behavior of `propagate` – the method will never remove indices from any table once they have been added. To remove the NA values in the `subjectMap` that were created in Figure 2.5, the `harmonize` method is called in Figure 2.9 after the experiment to subject relationship was established in Figure 2.8. The `harmonize` method ensures only indices that exist in experiments and are part of the intersection of indices across all tables will remain in the resulting `MultimodalExperiment` object, as shown in Figure 2.9.

```

ME <-
  harmonize(ME)

experimentMap(ME)

## DataFrame with 3 rows and 2 columns
##      type experiment
## <character> <character>
## 1 single-cell    scADTseq
## 2 single-cell    scRNAseq
## 3      bulk      pbRNAseq

subjectMap(ME)

## DataFrame with 3 rows and 2 columns
##  experiment    subject
## <character> <character>
## 1    pbRNAseq  SUBJECT-1
## 2    scADTseq  SUBJECT-1
## 3    scRNAseq  SUBJECT-1

sampleMap(ME)

## DataFrame with 1 row and 2 columns
##  subject    sample
## <character> <character>
## 1  SUBJECT-1  SAMPLE-1

cellMap(ME)

## DataFrame with 5000 rows and 2 columns
##      sample    cell
## <character> <character>
## 1    SAMPLE-1 AAACCTGAGAGCAATT
## 2    SAMPLE-1 AAACCTGAGGCTCTTA
## 3    SAMPLE-1 AAACCTGAGTGAACGC
## 4    SAMPLE-1 AAACCTGCAAACGCGA
## 5    SAMPLE-1 AAACCTGCAGCGTTCC
## ...      ...      ...
## 4996  SAMPLE-1 TTTGTCAGTTGGACCC
## 4997  SAMPLE-1 TTTGTCAGTTGGAGGT
## 4998  SAMPLE-1 TTTGTCAGTTTAGCTG
## 4999  SAMPLE-1 TTTGTCATCATGGTCA
## 5000  SAMPLE-1 TTTGTCATCTCGTTTA

```

Figure 2.9. Maps After Harmonization. The harmonize method is called on the ME object and the extraneous rows of the subjectMap from Figure 2.8 are removed. Inspection of maps demonstrates the absence of duplicated indices – specifically, the cell column of the cellMap is related to both cellData row names and the column names of scRNAseq and scADTseq data.

To gain better understanding of the relational nature of indices, it can be helpful to see the maps as a single non-normalized table; the `joinMaps` method provides this functionality. When called on the ME object constructed here, a DataFrame with 10,001 rows is returned – the number of rows equals the number of observations across all experiments in this case.

```
joinMaps(ME)

## DataFrame with 10001 rows and 5 columns
##           type experiment      subject      sample      cell
##    <character> <character> <character> <character> <character>
## 1      bulk      pbRNAseq  SUBJECT-1  SAMPLE-1      NA
## 2 single-cell    scADTseq  SUBJECT-1  SAMPLE-1 AAACCTGAGAGCAATT
## 3 single-cell    scADTseq  SUBJECT-1  SAMPLE-1 AAACCTGAGGCTCTTA
## 4 single-cell    scADTseq  SUBJECT-1  SAMPLE-1 AAACCTGAGTGAACGC
## 5 single-cell    scADTseq  SUBJECT-1  SAMPLE-1 AAACCTGCAAACGCGA
## ...      ...      ...      ...      ...      ...
## 9997 single-cell  scrNAseq  SUBJECT-1  SAMPLE-1 TTTGTCAGTTGGACCC
## 9998 single-cell  scrNAseq  SUBJECT-1  SAMPLE-1 TTTGTCAGTTGGAGGT
## 9999 single-cell  scrNAseq  SUBJECT-1  SAMPLE-1 TTTGTCAGTTAGCTG
## 10000 single-cell scrNAseq  SUBJECT-1  SAMPLE-1 TTTGTCATCATGGTCA
## 10001 single-cell scrNAseq  SUBJECT-1  SAMPLE-1 TTTGTCATCTCGTTTA
```

Figure 2.10. Outer Join of All Maps. When called on the ME object constructed here, the `joinMaps` method returns a non-normalized DataFrame representing the outer join of all maps.

When the ME object itself is called, the `MultimodalExperiment` `show` method is used to return a summary of relevant information about the object. As is shown in Figure 2.11, because no annotations have been added thus far, the `experimentData`, `subjectData`, `sampleData`, and `cellData` slots all contain `DataFrame` objects with zero columns.

```
ME
## MultimodalExperiment with 1 bulk and 2 single-cell experiment(s).
##
## experimentData: DataFrame with 3 row(s) and 0 column(s).
##
## subjectData: DataFrame with 1 row(s) and 0 column(s).
##
## sampleData: DataFrame with 1 row(s) and 0 column(s).
##
## cellData: DataFrame with 5000 row(s) and 0 column(s).
##
## bulkExperiments: ExperimentList with 1 bulk experiment(s).
## [1] pbRNAseq: matrix with 3000 row(s) and 1 column(s).
##
## singleCellExperiments: ExperimentList with 2 single-cell experiment(s).
## [1] scADTseq: matrix with 8 row(s) and 5000 column(s).
## [2] scRNAseq: matrix with 3000 row(s) and 5000 column(s).
##
## Need help? Try browseVignettes("MultimodalExperiment").
## Publishing? Cite with citation("MultimodalExperiment").
```

Figure 2.11. The ME Object Without Annotations. The `MultimodalExperiment` `show` method is used to display the ME object which has no annotations in the `experimentData`, `subjectData`, `sampleData`, or `cellData` slots. The row names represent indices at the relevant level of hierarchy, but without annotations the `DataFrame` objects all have zero columns.

What little metadata is provided with the 10x Genomics dataset can be organized into annotations at the various levels of hierarchy. Specifically, the scRNA-seq and scADT-seq were published on November 19, 2018, and the PBMCs are known to come from a single healthy subject. To serve as cell annotations, a naive definition of immune phenotypes can be constructed from scADT-seq data based on cell surface marker counts greater than zero. (A more thorough definition would be necessary for any purpose beyond demonstration). The creation of experiment, subject, sample, and cell annotations is shown in Figure 2.12.

```

experimentData(ME)[["published"]] <-
  c(NA_character_, "2018-11-19", "2018-11-19") |>
  as.Date()

subjectData(ME)[["condition"]] <-
  as.character("healthy")

sampleData(ME)[["sampleType"]] <-
  as.character("peripheral blood mononuclear cells")

cellType <- function(x) {
  if (x[["CD4"]] > 0L) {
    return("T Cell")
  }

  if (x[["CD14"]] > 0L) {
    return("Monocyte")
  }

  if (x[["CD19"]] > 0L) {
    return("B Cell")
  }

  if (x[["CD56"]] > 0L) {
    return("NK Cell")
  }

  NA_character_
}

cellData(ME)[["cellType"]] <-
  experiment(ME, "scADTseq") |>
  apply(2L, cellType)

```

Figure 2.12. Creations of ME Annotations. Experiment, subject, sample, and cell annotations are created from limited metadata; the function defines cell types based on cell surface markers.

When called again, the ME object will use the `MultimodalExperiment show` method to display the annotations created in Figure 2.12. The display of ME in Figure 2.13 shows only the first and last rows of each `DataFrame` of annotations, but this behavior can be changed by setting the `showRowsT` and `showRowB` options in R.

```
ME
## MultimodalExperiment with 1 bulk and 2 single-cell experiment(s).
##
## experimentData: DataFrame with 3 row(s) and 1 column(s).
##           published
##           <Date>
## pbRNAseq      NA
## ...           ...
## scRNAseq 2018-11-19
##
## subjectData: DataFrame with 1 row(s) and 1 column(s).
##           condition
##           <character>
## SUBJECT-1    healthy
##
## sampleData: DataFrame with 1 row(s) and 1 column(s).
##           sampleType
##           <character>
## SAMPLE-1 peripheral blood mononuclear cells
##
## cellData: DataFrame with 5000 row(s) and 1 column(s).
##           cellType
##           <character>
## AAACCTGAGAGCAATT    B Cell
## ...                 ...
## TTTGTCATCTCGTTA    NK Cell
##
## bulkExperiments: ExperimentList with 1 bulk experiment(s).
## [1] pbRNAseq: matrix with 3000 row(s) and 1 column(s).
##
## singleCellExperiments: ExperimentList with 2 single-cell experiment(s).
## [1] scADTseq: matrix with 8 row(s) and 5000 column(s).
## [2] scRNAseq: matrix with 3000 row(s) and 5000 column(s).
##
## Need help? Try browseVignettes("MultimodalExperiment").
## Publishing? Cite with citation("MultimodalExperiment").
```

Figure 2.13. The ME Object with Annotations. The `MultimodalExperiment show` method is used to display the ME object with experiment, subject, sample, and cell annotations that were created in by the steps detailed in Figure 2.12.

Discussion

Where bulk and single-cell techniques have been used to profile a set of samples and cells from a group of subjects, `MultimodalExperiment` provides an appropriate and efficient means to represent resulting data. While this experimental design is relatively uncommon as of this writing, such studies do exist and are expected to grow in number because of the resolution that single-cell sequencing provides.³² Yet, the `MultimodalExperiment` data structure is not limited to this scenario alone and can be used to represent any type of multimodal data that conform to the constraints of biological hierarchy it defines. As compared to managing modalities independently, the `MultimodalExperiment` API simplifies the tasks of storing, managing, and analyzing multimodal data using the R language. The data structure addresses shortcomings that have come to light with the rapid growth of single-cell sequencing and derives storage and computation benefits from normalization of annotations.

The data structure lends itself to exceptionally fast propagation and harmonization of indices given its architecture choices – internally, normalization allows for database-style joins to be used for these operations and they are written using base R and S4 vectors alone, which themselves are written largely in C++ for computation efficiency.^{17,27} Through the development of `MultimodalExperiment`, a novel type of database join was needed and created – the “hemi join” is an outer join of X and Y in which only the columns appearing in X are kept. This is similar to, but distinct from, `dplyr`’s “semi-join” – an inner join of X and Y in which only the columns appearing in X are kept.³³ The computational innovations of `MultimodalExperiment` are available as open-source software through Bioconductor.

Lastly, there is one final use case for MultimodalExperiment that merits a brief discussion: the storage and management of data resulting from the deconvolution and aggregation of RNA sequencing data. Methods to perform these tasks are the domain of other software, but resulting data suggest MultimodalExperiment.^{29,34} Where bulk RNA-seq data is deconvoluted to single-cell resolution for comparison to scRNA-seq data or scRNA-seq data is aggregated for comparison to bulk RNA-seq data, the MultimodalExperiment hierarchy is realized. Methods to project resulting data of the same resolution into shared manifold space for analysis could then be written around MultimodalExperiment.

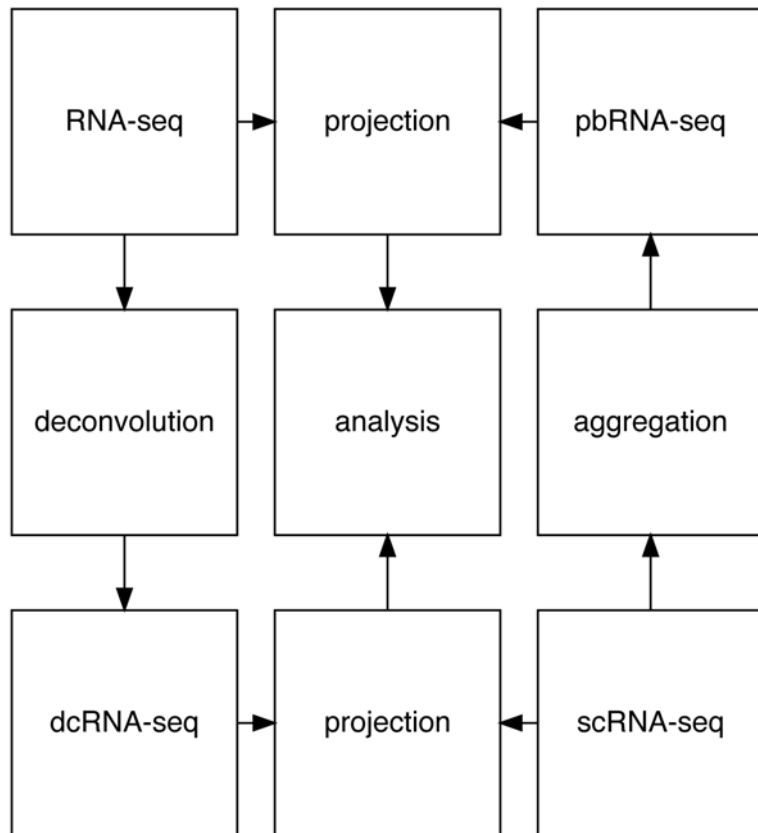


Figure 2.14. Potential RNA-seq Applications. From the top left downward, bulk RNA-seq data is deconvoluted into dcRNA-seq (deconvoluted RNA-seq) data; from the bottom right upward, scRNA-seq data is aggregated into pbRNA-seq data; from the corners inward and to the center, data of the same resolution are projected into shared manifold space for analysis.

CHAPTER THREE

Metabolomics of IO Therapy

Introduction

Prior to the development of immune checkpoint inhibitors, neoplastic lesions were generally treated by surgical interventions, radiotherapy, chemotherapy, or by other targeted therapies such as nanoparticle treatments. With the advent and approval of immuno-oncology (IO) therapy agents in the past decade, the treatment of some cancers has changed considerably to include them.³⁵ As compared to other types of therapy, IO therapy features a substantial divergence in how it seeks to bring about the death of tumor cells. Rather than intervening by an exogenous means, IO therapy works by soliciting immune responses to kill tumor cells and create remission. Where tumor proliferation has led to immune evasion, IO therapy can reactivate T lymphocytes to fight and arrest the growth of cancer cells.

A specific type of IO therapy, anti-programmed cell death protein 1/programmed cell death ligand 1 (PD-1/PD-L1) IO therapy, will be the focus here. This therapy targets and seeks to prevent the binding of PD-L1 to PD-1, a transmembrane protein that is expressed on the surface of T/B lymphocytes, natural killer cells, dendritic cells, macrophages, and monocytes.³⁶ When bound to PD-1 on T lymphocytes, PD-L1 decreases their activation, proliferation, and survival by acting as an immune checkpoint inhibitor or “brake” that diminishes the adaptive immune response.³⁷ In a number of cancer types, tumor cells are known to express high levels of PD-L1 to inhibit T cells, evade an immune response, and create a tumor microenvironment hospitable to proliferation.³⁸

The tumor microenvironment is also known to elicit immune privilege and suppress T cell responses when levels of the indoleamine 2,3-dioxygenase (IDO) 1 enzyme are elevated from the catabolism of tryptophan.³⁹⁻⁴⁷ Thus the study of tryptophan metabolism in cancer holds great potential for the improvement and development of therapeutics.^{40,48,49} Here longitudinal metabolomics analysis of a cohort of cancer patients undergoing treatment with anti PD-1/PD-L1 IO therapy at Boston Medical Center will be presented. Specifics of tryptophan metabolism have been extensively detailed elsewhere, but will be highlighted here in metabolic pathway diagrams that synthesize current knowledge.⁴⁹⁻⁵¹

Methods

Cancer patients diagnosed with all types of neoplasms who were being treated with anti PD-1/PD-L1 IO therapy in clinic at Boston Medical Center were eligible to enroll in this prospective cohort study. For patients who agreed to participate, consent was obtained during their baseline visit before beginning IO therapy under Boston University IRB #H-38024. Each participant provided a serum sample at baseline and during follow up at a non-specific time point, both samples were frozen until sample collection for the cohort was complete. An intake interview to ascertain demographic and clinical history was also conducted at baseline. Then during the course of 24 month follow up, tumor growth was monitored by a radiologist who measured tumor size following RECIST-like criteria to determine progression status.⁵² The dates of progression, survival, and censoring events were recorded along with treatment initiation and completion dates to enable survival analyses. Where study participants were lost to follow up, the date of their last appointment in clinic was used as the date of their censoring event.

When sample collection for the entire cohort was complete, serum samples were sent to the Boston University Chemistry Department Chemical Instrumentation Center for targeted metabolomics. Concentrations of 232 metabolites related to and including tryptophan were acquired by triple quadrupole mass spectrometry using electrospray flow injection in the positive mode – an Agilent 1100 Series instrument was used for high performance liquid chromatograph, and a Sciex API 4000 instrument was used for mass spectrometry. Samples were run in four separate batches due to instrument constraints; concentrations of each metabolite in each sample were returned as a spreadsheet.

All statistical analysis was performed in R and began with the importing of data using the readr package.^{17,53} Data cleaning was done using the dplyr, forcats, magrittr, purrr, tibble, tidyr, and withr packages before creating a SummarizedExperiment.^{30,33,54–59} The generalized log transformation defined by the MetaboAnalystR package was applied to metabolomics data prior to batch correction using ComBat.^{60,61} All primary (overall survival and progression-free survival) and secondary (all other measures) outcomes were evaluated for consistency of findings with published literature. The linear-mixed effect models presented in results were constructed using the lmerTest package; one for each metabolite with concentration (μmol) as the Y variable and the time after treatment initiation (days) as the X variable.⁶² An interaction term between the X variable and binary progression-free survival status was added to the models for stratification; additional covariates were adjusted for as outlined in the results; intercepts per subject were random.

Results

During the three-year study period from 2019 to 2021, 103 participants provided 155 samples and more than 100 person-years of follow up in total. Given the nature of the study, participants were older with a median age of 67 (IQR 57 – 71); the distribution of sex was relatively even, with slightly more male participants (54%). More than one-quarter of the study population (28%) was known to have type two diabetes mellitus. Diagnosed cancer type was recorded for each participant, but certain malignancies had too few samples for meaningful adjustment; therefore, cancer types were aggregated into cancer categories to ensure sufficient bin sizes. Of these categories, lung non-small cell carcinoma represented 20% of participants, adult liver carcinoma represented 16% of participants, melanoma represented 13% of patients, and malignant head/neck neoplasms represented 11% of participants. The some other cancer type category represented the remainder of participants (41%) and contained diverse carcinomas and sarcomas.

Race was recorded in a manner consistent with the US Census for comparability, and no participants reported being of more than one race. The study population was 40% White, 29% Black or African American, 17% Some Other Race, and 15% Asian – for comparison, the single race population of the United States is 69% White, 14% Black or African American, 9% Some Other Race, and 7% Asian.⁶³ While the study population was drawn by non-probability sampling, it yielded the desirable property of under sampling the White racial group and over-sampling minority racial groups. In theory, this removes bias to a degree by shrinking standard errors of minority race covariates in modeling to be more consistent with the standard error of the majority race group.

Characteristic	N = 103
Age, Median (IQR)	67 (59 – 71)
Sex, n (%)	
Male	56 (54)
Female	47 (46)
Race, n (%)	
White	41 (40)
Black or African American	30 (29)
Some Other Race	17 (17)
Asian	15 (15)
Diabetic Status, n (%)	
Non-Diabetic	74 (72)
Diabetic	29 (28)
Cancer Category, n (%)	
Some Other Cancer Type	42 (41)
Lung Non-Small Cell Carcinoma	21 (20)
Adult Liver Carcinoma	16 (16)
Melanoma	13 (13)
Malignant Head/Neck Neoplasm	11 (11)

Table 3.1. Demographic and Clinical Characteristics. Select characteristics of the study cohort; see the preceding paragraphs for a textual summary of these characteristics.

The demographic and clinical characteristics of the cohort are reported in Table 3.1, yet the table does not report anti PD-1/PD-L1 IO therapy agents. This is intentional, as the purpose of the study is a descriptive analysis of metabolomics related to progress status, rather than a comparison of therapeutic efficacy of various anti PD-1/PD-L1 IO therapy agents. Which anti PD-1/PD-L1 IO therapy agent participants were treated with (atezolizumab, cemiplimab, durvalumab, nivolumab, or pembrolizumab) was based upon diagnosis and indication. All the agents are known to work on the PD-1/PD-L1 axis, and the goal here is to illustrate how tryptophan and other metabolites are associated with progress status and therapeutic efficacy regardless of which is being administered.

Measure	N = 103
Overall Survival, Days, Median (95% CI)	–
Progression-Free Survival, Days, Median (95% CI)	223 (166 to 493)
Response Status, n (%)	
Complete/Partial Response	45 (45)
Stable Disease	21 (21)
Progressive Disease	24 (24)
Deceased	11 (11)
Unknown	2 (–)
Combined Status, n (%)	
Stable/Responsive Disease	66 (65)
Deceased/Progressive Disease	35 (35)
Unknown	2 (–)
Survival Status, n (%)	
Survived	64 (62)
Deceased	39 (38)
Progress Status, n (%)	
Responded	48 (47)
Progressed	55 (53)

Table 3.2. Primary and Secondary Outcome Measures. Overall and progression-free survival outcomes represent primary measures, and all other measures represent secondary outcomes. Response status is the best clinical response achieved during treatment, established by tumor measurements. Combined status is a condensation of response status into a binary variable, as are survival and progression statuses of overall survival and progression-free survival, respectively.

Primary and secondary outcomes are reported in Table 3.2 and include overall/progression-free survival (primary measures), response status, combined status, survival status, and progression status (secondary measures). Median overall survival could not be calculated because less than half of the cohort (38%) was deceased after 24 months of follow up, but median progression-free survival time was 223 days (95% CI, 166 to 493 days). Response status was ascertained through the measurement of tumor dimension (see Methods) and represented the clinical best response achieved during treatment. In 45% of participants tumor size decreased (complete/partial response), tumors exhibited neither growth nor

shrinkage (stable disease) in 21% of participants, tumor dimensions increased in 24% of participants (progressive disease), and 11% of participants were deceased before response status could be determined (deceased) – response status was unknown for 2 participants.

Three outcome measures in Table 3.2 (combined status, survival status, and progress status) represent other variables condensed into dichotomous binary variables. These variables were created because they offer greater possibilities of analyses, increase statistical power, and produce more tractable classification problems. Combined status was created by the condensation of response status – its levels, stable/responsive disease and deceased/progressive disease, represent 65 and 35% of participants, respectively. Survival status represents the eventual survival outcome (i.e., event) that was observed in 24 months of follow up – 62% of participants survived until the end of the observation period and 38% were deceased. Progress status is the same, but for progression-free survival – 47% of participants responded (i.e., did not progress) and 53% progressed (i.e., showed signs of tumor growth). For the creation of survival and progress status variables, censorship or loss to follow up events were considered as survival and response events.

Survival analysis of the primary outcomes was attempted but ultimately abandoned because the time-dependent high-dimensional nature of models necessitated by the longitudinal metabolomics data cannot be created as of this writing. Specifically, the survival package features infrastructure for Cox proportional hazards models and permits time-varying covariates as would be needed; yet, maximum likelihood estimation by Newton-Raphson iterations does not produce convergence no matter how parameters are specified – even where a vast number of iterations are undertaken.^{64,65} Thus, estimates of

betas for many covariates are infinite while the rest are inaccurate. The Machine Learning in R suite of packages also reports the ability to create survival models with time-varying covariates that would be necessary, but in practice these models produced errors when interval or counting censorship was specified.⁶⁶⁻⁷⁰ Therefore, no survival analysis is presented here because it cannot be completed using R as of this writing.¹⁷

It was possible to analyze response, combined, survival, and progress statuses as classification problems with machine learning using the h2o R package.⁷¹ However, the accuracy of these models was either exceptionally poor (response status and combined status) or deeply questionable in light of the small number of samples (survival status and progress status). Results from these models have been excluded here because they would demonstrate poor practices in machine learning regarding the number of samples per class that are needed for accuracy and generalizability.⁷²⁻⁷⁶ These outcomes would also lend themselves to analysis by multinomial logistic regression or logistic regression, but such models would be inferior to machine learning models and require tables of 250 lines each for complete reporting – consequently, they are also excluded here.

Therefore, regression models become the best alternative for the analysis of the response, combined, survival, and progress statuses. However, only the analysis of progress status is presented here using linear-mixed effects models (see Methods) for several reasons. The reporting of response status would again require a table 250 lines and the status has issues related to temporality. This is because progress status was defined as the clinical “best” response at *any* time during the follow up period – where measurements of tumor dimensions were taken at two close time points, a status of stable disease might easily be

achieved. Even where progression follows shortly after the assignment of a complete/partial or stable response status, the outcome would remain the “best” response. As mentioned in the methods, time points of follow up were non-specific and this creates inconsistency in the assignment of response and combined statuses that would introduce significant bias into their analysis. Finally, only the analysis of progress status is presented here because survival status represents a more distal endpoint for which progress status is a leading surrogate – models of progress status better illustrate which tryptophan metabolites, among others, are associated with progression and/or therapeutic efficacy.

Two volcano plots in Figure 3.1 show metabolites that are significantly ($P \geq 0.05$) associated with progress status; at the left (a) are metabolites with significantly different concentrations at baseline, and at the right (b) are metabolites with concentrations that change significantly throughout the course of treatment. In both plots, responded is the reference level such that red points represent metabolites that are significantly lower at baseline or decreasing in concentration over time among those who progressed, and green points represent metabolites that are significantly higher or increasing over time; gray points represent metabolites that were not significantly different between the progress status levels either at baseline or over time. Finally, as each point in Figure 3.1 (a) or (b) represents an individual linear mixed-effects model including terms for all covariates in Table 3.1 in addition to progress status and its interaction with time after treatment initiation (measured in days), correction for multiple testing was attempted but its application removed all significant findings – the P values represented by the Y axis of the Figure 3.1 are not corrected for multiple testing.

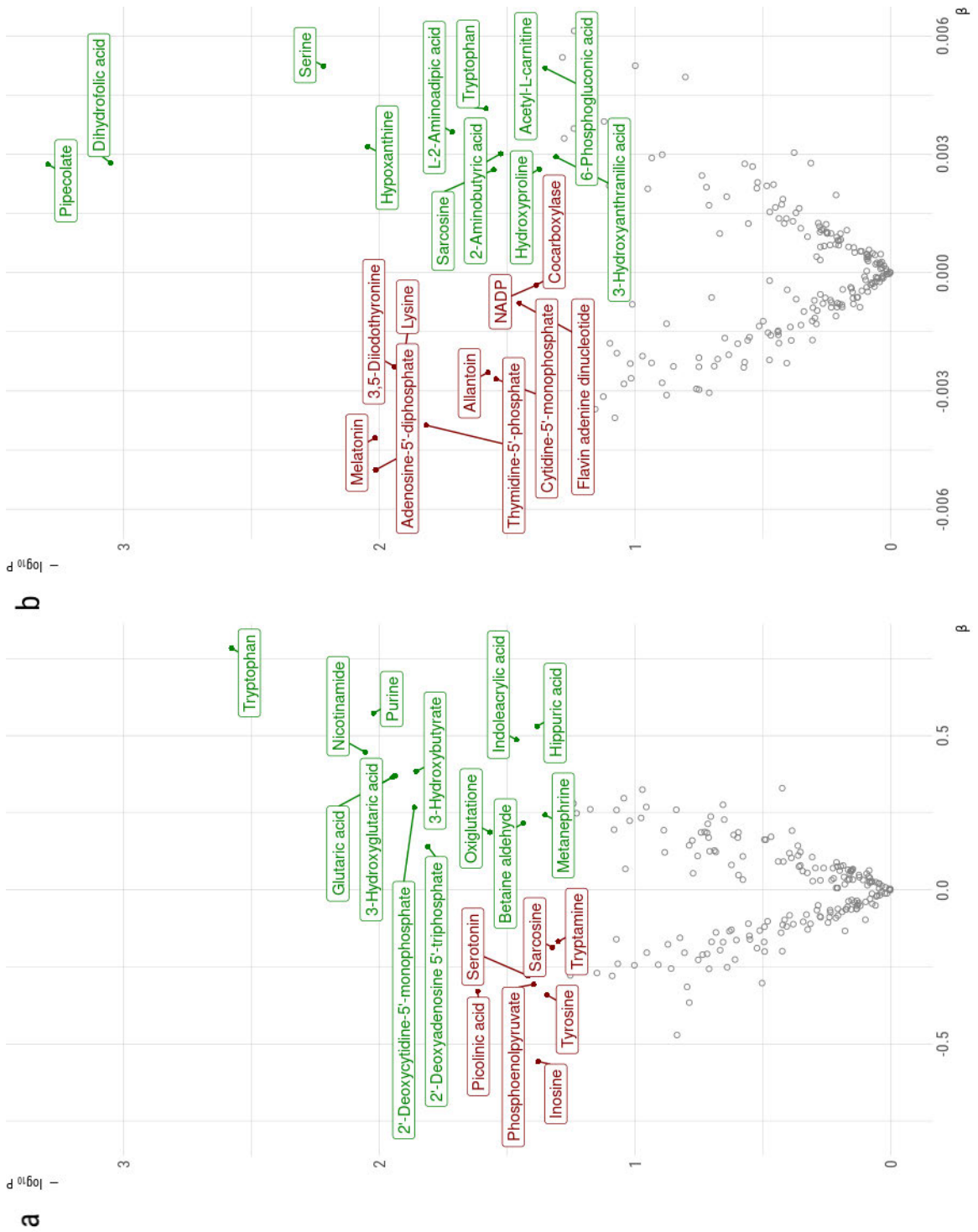


Figure 3.1. Metabolites Associated with Progression-Free Survival Status. Linear mixed-effects regression P values ($-\log$ transformed) versus effect size (β) estimates of progress status (a) and progress status \times time after treatment initiation (b). See previous page for complete description.

Key constituents of tryptophan, folic acid, lysine, and purine metabolism emerge from the significant metabolites of linear mixed-effects regression modeling when arranged into metabolic pathways, as in Figures 3.2 and 3.3 (text in these figures has been colored based on Figure 3.1). Analysis suggests tryptophan metabolism is associated with progress status and/or therapeutic efficacy of anti PD-1/PD-L1 IO agents and is likely creating immune privilege in tumor microenvironments – in Figure 3.1 (a), high levels of tryptophan at baseline are the best and most significant predictor of progress status. It appears the serotonin pathway is shunted in favor of the kynurenine and indole pathways (Figure 3.2); the former requiring high levels of IDO1 to catalyze the first reaction of the pathway which would grant immune privilege, and the later known to have metabolites capable of acting as aryl hydrocarbon receptor (AhR) agonist – whose complex roles in cancer are not yet completely understood.^{40,77,78}

Progressing through the kynurenine pathway in Figure 3.2, analysis suggests tryptophan is fated either to become the methyl acceptor nicotinamide to drive the methionine cycle or enter the tricarboxylic acid (TCA) cycle with lysine. As the methyl acceptor nicotinamide, the catalysis of S-adenosylmethionine (SAM) to S-adenosylhomocysteine (SAH) is driven to convert homocysteine to methionine – this reaction can also be catalyzed with epinephrine acting as a methyl acceptor and being converted to metanephrine (significantly higher in concentration at baseline among those who progress). Analysis suggests significant upregulation of the methionine cycle among those who progress, with the production of homocysteine catabolites oxoglutatione and 2-aminobutyric acid.^{79,80} This upregulation also seems to upregulate folate metabolism through choline metabolism.

Specifically, four metabolite intermediates, betaine aldehyde, sarcosine, serine, and hippuric acid, along the transition of choline to hippuric acid are upregulated among those who progressed (see Figure 3.2). The conversion of homocysteine to methionine requires catalysis from the betaine to dimethylglycine reaction which yields sarcosine which becomes serine. The transition of serine to glycine is a catalyst for the conversion of tetrahydrofolic acid (THF) to 5,10-methylene tetrahydrofolic acid (5,10-CH₂-THF) which becomes dihydrofolic acid (DHF) (significantly upregulated over time) when catalyzed by the 2'-deoxyuridine 5'-monophosphate (dUMP) to thymidine-5'-phosphate (dTMP) reaction. The precursors of dUMP, cytidine-5'-monophosphate (CMP) and deoxycytidine-5'-monophosphate (dCMP), are downregulated and upregulated respectively suggesting interconversion to drive the folate cycle. The downregulation of dTMP over time in those who progressed also suggest folate cycle activity and perhaps the exhaustion of this metabolite where it is being used as a DNA monomer.⁸¹⁻⁸⁵

Revisiting observations in the kynurenine pathway, it appears 3-hydroxyanthranilic acid is not converted to picolinic acid in those who progressed because these metabolites are significantly upregulated and downregulated respectively. This suggests tryptophan might progress towards 2-oxoadipic acid (sometimes also known as α -ketoacidic acid) to join lysine catabolites and enter the TCA cycle. Over time, analysis strongly suggest lysine is being consumed as an energy source, as it is converted to pipercolate (the most significantly different metabolite in Figure 3.1 (b)) and then L-2-aminoacidic acid. Intermediates between 2-oxoadipic acid and the TCA cycle are present, consistent with the increased energy demands of tumor cells, as is shown in Figure 3.2.

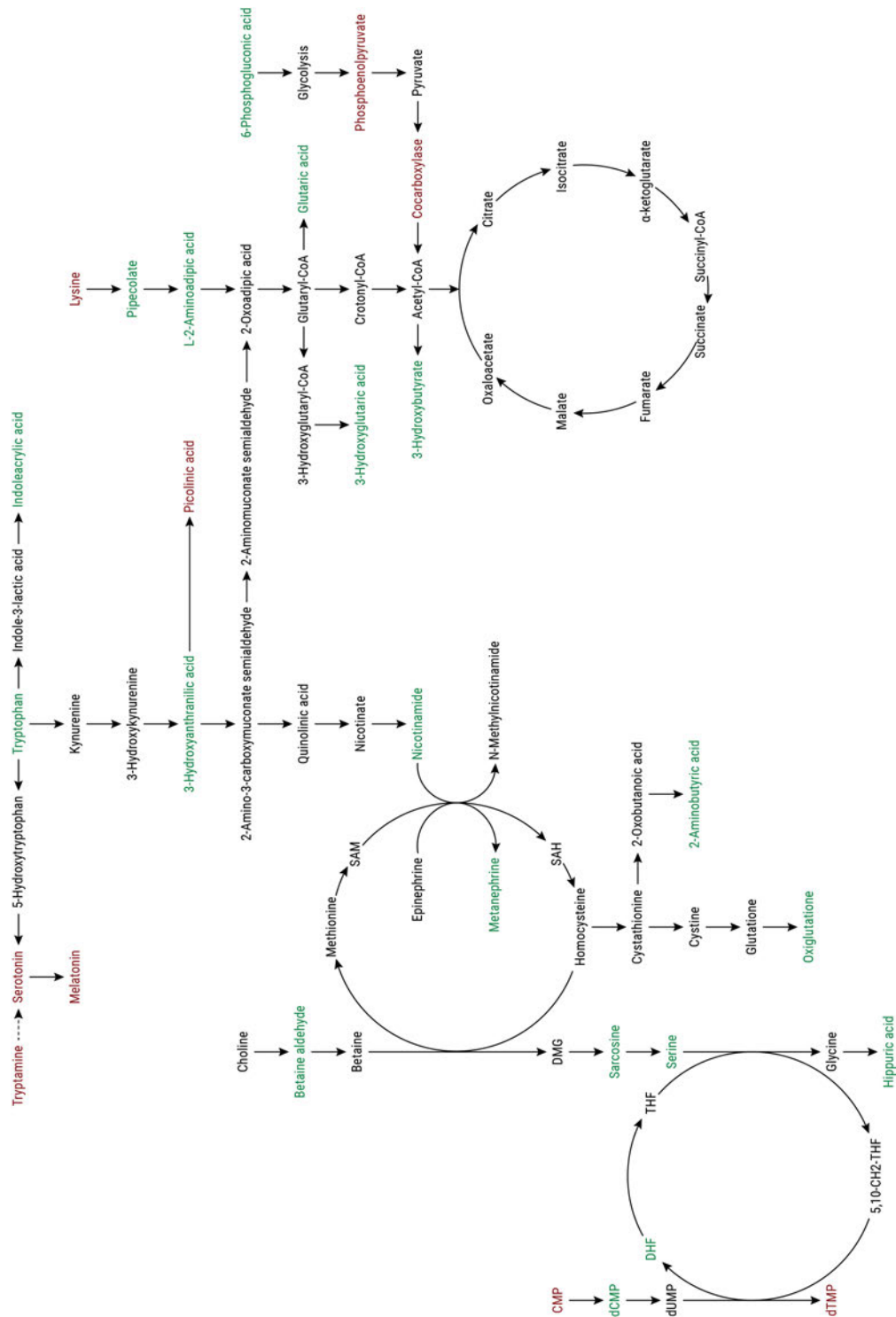


Figure 3.2. Tryptophan, Folic Acid, and Lysine Metabolism. From the top, tryptophan can become serotonin, enter the indole pathway, or become kynurenine. As nicotinamide, it catalyzes the methionine cycle and folate metabolism; to the right, it joins lysine heading to the TCA cycle.

Next, analysis suggests one of two fates for 6-phosphogluconic acid: either it enters the TCA cycle via glycolysis (Figure 3.2) or proceeds via the pentose phosphate pathway to participate in purine metabolism (Figure 3.3). While 6-phosphogluconic acid is increasing significantly over time among those who progress, the intermediates phosphoenolpyruvate and carboxylase are significantly downregulated suggesting either no or rapid conversion to acetyl-CoA. Yet, the upregulation of acetyl-CoA catabolite 3-hydroxybutyrate suggests acetyl-CoA is present and perhaps being consumed rapidly by tumor cells – the catabolite is synthesized in the liver when serum glucose is low (i.e., shunted to tumor cells).^{81–85}

To participate in purine metabolism, 6-phosphogluconic acid could be converted to phosphoribosyl pyrophosphate (PRPP) via the pentose phosphate pathway and proceed through intermediates to become inosine 5'-monophosphate (IMP) as Figure 3.3 shows. The conversion is possible but very few of the intermediates and constituents are up or downregulated, suggesting simply that molecular inosine joins hypoxanthine to become the nucleoside IMP. Then IMP is catabolized through succinyl AMP (S-AMP) to provide energy in the form of ATP/dATP (adenosine-5'-triphosphate/2'-deoxyadenosine 5'-triphosphate) from ADP (adenosine-5'-diphosphate). Given that allantoin (a distal metabolite of hypoxanthine) is downregulated, hypoxanthine is upregulated, and inosine is downregulated this explanation seems to be the most parsimonious. The downregulation and upregulation of the energy molecules ADP and dATP also suggest 6-phosphogluconic acid rather becomes acetyl-CoA for entry into the TCA cycle. Finally, the general label purine appears as a metabolite that is significantly higher in concentration at baseline among those who progressed – this likely related to ATP generation or DNA synthesis.

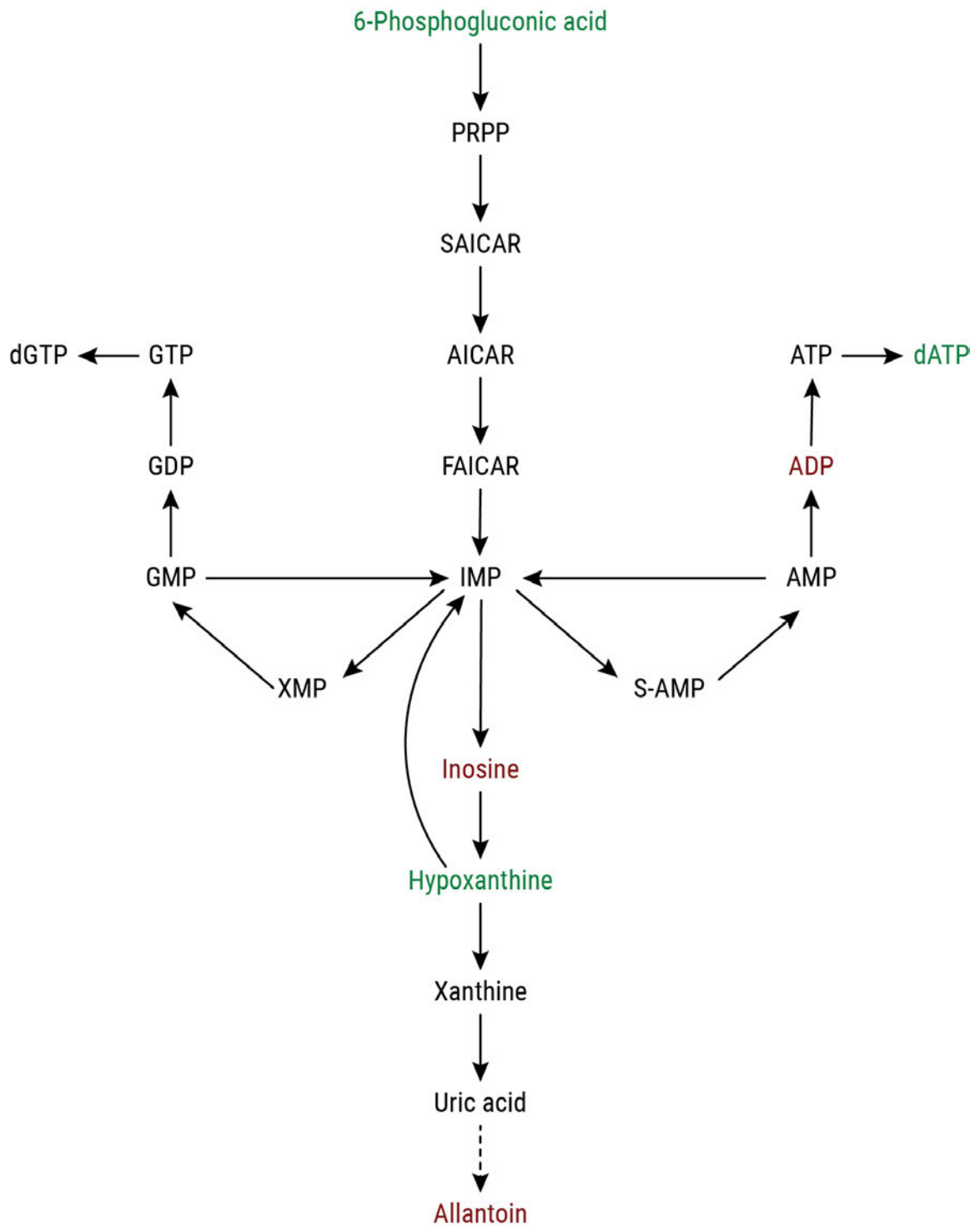


Figure 3.3. ATP Synthesis by Purine Synthesis and Salvage. Analysis suggests molecular inosine is bound to hypoxanthine to produce the inosine nucleoside IMP which is then consumed to produce S-AMP, AMP (adenosine monophosphate), ADP, ATP, and ultimately dATP.

Discussion

All but a small number of metabolites significantly associated with progress status can be accounted for through tryptophan, folic acid, and lysine metabolic pathways. Those not mentioned in results only include acetyl-L-carnitine, FAD/NADP (flavin adenine dinucleotide/nicotinamide adenine dinucleotide phosphate), 3,5 diiodothyronine, tyrosine, and hydroxyproline. Of these, acetyl-L-carnitine is likely present to detoxify high concentrations of 3-hydroxyglutaric acid and glutaric acid which become metabotoxins as their concentrations increase proportionally with tryptophan and lysine metabolism.⁸¹⁻⁸⁵ Otherwise, hydroxyproline has long been known to be higher in concentration in the serum of cancer patients – it is an amino acid unique to the synthesis of collagen.^{86,87} Current research suggests tumor cells can prevent immune infiltration by constructing a dense extracellular matrix of collagen impenetrable to lymphocytes.⁸⁸

The remaining metabolites require a slightly more complicated hypothesis and cannot yet be fully explained by current research. Tyrosine is an amino acid precursor to thyroid hormones like T3 and T4, but there are other variants of thyroid hormones such as T2 or 3,5 diiodothyronine. T2 is a potent energy modulator and animal studies have shown it to cause the rapid uptake of Ca^{2+} ; its cellular target is likely the mitochondria.^{89,90} Similarly, phosphoenolpyruvate is thought to correlate with Ca^{2+} levels in the cytosol and increase when oxidative stress is high because of intense glucose competition, a condition occurring in tumor microenvironments.⁹¹ Further, in the mitochondrial matrix, Ca^{2+} modulates the TCA cycle and through it the production of FAD and NADP both of which drive the electron transport chain and in turn the activity of APT synthase.⁹²

Thus, it is hypothesized that tyrosine is used to create T2 which targets mitochondria and causes the rapid uptake of Ca^{2+} to produce FAD/NADP and increase ATP synthase output. Studying the signs of coefficients in Figure 3.1, tyrosine and 3,5 diiodothyronine are negative suggesting either exhaustion of these metabolites in those who progressed or increasing concentrations of them among those who responded. If the hypothesis is correct, the former suggests tumor cells are using T2 to increase ATP synthase output for malignancy; whereas the latter suggests lymphocytes are using T2 to support energy demands of adaptive immunity. In either case, the current scientific literature as of this writing does not provide an answer as to which cells might be using T2 or soliciting its production, but analysis here begins to suggest some mechanism might be at play.

Analysis here might also suggest some additional rationale as to why IDO1 inhibitors achieved lackluster results in clinical trials.⁴⁰ Towards the bottom left of Figure 3.2, folate metabolism is highly active in those who progressed and requires methyl acceptors to catalyze the preceding reactions of the methionine cycle. If combination IDO1, IDO2 and tryptophan 2, 3-dioxygenase (TDO) inhibitors to diminish the conversion of tryptophan to kynurenine were to fail in clinical trials, observations of analysis here might suggest methyl acceptors beyond nicotinamide are driving the methionine and folate cycles in oncogenesis.^{40,48,93} The latter being an essential part of one-carbon metabolism and DNA synthesis in normal and tumor cells alike that could be targeted with immunosuppressants.

CHAPTER FOUR

tuberculosis

Introduction

Before the onset of the global coronavirus pandemic, tuberculosis had been the leading cause of infectious disease mortality worldwide for many years.⁹⁴ *Mycobacterium tuberculosis*, the causative agent of tuberculosis disease, is spread by aerosolization and, with sufficient exposure, results in latent or active tuberculosis.⁹⁵ The bacilli are acid-fast gram-ambiguous nonmotile rods with lengths of approximately 2µm for which humans are the reservoir.^{96–99} *M. tuberculosis* is exceptionally slow growing with a doubling time of about 24 hours, thus confirmation of tuberculosis disease by culture requires approximately six weeks.^{99,100} Therefore, the use of tuberculin skin tests, interferon-gamma release assays, or the Xpert MTB/RIF Ultra assay is preferred for diagnosis.^{96,99,101} Treatment of drug-susceptible infections has recently changed and requires a four month regimen of combination therapy in two phases – eight weeks of daily treatment with rifapentine, isoniazid, pyrazinamide, and moxifloxacin are followed by nine weeks of daily treatment with rifapentine, isoniazid, and moxifloxacin.¹⁰²

Diagnostics and therapies for tuberculosis are advanced and effective, yet even small improvements to them yield remarkable results given the burden of disease. In 2020, the most recent year for which data are available, an estimated 1.5 million deaths globally were attributed to tuberculosis and 10 million cases of tuberculosis were newly diagnosed. Even though most deaths and cases occurred in South-East Asia and Africa, the control of tuberculosis remains a high priority for public health authorities in all localities.⁹⁴

This is because one-quarter of the global population is estimated to have a latent tuberculosis infection (LTBI), where an individual has been infected with *M. tuberculosis* bacilli and would achieve a positive diagnostic test result but lacks clinical sequelae of disease.^{103,104} It is estimated that 80% of infected individuals will eventually develop symptoms, and recent theoretical work by Drain *et al.* suggests disease may progress along a continuum with biomarkers corresponding to severity and temporality.^{96,105} In their view, tuberculosis progresses from latent to incipient to subclinical to active and might cycle through these statuses in time or simply ascend through them at a rapid or slow pace.

The theory is alluring and offers a framework to study the etiology of tuberculosis as a classification or regression problem with machine learning, where highly predictive molecular features could suggest important targets for the development of diagnostic test or therapeutic interventions. Among the various types of multiomics data, host transcriptomics becomes an obvious modality to explore the pursuit because infections are known create distinct transcriptional changes and vast quantities of transcriptomics data sufficient for machine learning have been made publicly available.¹⁰⁶ However, exceptional difficulty in applying machine learning would arise from the formatting, cleanliness, and quality of data available to serve as the input of the modeling process.

To address this difficulty the tuberculosis R/Bioconductor package was developed and provides *all* transcriptomics data of human host infected with tuberculosis available from GEO (Gene Expression Omnibus) in a clean high-quality machine learning-ready format, given that samples did not come from cell lines, were not taken postmortem, and did not feature recombination.¹⁸ The package can access more than 10,000 samples from both

microarray and sequencing studies that have been processed from raw data through its hyper-standardized, reproducible pipeline. The companion data preprocessing pipeline, `tuberculosis.pipeline`, is containerized for reproducibility and provides novel data to the package twice a year through cloud storage infrastructure.¹⁹ The `tuberculosis` package uses the ExperimentHub platform to provide resources (i.e., expression matrices) from cloud storage which allows the software to remain lightweight and avoid freighting of data – it contains only instructions to access data rather than data itself.¹⁰⁷ An overview of data provenance in the `tuberculosis` package is shown in Figure 4.1.

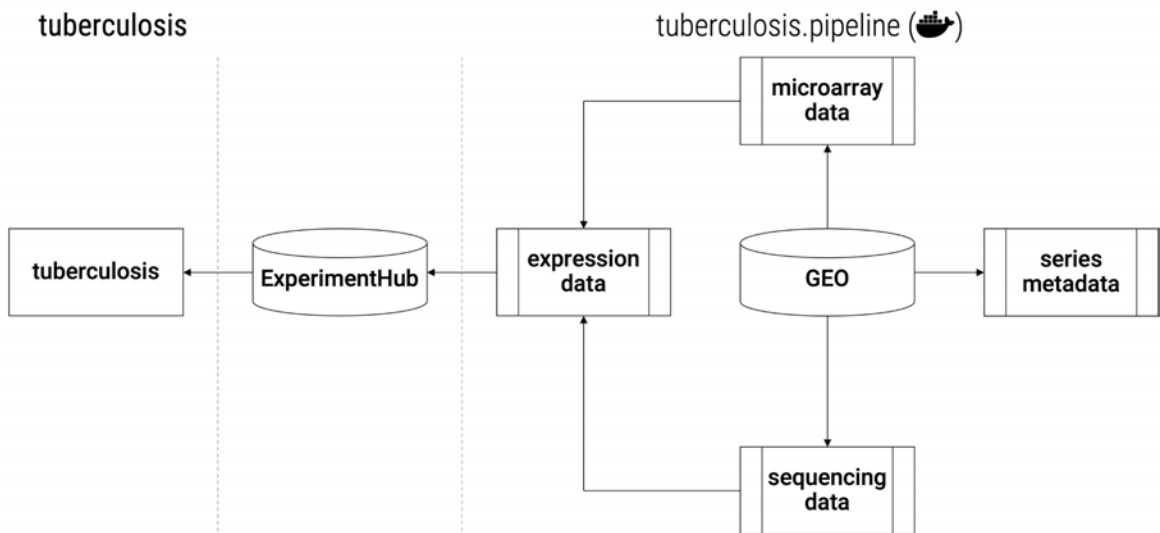


Figure 4.1. Data Provenance in the `tuberculosis` Package. On the left, the `tuberculosis` R/Bioconductor package provides resources (i.e., expression matrices) from the ExperimentHub platform – it is the only interface users interact with. In the center, the ExperimentHub platform from Bioconductor acts as cloud storage for resources which can be downloaded on an as-needed basis by `tuberculosis`. On the right, the `tuberculosis.pipeline` companion R package and Docker container are used to preprocess data from GEO – there are subroutines to process series metadata, microarray data, and sequencing data; output of the latter two become expression matrices that are uploaded to ExperimentHub through the `expression data` subroutine.

Methods

The data preprocessing pipeline, `tuberculosis.pipeline`, is run inside of a Docker container for reproducibility because it provides a stable and immutable environment for execution. The container is a modified version of the `schifferl/bioc-release` container which uses an older version of OpenBLAS (v0.3.3) that was needed for compatibility with `affy` package at the time of development.^{108,109} The data preprocessing pipeline is written in R and required functionality from other R packages for its development.^{17,33,53,55–58,108,110–130} All data that are preprocessed, or perhaps reprocessed, come from GEO.^{20,21}

Data preprocessing begins with the programmatic creation of a spreadsheet file on Google Drive of GEO series matching the following query: “Tuberculosis”[MeSH Terms] AND “Homo sapiens”[Organism] AND “gse”[Filter] AND (“Expression profiling by array”[Filter] OR “Expression profiling by high throughput sequencing”[Filter]). The spreadsheet, `series-metadata`, contains metadata about each GEO series (i.e., study) matching the query including accession numbers, experiment type (microarray or sequencing), and the number of applicable samples. It is also updated programmatically as new GEO series are released and is referenced to know which series to preprocess.

The preprocessing of individual microarray or sequencing series requires addressing issues regarding gene names and microarray annotations a priori. Specifically, gene names across microarray annotations are inconsistent and frequently outdated; to ensure consistent feature names for machine learning, they must be updated and standardized. Therefore, mappings from current HGNC-approved GRCh38 gene names to antiquated, alias, ensemble, entrez, ena, and refseq identifiers are created from the `genenames.org` REST API

first.¹³¹ Then, gene annotations of microarray probes are updated from the mappings; both the mappings and the microarray platform annotations are serialized to ensure consistent annotation when novel data are preprocessed. An identity mapping of the valid gene names is also serialized and used to ensure the gene names from sequencing series are correct and consistent with microarray platform annotations.

Preprocessing of both microarray and sequencing series begins with the retrieval of sample metadata from GEO; it is programmatically cleaned and reviewed to ensure each sample has raw data available, did not come from a cell line, was not taken postmortem, and did not feature recombination. For microarray series, raw data from imaging (e.g. CEL files) must be available as supplemental files; and for sequencing series, it must be possible to translate GEO sample accession numbers to Sequence Read Archive (SRA) run accession numbers.^{132,133} If samples of a given series meet these criteria, a spreadsheet of sample metadata per series is created on Google Drive to later be standardized against ontology vocabulary.

An expression matrix for each microarray series is generated from raw data by applying the normal-exponential background correction method from limma, with the saddle-point approximation to maximum likelihood specified.^{116,134,135} Where platforms necessitated it, the robust multichip average (RMA) algorithm, as implemented in affy or oligo, without background correction or normalization was used to generate the matrix.^{108,118,136–138} When multiple probes mapped to a single gene, a single probe was chosen at random *per* series as to not introduce bias by creating a probe to gene relationship across *all* series; expression values were not normalized before the matrix was upload to Google Drive for storage.

For each sequencing series, the SRA run accession numbers are used to write a bash script that processes raw sequencing data using the SRA Toolkit and the nf-core/rnaseq Nextflow pipeline.^{139–142} These bash scripts are run on the Boston University Shared Computing Cluster to create array jobs which download fastq files from SRA, run the nf-core/rnaseq pipeline, and create tab-separated files of gene expression measurements for each sample. The preprocessing pipeline is used to read in each tab-separated file related to a single series and construct a gene expression matrix that includes all samples; the gene names of which are validated using the identity mapping mentioned earlier before the matrix is upload to Google Drive for storage.

The expression matrices stored in Google Drive exist as comma-separated value files and require conversion to R objects before they can be uploaded to ExperimentHub storage. To accomplish this, the expression matrices for both types of series are downloaded from Google Drive and saved as R matrix files for upload to ExperimentHub storage – this last step requires manual interaction with a cloud storage client. When the R matrix files are uploaded to ExperimentHub storage, they must be made available by Bioconductor administrators – a file of metadata describing the R matrix files is sent to them and the resources are added to the ExperimentHub production database; this file is also included in the tuberculosis package.

The tuberculosis package is updated by the methods outline here with each Bioconductor release in April and October. When new data matching the inclusion/exclusion criteria outline here are released though GEO, they are included in the tuberculosis package using the data preprocessing pipeline, tuberculosis.pipeline, and the ExperimentHub platform.

Results

The initial release of the tuberculosis package (v1.0.0) provided *all* transcriptomics data of human host infected with tuberculosis available from GEO at the time – 12,614 samples from 135 series – in a clean high-quality machine learning-ready format and more data has been made available since. The microarray and sequencing data resources available through tuberculosis have been reprocessed from raw data by its companion pipeline (see Methods) and are hyper-standardized – even gene names and samples have been alphabetized. The intuitive interface of tuberculosis provides access to all resources with a single function and returns SummarizedExperiment objects with current HGNC-approved GRCh38 gene names that are consistent across every GEO series included in the package. Through the development of the tuberculosis package, a substantial barrier to applying machine learning to study the etiology of the namesake disease has been overcome.

To create the vast volume of data available in tuberculosis (v1.0.0), 251 GEO series records were first identified by the query detailed in the Methods section. Many of these series did not have raw data available and 106 of them were excluded based on this eligibility criteria, also outlined in the Methods section. Screening of the remaining 145 GEO series records found 10 GEO series for exclusion – there were 5 records where GEO sample accession numbers could not be translated to SRA run accession numbers and 5 records where raw data files could not be downloaded as expected. Samples of the remaining 135 records were then screened and only 3 were excluded; 1 sample was excluded because the expected raw data files could not be downloaded, and 2 samples were excluded because the genome alignment rate during data preprocessing was exceptionally poor.

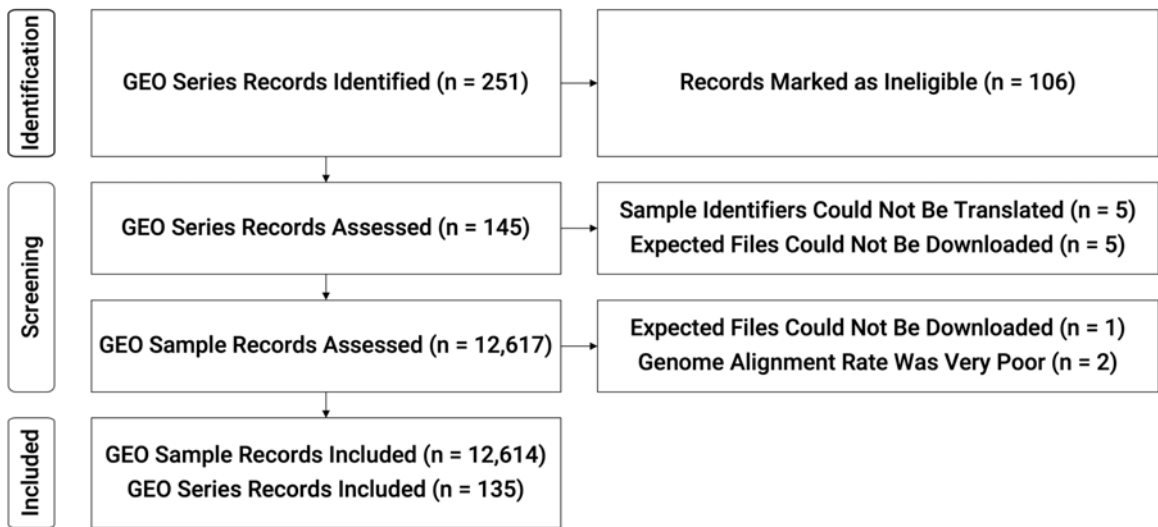


Figure 4.2. GEO Records Initially Included in tuberculosis. The flow chart shows the number of GEO series and sample records that were identified, screened, and ultimately included in v1.0.0 of the tuberculosis package – see the paragraph above for a narrative description.

An intuitive single function interface provides access to all the data resources available through tuberculosis. The tuberculosis function, as shown in Figure 4.3, has only two arguments and is used to both find and download data resources from cloud storage. The first argument, `pattern`, is a regular expression pattern to match against the names of available resources. The second argument, `dryrun`, specifies whether the names of data resources or data resources themselves should be returned. When `dryrun = TRUE`, the names of resources are printed as a message and returned invisibly as a character vector. When `dryrun = FALSE`, a list of resources is returned as `SummarizedExperiment` objects – these are downloaded from cloud storage on-demand and locally cached so that future use of the same resources does not require redownload. Finally, just as a note, a creation date is prefixed to the name of each resource – if a resource has multiple creation dates, the most recent is selected by default. This implementation detail can largely be ignored but adding a date to `pattern` will override this behavior; the feature exists for reproducibility.

```

library(tuberculosis)

tuberculosis("GSE103147")

## 2021-09-15.GSE103147

tuberculosis("GSE103147", dryrun = FALSE)

## `$` 2021-09-15.GSE103147`
## class: SummarizedExperiment
## dim: 24353 1649
## metadata(0):
## assays(1): exprs
## rownames(24353): A1BG A1BG-AS1 ... ZZEF1 ZZZ3
## rowData names(0):
## colnames(1649): SRR5980424 SRR5980425 ... SRR5982072 SRR5982073
## colData names(0):

tuberculosis("GSE13293.")

## 2021-09-15.GSE132931
## 2021-09-15.GSE132932

tuberculosis("GSE13293.", dryrun = FALSE)

## `$` 2021-09-15.GSE132931`
## class: SummarizedExperiment
## dim: 24353 10
## metadata(0):
## assays(1): exprs
## rownames(24353): A1BG A1BG-AS1 ... ZZEF1 ZZZ3
## rowData names(0):
## colnames(10): SRR9320541 SRR9320545 ... SRR9320573 SRR9320577
## colData names(0):
##
## `$` 2021-09-15.GSE132932`
## class: SummarizedExperiment
## dim: 24353 60
## metadata(0):
## assays(1): exprs
## rownames(24353): A1BG A1BG-AS1 ... ZZEF1 ZZZ3
## rowData names(0):
## colnames(60): SRR9320820 SRR9320825 ... SRR9321106 SRR9321111
## colData names(0):

```

Figure 4.3. Example Usage of the tuberculosis Package. The tuberculosis package has an intuitive single function interface to find and return resources – when no dryrun argument is specified, only the names of available resources are returned. When dryrun = FALSE, data resources are returned as a named list of SummarizedExperiment objects. The first argument, pattern, is a regular expression pattern to match against the names of available resources – it can be used to return multiple resources simultaneously. See the paragraph above for notes about the creation date prefix.

Discussion

The tuberculosis package is original software for the mega-analysis of host transcriptomics relevant to etiology of its namesake disease. It overcomes the first significant barrier to applying machine learning to relevant molecular classification and regression problems by providing more than 10,000 samples from human host infected with tuberculosis. The data resources accessible through tuberculosis hold great promise for the improvement of diagnostic and therapeutic tools, where any advances are instantly meaningful because the burden of tuberculosis disease is immense. The development of the package also embodies the *primum non nocere* doctrine of bioethics and represent the optimal final resting place for biomedical data – whose generation is annoying to research participants at best and harmful at worst. Priming data for reuse represents both the potential reduction of harms and direction of limited new resources towards their highest purpose.

The second significant barrier to applying machine learning and the current limitation of the tuberculosis package is the provision of metadata for every sample. Accomplishing the task would enable supervised machine learning, where only unsupervised analyses are possible now; yet the endeavor parallels in difficulty to the development of tuberculosis itself. This is because to achieve the same level of hyper-standardization would require an entirely different infrastructure to curate metadata from each study (i.e., series) independently. Inconsistencies in the quality of metadata abound in the same way as they do with data (e.g., incongruent gene names), but they are far less amenable to programmatic resolution. Software to accomplish this task would feature a script for every study and need to consistently align every metadata variable with ontological vocabulary.

This piece of software, tuberculosis.curation, the metadata preprocessing pipeline for tuberculosis is imagined and will be developed in years to come. When complete, the final challenge in applying machine learning to host transcriptomics of human host infected with tuberculosis will be revealed – this is, it will be possible to develop sensitive but not specific diagnostic models of the disease. Where labels of disease severity or status are accurate, it will very likely be possible to uncover salient molecular (i.e., transcriptomic) features that define them, but it will be not be possible to definitively know if features are related to tuberculosis alone in the absence data from other conditions. Determining the cosmic background transcriptional activity of infection in general suggests additional packages like tuberculosis (e.g., influenza) would be needed and that a symbiotic ecosystem of them could leverage great synergy.

CHAPTER FIVE

Conclusions

This dissertation has presented perspectives on biomedical data science as a discipline and helps to bring clarity to multiomics analyses with new and redrawn illustrations that demonstrate interrelationships between the seven modalities of data representing the epigenotype, genotype, and phenotype. The research projects presented here have explored current challenges related to multimodal, longitudinal, and mega-analysis of biomedical data in the context of a paradigm shift towards single-cell sequencing. Through the creation of novel software and applied translation analysis, several advancements and insights have been gained – both of which should confer benefits to future research.

Chapter two presented the MultimodalExperiment data structure as means to appropriately and efficiently represent hierarchical multimodal multiomics data. In comparison to alternative data structures, existing storage and management challenges were overcome by employing normalization and creating a solution informed by database architecture. This required considering biomedical data in a relational framework and separating experiment, subject, sample, and cell annotations to optimize storage efficiency. In doing so, the ease of data management was improved and made to require essentially only two verbs, propagate and harmonize, with minimal manual management of maps. As of this writing, few studies are sufficiently complex to take full advantage of MultimodalExperiment, but their numbers are expected to grow in coming years. If this projection does not come to fruition, MultimodalExperiment still provides an elegant solution for the storage and management of deconvoluted bulk and aggregated single-cell RNA-seq data.

Chapter three presented longitudinal analysis of a cohort study of cancer patients being treated with anti PD-1/PD-L1 immune checkpoint inhibitors at Boston Medical Center. Progression-free survival status of study participants was analyzed using linear mixed-effects models to describe metabolites that were significantly different, among those who progressed, both at baseline and over time. Of metabolites associated with progression-free survival status, tryptophan metabolites were found to be particularly important, and evidence suggested they serve as methyl acceptors to catalyze methionine and folate metabolism in oncogenesis. Evidence also suggested lysine catabolites enter the TCA cycle to meet energy demands under the intense glucose competition conditions of cancer. Finally, it was hypothesized that tumors metabolize hydroxyproline to build a dense extracellular matrix and prevent immune infiltration, and that T2 thyroid hormone is involved in increased ATP synthase output during cancer by some unknown mechanism.

Chapter four provided extensive details regarding the development of the tuberculosis package, and its companion data preprocessing pipeline, `tuberculosis.pipeline`.^{18,19} The provenance of data was established and extensive methods detailing how data were preprocessed were provided—microarray data were reprocessed from raw image files and consistently background corrected using the normal-exponential method with the saddle-point approximation to maximum likelihood specified; sequencing data were reprocessed from raw data using the `nf-core/rnaseq` pipeline.^{135,142} The package provides more than 10,000 samples of clean high-quality machine learning ready transcriptomics data from human host infected with tuberculosis. These data are important to applying machine learning to study the etiology of tuberculosis and may help improve outcomes in the future.

The advancements and insights gained through and elaborated upon in this dissertation will support the further development of integrative analyses of biomedical data. Specifically, the development of the MultimodalExperiment has enabled the storage, management, and analysis of multimodal multiomics data to a greater extent that was previously possible; longitudinal analysis of the anti PD-1/PD-L1 IO cohort study presented here has suggested a few putative drug targets to pursue that might improve therapeutic efficacy; and the tuberculosis package enables machine learning mega-analysis of host transcriptomics to study the etiology of the namesake disease for the first time.

LIST OF JOURNAL ABBREVIATIONS

Am. J. Cancer Res.	American journal of cancer research
Am. J. Transplant	American Journal of Transplantation
Ann. Stat.	The Annals of Statistics
arXiv	arXiv
Biochem. J	Biochemical Journal
Biochim. Biophys. Acta Mol. Cell Res.	Biochimica et Biophysica Acta - Molecular Cell Research
Bioinformatics	Bioinformatics
bioRxiv	bioRxiv
Biostatistics	Biostatistics
Blood	Blood
BMC Bioinformatics	BMC Bioinformatics
Br. J. Cancer	British Journal of Cancer
Br. Med. J.	British Medical Journal
Can. Assoc. Radiol. J.	Canadian Association of Radiologists Journal
Cancer Res.	Cancer Research
Cell Host Microbe	Cell Host and Microbe
Cells	Cells
Clin. Microbiol. Rev.	Clinical Microbiology Reviews
Commun Biol	Communications Biology

Eur. J. Cancer	European Journal of Cancer
Eur. Respir. J.	European Respiratory Journal
Expert Opin. Ther. Pat.	Expert Opinion on Therapeutic Patents
F1000Research	F1000Research
Front. Endocrinol.	Frontiers in Endocrinology
Front. Immunol.	Frontiers in Immunology
Genome Biol.	Genome Biology
Int. J. Epidemiol.	International journal of epidemiology
Int. J. Mol. Sci.	International Journal of Molecular Sciences
Int. J. Tryptophan Res.	International Journal of Tryptophan Research
J. Card. Fail.	Journal of Cardiac Failure
J. Clin. Invest.	Journal of Clinical Investigation
J. Clin. Microbiol.	Journal of Clinical Microbiology
J. Immunol.	The Journal of Immunology
J. Infect. Dis.	Journal of Infectious Diseases
J. Open Source Softw.	Journal of Open Source Software
J. Stat. Softw.	Journal of Statistical Software
JAMA	JAMA: The Journal of the American Medical Association
Lancet Infect. Dis.	The Lancet Infectious Diseases
Metabolites	Metabolites
Methods Mol. Biol.	Methods in Molecular Biology

MMWR Morb. Mortal. Wkly. Rep.	MMWR Recommendations and Reports
Nat. Biotechnol.	Nature Biotechnology
Nat. Genet.	Nature Genetics
Nat. Methods	Nature Methods
Nat. Rev. Drug Discov.	Nature Reviews Drug Discovery
Nat. Rev. Genet.	Nature Reviews Genetics
Nature	Nature
Neural Comput.	Neural Computation
Nucleic Acids Res.	Nucleic acids research
Pharm. J.	Pharmaceutical Journal
PLoS Genet.	PLoS Genetics
PLoS Med.	PLoS Medicine
Sci. Rep.	Scientific Reports
Science	Science
The R Journal	The R Journal
Trends Genet.	Trends in Genetics
Trends Pharmacol. Sci.	Trends in Pharmacological Sciences
Tuberculosis	Tuberculosis

BIBLIOGRAPHY

1. Huber, W. *et al.* Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods* **12**, 115–121 (2015).
2. Waddington, C. H. The epigenotype. 1942. *Int. J. Epidemiol.* **41**, 10–13 (2012).
3. Hasin, Y., Seldin, M. & Lusic, A. Multi-omics approaches to disease. *Genome Biol.* **18**, 83 (2017).
4. Crick, F. Central dogma of molecular biology. *Nature* **227**, 561–563 (1970).
5. Egger, G., Liang, G., Aparicio, A. & Jones, P. A. Epigenetics in human disease and prospects for epigenetic therapy. *Nature* **429**, 457–463 (2004).
6. Liebers, R., Rassoulzadegan, M. & Lyko, F. Epigenetic regulation by heritable RNA. *PLoS Genet.* **10**, e1004296 (2014).
7. Chen, Q., Yan, W. & Duan, E. Epigenetic inheritance of acquired traits through sperm RNAs and sperm RNA modifications. *Nat. Rev. Genet.* **17**, 733–743 (2016).
8. Duempelmann, L., Skribbe, M. & Bühler, M. Small RNAs in the Transgenerational Inheritance of Epigenetic Information. *Trends Genet.* **36**, 203–214 (2020).
9. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
10. Peterson, V. M. *et al.* Multiplexed quantification of proteins and transcripts in single cells. *Nat. Biotechnol.* **35**, 936–939 (2017).
11. Li, G. *et al.* Joint profiling of DNA methylation and chromatin architecture in single cells. *Nat. Methods* **16**, 991–993 (2019).

12. Chen, A. F. *et al.* NEAT-seq: simultaneous profiling of intra-nuclear proteins, chromatin accessibility and gene expression in single cells. *Nat. Methods* **19**, 547–553 (2022).
13. Eng, C.-H. L. *et al.* Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH. *Nature* **568**, 235–239 (2019).
14. Lee, D.-S. *et al.* Simultaneous profiling of 3D genome structure and DNA methylation in single human cells. *Nat. Methods* **16**, 999–1006 (2019).
15. Saikia, M. *et al.* Simultaneous multiplexed amplicon sequencing and transcriptome profiling in single cells. *Nat. Methods* **16**, 59–62 (2019).
16. Pan, L., Ku, W. L., Tang, Q., Cao, Y. & Zhao, K. scPCOR-seq enables co-profiling of chromatin occupancy and RNAs in single cells. *Commun Biol* **5**, 678 (2022).
17. R Core Team. R: A Language and Environment for Statistical Computing. (2022).
18. Schiffer, L. *tuberculosis: Tuberculosis Gene Expression Data for Machine Learning*. (Bioconductor, 2021). doi:10.18129/B9.BIOC.TUBERCULOSIS.
19. Schiffer, L. *tuberculosis.pipeline: Data Processing Pipeline for the tuberculosis Package*. (Github).
20. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**, 207–210 (2002).
21. Barrett, T. *et al.* NCBI GEO: archive for functional genomics data sets--update. *Nucleic Acids Res.* **41**, D991-5 (2013).

22. Ramos, M. *et al.* Software for the Integration of Multiomics Experiments in Bioconductor. *Cancer Res.* **77**, e39–e42 (2017).
23. Hernandez-Ferrer, C., Ruiz-Arenas, C., Beltran-Gomila, A. & González, J. R. MultiDataSet: an R package for encapsulating multiple data sets with application to omic data integration. *BMC Bioinformatics* **18**, 36 (2017).
24. Cancer Genome Atlas Research Network *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).
25. Wickham, H. *R Packages*. (O’Reilly Media, Incorporated, 2015).
26. Wickham, H. *Advanced R, Second Edition*. (CRC Press, 2019).
27. Pagès, H., Lawrence, M. & Aboyoun, P. *S4Vectors: Foundation of vector-like and list-like containers in Bioconductor*. (Bioconductor, 2017). doi:10.18129/B9.BIOC.S4VECTORS.
28. Genomics, 10x. PBMCs of a Healthy Donor - 5’ Gene Expression with a Panel of TotalSeq™-C Antibodies. *Single Cell Immune Profiling Dataset by Cell Ranger 3.0.0* (2018).
29. McCarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33**, 1179–1186 (2017).
30. Martin Morgan, Valerie Obenchain, Jim Hester, Hervé Pagès. *SummarizedExperiment*. (Bioconductor, 2017). doi:10.18129/B9.BIOC.SUMMARIZEDEXPERIMENT.

31. Amezquita, R. A. *et al.* Orchestrating single-cell analysis with Bioconductor. *Nat. Methods* **17**, 137–145 (2020).
32. Sorini, C. *et al.* Metagenomic and single-cell RNA-seq survey of the *H. pylori*-infected stomach in asymptomatic individuals. *bioRxiv* (2021) doi:10.1101/2021.12.04.21267139.
33. Wickham, H., François, R., Henry, L. & Müller, K. dplyr: A Grammar of Data Manipulation. (2022).
34. Newman, A. M. *et al.* Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat. Biotechnol.* **37**, 773–782 (2019).
35. Carter, S. & Thurston, D. E. Immuno-oncology agents for cancer therapy. *Pharm. J.* (2020) doi:10.1211/pj.2020.20207825.
36. Ahmadzadeh, M. *et al.* Tumor antigen-specific CD8 T cells infiltrating the tumor express high levels of PD-1 and are functionally impaired. *Blood* **114**, 1537–1544 (2009).
37. Han, Y., Liu, D. & Li, L. PD-1/PD-L1 pathway: current researches in cancer. *Am. J. Cancer Res.* **10**, 727–742 (2020).
38. Chen, L. & Han, X. Anti-PD-1/PD-L1 therapy of human cancer: past, present, and future. *J. Clin. Invest.* **125**, 3384–3391 (2015).
39. Munn, D. H. *et al.* Prevention of allogeneic fetal rejection by tryptophan catabolism. *Science* **281**, 1191–1193 (1998).
40. Opitz, C. A. *et al.* The therapeutic potential of targeting tryptophan catabolism in cancer. *Br. J. Cancer* **122**, 30–44 (2020).

41. Brenk, M. *et al.* Tryptophan deprivation induces inhibitory receptors ILT3 and ILT4 on dendritic cells favoring the induction of human CD4⁺CD25⁺ Foxp3⁺ T regulatory cells. *J. Immunol.* **183**, 145–154 (2009).
42. Chen, W., Liang, X., Peterson, A. J., Munn, D. H. & Blazar, B. R. The indoleamine 2,3-dioxygenase pathway is essential for human plasmacytoid dendritic cell-induced adaptive T regulatory cell generation. *J. Immunol.* **181**, 5396–5404 (2008).
43. Chung, D. J. *et al.* Indoleamine 2,3-dioxygenase-expressing mature human monocyte-derived dendritic cells expand potent autologous regulatory T cells. *Blood* **114**, 555–563 (2009).
44. Curti, A. *et al.* Modulation of tryptophan catabolism by human leukemic cells results in the conversion of CD25⁻ into CD25⁺ T regulatory cells. *Blood* **109**, 2871–2877 (2007).
45. Fallarino, F. *et al.* The combined effects of tryptophan starvation and tryptophan catabolites down-regulate T cell receptor zeta-chain and induce a regulatory phenotype in naive T cells. *J. Immunol.* **176**, 6752–6761 (2006).
46. Hippen, K. L. *et al.* In vitro induction of human regulatory T cells using conditions of low tryptophan plus kynurenines. *Am. J. Transplant* **17**, 3098–3113 (2017).
47. Sharma, M. D. *et al.* Plasmacytoid dendritic cells from mouse tumor-draining lymph nodes directly activate mature Tregs via indoleamine 2,3-dioxygenase. *J. Clin. Invest.* **117**, 2570–2582 (2007).

48. Platten, M., Nollen, E. A. A., Röhrig, U. F., Fallarino, F. & Opitz, C. A. Tryptophan metabolism as a common therapeutic target in cancer, neurodegeneration and beyond. *Nat. Rev. Drug Discov.* **18**, 379–401 (2019).
49. Modoux, M., Rolhion, N., Mani, S. & Sokol, H. Tryptophan Metabolism as a Pharmacological Target. *Trends Pharmacol. Sci.* **42**, 60–73 (2021).
50. Jones, S. P., Guillemin, G. J. & Brew, B. J. The kynurenine pathway in stem cell biology. *Int. J. Tryptophan Res.* **6**, 57–66 (2013).
51. Hsu, C.-N. & Tain, Y.-L. Developmental Programming and Reprogramming of Hypertension and Kidney Disease: Impact of Tryptophan Metabolism. *Int. J. Mol. Sci.* **21**, (2020).
52. Eisenhauer, E. A. *et al.* New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1). *Eur. J. Cancer* **45**, 228–247 (2009).
53. Wickham, H., Hester, J. & Bryan, J. readr: Read Rectangular Text Data. (2022).
54. Wickham, H. forcats: Tools for Working with Categorical Variables (Factors). (2021).
55. Bache, S. M. & Wickham, H. magrittr: A Forward-Pipe Operator for R. (2022).
56. Henry, L. & Wickham, H. purrr: Functional Programming Tools. (2020).
57. Müller, K. & Wickham, H. tibble: Simple Data Frames. (2022).
58. Wickham, H. & Girlich, M. tidyr: Tidy Messy Data. (2022).
59. Hester, J. *et al.* withr: Run Code “With” Temporarily Modified Global State. (2022).
60. Pang, Z., Chong, J., Li, S. & Xia, J. MetaboAnalystR 3.0: Toward an optimized workflow for global metabolomics. *Metabolites* **10**, 186 (2020).

61. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127 (2007).
62. Kuznetsova, A., Brockhoff, P. B. & Christensen, R. H. B. lmerTest Package: Tests in Linear Mixed Effects Models. *J. Stat. Softw.* **82**, 1–26 (2017).
63. U.S. Census Bureau. [No title].
<https://data.census.gov/cedsci/table?q=United%20States&g=0100000US&tid=DECENNIALPL2020.P1>.
64. Therneau, T. M. & Grambsch, P. M. *Modeling Survival Data: Extending the Cox Model*. (Springer New York).
65. Therneau, T. M. *A Package for Survival Analysis in R*. <https://CRAN.R-project.org/package=survival> (2022).
66. Lang, M. *et al.* mlr3: A modern object-oriented machine learning framework in R. *J. Open Source Softw.* **4**, 1903 (2019).
67. Sonabend, R., Király, F. J., Bender, A., Bischl, B. & Lang, M. mlr3proba: An R Package for Machine Learning in Survival Analysis. *Bioinformatics* (2021) doi:10.1093/bioinformatics/btab039.
68. Becker, M., Lang, M., Richter, J., Bischl, B. & Schalk, D. mlr3tuning: Tuning for “mlr3.” (2022).
69. Sonabend, R., Schratz, P. & Fischer, S. mlr3extralearners: Extra Learners For mlr3. (2022).
70. Andersen, P. K. & Gill, R. D. Cox’s Regression Model for Counting Processes: A Large Sample Study. *Ann. Stat.* **10**, 1100–1120 (1982).

71. LeDell, E. *et al.* h2o: R Interface for the “H2O” Scalable Machine Learning Platform. (2022).
72. Baum, E. B. & Haussler, D. What size net gives valid generalization? *Neural Comput.* **1**, 151–160 (1989).
73. Vapnik, V. N. *The Nature of Statistical Learning Theory*. (Springer New York, 2000).
74. Cho, J., Lee, K., Shin, E., Choy, G. & Do, S. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv [cs.LG]* (2015).
75. Rokem, A., Wu, Y. & Lee, A. Assessment of the need for separate test set and number of medical images necessary for deep learning: a sub-sampling study. *bioRxiv* 196659 (2017) doi:10.1101/196659.
76. Balki, I. *et al.* Sample-Size Determination Methodologies for Machine Learning in Medical Imaging Research: A Systematic Review. *Can. Assoc. Radiol. J.* **70**, 344–353 (2019).
77. Wlodarska, M. *et al.* Indoleacrylic Acid Produced by Commensal Peptostreptococcus Species Suppresses Inflammation. *Cell Host Microbe* **22**, 25-37.e6 (2017).
78. Paris, A., Tardif, N., Galibert, M.-D. & Corre, S. AhR and Cancer: From Gene Profiling to Targeted Therapy. *Int. J. Mol. Sci.* **22**, (2021).
79. Toh, R. *et al.* 2-Aminobutyric Acid, a Potential Indicator of Oxidative Stress in Failing Hearts. *J. Card. Fail.* **20**, S186 (2014).
80. Irino, Y. *et al.* 2-Aminobutyric acid modulates glutathione homeostasis in the myocardium. *Sci. Rep.* **6**, 36749 (2016).

81. Wishart, D. S. *et al.* HMDB: the Human Metabolome Database. *Nucleic Acids Res.* **35**, D521-6 (2007).
82. Wishart, D. S. *et al.* HMDB: a knowledgebase for the human metabolome. *Nucleic Acids Res.* **37**, D603-10 (2009).
83. Wishart, D. S. *et al.* HMDB 3.0--The Human Metabolome Database in 2013. *Nucleic Acids Res.* **41**, D801-7 (2013).
84. Wishart, D. S. *et al.* HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Res.* **46**, D608–D617 (2018).
85. Wishart, D. S. *et al.* HMDB 5.0: the Human Metabolome Database for 2022. *Nucleic Acids Res.* **50**, D622–D631 (2022).
86. HYDROXYPROLINE EXCRETION IN CANCER. *JAMA* **198**, 1208–1209 (1966).
87. Powles, T. J., Leese, C. L. & Bondy, P. K. Hydroxyproline excretion in patients with breast cancer and response to treatment. *Br. Med. J.* **2**, 164–166 (1975).
88. Rømer, A. M. A., Thorseth, M.-L. & Madsen, D. H. Immune Modulatory Properties of Collagen in Cancer. *Front. Immunol.* **12**, 791453 (2021).
89. Senese, R. *et al.* 3,5-Diiodothyronine: A Novel Thyroid Hormone Metabolite and Potent Modulator of Energy Metabolism. *Front. Endocrinol.* **9**, 427 (2018).
90. Hummerich, H. & Soboll, S. Rapid stimulation of calcium uptake into rat liver by L-tri-iodothyronine. *Biochem. J* **258**, 363–367 (1989).
91. Moreno-Felici, J. *et al.* Phosphoenolpyruvate from Glycolysis and PEPCK Regulate Cancer Cell Fate by Altering Cytosolic Ca²⁺. *Cells* **9**, (2019).

92. Rossi, A., Pizzo, P. & Filadi, R. Calcium, mitochondria and cell metabolism: A functional triangle in bioenergetics. *Biochim. Biophys. Acta Mol. Cell Res.* **1866**, 1068–1078 (2019).
93. Cheong, J. E., Ekkati, A. & Sun, L. A patent review of IDO1 inhibitors for cancer. *Expert Opin. Ther. Pat.* **28**, 317–330 (2018).
94. *Global tuberculosis report 2021*. (World Health Organization, 2021).
95. Churchyard, G. *et al.* What We Know About Tuberculosis Transmission: An Overview. *J. Infect. Dis.* **216**, S629–S635 (2017).
96. *Handbook of Tuberculosis*. (Adis, Cham, 2017).
97. Fu, L. M. & Fu-Liu, C. S. Is Mycobacterium tuberculosis a closer relative to Gram-positive or Gram-negative bacterial pathogens? *Tuberculosis* **82**, 85–90 (2002).
98. Nelson, K. E. & Williams, C. M. *Infectious Disease Epidemiology: Theory and Practice*. (Jones & Bartlett Learning, 2013).
99. *Control of Communicable Diseases Manual*. (Amer Public Health Assn, 2014).
100. Pfyffer, G. E. & Wittwer, F. Incubation time of mycobacterial cultures: how long is long enough to issue a final negative report to the clinician? *J. Clin. Microbiol.* **50**, 4188–4189 (2012).
101. Dorman, S. E. *et al.* Xpert MTB/RIF Ultra for detection of Mycobacterium tuberculosis and rifampicin resistance: a prospective multicentre diagnostic accuracy study. *Lancet Infect. Dis.* **18**, 76–84 (2018).

102. Carr, W. *et al.* Interim Guidance: 4-Month Rifapentine-Moxifloxacin Regimen for the Treatment of Drug-Susceptible Pulmonary Tuberculosis - United States, 2022. *MMWR Morb. Mortal. Wkly. Rep.* **71**, 285–289 (2022).
103. Houben, R. M. G. J. & Dodd, P. J. The Global Burden of Latent Tuberculosis Infection: A Re-estimation Using Mathematical Modelling. *PLoS Med.* **13**, e1002152 (2016).
104. Cohen, A., Mathiasen, V. D., Schön, T. & Wejse, C. The global prevalence of latent tuberculosis: a systematic review and meta-analysis. *Eur. Respir. J.* **54**, (2019).
105. Drain, P. K. *et al.* Incipient and Subclinical Tuberculosis: a Clinical Review of Early Stages and Progression of Infection. *Clin. Microbiol. Rev.* **31**, (2018).
106. Berry, M. P. R. *et al.* An interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis. *Nature* **466**, 973–977 (2010).
107. Morgan, M. & Lori, S. *ExperimentHub: Client to access ExperimentHub resources.* (Bioconductor, 2022). doi:10.18129/B9.BIOC.EXPERIMENTHUB.
108. Gautier, L., Cope, L., Bolstad, B. M. & Irizarry, R. A. affy--analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* **20**, 307–315 (2004).
109. Xianyi, Z. *OpenBLAS: OpenBLAS is an optimized BLAS library based on GotoBLAS2 1.13 BSD version.* (Github).
110. Vaughan, D. & Dancho, M. *furrr: Apply Mapping Functions in Parallel using Futures.* (2021).
111. D’Agostino McGowan, L. & Bryan, J. *googledrive: An Interface to Google Drive.* (2021).

112. Bryan, J. googlesheets4: Access Google Sheets using the Sheets API V4. (2021).
113. Wickham, H. httr: Tools for Working with URLs and HTTP. (2020).
114. Smith *et al.* illuminaio: An open source IDAT parsing tool for Illumina microarrays. *F1000Research* vol. 2 (2013).
115. Ooms, J. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. *arXiv:1403.2805 [stat. CO]* (2014).
116. Ritchie, M. E. *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **43**, e47 (2015).
117. Irizarry, R. A., Gautier, L., Huber, W. & Bolstad, B. makecdfenv: CDF Environment Maker. (2021).
118. Carvalho, B. S. & Irizarry, R. A. A Framework for Oligonucleotide Microarray Preprocessing. *Bioinformatics* vol. 26 2363–2367 (2010).
119. MacDonald, J. W. pd.clariom.d.human: Platform Design Info for Affymetrix Clariom_D_Human. (2016).
120. MacDonald, J. W. pd.hta.2.0: Platform Design Info for Affymetrix HTA-2_0. (2017).
121. Carvalho, B. pd.huex.1.0.st.v2: Platform Design Info for Affymetrix HuEx-1_0-st-v2. (2015).
122. Carvalho, B. pd.hugene.1.0.st.v1: Platform Design Info for Affymetrix HuGene-1_0-st-v1. (2015).
123. Carvalho, B. pd.hugene.2.1.st: Platform Design Info for Affymetrix HuGene-2_1-st. (2015).

124. Winter, D. J. rentrez: an R package for the NCBI eUtils API. *The R Journal* vol. 9 520–526 (2017).
125. Henry, L. & Wickham, H. rlang: Functions for Base Types and Core R and “Tidyverse” Features. (2021).
126. Grosser, M. snakecase: Convert Strings into any Case. (2019).
127. Wickham, H. stringr: Simple, Consistent Wrappers for Common String Operations. (2019).
128. Henry, L. & Wickham, H. tidysselect: Select from a Set of Strings. (2021).
129. Wickham, H. & Bryan, J. usethis: Automate Package and Project Setup. (2021).
130. Wickham, H., Hester, J. & Ooms, J. xml2: Parse XML. (2020).
131. Tweedie, S. *et al.* Genenames.org: the HGNC and VGNC resources in 2021. *Nucleic Acids Res.* **49**, D939–D946 (2021).
132. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **49**, D10–D17 (2021).
133. Sayers, E. W., O’Sullivan, C. & Karsch-Mizrachi, I. Using GenBank and SRA. *Methods Mol. Biol.* **2443**, 1–25 (2022).
134. Ritchie, M. E. *et al.* A comparison of background correction methods for two-colour microarrays. *Bioinformatics* **23**, 2700–2707 (2007).
135. Silver, J. D., Ritchie, M. E. & Smyth, G. K. Microarray background correction: maximum likelihood estimation for the normal-exponential convolution. *Biostatistics* **10**, 352–363 (2009).

136. Irizarry, R. A. *et al.* Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.* **31**, e15 (2003).
137. Bolstad, B. M., Irizarry, R. A., Astrand, M. & Speed, T. P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**, 185–193 (2003).
138. Irizarry, R. A. *et al.* Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4**, 249–264 (2003).
139. Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316–319 (2017).
140. *sra-tools: SRA Tools*. (Github).
141. Ewels, P. A. *et al.* The nf-core framework for community-curated bioinformatics pipelines. *Nat. Biotechnol.* **38**, 276–278 (2020).
142. Ewels, P. *et al.* *nf-core/rnaseq: nf-core/rnaseq version 1.4.2*. (2019). doi:10.5281/zenodo.3503887.

CURRICULUM VITAE

