A Practical Guide to

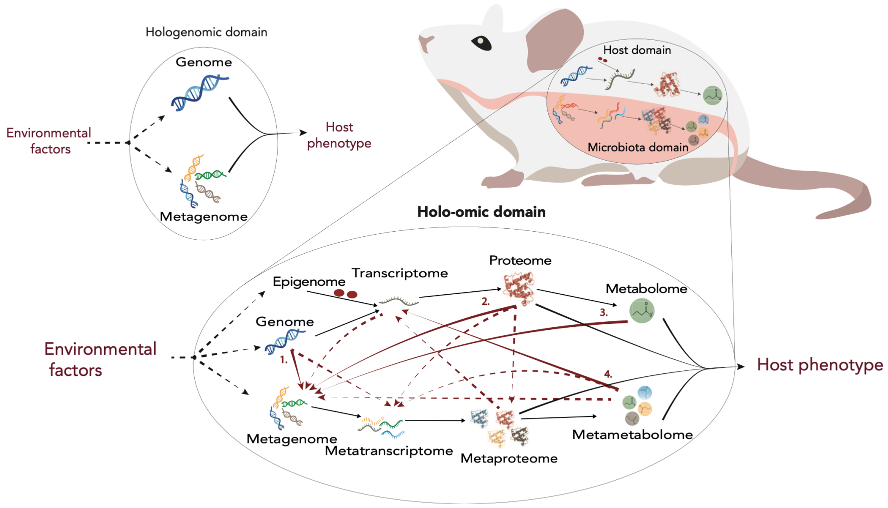
Holo-omics

April 28, 2023

CEH | CENTER FOR EVOLUTIONARY HOLOGENOMICS

Contents

About this guidebook



Holo-omics overview. Modified from Nyholm et al. 2020 [Nyholm et al., 2020]

The **practical guide to holo-omics** is a compilation of methodological procedures to generate, analyse and integrate holo-omic data, i.e., multi-omic data jointly generated from hosts and associated microbial communities [Nyholm et al., 2020, Limborg et al., 2018]. This guide extends the contents of the article **"A practical introduction to holo-omics"**, which aims at guiding researchers to the main critical steps and decision points to perform holo-omic studies. While the article focuses on discussing pros and cons of using multiple available options, the aim of this guide is to compile protocols and pipelines to be implemented by researchers. The **practical guide to holo-omics** is presented in two formats:

- Website (http://www.holo-omics.science/)
- PDF document (http://www.holo-omics.science/holo_omics_guidebook.pdf)

The guidebook is meant to be a live resource under continuous development, whose contents are updated, replaced and improved as technology and knowledge advances. So, changes are to be expected. However, frozen PDF versions will be released periodically to ensure traceability of the contents.

This guide is presented as a final output of the H2020 project HoloFood. More information about this EU Innovation Action that ran between 2019 and 2023 can be found in the HoloFood Website and the CORDIS website.

Contents

- **Introduction**: general information about holo-omics, employed data types and study design considerations.
- **Laboratory procedures**: methods and procedures for generating raw omic data of hosts and microbial communities.
- **Bioinformatic procedures**: methods and procedures for processing raw omic data into quantitative datasets to be analysed through statistics.
- **Statistical procedures**: methods and procedures for analysing and integrating holo-omic data.

Protocols, exercises and tutorials

This guide contains example data and bits of code (mostly shell and R) to reproduce data generation and analysis procedures. Code boxes look like the following:

```
shao4d_perm <- shao4d %>%
  tax_transform("identity", rank = "genus") %>%
  dist_calc("aitchison") %>%
  dist_permanova(
    variables = c("breed", "sex", "number_reads"),
    n_perms = 99, # you should use more permutations in your real analyses!
    n_processes = 1
  )
#> Dropping samples with missings: 15
#> 2022-11-24 01:15:20 - Starting PERMANOVA with 99 perms with 1 processes
#> 2022-11-24 01:15:21 - Finished PERMANOVA
```

Data sets

The data sets employed in the guidebook are derived from chicken and salmon intestinal samples produced in the H2020 project HoloFood. All the raw omic data, as well as relevant metadata and complementary information can be found in the **HoloFood Data Portal**.

About the authors

The authors of the **Practical Guide to Holo-omics** belong primarily to the Center for Evolutionary Hologenomics, at the Globe Institute of the University of Copenhagen (Denmark). The CEH is a research centre dedicated to studying host-microbiota interactions and their impact on basic and applied biological processes.

Antton Alberdi

Antton is an Associate Professor at the University of Copenhagen whose research is focused on understanding how animal-microbiota interactions shape basic and applied biological processes. Antton is the corresponding author of the **Practical Guide to Holo-omics**.

antton.alberdi@sund.ku.dk | www.alberdilab.dk

Morten Limborg

Morten is an Associate Professor at the University of Copenhagen and the Head of the Section for Hologenomics at the Globe Institute. His research focuses on host-microbiota interactions in aquaculture fish species.

Raphael Eisenhofer

Raphael is a postdoctoral researcher working under the supervision of Alberdi. His research activity is mainly focused on metagenomic analyses and related tool development, as well as marsupial microbiome analysis.

Iñaki Odriozola

Iñaki is a postdoctoral researcher working at the Alberdi lab. He implements his statistical background in community ecology to multi-omic data integration in host-microbiota systems.

Jacob A Rasmussen

Jacob is a postdoctoral researcher supervised by Limborg. He researches on fish-microbiota interactions using multi-omic tools, including genomics, metagenomics and metabolomics.

Protocol and script contributors

- Carlotta Pietroni, University of Copenhagen
- Jorge Langa, University of Copenhagen

Other relevant people

- Tom Gilbert (HoloFood project coordinator), University of Copenhagen
- Anna Fotaki (HoloFood project manager), University of Copenhagen

How to cite this work

Instructions to cite this work will be eventually added.

Acknowledgement

HoloFood
Hologenomics for sustainable food production
Horizon2020
Innovation action
2019-2022

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 817729.

Part I

INTRODUCTION

Chapter 1

Introduction to holo-omics

Why do we need holo-omics?

Every multicellular organism is a host ‘environment’ for which microbes pass through, persist, replicate, and/or influence the host phenotype. Evidence, collected from rainforest swamps to research labs, and from farm stables to patient bedsides, has made it clear that no fauna or flora live alone. Although each have their own peculiar characteristics, animals and plants are incontrovertible assemblages of multiple lifeforms. They compositionally form holobionts with their diverse microbial associates, whether they are transient or stably present [Theis et al., 2016]. Holobionts can thus change in time and space, and the collective gene catalog of a holobiont in turn forms a hologenome, which can yield variation in phenotypes with fitness, performance, or disease consequences. The prefix “holo” derives from the Greek word holos for entire or whole. Holobiont and hologenome are thus structural terms that help us view and study biological systems in an integrated community context, that are subject to diverse ecological and evolutionary forces with harmful, helpful, or harmless consequences [Rosenberg and Zilber-Rosenberg, 2013]. The terms also recognize that hosts often outsource or intertwine metabolism to stable or transient microbial associates, and that hosts have evolved a gradient of dependencies and antagonisms with microorganisms in or on their surfaces and surroundings across the plant and animalia kingdoms.

What is holo-omics?

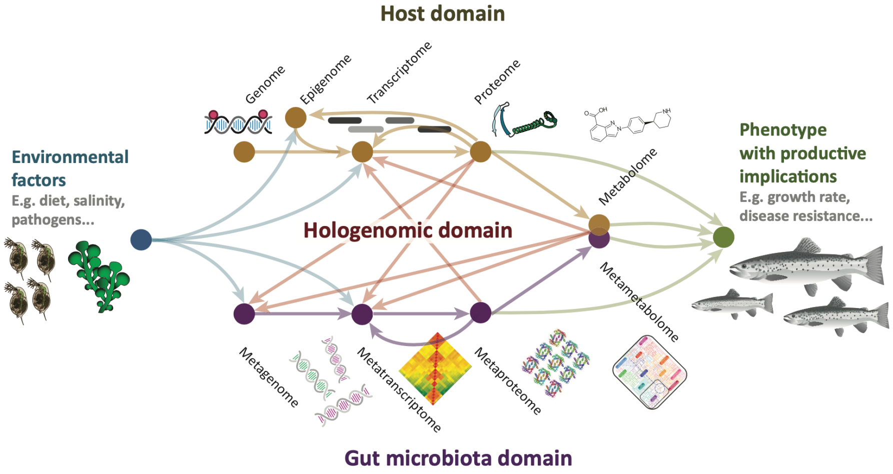
Holo-omics refers to the methodological approach that jointly generates and analyses multi-omic data from hosts and associated microbial communities [Nyholm et al., 2020]. Holo-omics leverage current knowledge and methods in the fields of molecular biology and microbiology into a novel framework integrating molecular data including genomes, transcriptomes, epigenomes, proteomes, and metabolomes for analyzing host organisms and their gut microbiota as interconnected and coregulated systems. The advantage of holo-omics is that it is supposed to overcome the limited functional insights of current analytical strategies by simultaneously considering the holobiont at multiple molecular levels. This involves deciphering interactions between not only the host genome but also its epigenome and transcriptome, as well as its microbial metagenome and metatranscriptome. Studies would ideally also incorporate analyses of the associated proteomes and metabolomes, and metaproteomes and metametabolomes, to fully recover the functional pathways controlling the observable phenotype of a host organism. Successful integration of such data into a holo-omic framework will reveal mechanisms such as how host genomes regulate the composition of the microbial community, or, conversely, how specific microbes interact to control host gene expression patterns. Finally, the holo-omic approach to study host-microbiota interactions relies on three major assumptions of the study system:

1. Host-associated microorganisms interact not only with each other but also with their host [Fischer et al., 2017].
2. These interactions affect, either positively or negatively, central biological processes of hosts and microorganisms [Wu and Wu, 2012].
3. The interplay can be traced using biomolecular tools.

Contents of this section were created by Antton Alberdi and Morten Limborg.

1.1 Omic layers

Nucleic acid sequencing and mass spectrometry technologies that enable tracking the biomolecular pathways linking host and microbial genomic sequences with biomolecular phenotypes by generating (meta)transcriptomes, (meta) proteomes, and (meta)metabolomes. The same technologies also enable epigenomic and exposomic profiling, which can further contribute to disentangling the biochemical associations between host-microbiota-environment interactions and their effect on host phenotypes.



Overview of omic layers. Modified from Limborg et al. 2018 [Limborg et al., 2018].

In this workbook we consider seven omic layers that require specific data generation and analysis strategies before integrating them into multi-omic statistical models:

- Nucleic acid sequencing-based
  - Host genomics - **HG**
  - Host transcriptomics - **HT**
  - Microbial metagenomics - **MG**
  - Microbial metatranscriptomics - **MT**
- Mass spectrometry-based
  - Host proteomics - **HP**
  - Microbial metaproteomics - **MP**
  - (Meta)metabolomics - **ME**

Acknowledging the distinct biological and structural characteristics of these seven omic layers is essential to design experiments and analytical pipelines for better solving the complex puzzle of host-microbiota interactions.

Host genomics (HG)

Host genomics refers to the study of an organism's genetic information (its genome) and how it relates to that organism's traits and characteristics. In the case of host genomics, we are specifically looking at the genetic information of a host organism, such as a human or an animal, and how that genetic information can influence things like susceptibility to certain diseases, response to treatments, and traits related to its microbiome composition and function.

Host transcriptomics (HT)

Host transcriptomics refers to the study of the full set of RNA molecules produced by an organism's cells (known as the transcriptome), and how it relates to that organism's traits and characteristics. In the case of host transcriptomics, we are specifically looking at the RNA molecules produced by the cells of a host organism, such as a human or an animal, and how those RNA molecules can influence things like gene expression, protein production, and how this affects the host's associated microbiome.

Microbial metagenomics (MG)

Microbial metagenomics is the study of genetic material from a mixed community of microorganisms, without the need to isolate and culture individual organisms. This approach involves extracting DNA from an environmental sample, including e.g. from the intestinal environment of a host organism, and sequencing it to obtain a snapshot of the genetic diversity and potential functions of the microbial community present in that sample. By analysing the genetic information obtained through metagenomics, researchers can gain insights into the metabolic capabilities, ecological roles, and evolutionary relationships of the microorganisms living in a particular environment.

Microbial metatranscriptomics (MT)

Microbial metatranscriptomics is the study of all the genetic material expressed as RNA transcripts by a community of microorganisms living in a particular environment. This approach allows researchers to understand which genes are active and which metabolic pathways are being used by the microorganisms in a specific ecosystem. Essentially, it involves analyzing the RNA molecules produced by the microorganisms in a sample to gain insight into their activities and behaviors.

Host proteomics (HP)

Host proteomics is the study of all the proteins produced by a specific host organism in response to various stimuli, including disease or infection. Proteins are the workhorses of the body and perform many important functions, such as regulating cell growth, repairing damaged tissues, and fighting infections. Proteomics techniques involve analysing the entire set of proteins, or proteome, of a particular organism or tissue sample, to understand how they are produced, modified, and interact with each other. The most popular method to generate proteomics data today is mass spectrometry which involves ionizing protein samples and analyzing the resulting ions based on their mass-to-charge ratio to identify the protein and its modifications.

Microbial metaproteomics (MP)

Microbial metaproteomics is the large-scale study of all the proteins produced by a community of microorganisms living in a particular environment such as a host organism. This approach involves extracting and analysing proteins from a mixed microbial sample, without the need to isolate and culture individual microorganisms. By identifying and quantifying the proteins present in the sample, microbial metaproteomics can provide insights into the functional roles and metabolic activities of the microorganisms in the community, as well as their interactions with each other and with their environment. These data are also often generated using mass spectrometry methods as described for host proteomics above.

(Meta)metabolomics (ME)

Host as well as Meta-metabolomics is the study of all the small molecules, or metabolites, produced by an organism and/or microbial community under different physiological conditions. These metabolites include molecules such as sugars, amino acids, and lipids, which are the building blocks and energy sources for cells. Metabolomics techniques involve the identification and quantification of these molecules using advanced analytical methods, such as mass spectrometry. By analysing the metabolic profile of an organism and/or microbial community, researchers can gain insights into the biochemical pathways and metabolic networks that regulate various physiological processes in e.g. the intestinal environment of animals.

Contents of this section were created by Morten Limborg.

Chapter 2

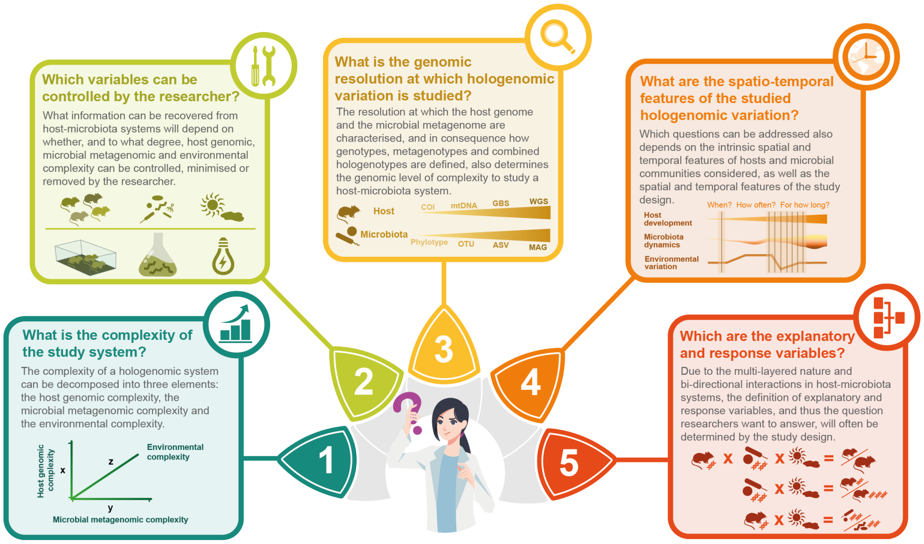
Study design considerations

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

Holo-omic approaches can be used to understand how the combined features of hosts and microorganisms shape biological processes relevant for hosts (such as adaptation), for microorganisms (such as meta-community dynamics) or both [Alberdi et al., 2022].

Depending on the aims and features of the study system, holo-omics can be implemented using different study designs, model systems and techniques. This landscape of possibilities is shaped around five essential questions that need to be considered when designing and interpreting hologenomic studies, which relate to five core topics:

1. **Hologenomic complexity**
2. **Control of variables**
3. **Molecular resolution**
4. **Spatiotemporal factors**
5. **Explanatory and response variables**



What is the complexity of the study system?

The complexity of a hologenomic system can be decomposed into three elements: the host genomic complexity, the microbial metagenomic complexity and the environmental complexity.

Which variables can be controlled by the researcher?

What information can be recovered from host-microbiota systems will depend on whether, and to what degree, host genomic, microbial metagenomic and environmental complexity can be controlled, minimised or removed by the researcher.

What is the genomic resolution at which hologenomic variation is studied?

The resolution at which the host genome and the microbial metagenome are characterised, and in consequence how genotypes, metagenotypes and combined hologenotypes are defined, also determines the genomic level of complexity to study a host-microbiota system.

What are the spatio-temporal features of the studied hologenomic variation?

Which questions can be addressed also depends on the intrinsic spatial and temporal features of hosts and microbial communities considered, as well as the spatial and temporal features of the study design.

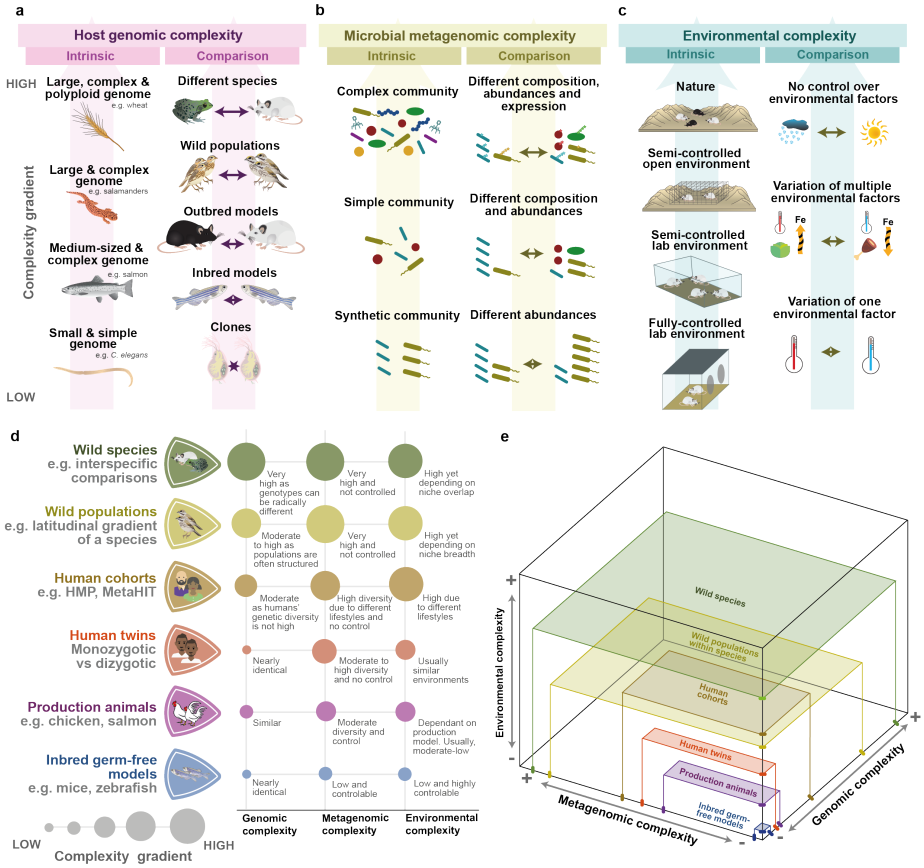
Which are the explanatory and response variables?

Due to the multi-layered nature and bi-directional interactions in host-microbiota systems, the definition of explanatory and response variables, and thus the question researchers want to answer, will often be determined by the study design.

2.1 Hologenomic complexity

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

Hologenomic complexity can be broadly defined as the amount of information relevant to the study that the biological system under analysis contains and it can be decomposed into three major elements: host genomic, microbial metagenomic and environmental complexity [Alberdi et al., 2022]. Within each of these elements, two sources of complexity can be defined: the intrinsic complexity of the system under study, including host genome size and number of bacterial genomes, and the complexity introduced by the degree of difference between the organisms under comparison such as gene expression differences versus distinct genomes.


a
Host genomic complexity
Intrinsic
Comparison
HIGH
Large, complex & polyploid genome
e.g. wheat
Different species
Wild populations
Large & complex genome
e.g. salamanders
Outbred models
Medium-sized & complex genome
e.g. salmon
Inbred models
Complexity gradient
Small & simple genome
e.g. C. elegans
Clones
LOW
b
Microbial metagenomic complexity
Intrinsic
Comparison
Complex community
Different composition, abundances and expression
Simple community
Different composition and abundances
Synthetic community
Different abundances
c
Environmental complexity
Intrinsic
Comparison
Nature
No control over environmental factors
Semi-controlled open environment
Variation of multiple environmental factors
Fe
Semi-controlled lab environment
Fe
Variation of one environmental factor
Fully-controlled lab environment
d
Wild species
e.g. interspecific comparisons
Very high as genotypes can be radically different
Very high and not controlled
High yet depending on niche overlap
Wild populations
e.g. latitudinal gradient of a species
Moderate to high as populations are often structured
Very high and not controlled
High yet depending on niche breadth
Human cohorts
e.g. HMP, MetaHIT
Moderate as humans' genetic diversity is not high
High diversity due to different lifestyles and no control
High due to different lifestyles
Human twins
Monozygotic vs dizygotic
Nearly identical
Moderate to high diversity and no control
Usually similar environments
Production animals
e.g. chicken, salmon
Similar
Moderate diversity and control
Dependant on production model. Usually, moderate-low
Inbred germ-free models
e.g. mice, zebrafish
Nearly identical
Low and controlable
Low and highly controlable
Genomic complexity
Metagenomic complexity
Environmental complexity
LOW
Complexity gradient
HIGH
e
+
Environmental complexity
-
Wild species
Wild populations within species
Human cohorts
Human twins
Production animals
Inbred germ-free models
+
Metagenomic complexity
-
-
Genomic complexity
+

Decomposition of hologenomic complexity. **(a-c)** The design and interpretation of hologenomic studies depend on the host genomic (part a), microbial metagenomic (part b) and environmental (part c) complexity of the system under study. Within each axis of complexity, two types of gradients can be defined based on whether the features are intrinsic to the system or introduced by the researcher through the selection of groups under comparison. **(d)** Six examples of study systems with different levels of genomic, metagenomic and environmental complexity. **(e)** Three-dimensional representation of the complexity of the examples. The area of the plain represents the combined host genomic and microbial metagenomic complexity of the system, while the height represents the environmental complexity. The combined three-dimensional volume represents the overall hologenomic complexity of the system. HMP: Human Microbiome Project.

2.2 Control of variables

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

Controlling the complexity of hologenomic variables is essential for addressing specific research questions. Broadly speaking, the more detailed and mechanistic the question under study, the greater the required control. For instance, research on specific biomolecular processes using laboratory models will require a higher level of control than studying biogeographical patterns of host–microbiota interactions in wild organisms. The control of hologenomic variables can be achieved through a number of strategies.

2.2.1 Controlling host genomes

The control over host genomic complexity largely depends on the model organisms studied and the technical approaches employed. In laboratory organisms that can reproduce asexually, such as water fleas (Daphnia, Crustacea) and Lamiaceae plants, absolute control over host genotypes can be achieved by using clonal organisms [Mushegian et al., 2019]. When clones cannot be used, inbred laboratory animals can provide a high level of genomic homogeneity. The use of groups of genetically homogeneous hosts allows the effects of contrasting environmental conditions or specific microbial comunities to be compared. Clonal and inbred models also enable the effects of a specific host genetic factor to be studied in a controlled genomic background through the application of targeted techniques for modulating gene expression (such as RNA-mediated interference) or for genomic engineering (such as CRISPR–Cas9). Working with humans and wild organisms does not enable such a degree of control over the genotypes studied unless in vitro models, such as organ-on-a-chip co-cultures of animal tissues and microbial communities, are generated62. When this level of control is not possible, coarse control over host genotypes can be achieved through contrasting animals from different populations or from closely related species63, while greater control can be achieved through comparing individuals across different degrees of kinship, such as monozygotic versus dizygotic twins38, and family members to other individuals64.

2.2.2 Controlling microbial metagenomes

Control over microbial metagenomic complexity is usually achieved through modulating microbial communities. Some strategies, such as modification of dietary regimes or the administration of microbiota-targeted additives or prebiotics, aim to modify microbial ecosystems by changing nutrient availability. However, unless compounds that match unique enzymatic capabilities of specific microorganisms are used, it is difficult to accurately modulate the microbiota owing to the complexity of ecological relationships among microorganisms. Alternative approaches to modify microbial communities include inoculation of target bacteria (such as probiotics) and faecal microbiota transplantation. The efficacy and accuracy of these methods is also variable; there is no guarantee that inoculated bacteria will establish or modulate the microbiota, while transplantation does not enable accurate control over the microbial community introduced or the secondary elements that are transplanted along with bacteria. These issues complicate interpretation of results; for example, bacteriophages transferred alongside bacteria may severely impact the gut microbiota composition. A higher level of control could potentially be achieved through transplanting synthetic microbial communities. While this approach has been successfully implemented in diverse in vitro setups the complexity of microbial communities still hinders its efficient use as a routine scientific procedure in live animals.

2.2.3 Controlling the environment

In most laboratory studies, environmental complexity is reduced so that no, or very few, environmental parameters (usually only experimental treatments) vary among groups and subjects. Climate chambers and aquaria enable experiments by providing absolute control of abiotic conditions, such as light/dark cycles, humidity and temperature variations. Outdoor common garden experiments do not provide full control over environmental factors, but they ensure the effect on the systems being compared is identical. Some natural systems can also provide special conditions that enable environmental features to be controlled, such as cuckoo nestlings that are bred by other birds or salmon populations that breed in the same rivers in alternating years. Research on wild organisms usually incorporates more complex and dynamic environmental conditions. When controlling them is not possible, collection of relevant environmental metadata to be incorporated as covariates in the statistical analyses is useful. A century of ecological research has revealed the advantages of each of these approaches. On the one extreme, laboratory microcosms allow the most reductive control. On the other extreme, studies in the macrocosm of the real world provide perspective on emergent properties of natural ecosystems that cannot be anticipated solely based on microcosms.

2.3 Molecular resolution

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

The complexity of a study system is not only determined by its inherent properties and study design, but also the techniques and procedures employed to analyse it. Researchers can decide how much a system is simplified by altering the resolution of the hologenomic features under study; in essence, zooming in or zooming out.

2.3.1 Resolution of host genotypes

In host-microbiota studies, host genotypes can be defined at different levels, including species, breeds, populations, strains, sex or individuals. Genotypes can be defined as categorical variables, without analysing the differences between them, or can be studied in more detail through considering their actual genetic content and establishing correlations among them. When using an evolutionary perspective, phylogenetic relationships between genotypes are established based on phylogenomic markers, which usually vary above population and species level, but not among individuals. This implies that genomic variability among the individuals included within each genotype is overlooked. Studying the effect of interindividual genomic variability on host-microbiota systems, such as identifying candidate host genomic variants associated with microbial features, requires a higher level of resolution. This is achieved through defining genotypes at the individual level, and using techniques based on whole genome resequencing that enable the complexity of host genomes to be screened at a much finer level, so that differences between the individuals contrasted are not only defined based on their kinship, but also the functional properties of their genomic variants. Currently, this approach requires high quality reference genomes from which high density SNP profiles of individuals can be generated, for example through SNPchip or resequencing studies. The genomic resolution could be further refined by incorporating structural variants, methylation patterns, or even, we hypothesise, chromosome 3D folding structure as revealed through techniques such as Hi-C. In doing so, researchers can identify associations between SNPs or gene variants and specific microbiota traits, such as the relative abundance of certain taxa or the enrichment of a given function, and thus identify mechanisms by which a host exerts control over composition and function of its associated microbiota

Resolution of microbial metagenotypes

The structure and resolution at which microbial metagenotypes are defined also affects the complexity of the metagenome under analysis. Metagenotypes can be defined as arrays of microbial taxa, microbial genes or a combination of both. The most common approach to define them is to rely on short marker sequences targeted for metabarcoding purposes, such as the 16S rRNA or the internal transcribed spacer (ITS). However, these procedures often do not enable reliable taxonomic assignment at genus or species level, do not capture strain level community dynamics, and are prone to generate biased functional inferences, as bacteria with identical marker genes (particularly those associated with wild taxa) might carry very different catalogues of genes. Thus, while useful for estimating microbial diversity and obtaining preliminary insights into functionality, targeted sequencing approaches do not provide conclusive evidence about the metabolic capabilities of the microbiota, particularly when working with non-human systems.

By contrast, if appropriate strategies and adequate sequencing depths are employed, shotgun metagenomics enables bacterial genome sequences to be recovered, from which genes can be predicted and annotated to create a gene catalogue that can define a metagenotype. However, these genes are not randomly distributed, but enclosed within genomes of specific bacteria or other microorganisms, with a particular combination of genes that shape their expression and the specific biological features (such as oxygen affinity, reproduction time, metabolic capacity) that determine their ecology. Hence, a more refined characterisation of microbial metagenotypes can be achieved through binning algorithms that enable bacterial genome reconstruction from metagenomic mixtures, yielding metagenome-assembled genomes (MAGs). Nevertheless, unless short-read sequencing is combined with long-read approaches, it is challenging to capture multi-copy genes such as the 16S rRNA marker gene 103, which is often employed in metabarcoding studies and therefore represents a useful link to a large number of existing studies. Machine learning-based solutions to link 16S rRNA marker gene sequences with MAGs are, however, being developed 104. Finally, regardless of the approach used to define the microbial metagenotype, the complexity of microbial communities will often require dimensionality reduction to increase statistical power 105,106. This can be achieved by defining co-abundance clusters, ecological guilds or more complex strategies that also consider temporal features of microbiota variation, such as compositional tensor factorisation.

Resolution of envirotypes

Characterisation of environmental factors that affect the host-microbiota system under study enable the definition of envirotypes, a term drawn from crop sciences that is useful for accounting for the environmental factors in the hologenomic context. Any different physical place, or place sampled at different time points, will be exposed to a different environment, as conditions will seldom be identical between two spatial and temporal points. Hence, the resolution at which the composite of environmental factors is considered will define whether these two environments will be considered different envirotypes or not. For example, if only considering water temperature, killer whales sampled in the Arctic and the Antarctic seas experience the same envirotype. However, if the biotic composition is also considered in the definition of the environment, the Arctic and the Antarctic will need to be split into two distinct envirotypes, as some killer whales will have access to penguins while others will not. The same principle applies to laboratory setups or mesocosm experiments: a temperature shift of 2-3 ºC might not be considered relevant under some experimental setups, while it can define different envirotypes under other study designs. Finally, failure to recognise environmental factors that affect host-microbiota interactions, and thus define relevant envirotypes, can lead to increased noise and decreased capacity to achieve statistical significance.

2.4 Spatiotemporal factors

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

Spatial factors

Spatial resolution. Microbial communities associated with animal and plant hosts vary not only across coarse body parts, but also at the micro-scale, such as between the lumen and the intestinal crypts. Thus, the resolution at which a body site is defined will also determine how a hologenomic system is characterised. For example, the animal gastrointestinal tract can be considered a single sampling unit, 4-5 units or hundreds of micro-units, depending on the sampling and data processing strategies employed. Naturally, each level of resolution will allow different questions to be addressed and will require the use of different technologies and analytical approaches.

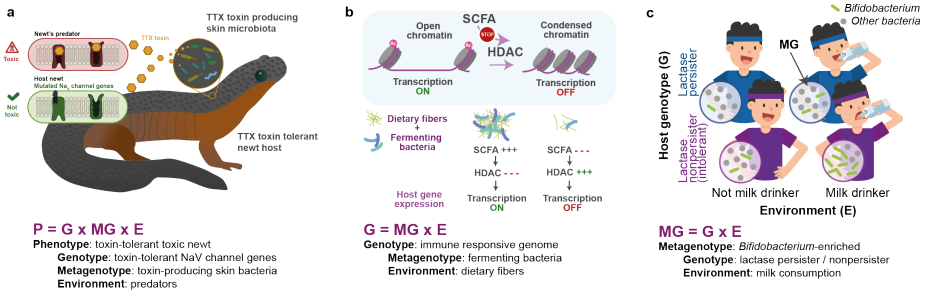
Temporal factors

Temporal features to be considered include when, how often, and for how long host-microbiota systems are to be analysed. When a host is first exposed to microbes with regard to temporal benchmarks (number of days or years) must be considered, as should the order in which it is exposed to them. Priority effects relate to how the order of species arrivals in an ecosystem shape the potential for subsequently arriving taxa to establish themselves. Although originally discussed at the macroorganismal level in the context of plant communities, the phenomenon is also relevant for building host-associated microorganism communities, for example as documented in the human gut. In addition, microbial communities are known to vary daily, seasonally and relative to life-stage patterns. Hence, the extent and frequency of sampling determine which of these dynamics will be observed or, conversely, missed. Finally, it is important to consider that the consequences of changes at one time period or life stage may appear only later in time, thus detection of such effects obviously requires that the subsequent period is also studied. For example, interventional animal experiments show that when the immune system develops early in life, there is a window of opportunity where the gut microbiota composition shapes the risk of developing diseases in the future.

2.5 Explanatory and response variables

The contents of this section have been extracted and modified from the article Disentangling host–microbiota complexity through hologenomics published in ***Nature Reviews Genetics*** in 2022 by the authors of the **Holo-omics Workbook**.

Host genomic and microbial metagenomic data generated under hologenomic setups can take on different roles when generating statistical models. While the environment is most often considered as an explanatory variable (though one can also study how the hologenome affects the environment), the host genome and the microbial metagenome are sometimes viewed as explanatory and sometimes as response variables, depending on the aim of the research. In many cases, directionality is set by the researcher rather than the biological system itself, as host-microbiota systems contain many bi-directional interactions and circular processes, which complicate the establishment of causal relationships. Here, we define three basic models in which the three main variables (genome, metagenome and environment) are assigned different roles to address different types of fundamental questions.


a
TTX toxin producing skin microbiota
Newt's predator
TTX toxin
Toxic
Host newt
Mutated Na_V channel genes
Not toxic
TTX toxin tolerant newt host
P = G x MG x E
Phenotype: toxin-tolerant toxic newt
Genotype: toxin-tolerant NaV channel genes
Metagenotype: toxin-producing skin bacteria
Environment: predators
b
Open chromatin
SCFA
STOP
HDAC
Condensed chromatin
Ac
Ac
Transcription ON
Transcription OFF
Dietary fibers + Fermenting bacteria
SCFA +++
HDAC - - -
Transcription ON
SCFA - - -
HDAC +++
Transcription OFF
Host gene expression
G = MG x E
Genotype: immune responsive genome
Metagenotype: fermenting bacteria
Environment: dietary fibers
c
Bifidobacterium
Other bacteria
MG
Host genotype (G)
Lactase persister
Lactase nonpersister (intolerant)
Not milk drinker
Milk drinker
Environment (E)
MG = G x E
Metagenotype: Bifidobacterium-enriched
Genotype: lactase persister / nonpersister
Environment: milk consumption

Examples of biological processes addressed by the different models of host-microbiota interactions. **a)** How does the hologenome shape animal phenotypes? Only the combination of specific host genomic (G) and microbial metagenomic (MG) features, probably developed due to a selective force exerted by the presence of predators (E) enables rough-skinned newts to have skin toxicity, an ecologically relevant phenotypic trait (P). **b)** How do the microbial metagenome and environment shape host genomic features? SCFA-producing bacteria along with a fibre-rich diet enhance chromatin accessibility and thus activate immune gene expression. **c)** How do the host genome and the environment shape microbial genomic features? Only the combination of a lactase nonpersister genotype combined with the milk-drinking envirotype generates a microbial metagenotype characterised by enrichment of Bifidobacterium.

Phenotype as a product of genotype, metagenotype and envirotype

This is the main model used when hologenomics is conducted to ascertain how genome-metagenome-environment interactions affect the biological properties of a host, such as disease susceptibility, performance or fitness. It is an especially common and relevant model for health, agricultural, and ecological and evolutionary research 19,125–127. One clear example of a phenotype shaped by host genomic, microbial metagenomic and environmental factors was recently reported for rough-skinned newts. The study showed that bacteria on the skin of the newts produce a deadly neurotoxin from which the newt is protected by mutations in five host genes that encode the NaV channels normally targeted by the toxin. Thus, this 'toxic newt' phenotype is the result of both host and microbial genes, which likely evolved under the pressure exerted by an environmental factor, namely the presence of predators.

Genotype expression influenced by metagenotype and envirotype

When studying how core host genomic features, which contribute to shaping phenotypes, are affected by the microbiota, host genomic features become the response variable. Unlike the microbial metagenome, the genome sequence of the host organism is not variable, but microorganisms can induce chromatin remodelling and DNA methylation, and thus modulate the bioactivity of molecular receptors and host gene expression. A well-studied pathway that links the microbiota with host gene expression involves modulation of the activity of host histone deacetylases (HDAC) by short chain fatty acids (SCFA) produced by intestinal microorganisms. HDACs remove histone lysine acetyl groups, which leads to chromatin condensation and transcriptional silencing of genes. Increased SCFA concentrations inhibit histone deacetylases, thereby enhancing chromatin accessibility and activating gene expression. A metagenotype with a higher capacity to produce SCFAs combined with an envirotype characterised as a fibre-rich diet (required to produce SCFAs), therefore contributes to boost immune response through activating host immune gene expression.

Metagenotype as a product of genotype and envirotype

This model assumes the inverse causal directionality between the host genome and microbial metagenome to that described above. Candidate host genes related to microbiota features can be identified through GWAS in which the metagenotype (or derived metrics such as diversity or abundance of specific microbial taxa, genes or metabolic functions) are treated as a phenotypic trait. For instance, the increased abundance of lactose degrader Bifidobacteria in humans has been shown to be associated with lactase nonpersister genotype and consumption of milk (envirotype). Once candidate genes are known, targeted analyses in which natural or human-controlled genomic variability (such as the number of copies of the amylase-encoding gene in humans) can be contrasted under controlled environmental conditions to ascertain the effect on metagenotypes (such as the abundance of Ruminococcaceae bacteria in the gut microbiota).

Part II

LABORATORY PROCEDURES

Chapter 3

About labwork

General considerations

Although the generation of each omic data layer requires dedicated protocols to be implemented, there are multiple general considerations that apply to all laboratory processes. In the following we list three of the most relevant ones.

External contamination

The risk of external contamination is a relevant issue that must be actively tackled when generating multi-omic data. External contamination refers to any molecule of interest that unintendedly is added to the sample, and analysed with the target molecules. As shotgun sequencing entails analysing all available nucleic acids in a sample, human saliva, skin microbiome, or microbes present in water and reagents are some of the sources of external contamination. Incorporating DNA and RNA from these sources can distort the biological signal, which can lead researchers into incorrect conclusions. The following measures contribute to minimise external contamination:

- Always wear gloves and work in sterile environments, such as clean laminar flow cabinets.
- Use filtered pipette tips.
- Separate pre-PCR and post-PCR laboratories.
- Process and sequence blank controls.

Internal cross-contamination

Another common type of contamination is that happening among samples. During the many pipetting actions laboratory protocols entail, is not uncommon to transfer small amounts of samples to adjacent tubes or wells. This can obviously distort the sample, and lead researchers into incorrect conclusions. The following measures contribute to minimise internal cross-contamination:

- Process all batches in an identical way for errors to be detectable.
- Avoid pipetting from the top of the tube to minimise sprays.
- Process and sequence blank controls.

Batch effects

The last global consideration is to be aware of batch effects and try to minimise or account for their impact in downstream analyses. Batch effects are almost unavoidable in holo-omic data generation, because samples are usually processed in batches. Each batch can suffer different types of technical biases, including the aforementioned contamination issues, but also other problems derived from the use of different reagent stocks, different researchers executing identical protocols in slightly different ways, or storing samples for variable time periods. The critical measure to minimise the impact of batch effects and account for them in downstream analysis is to randomise samples. Randomising means randomly assigning samples from different contrasting groups to the different batches, minimising correlation between batches and experimental groups. If this is done, laboratory batches can be included as covariates in statistical analyses, which enable accounting and controlling for batch effects in the final results.

Procedures for generating multi-omic data

This chapter contains sections dedicated to each of the omic layers included in the workbook.

- **Nucleic-acid sequencing-based approaches**
  - DNA/RNA extraction for **HG**, **HT**, **MG** and **MT**
  - Sequencing library preparation for **HG** and **MG**
  - Sequencing library preparation for **HT**
  - Sequencing library preparation for **MT**
- **Mass spectrometry-based approaches**
  - Protein extraction for **HP** and **MP**
  - Metabolite extraction for **ME**

Chapter 4

DNA/RNA extraction

Hundreds or (probably) thousands of different protocols and variations exist for extracting and purifying nucleic acids. Protocols can be classified based on methodological (e.g., chemical vs. physical DNA/RNA isolation, column-based vs bead-based, commercial vs. open-access).

Sample preprocessing

Bead-beating

Bead-beating is a mechanical disruption method performed before standard DNA extraction, where ceramic or glass beads are added to a tube containing microbial samples. Subsequently, moderate to high-speed shaking is applied to create collisions between the beads and samples. Bead-beating is widely used in microbial metagenomics studies for bacterial cell lysis, and various bead-beating protocols have been utilized to extract microbial DNA from stool samples. Literature has investigated the effects of different bead-beating techniques on downstream analyses [Zhang et al., 2021, Fiedorová et al. [2019]].

Freeze-heat shock

Temperature shocks are one of the most damaging processes for tissue, cell and DNA integrity. While such events are commonly avoided to preserve the quality of the samples, heat-shocks have been shown to improve nucleic acid extractions in various contexts. This is because freezing induces crystallisation of water inside cells which leads to destruction of cytoplasmic structures.

Tissue digestion

After tissue disaggregation, a typical approach involves treating samples with a detergent and salt (such as SDS) to rupture cell membranes and sometimes with enzymes for cellular and organelle disruption and the elimination of impurities. Proteinase K is a popular choice for DNA isolation from mammalian tissues and cells, whereas lyticase and lysozyme are enzymes used to break down the cell walls of yeast and bacteria and are commonly featured in microbial DNA extraction kits.

Chemical isolation

Once the DNA is released, proteins and other contaminants must be removed. When using chemical approaches, this is typically done by adding a precipitating agent like alcohols (e.g., ethanol) or salts (e.g., ammonium acetate). This process separates DNA and contaminants in different phases, which enables the contaminants to be removed from the sample, thus purifying the DNA. Purely chemical procedures for DNA isolation are becoming less common for the challenges they entail for high-throughput sample processing and automatisation.

Physicochemical isolation

Physicochemical procedures are the most commonly employed strategies for DNA and RNA isolation. Two main strategies exist, either column-based or bead-based isolation.

Column-based

The spin column-based nucleic acid purification method is a rapid solid-phase extraction technique that purifies nucleic acids. The principle underlying this method is that under specific ionic conditions, nucleic acids bind to the solid-phase silica. A binding buffer is used to establish the optimal pH or salt concentration required for DNA to bind to silica. The sample in the binding buffer is then transferred to a spin column, which is placed in a centrifuge or attached to a vacuum. The centrifuge or vacuum forces the solution through a silica membrane within the spin column, where the nucleic acids bind to the silica membrane while the rest of the solution flows through. Once the target material is bound, the flow-through can be discarded.

The next step involves washing the column by adding a new buffer, which typically contains alcohol, to maintain binding conditions while removing binding salts and any remaining contaminants. This process requires several washes, often with increasing concentrations of ethanol or isopropanol, until the nucleic acids on the silica membrane are free of contaminants. Finally, elution is the process of adding an aqueous solution to the column, allowing the hydrophilic nucleic acid to leave the column and enter the solution. This step can be enhanced by altering the salt, pH, time, or temperature. Finally, to collect the extract, the column is transferred to a clean microtube prior to a final centrifugation step.

Bead-based

Bead-based nucleic acid extractions are based on the magnetic properties of very small (20 to 30 nm) iron oxide particles that only display magnetic behaviour in the presence of an external magnetic field. Several types of magnetic beads with different binding properties exist, which can be used for DNA and RNA purification as well as proteins and other biomolecules, depending on their surface coatings and chemistries. For instance, streptavidin-coated magnetic beads are commonly used for nucleic acid extractions, due to their capacity to bind biotinylated ligands such as nucleic acids.

First, beads are mixed with the sample along with a binding buffer that provokes DNA to get attached to the magnetic beads. Subsequently, particles (with attached DNA) are dragged to an edge of the tube by the magnetic force of an external magnet, thus immobilising them. While the beads are immobilised, the rest of the sample is removed. The bead-bound DNA is retained during the washing steps, and finally released to an aqueous solution when the sample has been purified. The main advantage offered by bead-based strategies is their capacity to upscale and automatise, because there is no need for vacuum or centrifugation.

Available protocols

Hundreds of protocols, both open access and commercial, are currently available to extract nucleic acids from different types of samples.

Contents of this section were created by Antton Alberdi.

Chapter 5

Protein/metabolite extraction

To ensure accurate and reliable analysis, it is crucial to apply cold processing when dealing with highly volatile subjects such as small metabolites. This approach helps to minimize any potential biases in the composition. In order to obtain efficient precipitation, cell lysis techniques such as sonication or homogenisation of tissue by freezing, grinding, or bead-beating may be necessary. Additionally, it is important to filter tissue samples to remove any large debris, and purify the lysate for either high-performance (HP), medium-performance (MP), or low-performance (ME) analysis.

For ME, a suitable solvent is required for the precipitation of the metabolites, which should be chosen based on the desired detection spectrum of the metabolites. The polarity of the solvent can influence the target component of the ME. In contrast, for HP and MP, two main methods are currently used: acetone/TCA precipitation and phenol extraction.

To increase the detection of rare abundant proteins and metabolites, it is necessary to remove highly abundant metabolites or proteins. However, it is crucial to include pooled quality controls during data acquisition to detect all metabolites and correct stochastic drift.

Overall, to ensure the high quality of data for further processing, it is important to apply appropriate techniques for sample preparation and analysis of small metabolites. Proper purification and removal of contaminants can significantly improve the accuracy and reliability of results.

Chapter 6

Sequencing library preparation

Sequencing of DNA and RNA molecules require sequencing libraries to be prepared, which entails modifying original nucleic acid molecules to allow sequencing platforms to identify the target molecules and perform the sequencing.

Sequencing strategies and platforms

Library features are specific to each sequencing platform, which requires selecting in advance the sequencing strategy to be employed. Pure nucleic acid sequencing-based strategies can be broadly divided in two groups. Short-read sequencing (SRS) platforms provide large amounts of data yet with short sequencing reads (typically 150 nucleotides). In contrast, long-read sequencing (LRS) platforms yield much longer sequences (thousands or even million of nucleotides), yet with a lower throughput, and typically lower sequence quality. The SRS market is dominated by two main companies with proprietary platforms, namely Illumina and BGI, although PacBio recently released their own SRS platform called ONSO. The LRS market is also dominated by two different companies with proprietary technologies, which are Oxford Nanopore (ONT) and Pacific Biosciences (Pacbio).

Sequencing enterprises, as well as auxiliary biotechnological companies, provide library preparation kits that can be more or less customised for different purposes.

Technology	Platforms	Sequencing type	Company
Sequencing by synthesis (SBS)	MiSeq, NovaSeq	Short-read sequencing	Illumina

Technology	Platforms	Sequencing type	Company
Combinatorial probe-anchor synthesis (cPAS)	DNBSeq	Short-read sequencing	BGI
Sequencing by binding (SBB) technology	Onso	Short-read sequencing	PacBio
Single Molecule Real-Time sequencing (SMRT)	Sequel, Revio	Long-read sequencing	PacBio
Nanopore sequencing	MinION, GridION, PromethION	Long-read sequencing	Oxford Nanopore

Some of the most widely used sequencing technologies and platforms.

PCR-based vs. PCR-free library preparation

Sequencing library preparation procedures can be split into two main groups depending on whether they PCR-amplify or not the DNA templates. Unlike in the case of targeted amplicon sequencing, in which the objective is to amplify a specific target region, the aim of including a PCR step in shotgun-based library preparation is to increase the molarity of the library and/or to attach indices (see below) to the adaptors.

Learn more about PCR-based and PCR-free library preparation in this article by Jones et al. [Jones et al., 2015].

Indices and multiplexing

Usually, library preparation also entails tagging molecules with unique sample identifiers known as indices, which enable pooling molecules derived from multiple samples in a single sequencing run. This can be achieved in PCR-free protocols by using adaptors containing unique indices per sample, or by using indexed amplification primers in PCR-based library preparation protocols.

Learn more about indices and multiplexing in this article by Kircher et al. [Kircher et al., 2012].

Unique molecular identifiers (UMIs)

Unique molecular identifiers (UMIs) are a type of molecular barcoding that provides error correction and increased accuracy during sequencing by uniquely tag each molecule (rather than each pool of molecules derived from a sample) in a sample library. UMIs are used for a wide range of sequencing applications, many around PCR duplicates in DNA and cDNA. UMI deduplication is also useful for RNA-seq gene expression analysis and other quantitative sequencing methods.

Learn more about unique molecular identifiers in this article by Kivioja et al. [Kivioja et al., 2011].

Contents of this section were created by Antton Alberdi.

6.1 Host genomics and microbial metagenomics

The library preparation strategies for generating host genomic (HG) and microbial metagenomic (MG) data are generally the same.

DNA fragmentation

Short-read sequencing libraries requires DNA to be sheared to the desired fragment-length (usually 400-500 nucleotides), which can be achieved using either chemical (e.g., restriction enzymes) or physical (e.g. ultrasonication) procedures. Some long-read sequencing libraries intend to keep the largest DNA molecules possible, although some others recommend fragmenting to optimal mid-length molecules (e.g., around 10,000 nucleotides for Pacbio HiFi). After fragmentation, many library preparation protocols require repairing molecule ends by converting 5'-protruding and/or 3'-protruding ends to 5'-phosphorylated, blunt-end (see below) molecules.

Adaptor ligation

In shotgun libraries adaptors are merged to DNA template molecules through chemical ligation (e.g., using a ligase enzyme). The ligation process is slightly different depending on whether the DNA template has blunt- or sticky-ends. In blunt ends, both strands are of equal length – i.e. they end at the same base position, leaving no unpaired bases on either strand, while in sticky ends, one strand is longer than the other. Some protocols deliberately create sticky-ends from blunt-end fragmented DNA molecules by adding a single adenine base to form an overhang by an A-tailing reaction. This A overhang allows adapters containing a single thymine overhanging base to pair with the DNA fragments.

An example of a blunt-end molecule:

```
5'-GATCTGACTGATGCGTATGCTAGT-3'
3'-CTAGACTGACTACGCATACGATCA-5'
```

An example of a sticky-end molecule:

```
5'-GATCTGACTGATGCGTATGCTAGT-3'
3'-CTAGACTGACTACGCATACGATC-5'
```

List of available protocols

Type	Name	Author/owner	Protocol/Article
SRS	Blunt-End Single-Tube (BEST) library prep protocol	Open access	Article
SRS	Santa Cruz Reaction (SCR) single-stranded library prep protocol	Open access	Article
SRS		Illumina	Protocol
LRS	SMRTbell prep kit 3.0 for PacBio HiFi Sequencing	Pacbio	Protocol

6.2 Host transcriptomics

Library preparation for host transcriptomics (HT) requires some extra steps to the already described procedures for host genomics and microbial metagenomics. This is due to two main reasons. First, because RNA molecules cannot be directly built into most sequencing libraries, and require instead to generate complementary DNA (cDNA) before library preparation. Second, because gene transcripts tend to be overwhelmingly dominated by rRNA and mtDNA genes, which are often not of interest for the researcher.

Sample quality assessment

Before starting any library preparation protocol assessing the quality of RNA samples is strongly recommended. While traditionally assessed through agarose gel electrophoresis, nowadays RNA quality assessment is performed on electropherogram profiles, which are produced by nucleic acid fragment analysis instruments (e.g. Bioanalyzer, Fragment Analyzer). Traditionally, a simple model evaluating the 28S to 18S rRNA ratio was used as a criterion for RNA quality. However, the most common metric currently employed for assessing the preservation quality of RNA is the RNA integrity number (RIN), which accounts for more RNA features for assessing sample quality [Schroeder et al., 2006]. RIN values range from 10 (intact RNA) to 1 (totally degraded RNA). For example, the poly(A) enrichment procedures explained below require high quality RNA (RIN > 8), because RNA degradaation to breaks within the transcript body and due to the selection of the poly(A) tail, the 3' ends are enriched while the more 5' sequences would not be captured, leading to a strong 3' bias for degraded RNA inputs.

DNA removal

Depending on the RNA extraction method employed, it is not rare trace amounts of genomic DNA (gDNA) to be co-purified with RNA. Contaminating gDNA can interfere with reverse transcription and may lead to false positives, higher background, or lower detection in sensitive applications such as RT-qPCR. The traditional method of gDNA removal is the addition of DNase I to RNA extracts. DNase I must be removed prior to cDNA synthesis since any residual enzyme would degrade single-stranded DNA. Unfortunately, RNA loss or damage can occur during DNase I inactivation treatment. As an alternative to DNase I, double-strand–specific DNases are available to eliminate contaminating gDNA without affecting RNA or single-stranded DNAs.

Stranded vs. non-stranded transcriptomics

RNA-Seq libraries can be stranded or non-stranded (unstranded), a decision that affects data analysis and interpretation. Stranded RNA-Seq (also referred to as strand-specific or directional RNA-Seq) enables researchers to determine the orientation of the transcript, whereas this information is lost in non-stranded, or standard, RNA-Seq. Non-stranded RNA-Seq is often sufficient for measuring gene expression in organisms with well-annotated genomes, as with a reference transcriptome, it is possible to infer orientation for most of the sequencing reads. As there are fewer steps than stranded library preparation, the benefits of this approach are lower cost, simpler execution, and greater recovery of material, which renders non-stranded RNA-Seq the preferred option for holo-omic analyses. In contrast, stranded RNA-Seq is useful if the aims include annotating genomes, identifying antisense transcripts or discovering novel transcripts.

cDNA conversion

Most RNA-Seq experiments are carried out on instruments that sequence DNA molecules, rather than RNA. This implies that RNA conversion to cDNA is a required step before library preparation. The synthesis of cDNA from an RNA template is carried out via reverse transcription using reverse transcriptases. In nature, these enzymes convert the viral RNA genome into a complementary DNA (cDNA) molecule, which can integrate into the host's genome, among other processes.

Reverse transcription, similar to PCR, requires the use of primers. Two main types of primers:

- **Random primers**: this type of primers are oligonucleotides with random base sequences. They are often six nucleotides long and are usually referred to as random hexamers. While random primers help improve cDNA synthesis for detection, they are not suitable for full-length reverse transcription of long RNA. Increasing the concentration of random hexamers in reverse transcription reactions improves cDNA yield but results in shorter cDNA fragments due to increased binding at multiple sites on the same template

- **oligo(dT) primers**: this type of primers consist of a stretch of 12–18 deoxythymidines that anneal to poly(A) tails of eukaryotic mRNAs (see the section below for further details).

Reverse transcription reactions for cDNA library construction and sequencing involve two main steps: first-strand cDNA synthesis and second-strand cDNA synthesis.

- **First-strand cDNA synthesis**: this initial step generates a cDNA:RNA hybrid through the below-described three-step process.
  - **Primer annealing**: in this step primers are attached to the RNA template, which usually happens before reverse transcriptase and necessary components (e.g., buffer, dNTPs, RNase inhibitor) are added.
  - **DNA polymerisation**: in this step the complementary DNA is polymerised by the reverse transcriptase enzyme. With oligo(dT) primers (Tm ~35–50°C), the reaction is often incubated directly at the optimal temperature of the reverse transcriptase (37–50°C), while random hexamers typically have lower Tm (~10–15°C) due to their shorter length. Using a thermostable reverse transcriptase allows, a higher reaction temperature (e.g., 50°C), to help denature RNA with high GC content or secondary structures without impacting enzyme activity. With such enzymes, high-temperature incubation can result in an increase in cDNA yield, length, and representation.
  - **Enzyme deactivation**: in this final step temperature is increased to 70–85°C, depending upon the thermostability of the enzyme, to deactivate the reverse transcriptase.
- **Second-strand cDNA synthesis**: in this second step the first-strand cDNA is used as a template to generate double-stranded cDNA representing the RNA targets. Synthesis of double-stranded cDNA often employs a different DNA polymerase to produce the complementary strand of the first cDNA strand.

rRNA depletion through poly-A enrichment

Ribosomal RNA (rRNA) helps translate the information in messenger RNA (mRNA) into protein. It is the predominant form of RNA found in most cells, which can make over 80% of cellular RNA despite never being translated into proteins itself. In consequence, most reads derived from RNA belong to rRNAs, unless depletion strategies are implemented.

Excessively abundant rRNA sequences can be depleted using multiple strategies, which are covered in the Microbial metatranscriptomics. The most broadly employed enrichment strategy when dealing with eukaryotic organisms is rRNA depletion through poly-A enrichment. This strategy relies on the fact that mature coding mRNAs of eukaryotic organisms contain polyA tails, long chains (tens to hundreds) of adenine nucleotides that are added to primary RNA transcripts to increase the stability of the molecule. However, not all transcripts contain poly(A) tails. microRNAs, small nucleolar RNAs (snoRNAs), transfer RNAs (tRNAs), some long non-coding RNAs (lncRNAs), and even protein-coding mRNAs such as histone mRNAs do not contail poly(A) tails, thus will be removed together with rRNA during poly(A) selection. If interested in quantifying expression of such transcripts the use of alternative methods is recommended.

The most broadly employed strategies to deplete rRNA through poly-A enrichment rely either on hybridisation with Oligo(dT)-attached magnetic beads or oligo(dT) priming during cDNA conversion step. In the former strategy, poly(A)-containing RNA molecules hybridise with Oligo(dT) stretches attached to magnetic beads. Following hybridisation, the supernatant consisting of non-polyadenylated molecules is removed. The beads are washed prior to elution of the poly(A)-selected RNA in water or buffer.

List of available protocols

Type	Name	Author/owner	Protocol/Article
oligo(dT) hybridisation oligo(dT) priming	Dynabeads Oligo (dT)25-61005	Thermo Fisher	Protocol

Contents of this section were created by Antton Alberdi.

6.3 Microbial metatranscriptomics

Sequencing library preparation for microbial metatranscriptomics faces the same challenges as host genomics, but the fact that prokaryotic mRNA have no poly-A tails makes it impossible to apply oligo(dT)-based rRNA depletion strategies. There are three other alternatives through which prokaryotic rRNA can be depleted. These three strategies require designing oligos, probes or guides whose sequences complement the DNA sequences that should be removed. Most commercial kits contain probes designed to remove rRNA sequences of the most commonly employed animal hosts (Human/Mouse/Rat), as well as bacteria, but custom probes targeting any genes could be employed. The first two methods shown below are implemented before library-preparation, thus independent reactions must be ran for each sample. The last strategy is implemented after library preparation, which enables multiple indexed libraries to be pooled, and thus performing a single reaction per pool.

Capture-based rRNA depletion

This method relies on capture rRNA with complimentary oligos that are coupled to paramagnetic beads. Unwanted transcripts get bound to beads, which can then be retained using a magnet, while the non-hybridasing transcripts remain in the elution.

RNAse-based rRNA depletion

A more recent technological upgrade to capture-based rRNA depletion is to, instead of using paramagnetic beads, degrade RNA:DNA hybrids using RNase H [Huang et al., 2020].

CRISPR/Cas9-based rRNA depletion

The newest method of all three relies on the DNA claveage capacity of the Cas9 enzyme [Gu et al., 2016]. In this method, custom-designed guides are used for the Cas9 enzyme to cleveage unwanted sequences. This strategy is applied once libraries are prepared, and before the final PCR amplification is conducted. When the targetted molecules are cleaveged, they lack one of the two adaptors, and therefore they are not amplified, resulting in a considerable depletion compared to the rest of the library.

List of available protocols

Name	Strategy	Author/owner	Protocol/Article
Custom capture-based depletion	Capture-based	Open Source	Article [Kraus et al., 2019]
Legacy Ribo-Zero	Capture-based	Illumina	
Custom RNAse-based depletion	RNAse-based	Open Source	Article [Huang et al., 2020]
Ribo-Zero Plus	RNAse-based	Illumina	
NEBNext® rRNA Depletion Kit	RNAse-based	NEB	
DASH	Cas9-based	Open Source	Article [Prezza et al., 2020]

Contents of this section were created by Antton Alberdi.

Part III

BIOINFORMATIC PROCEDURES

Chapter 7

About bioinformatics

Bioinformatic processing of raw sequencing and mass spectrometry data is the computational step that precedes statistical analyses and integration of multi-omic data. Through bioinformatic processing raw data are converted into meaningful bits of information, usually drastically decreasing the size of the data sets that are used for downstream analyses.

Raw sequencing and mass spectrometry-based data files used in holo-omic analyses are typically in the realm of gigabytes (Gb) or even terabytes (Tb). Many of the performed operations require large amounts of memory (some more than 1Tb), which makes it impossible to process data in personal computers. Instead, most bioinformatics tasks are performed in computational clusters with access to large amounts of memory and many CPUs and GPUs, which enable parallelising computational tasks thus speeding up data processing time.

However, for the sake of simplicity and practicality, the example datasets included in this Workbook have been considerably downscaled to enable reproducing the exercises in personal computers.

All bioinformatic analyses included in the **Holo-omics workbook** are conducted in a Unix command line Shell environment (BASH/SH). You can find the details to set-up your SHELL environment in the section Prepare your Shell environment.

7.1 Prepare your shell environment

If the comment chunks of the code (text after #) is creating you problems, use the following code to disable interactive comments and avoid issues when copy-pasting code:

```
setopt interactivecomments
```

Required software

Bioinformatic pipelines for processing omic data require the use of dozens of software. All the required software are listed in the conda environment installation file available here.

Install conda / miniconda

Conda is an open-source package management system and environment management system that quickly installs, runs, and updates packages and their dependencies. If **conda** is not installed in your system, the first step is to install **miniconda** (a free minimal installer for conda, enough to run the bioinformatic analyses explained in the workbook). Miniconda installers for Linux, Mac and Windows operating systems can be found in the following website: https://docs.conda.io/en/latest/miniconda.html

Once conda or miniconda is installed in your system, you should be able to create and manage your conda environments. You can test whether conda has been succesfully installed using the following code:

```
conda -V
#> conda 22.11.1 #or whatever version you have installed
```

Install mamba (optional)

An optional step is to install **mamba**, which is a reimplementation of the conda package manager in C++, which speeds up many of the processes. Mamba can be installed through the command line using the `conda install` option.

```
conda install mamba -n base -c conda-forge
```

Create a conda environment

All the bioinformatic analyses explained in this workbook will be run within an environment containing all the necessary software. The file that specifies which software to install in the environment is available here, and can be retrieved using wget (as shown in the code below), or downloading from the Internet browser. If using the latter option, don't forget to provide the absolute path to the 'holo-omics-env.yaml' file in the `mamba create` command.

```
wget https://raw.githubusercontent.com/holo-omics/holo-omics.github.io/main/bin/holo-omics-env.y
conda update conda #to ensure everything is updated
conda deactivate #deactivate any conda environment before creating a new one
conda env create -f holo-omics-env.yml
rm holo-omics-env.yaml #remove installer file
```

As the environment contains dozens of softwares, the process of creating it will take a while. It is recommended to have a good Internet connection to speed-up software download. Once the installation is over, you can double-check whether the environment has been successfully created using the following script:

```
conda activate holo-omics
#> (holo-omics) anttonalberdi@Anttons-MBP ~ %
```

The (holo-omics) specifies the environment you are at. To get out of the environment use:

```
conda env list
#> base                  *  /Users/anttonalberdi/miniconda3
#> holo-omics               /Users/anttonalberdi/miniconda3/envs/holo-omics
```

Activate the holo-omics conda environment

Whenever running the holo-omic analyses explained in this workbook, it will be necessary to activate the holo-omics environment through the following command:

```
conda activate holo-omics
#> (holo-omics) anttonalberdi@Anttons-MBP ~ %
```

Install software in conda environment

Once the environment is activated, you can install the required software using the Conda package manager. For example, to install **metawrap**, run the following command:

```
conda activate holo-omics
conda install -y -c ursky metawrap-mg=1.2.2
```

7.2 Using snakemake for workflow management

Snakemake is a workflow management system that helps automate the execution of computational workflows. It is designed to handle complex dependencies between the input files, output files, and the software tools used to process the data. Snakemake is based on the Python programming language and provides a simple and intuitive syntax for defining rules and dependencies.

Here is a brief overview of how Snakemake works and its basic usage:

1. **Define the input and output files:** In Snakemake, you define the input and output files for each step in your workflow. This allows Snakemake to determine when a step needs to be executed based on the availability of its inputs and the freshness of its outputs.
2. **Write rules:** Next, you write rules that describe the software tools and commands needed to process the input files into the output files. A rule consists of a name, input and output files, and a command to run.

3. **Create a workflow:** Once you have defined the rules, you create a workflow by specifying the order in which the rules should be executed. Snakemake automatically resolves the dependencies between the rules based on the input and output files.
4. **Run the workflow:** Finally, you run the workflow using the snakemake command. Snakemake analyzes the input and output files and executes the rules in the correct order to generate the desired output files.

```
rule count_reads:
    input:
        "reads/sample1.fastq.gz",
        "reads/sample2.fastq.gz"
    output:
        "counts.txt"
    shell:
        "fastqc {input} -o {output} -f fastq"

rule trim_reads:
    input:
        "reads/sample1.fastq.gz",
        "reads/sample2.fastq.gz"
    output:
        "trimmed/sample1.trimmed.fastq.gz",
        "trimmed/sample2.trimmed.fastq.gz"
    shell:
        "trimmomatic SE {input} {output} -threads 4"

rule align_reads:
    input:
        "trimmed/sample1.trimmed.fastq.gz",
        "trimmed/sample2.trimmed.fastq.gz"
    output:
        "aligned.bam"
    shell:
        "bwa mem -t 4 genome.fa {input} | samtools view -Sb - > {output}"

rule call_variants:
    input:
        "aligned.bam"
    output:
        "variants.vcf"
    shell:
        "freebayes -f genome.fa {input} > {output}"

workflow:
    rule count_reads
    rule trim_reads
    rule align_reads
    rule call_variants
```

To run this workflow, save the code to a file named Snakefile and execute the following command in your terminal:

```
snakemake
```

Chapter 8

Sequencing data preprocessing

The first step of the bioinformatic pipeline is to pre-process the raw sequencing data to prepare them for downstream analyses.

Preprocess the reads using fastp

Raw sequencing data require an initial preprocessing to get rid off low-quality nucleotides and reads, as well as any remains of sequencing adaptors that can mess around in the downstream analyses. An efficient way to do so is to use the software **fastp**, which can perform all above-mentioned operations in a single go and directly on compressed files.

fastP documentation can be found here.

```
fastp \
    --in1 {input.r1i} --in2 {input.r2i} \
    --out1 {output.r1o} --out2 {output.r2o} \
    --trim_poly_g \
    --trim_poly_x \
    --low_complexity_filter \
    --n_base_limit 5 \
    --qualified_quality_phred 20 \
    --length_required 60 \
    --thread {threads} \
    --html {output.fastp_html} \
    --json {output.fastp_json} \
    --adapter_sequence {params.adapter1} \
    --adapter_sequence_r2 {params.adapter2}
```

Splitting host and non-host data

Depending on the sample type employed for data generation, sequencing data might contain only host reads, only microbial reads, or a mixture of both. For example, blood sampled from an animal is expected to only contain host DNA/RNA reads (unless an infection is ongoing), while DNA extracted from a microbial culture is only expected to contain microbial DNA/RNA reads (unless human contamination has happened). In contrast, intestinal content samples, faecal samples, leave samples or root samples can contain both host and microbial nucleic acids.

Index host genome

In order to map metagenomic reads to a reference host genome, it is necessary to index the genome. An index is a data structure that allows for efficient searching of the reference genome by breaking it down into smaller, more manageable pieces. Without an index, aligning reads to a reference genome would be prohibitively slow, especially for large genomes. Bowtie2 is a popular software tool for aligning reads to a reference genome, and it requires an index of the reference genome before alignment can be performed. Bowtie2 uses an index based on the Burrows-Wheeler transform (BWT) algorithm, which enables it to efficiently align reads to the reference genome. Here are the basic command to create a Bowtie2 index for a reference genome:

```
bowtie2-build \
    --large-index \
    --threads {threads} \
        {input.genome} {output.index}
```

Map samples to host genomes

The next step is to map the reads against the reference genome, followed by a split between reads that have been mapped (in the example below are retained in a BAM/SAM file) and the reads that were not mapped (in the example below outputed to fastq files). The mapped reads can be used for performing population genomic analyses, while the unmapped reads can be used for metagenomic analyses.

```
# Map reads to the reference genome using Bowtie2
bowtie2 \
    --time \
    --threads {threads} \
    -x {indexed.genome} \
    -1 {input.r1i} \
    -2 {input.r2i} \
| samtools view -b -@ {threads} - | samtools sort -@ {threads} -o {output.all_bam} -

# Extract non-host reads
samtools view -b -f12 -@ {threads} {output.all_bam} \
| samtools fastq -@ {threads} -1 {output.non_host_r1} -2 {output.non_host_r2} -

# Send host reads to BAM
samtools view -b -F12 -@ {threads} {output.all_bam} \
| samtools sort -@ {threads} -o {output.host_bam} -
```

Chapter 9

Host genomics (HG) data processing

Bioinformatic procedures for host genomics can be grouped in two main tasks, the generation of the reference genome and the resequencing of the genome. The first one aims at creating a (often single) reference genome sequence for a species, a breed or a population, which is then used to generate genomic profiles for multiple individuals. Reference genomes generation requires a considerably higher sequencing effort than resequencing, and the use of multiple sequencing technologies (long-read, Hi-C, etc.) is often needed to resolve the structural complexities of most eukaryotic organisms. Genome resequencing, in contrast, usually relies on short-read sequencing, which provide sufficient information for calling nucleotide variarions.

Overviews of both procedures are shown in the following chapters:

- **Host reference genome**
- **Host genome resequencing**

9.1 Host reference genome

Genomes of eukaryotic organisms are generally complex, because they carry multiple copies of the same genome, genomes contain duplications, repetitive sequences, mobile elements, etc. In consequence, generating a high-quality reference genome that represents all this complexity is a complex effort, that today, requires multiple complementary molecular techniques to be merged. Although multiple genome assembly protocols exist, in this guidebook we will focus on the one employed in the Vertebrates Genomes Project, the largest consortium aiming at generating animal reference genomes in a standardised way [Rhie et al., 2021]. The VGP assembly pipeline uses data generated by a variety of technologies, including PacBio HiFi reads, Bionano optical maps, and Hi-C chromatin interaction maps.

9.1.1 Genome quality

Before advancing with genome generation procedures, it is important to acknowledge that reference genomes can have different qualities. Quality is measured by assembly statistics, such as the N50 and L90 metrics, which provide an overview of the completeness and accuracy of the genome. Based on those metrics, eukaryotic genomes are usually categorised in three levels:

Contig level: Contig level refers to the lowest level of genome assembly, where the genome is fragmented into small pieces called contigs. Contigs are contiguous sequences of DNA that are typically hundreds to thousands of base pairs in length. Contig-level genome assemblies lack information about the order and orientation of the contigs and may contain gaps between them. **Scaffold level**: Scaffold level is the next level of genome assembly, where contigs are linked together using paired-end reads or other genomic information to form larger structures called scaffolds. Scaffolds provide information about the order and orientation of contigs but may still contain gaps between them. **Chromosome level**: Chromosome level is the highest level of genome assembly, where the genome is fully assembled into chromosomes. Chromosome-level assemblies provide the most complete and accurate representation of the genome, with few gaps and accurate order and orientation of genomic elements. These assemblies typically require multiple sources of genomic information and sophisticated computational tools to produce.

9.1.2 Genome profile analysis

Gathering metrics on genome properties before initiating a de novo genome assembly project is very helpful in setting expectations for the assembly. In the past, DNA flow cytometry was commonly used to estimate genome size, but computational approaches have become the preferred method in recent times [Wang et al., 2020]. Currently, genome profiling is based on k-mer frequency analysis, which not only provides information on the genome's complexity, such as its size and levels of heterozygosity and repeat content, but also on the quality of the data.

k-mer spectra can be generated with Meryl, which generates k-mer profile by decomposing the sequencing data into k-length substrings, counting the occurrence of each k-mer and determining its frequency.

```
#Create a k-mer database
meryl count k=31 mer=both output reads.meryl threads=4 \
     input reads_1.fastq reads_2.fastq

#Generate a k-mer spectrum
meryl histogram reads.meryl > reads.hist
```

The k-mer histogram produced by Meryl can be used to deduce genome properties with the help of GenomeScope2. This tool utilises a nonlinear least-squares optimisation to fit a combination of negative binomial distributions, providing estimates for genome size, repetitiveness, and heterozygosity rates [Ranallo-Benavidez et al., 2020].

```
./genomescope2.pl -k 31 -i reads.hist -o reads_genomescope
```

9.1.3 Genome assembly using hifiasm

Hifiasm is a powerful de novo assembler specifically developed for PacBio HiFi reads. One of the key advantages of hifiasm is that it allows us to resolve near-identical, but not exactly identical, sequences, such as repeats and segmental duplications [Cheng et al., 2021]. Hifiasm can be run in multiple modes depending on data availability:

Solo mode

The solo mode generates a pseudohaplotype assembly, resulting in a primary and an alternate assembly solely using HiFi reads.

Hi-C-phased mode

The Hi-C-phased mode generates a hap1 assembly and a hap2 assembly, which are phased using the Hi-C reads from the same individual.

Trio mode

The trio mode requires long-read PacBio HiFi reads from child, and Illumina short-reads from both parents to generate a maternal assembly and a paternal assembly, which are phased using reads from the parents.

9.1.4 Assembly evaluation

Assemblies can be evaluated using a variety of approaches that assess different parameters of the assembled genomes.

gfastats can be used or summary statistics (e.g., contig count, N50, NG50, etc.)

BUSCO assesses genome completeness based on an evolutionary functional perspective. BUSCO genes are anticipated to exist in a single-copy haplotype for a particular clade, and their presence, absence, or duplication can help researchers determine whether an assembly is deficient in significant regions or has multiple copies, which may necessitate purging [Simão et al., 2015].

Merqury performs a reference-free assessment of assembly completeness and phasing based on k-mers. Merqury compares k-mers in the reads to the k-mers found in the assemblies, as well as the copy number (CN) of each k-mer in the assemblies [Rhie et al., 2020].

9.1.5 Assembly scaffolding

The following step in the process is to assemble contigs into scaffolds, i.e., to connect contigs interspaced with gaps. While traditionally, this process has been performed using paired-end short-read data with long insert-sizes, the VGP pipeline currently scaffolds using two more advanced technologies: Bionano optical maps and Hi-C data.

Scaffolding using Bionano optical maps

Content to be added.

Scaffolding using Hi-C data

Content to be added.

9.1.6 Final genome evaluation

Content to be added.

Contents of this section were created by Antton Alberdi.

9.1.7 Reference genome annotation

Content to be added.

9.2 Host genome resequencing

Once a reference genome is available, short-read sequencing data can be used for generating single nucleotide polymorphism (SNP) data. Although multiple options exists, the pipeline below describes a typical workflow to process data using Bowtie2 for read mapping, Picard for marking duplicates, and GATK for performing variant calling. The resulting SNP data can be used for a wide range of downstream analyses, such as identifying genetic variants associated with diseases, studying population genetics, and performing genome-wide association studies (GWAS). The pipeline is customisable and can be modified to suit the specific needs of the researcher, such as changing the parameters of the tools used or incorporating additional analysis steps. Overall, this pipeline is a powerful tool for investigating genetic variation in genomes and can provide valuable insights into the genetic basis of various biological processes.

The first step is to map the reads agains the reference genome:

```
bowtie2 -x reference_genome_index \
    -1 forward_reads.fq \
    -2 reverse_reads.fq \
    -S mapped_reads.sam
```

If the mapping file is saved to an uncompressed SAM file, this should be compressed, and sorted for downstream analyses.

```
samtools view -bS mapped_reads.sam > mapped_reads.bam
samtools sort mapped_reads.bam -o sorted_mapped_reads.bam
```

Picard can be then used to mark duplicates in the sorted BAM file.

```
java -jar picard.jar MarkDuplicates \
      INPUT=sorted_mapped_reads.bam \
      OUTPUT=dedup_sorted_mapped_reads.bam \
      METRICS_FILE=metrics.txt VALIDATION_STRINGENCY=LENIENT
```

The deduplicated BAM file without redundant reads must be done indexed before starting the variant calling.

```
samtools index dedup_sorted_mapped_reads.bam
```

GATK4 is the used to perform local realignment around indels.

```
gatk --java-options "-Xmx4g" IndelRealigner \
      -R reference_genome.fa \
      -I dedup_sorted_mapped_reads.bam
      -O realigned_reads.bam \
      -targetIntervals intervals.list
```

Then, base quality score recalibration is performed using GATK4.

```
gatk --java-options "-Xmx4g" BaseRecalibrator \
      -R reference_genome.fa
      -I realigned_reads.bam
      --known-sites known_snps.vcf \
      -O recal_data.table
```

Subsequently, base quality score recalibration is applied to the.

```
gatk --java-options "-Xmx4g" ApplyBQSR \
    -R reference_genome.fa
    -I realigned_reads.bam
    --bqsr-recal-file recal_data.table \
    -O recal_reads.bam
```

Chapter 10

Microbial metagenomics (MG) data processing

Microbial metagenomic data processing can be conducted following different strategies. Decision on which approach to use should be based on the aims of the study, available reference data, amount of generated data, and many other criteria. In this workbook we consider three main approaches that require different bioinformatic pipelines to be implemented.

- **Reference-based approach:** it relies on a reference database of microbial genomes to which sequencing reads can be mapped to obtain estimations of relative proportion of reads belonging to each of the genomes available in the reference database. It is the simplest and computationally less expensive approach, yet it completely relies on a complete and representative reference database.
- **Assembly-based approach:** it is based on assembling sequencing reads into longer DNA sequences known as contigs, which can then be used to predict genes and perform functional analyses. The main limitation of this approach is that the entire metagenome (set of contigs) in each sample is considered as a single unit, thus overlooking which bacterial genome each detected gene belongs to.
- **Genome-resolved approach:** it is the most advanced of the three approaches, and the strategy that provides the largest amount of information, as the aim of this approach is to directly reconstruct all the genomes in a metagenome. This is achieved by binning contigs into Metagenome-Assembled Genomes (MAGs), which can then be taxonomically and functionally annotated to perform sound community-level analyses.

Contents of this section were created by Antton Alberdi.

10.1 Reference-based

Reference-based metagenomics is the approach that aims at characterising metagenomes based on existing genome sequences that are used as reference. This approach is meaningful when the microbial community under study is well known, such as the human microbiome, or when a reference MAG catalogue has been generated from a subset of samples under analysis. When the genome catalogue used as a reference has not been generated from the same environment (e.g., using human gut microorganisms as reference for vulture gut microbiomes), there are two major risks. The first one is that some of the microorganisms present in the studied environment might not be represented in the reference catalogue, which results in diversity underestimations. The second one is that the microorganisms present in the reference catalogue are similar but not identical to the ones in the target sample, which might result in incorrect taxonomic and functional inferences. It is therefore important to keep these caveats in mind when performing reference-based metagenomic analyses, mainly when dealing with non-model host organisms.

10.2 Assembly-based

Assembly-based approaches for processing metagenomic data are based on assembling sequencing reads into longer DNA sequences known as contigs, which can then be used to predict genes and perform functional analyses. The main limitation of this approach is that the entire metagenome (set of contigs) in each sample is considered as a single unit, thus overlooking which bacterial genome each detected gene belongs to. Assembly-based approaches can be divided in two main strategies:

- Individual assembly-based
- Coassembly-based

Individual assembly-based

Two of the most popular metagenome assemblers are **Megahit** and **MetaSpades**. Metaspades is considered superior in terms of assembly quality, yet memory requirements are much larger than those of Megahit. Thus, one of the most relevant criteria to choose the assembler to be employed is the balance between amount of data and available memory. Another minor, yet relevant difference between both assemblers is that Megahit allows removing contings below a certain size, while MetaSpades needs to be piped with another software (e.g. bbmap) to get rid off barely informative yet often abundant short contigs.

Individual assembly using Megahit

```
megahit \
    -t {threads} \
    --verbose \
    --min-contig-len 1500 \
    -1 {input.r1} -2 {input.r2} \
    -o {params.workdir}
    2> {log}
```

Individual assembly using MetaSpades

```
metaspades.py \
    -t {threads} \
    -k 21,33,55,77,99 \
    -1 {input.r1} -2 {input.r2} \
    -o {params.workdir}
    2> {log}

# Remove contigs shorter than 1,500 bp using bbmap
reformat.sh \
    in={params.workdir}/scaffolds.fasta \
    out={output.assembly} \
    minlength=1500
```

Assembly statistics using Quast

The metagenome assemblies can have very different properties depending on the amount of data used for the assembly, the complexity of the microbial community, and other biological and technical aspects. It is therefore convenient to obtain some general statistics of the assemblies to decide whether they look meaningful to continue with downstream analyses. This can be easily done using the software **Quast**.

```
quast \
    -o {output.report} \
    --threads {threads} \
    {input.assembly}
```

Coassembly-based

Coassembly is the process of assembling input files consisting of reads from multiple samples, as opposed to performing an independent assembly for each sample, where the input would only include reads from that particular sample. Coassembly has several advantages, such as increased read depth, simplified comparison across samples by utilizing a single reference assembly for all, and frequently, a better capability to recover genomes from metagenomes by obtaining differential coverage information. However, it can also limit the capacity to recover strain-level variation.

Coassembling multiple samples does not require special assemblers, but only preparing the input files in the correct way to enable assemblers to perform the assembly over multiple samples. An example for metaspades is shown below:

```
#Concatenate input reads into a single big input file
cat {input.reads}/*_1.fq.gz > {params.r1_cat}
cat {input.reads}/*_2.fq.gz > {params.r2_cat}

# Run metaspades
metaspades.py \
    -t {threads} \
    -k 21,33,55,77,99 \
    -1 {params.r1_cat} -2 {params.r2_cat} \
    -o {params.workdir}
    2> {log}

# Remove contigs shorter than 1,500 bp using bbmap
reformat.sh \
    in={params.workdir}/scaffolds.fasta \
    out={output.Coassembly} \
    minlength=1500
```

Note that the genome-resolved metagenomic approach also relies on assemblies or co-assemblies, but downstream binning procedures are explained in the **genome-resolved approach** section.

Gene annotation

Gene annotation refers to the process of identifing and assigning function to genes present in an assembly. In the first step, protein-coding and other types of genes are identified using tools such as Prodigal based on structural information of the DNA sequences. These software also predict the protein sequences these genes are expected to yield, which are then used to assign functions by contrasting them with functionally annotated reference databases. Due to the amount of reference databases available, it is common practice to match the genes against multiple databases and yield multiple annotations per gene. Currently, multiple tools exist that perform all these procedures in a single pipeline, such as DFAST [Tanizawa et al., 2017] and DRAM [Shaffer et al., 2020]. DFAST annotates genes against the TIGRFAM and Clusters of Orthologous Groups (COG) databases, while DRAM performs the annotation using Pfam, KEGG, UniProt, CAZY and MEROPS databases.

```
DRAM.py annotate \
      -i {input.assembly} \
      -o {outdir} \
      --threads {threads} \
      --min_contig_size 1500
```

The procedure for annotating MAGs, which is explained in the **genome-resolved approach** section, is identical to this one, with the only difference that the a MAG or a set of MAGs are used as input data rather than a metagenomic assembly.

Read mapping

The aim of assembly-based analyses is often to obtain gene-abundance information per studied sample, to characterise the functional properties of the entire metagenome as a whole (as opposed to the **genome-resolved approach** approach, in which functional attributes are assigned to each MAG). This requires reads from each sample to be mapped against the sequence of all protein-coding genes identified during the annotation process. All gene prediction software and annotation pipelines produce a FASTA file only containing gene sequences, which is used as the reference material.

The gene catalogue needs to be indexed before the mapping.

```
bowtie2-build \
      --large-index \
      --threads {threads} \
       {all_genes}.fa.gz
```

Then, the following step needs to be iterated for each sample, yielding a BAM mapping file for each sample.

```
bowtie2 \
      --time \
      --threads {threads} \
      -x {all_genes} \
      -1 {input.r1} \
      -2 {input.r2} \
      | samtools sort -@ {threads} -o {output}
```

Or relative abundance per gene per sample.

```
coverm genome \
      -b {params.BAMs}/*.bam \
      -s ^ \
      -m relative_abundance \
      -t {threads} \
      --min-covered-fraction 0 \
      > {output.mapping_rate}
```

Contents of this section were created by Antton Alberdi.

10.3 Genome-resolved

Genome-resolved metagenomics aims at recovering near-complete bacterial genomes from metagenomic mixtures. It relies on the assembling and read-mapping procedures explained in **assembly-based approach** section, which is followed by a binning procedure to produce the so-called Metagenome-Assembled Genomes (MAGs).

Note that entire suites and pipelines are available for conducting all the steps outlined in this section, and often more. Some of them include:

- Anvi'o
- metaWRAP
- ATLAS

Binning

Metagenomic binning is the bioinformatic process that attempts to group metagenomic sequences by their organism of origin {Goussarov}. In practice, what binning does is to cluster contigs of a metagenomic assembly into putative bacterial genomes. In the last decade over a dozen of binning algorithms have been released, each relying on different structural and mathematical properties of the input data.

Two of the most relevant structural properties to group contigs into bins are oligonucleotide composition of contigs and present of universally conserved genes in contigs. MaxBin, for example, relies on such universally conserved genes to initialize clusters, which are then expanded using the oligonucleotide composition of contigs. Besides structural attributes of contigs, the main quantitative measure used for binning is differential coverage, which is computed by counting the number of reads from different samples mapped to the assembly. This information is used by binning algorithms CONCOCT and MetaBat, for example.

Metabat and Maxbin require a depth file to be generated first.

```
jgi_summarize_bam_contig_depths \
    --outputDepth {output.depth} \
    {input.assemblybampath}
```

Example code for launching metabat2.

```
metabat2 \
    -i {input.assemblypath} \
    -a {input.depth} \
    -o {output.basepath} \
    -m 1500 \
    -t {threads} \
    --unbinned
```

Example code for launching MaxBin.

```
run_MaxBin.pl \
    -contig {input.assemblypath} \
    -abund {input.depth} \
    -out {output.basepath} \
    -thread {threads}
```

Bin refinement

The performance of the binning algorithms is largely dependent on the specific properties of each sample. A software that performs very well with a given sample can be easily outcompeted by another one in the next sample. In consequence, many researchers opt for ensemble approaches whereby assemblies are binned using multiple algorithms, followed by a refinement step that merges all generated information to yield consensus bins. This final step is ofter referred to as "bin refinement", and can be performed using tools like metaWRAP [Uritskiy et al., 2018] or Dastool [Sieber et al., 2018]. Several benchmarking studies have shown that such ensemble approaches are usually better than individual binning tools.

The following code can be used to run an ensemble binning using metaWRAP.

```
metawrap binning -o {params.outdir} \
    -t {threads} \
    -m {params.memory} \
    -a {params.assembly} \
    -l 1500 \
    --metabat2 \
    --maxbin2 \
    --concoct \
```

The following code can be used to refine binds using metaWRAP.

```
metawrap bin_refinement \
    -m {params.memory} \
    -t {threads} \
    -o {params.outdir} \
    -A {params.concoct} \
    -B {params.maxbin2} \
    -C {params.metabat2} \
    -c 70 \
    -x 10
```

Bin quality assessment

Metagenomic binning is a powerful yet complex procedure that yields many bins that do not properly represent bacterial genomes. It is therefore essential to assess the quality of those bins before considering them representative of bacterial genomes. The two main parameters used for bin assessment are completeness and contamination. Completeness refers to the fraction of a given bacterial genome estimated to be represented in the bin, while contamination refers to the proportion of the bin estimated to belong to a different genome. The most commonly employed software to assess bin quality is CheckM, which yields completeness and contamination metrics based on single-copy core genes.

Based on completeness and contamination metrics, a group of experts proposed some community standards to classify bins according to their quality and establish minimum quality requirements for considering a bin as a MAG [Bowers et al., 2017].

Bin curation

Contamination is an issue that in certain cases can be minimised by curating bins. The Anvi'o suite [Eren et al., 2015] provides a powerful visual interface to manually curate bins by dropping contigs that display distinct features (e.g., taxonomic annotation, coverage, GC%) to the rest of the contigs included in a bin. GUNC provides a way to implement a similar curation step in a more automatised manner [Orakov et al., 2021].

Dereplication

Dereplication is the reduction of a set of MAGs based on high sequence similarity between them [Evans and Denef, 2020]. Although this step is neither essential nor meaningful in certain cases (e.g., when studying straing-level variation or pangenomes), in most cases it contributes to overcome issues such as excessive computational demans, inflated diversity or inspecific read mapping. If the catalogue of MAGs used to map sequencing reads to (see read mapping section below) contains many similar genomes, read mapping results in multiple high-quality alignments. Depending on the software used and parameters chosen, this leads to sequencing reads either being randomly distributed across the redundant genomes or being reported at all redundant locations. This can bias quantitative estimations of relative representation of each MAG in a given metagenomic sample.

Dereplication is based on pairwise comparisons of average nucleotide identity (ANI) between MAGs. This implies that the number of comparisons scales quadratically with an increasing amount of MAGs, which requires for efficient strategies to perform dereplication in a cost-efficient way. A popular tool used for dereplicating MAGs is dRep [Olm et al., 2017], which combines the fast yet innacurate algorithm MASH with the slow but accurate gANI computation to yield a fast and accurate estimation of ANIs between MAGs. An optimal threshold that balances between retaining genome diversity while minimising cross-mapping issues has been found to be 98% ANI.

Taxonomic annotation

Although not necessary for conducting most of the downstream analyses, taxonomic annotation of MAGs is an important step to provide context, improve comparability and facilitate result interpretation in holo-omic studies. MAGs can be taxonomically annotated using different algorithms and reference databases, but the Genome Taxonomy Database (GTDB) [Parks et al., 2022] and associated taxonomic classification toolkit (GTDB-Tk) [Chaumeil et al., 2022] have become the preferred option for many researchers.

Functional annotation

Functional annotation refers to the process of identifying putative functions of genes present in MAGs based on information available in reference databases. As explained in the **assembly-based approach**, the first step is to predict genes in the MAGs (unless these are available from the assembly), followed by functional annotation by matching the protein sequences predicted from the genes with reference databases. Currently, multiple tools exist that perform all these procedures in a single pipeline, such as DFAST [Tanizawa et al., 2017] and DRAM [Shaffer et al., 2020]. DFAST annotates genes against the TIGRFAM and Clusters of Orthologous Groups (COG) databases, while DRAM performs the annotation using Pfam, KEGG, UniProt, CAZY and MEROPS databases.

```
DRAM.py annotate \
      -i {input.MAG} \
      -o {outdir} \
      --threads {threads} \
      --min_contig_size 1500
```

These functional annotations can be used for performing functional gene enrichment analyses, distilling them into genome-inferred functional traits, and many other downstrean operations explained in the statistics part.

Read mapping

When the objective of a genome-resolved metagenomic analysis is to reconstruct and analyse a microbiome, researchers usually require relative abundance information to measure how abundant or rare each bacteria was in the analysed sample. In order to achieve this, it is necessary to map the reads of each sample back to the MAG catalogue and retrieve mapping statistics. The procedure is identical to that explained in the assembly read-mapping section, yet using the MAG catalogue as a reference database rather than the metagenomic assembly. This procedure usually happens in two steps. In the first step, reads are mapped to the MAG catalogue to generate BAM or CRAM mapping files. In the second step, these mapping files are used to extract quantitative read-abundance information in the form of a table in which the amount of reads mapped to each MAG in each sample is displayed.

First, all MAGs need to be concatenated into a single file, which will become the reference MAG catalogue or database.

```
cat {MAG.path}/*.fa.gz > {all_MAGs}.fa.gz
```

The MAG catalogue needs to be indexed before the mapping.

```
bowtie2-build \
      --large-index \
      --threads {threads} \
       {all_MAGs}.fa.gz
```

Then, the following step needs to be iterated for each sample, yielding a BAM mapping file for each sample.

```
bowtie2 \
      --time \
      --threads {threads} \
      -x {all_MAGs} \
      -1 {input.r1} \
      -2 {input.r2} \
      | samtools sort -@ {threads} -o {output}
```

Finally, CoverM can be used to extract the required stats, such as covered fraction per MAG per sample.

```
coverm genome \
      -b {input} \
      -s ^ \
      -m count covered_fraction length \
      -t {threads} \
      --min-covered-fraction 0 \
      > {output.count_table}
```

Or relative abundance per MAG per sample.

```
coverm genome \
      -b {params.BAMs}/*.bam \
      -s ^ \
      -m relative_abundance \
      -t {threads} \
      --min-covered-fraction 0 \
      > {output.mapping_rate}
```

Contents of this section were created by Antton Alberdi.

Chapter 11

Host transcriptomics (HT) data processing

The analysis of host transcriptomic data can be conducted following two main strategies, which depend on whether a well-annotated reference genome that contain all gene sequences of the studied transcripts is available or not. This is seldom the case in non-model organisms that lack complete reference genomes with high-quality annotation of genetic information, although many reference genome generation initiatives are rapidly increasing the number of available reference genomes. The pros and cons of using either approach have been addressed in the literature [Lee et al., 2021].

In the following two chapters, you will find example pipelines to process host transcriptomic data through both strategies:

- **Reference-based host transcriptomics (HT) data processing**
- **Reference-free host transcriptomics (HT) data processing**

11.1 Reference-based host transcriptomics (HT) data processing

Quality-filtering

Fastp is a high-performance FASTQ preprocessor that can be used to clean up raw sequencing reads from Illumina platforms. It provides various quality control, filtering, and trimming options to remove low-quality bases, contaminants, and adapter sequences. The code provided performs a number of these steps, including trimming of poly-G and poly-X tails, which are commonly observed in Illumina reads, filtering reads based on quality and length, and removing adapter sequences. The resulting cleaned reads are then written to the specified output files. Additionally, fastp provides comprehensive quality control reports in both HTML and JSON formats, which can be used to assess the quality of the input reads and the impact of the processing steps. Overall, pre-processing raw sequencing reads with fastp is a critical step in ensuring the accuracy and reliability of downstream bioinformatics analyses.

```
fastp \
      --in1 {input.read1} --in2 {input.read2} \
      --out1 {output.read1} --out2 {output.read2} \
      --trim_poly_g \
      --trim_poly_x \
      --low_complexity_filter \
      --n_base_limit 5 \
      --qualified_quality_phred 20 \
      --length_required 60 \
      --thread {threads} \
      --html {output.fastp_html} \
      --json {output.fastp_json} \
      --adapter_sequence AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
      --adapter_sequence_r2  AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT
```

Ribosomal RNA removal

Ribodetector can be used for efficient removal of rRNA sequences from host transcriptomics data, improving accuracy and reducing computational time. This tool helps to better identify transcripts and understand gene expression in complex microbiomes.

```
ribodetector_cpu \
      -t 24 \
      -l 150 \
      -i {input.r1} {input.r2} \
      -e rrna \
      -o {output.non_rna_r1} {output.non_rna_r2}
```

Reference genome indexing

In order to use STAR for host transcriptomics, it is neccesary to first generates a genome index, which can be used for multiple RNA-seq experiments.

```
STAR \
      --runMode genomeGenerate \
      --runThreadN {threads} \
      --genomeDir {input} \
      --genomeFastaFiles {input}/*.fna \
      --sjdbGTFfile {input}/*.gtf \
      --sjdbOverhang {params.readlength}
```

Read mapping against reference genome

STAR (Spliced Transcripts Alignment to a Reference) is a fast and efficient method for aligning RNA-seq reads to a reference genome. It uses a two-pass alignment approach to detect spliced transcripts, improve accuracy and speed up the alignment process.

```
STAR \
      --runMode alignReads \
      --runThreadN {threads} \
      --genomeDir {params.genome} \
      --readFilesIn {input.read1} {input.read2} \
      --outFileNamePrefix {wildcards.sample} \
      --outSAMtype BAM SortedByCoordinate \
      --outReadsUnmapped Fastx \
      --readFilesCommand zcat \
      --quantMode GeneCounts
```

Contents of this section were created by Antton Alberdi and Raphael Eisenhofer.

11.2 Reference-free host transcriptomics (HT) data processing

Contents will be added shortly.

Chapter 12

Microbial metatranscriptomics (HT) data processing

Metatranscriptomics is a powerful tool for investigating gene expression patterns in complex microbial communities. It allows researchers to explore the functional diversity of microbial populations, providing insights into the metabolic pathways and interactions that drive community dynamics. The analysis of microbial metatranscriptomic data can be conducted following two main strategies, which depend on whether a reference catalogue of annotated bacterial genomes is available or not.

In the following two chapters, you will find example pipelines to process microbial metatranscriptomic data through both strategies:

- **Reference-based microbial metatranscriptomics (MT) data processing**
- **Reference-free microbial metatranscriptomics (MT) data processing**

12.1 Reference-based microbial metatranscriptomics (MT) data processing

In reference-based metatranscriptomics, sequencing reads are aligned to a reference genome or transcriptome, allowing for the identification and quantification of transcripts from known genes. This approach provides a more focused analysis of gene expression in microbial communities and can be particularly useful when studying well-characterized microbial systems.

Quality filtering

Fastp is a high-performance FASTQ preprocessor that can be used to clean up raw sequencing reads from Illumina platforms. It provides various quality control, filtering, and trimming options to remove low-quality bases, contaminants, and adapter sequences. The code provided performs a number of these steps, including trimming of poly-G and poly-X tails, which are commonly observed in Illumina reads, filtering reads based on quality and length, and removing adapter sequences. The resulting cleaned reads are then written to the specified output files. Additionally, fastp provides comprehensive quality control reports in both HTML and JSON formats, which can be used to assess the quality of the input reads and the impact of the processing steps. Overall, pre-processing raw sequencing reads with fastp is a critical step in ensuring the accuracy and reliability of downstream bioinformatics analyses.

```
fastp \
      --in1 {input.r1i} --in2 {input.r2i} \
      --out1 {output.r1o} --out2 {output.r2o} \
      --trim_poly_g \
      --trim_poly_x \
      --n_base_limit 5 \
      --qualified_quality_phred 20 \
      --length_required 60 \
      --thread {threads} \
      --html {output.fastp_html} \
      --json {output.fastp_json} \
      --adapter_sequence CTGTCTCTTATACACATCT \
      --adapter_sequence_r2 CTGTCTCTTATACACATCT
```

Ribosomal RNA removal

Ribodetector can be used for efficient removal of rRNA sequences from microbial metatranscriptomics data, improving accuracy and reducing computational time. This tool helps to better identify transcripts and understand gene expression in complex microbiomes.

```
#Code to be added here
```

Host genome indexing

In order to use STAR for host transcriptomics, it is neccesary to first generates a genome index, which can be used for multiple RNA-seq experiments.

```
#Code to be added here
```

Host genome mapping

STAR (Spliced Transcripts Alignment to a Reference) can be used for efficiently mapping host reads against a selected reference genome, and thus filter them out from subsequent metatranscriptomic analyses.

```
#Code to be added here
```

Generating and indexing the microbial genome catalogue

Explanations to be added here.

```
#Code to be added here
```

Mapping against the microbial genome catalogue

Explanations to be added here.

```
#Code to be added here
```

Calculate gene counts

Explanations to be added here.

```
#Code to be added here
```

12.2 Reference-free microbial metatranscriptomics (MT) data processing

Contents to be added here.

Chapter 13

Host proteomics (HP) data processing

Proteins are highly complex molecules that require extensive processing to derive meaningful biological insights from the large number of spectra generated by mass spectrometers, in contrast to nucleic acids. In quantitative proteomics and metabolomics, tandem mass-spectrometry (MS/MS) is the most widely used form of data collection. MS1-level quantification and MS2-level identification are used to identify and quantify features that elute from the chromatograph column at an expected retention time. This is achieved through area under the curve (AUC) or peak height calculation for each feature. The corresponding features in HP, MP, and ME are then quantified using MS1 detection, and feature identification is achieved through MS2 using search algorithms that compare the recorded MS2 spectrum to a feature spectrum from a predefined database. HP and MP databases are typically protein databases translated from genomic data, although other strategies such as spectral libraries or mRNA databases have also been successful. However, assembling the identified peptides into proteins can be challenging, especially when dealing with redundant peptides or spliced proteins. Recent advances in computational methods for predicting protein structures are expected to expand the reference databases for proteomics.

Chapter 14

Microbial metaproteomics (MP) data processing

Proteins are highly complex molecules that require extensive processing to derive meaningful biological insights from the large number of spectra generated by mass spectrometers, in contrast to nucleic acids. In quantitative proteomics and metabolomics, tandem mass-spectrometry (MS/MS) is the most widely used form of data collection. MS1-level quantification and MS2-level identification are used to identify and quantify features that elute from the chromatograph column at an expected retention time. This is achieved through area under the curve (AUC) or peak height calculation for each feature. The corresponding features in HP, MP, and ME are then quantified using MS1 detection, and feature identification is achieved through MS2 using search algorithms that compare the recorded MS2 spectrum to a feature spectrum from a predefined database. HP and MP databases are typically protein databases translated from genomic data, although other strategies such as spectral libraries or mRNA databases have also been successful. However, assembling the identified peptides into proteins can be challenging, especially when dealing with redundant peptides or spliced proteins. Recent advances in computational methods for predicting protein structures are expected to expand the reference databases for proteomics.

Part IV

STATISTICAL PROCEDURES

Chapter 15

About statistics

Statistics is probably the most challenging step of holo-omic studies, due to two main factors: the extreme complexity of the data, often containing thousands of features, and the limited sample size, often in the realm of the dozens of sampling units. This combination renders many holo-omic datasets rather statistics unfriendly.

A step-by-step approach

In this workbook we strongly encourage researchers to proceed step-by-step when dealing with holo-omics data and biological questions.

Initial quantitative exploration of omic layers

The analysis of any multi-omic data should begin with independent analysis of each omic layer to learn about its structure and variability before jumping to multi-omic data integration.

- **Data transformations:** multivariate datasets consist of different data types (e.g., presence-absence of taxa, counts of genes, community-level metabolic capacity index of a function, concentrations of metabolites across samples) that may require specific transformation before applying statistical techniques.
- **Unsupervised exploration of omic layers:** include exploratory techniques, such as cluster analysis and ordination-based visualisation methods, which reveal the structure and main patterns of the omic datasets without prior information about experimental design. These procedures might reveal that the observations are structured into meaningful groups or that variables can be reduced to fewer dimensions.
- **Supervised analysis of omic layers:** this type of analyses incorporate information of experimental design and aim at testing and estimating the

effects of the experimental factors (e.g., dietary treatment, drug administration) or variables of interest (e.g., age of the experimental subjects, geographic location of studied populations) on different omic layers.

Multi-omic data integration

When it comes to multi-omic data integration, the approaches can be broadly categorised into two types: multi-staged analysis and meta-dimensional or simultaneous analysis.

- **Multi-staged integration:** leverages the central dogma of molecular biology to assume that the variation in omic datasets is hierarchical, such that variation in DNA leads to variation in RNA and so on to determine the phenotype
- **Meta-dimensional integration:** considers the possibility that the phenotype is the product of the combination of variation across all omic layers, with the presence of complex inter-omic interactions.

All statistical analyses included in the **Holo-omics workbook** are conducted in R environment. You can find the details to set-up your R environment in the section Prepare your R environment.

15.1 Prepare your R environment

All statistical analyses included in the **Holo-omics workbook** are conducted in R environment [R Development Core Team, 2008]. R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS, and in order to use it, R or RStudio must be installed in your local computer or remote server.

Required packages

In order to reproduce the analyses shown in the workbook, a rather long list of R packages must be installed. Packages are the fundamental units of reproducible R code, which include reusable R functions, the documentation that describes how to use them, and sample data.

- ape
- DESeq2
- distillR
- ggplot2
- tidyverse
- vegan
- (…)

Package installation

Packages are installed programatically using three main ways: through CRAN, Bioconductor or Github.

Install package from CRAN

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Packages stored in CRAN can be installed using the following code:

```r
install.packages("package_name")
#e.g.
install.packages("vegan")
```

Install package from Bioconductor

Bioconductor is a free, open source and open development software project for the analysis and comprehension of genomic data generated by wet lab experiments in molecular biology. Packages included in Bioconductor can be installed using the following code:

```r
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("package_name")
#e.g.
BiocManager::install("DESeq2")
```

Install package from Github

GitHub is a code hosting platform for version control and collaboration. Packages stored in R can be installed using the following code after installing the package devtools:

```r
library(devtools)
install_github("github_repository_name_of_the_package")
#e.g.
install_github("anttonalberdi/distillR")
```

Chapter 16

Single omic analyses

Multi-omic data analysis should start by evaluating each individual layer separately to gain insight into its structure and variability, before combining all layers. Despite the varying nature of the seven omic layers discussed in this guidebook, they all possess the common attribute of being multivariate, meaning they consist of multiple features, such as genomic variants, genes, metabolic pathways, proteins or metabolites, collected from multiple observations. This section contains the following three chapters:

- **Data transformations:** multivariate datasets consist of different data types (e.g., presence-absence of taxa, counts of genes, community-level metabolic capacity index of a function, concentrations of metabolites across samples) that may require specific transformation before applying statistical techniques.
- **Unsupervised exploration of omic layers:** include exploratory techniques, such as cluster analysis and ordination-based visualisation methods, which reveal the structure and main patterns of the omic datasets without prior information about experimental design. These procedures might reveal that the observations are structured into meaningful groups or that variables can be reduced to fewer dimensions.
- **Supervised analysis of omic layers:** this type of analyses incorporate information of experimental design and aim at testing and estimating the effects of the experimental factors (e.g., dietary treatment, drug administration) or variables of interest (e.g., age of the experimental subjects, geographic location of studied populations) on different omic layers.

Chapter 17

Data transformations

Before conducting any analysis on multivariate datasets, it is important to note that the different data types present (such as the presence/absence of taxa, gene counts, metabolic capacity indices, and metabolite concentrations) may need to undergo transformation. Data transformation is the process of changing the scale or distribution of data in order to meet the assumptions of a statistical model or to improve the interpretability of the data. Transformations can be applied to individual variables or to the entire dataset, and can involve a variety of mathematical operations, such as scaling, centring, and rescaling. Data transformations can be categorised by the objective they follow:

- **Transformations to account for statistical assumptions**
- **Transformations to account for compositional data**
- **Transformations to account for scaling**

17.1 Transformations to account for statistical assumptions

Most of the statistical techniques applied to omic datasets have a set of requirements, known as assumptions, which is necessary the aalysed data meet for statistical models to make accurate and unbiased predictions based on the available data. Violating these assumptions can lead to biased and inaccurate results, and statistical tests and methods should be selected and applied carefully, taking into account the specific assumptions required for each analysis.

The specific assumptions required vary depending on the type of statistical analysis being performed, but some common examples include:

1. **Normality:** The assumption that the distribution of the data is approximately normal.

2. **Independence:** The assumption that the observations are independent of each other.
3. **Homogeneity of variance:** The assumption that the variance of the data is the same across all levels of the independent variable.
4. **Linearity:** The assumption that there is a linear relationship between the independent and dependent variables.
5. **Randomness:** The assumption that the data was obtained randomly and that there are no systematic biases in the sample selection process.
6. **Stationarity:** The assumption that the statistical properties of the data do not change over time.

Unfortunately, biological datasets rarely meet these assumptions. However, original values can be transformed so that the modified values conform better to those assumptions. Some of the most typical transformations include:

1. **Log transformation:** This transformation is used to reduce the effect of extreme values in the data and to stabilize the variance. It is often used when the data is highly skewed or when the relationship between the variables is multiplicative rather than additive.
2. **Square root transformation:** This transformation is similar to the log transformation, but is less extreme. It is often used when the data is moderately skewed and when the variance increases with the mean.
3. **Box-Cox transformation:** This is a more general transformation that allows for a range of power transformations to be applied to the data. It is used when the data is highly skewed or when the variance is not constant across the range of values.
4. **Arcsine transformation:** This transformation is used for data that is bounded between 0 and 1, such as proportions or percentages. It is used to stabilise the variance and to improve the normality of the data.
5. **Rank transformation:** This transformation involves converting the data to ranks, which can be useful for non-parametric tests. It is often used when the data is highly skewed or when there are outliers.

Transforming data to meet normality assumption

In this example, we first load the multivariate dataset and use the Shapiro-Wilk test to check for normality. We then loop through each variable in the dataset and apply the Box-Cox transformation using the boxcox function from the MASS package if the Shapiro-Wilk test indicates that the variable is not normally distributed. Finally, we use the Shapiro-Wilk test again to check for normality of the transformed data.

```r
# Load the multivariate dataset
data <- read.csv("mydata.csv")

# Check for normality using Shapiro-Wilk test
shapiro.test(data)
```

```
# Apply Box-Cox transformation to each variable
library(MASS)
data_transformed <- data
for (i in 1:ncol(data)) {
  if (shapiro.test(data[,i])$p.value < 0.05) { # if not normal
    data_transformed[,i] <- boxcox(data[,i]) # apply Box-Cox transformation
  }
}

# Check for normality again using Shapiro-Wilk test
shapiro.test(data_transformed)
```

17.1.1 Transformations to account for compositional data

When dealing with multi-omic datasets, it's important to determine if the measurements represent absolute or relative values. While HG provides qualitative information about host genomes, MG, HT, and MT provide quantitative information that's dependent on the amount of sequencing performed and thus are compositional. Consequently, raw quantitative values of genome abundance or gene expression across samples cannot be compared directly. To address this, the most common solution is to transform the raw abundance values into relative abundance data for comparison. However, this transformation reduces the independence of individual variables, which is an assumption for many statistical methods. An alternative is using ratio transformations like the centred log-ratio, which better suit compositional data analysis by removing the effect of the constant-sum constraint on the covariance and correlation matrices.

Transforming data using centred log-ratio

In this example, we first load the multivariate dataset and then apply the CLR transformation using the `clr()` function from the **`compositions`** package. The CLR transformation is a commonly used method for analyzing compositional data, where the data represents proportions or percentages that add up to a constant sum. The CLR transformation is designed to remove the constant sum constraint and make the data amenable to standard multivariate statistical methods.

The resulting `data_transformed` object will be a transformed version of the original dataset, with each variable now representing the logarithm of the ratio between that variable and the geometric mean of the other variables. Note that the CLR transformation assumes that the data is non-negative, so it may not be appropriate for all types of multivariate data.

```
# Load the multivariate dataset
data <- read.csv("mydata.csv")
```

```
# Apply CLR transformation using the compositions package
library(compositions)
data_transformed <- clr(data)

# View the transformed data
head(data_transformed)
```

17.1.2 Transformations to account for scaling

It's important to keep in mind that scaling can also impact the results when analsing multi-omic data. For instance, in ME data, certain metabolites such as ATP may have much higher concentrations than other important metabolites like signaling molecules, potentially overshadowing significant differences in the less abundant yet meaningful metabolites. In Transcriptomics, transcript length biases may also cause similar distortions. One solution is to standardise the features by using transformations like z-score normalisation, but this can amplify the influence of measurement error that is typically higher for less abundant features.

Transforming data using z-score normalisation

In this example, we first load the dataset and then use the `apply()` function to apply the z-score normalization to each variable in the dataset. The `apply()` function applies a function to either the rows or columns of a matrix, and the 2 argument specifies that we want to apply the function to the columns (i.e., variables) of the dataset. The function applied to each column calculates the z-score of each observation by subtracting the mean of the variable from each observation and dividing by the standard deviation of the variable. This centers the data around 0 and scales it to have a standard deviation of 1. The resulting `data_norm` object will be a normalized version of the original dataset, with each variable now having a mean of 0 and a standard deviation of 1.

```
# Load the dataset
data <- read.csv("mydata.csv")

# Apply z-score normalization to each variable
data_norm <- apply(data, 2, function(x) (x - mean(x)) / sd(x))

# View the normalized data
head(data_norm)
```

Chapter 18

Unsupervised exploration

Unsupervised methods in multi-omic data analysis involve techniques for exploring the structure and patterns of the data without prior knowledge of the experimental design. These include cluster analysis and visualisation methods based on ordination. These procedures can reveal meaningful groupings among observations or allow for reducing the complexity of the data by reducing the number of dimensions. Researchers can then use the results from these exploratory techniques in further multi-omic data integration. It's crucial to properly pre-process the data and choose the appropriate association coefficient when computing these methods, as this has a significant impact on the final outcome.

- Cluster analysis
- Dimension reduction and ordination

18.1 Cluster analysis

Clustering procedures group features or observations into homogeneous sets by minimising within-group and maximising among-group distances

18.1.1 Hierarchical clustering

Hierarchical clustering produces a stratified organisation of features or observations where relatively similar objects are grouped together. The clustering can be performed using different criteria to measure the distance between clusters, which will affect the final outcome of the analysis (e.g., single linkage, complete linkage, average linkage and Ward's minimum variance).

```r
# Load the dataset
data <- read.csv("mydata.csv")
```

```r
# Perform hierarchical clustering
dist_matrix <- dist(data)  # calculate distance matrix
hc <- hclust(dist_matrix)  # perform hierarchical clustering

# Plot dendrogram of clustering
plot(hc, hang=-1)
```

A useful exploratory analysis to reveal general patterns in an omic layer can be obtained by simultaneous application of hierarchical clustering to the rows and columns of the data matrix, and visualising the results in a heatmap.

```r
# Load the dataset
data <- read.csv("mydata.csv", row.names=1)

# Perform hierarchical clustering of rows and columns
row_clusters <- hclust(dist(data))
col_clusters <- hclust(dist(t(data)))

# Plot heatmap with row and column dendrograms
library(gplots)
heatmap.2(as.matrix(data),
          Rowv=row_clusters,
          Colv=col_clusters,
          scale="row",
          dendrogram="both",
          key=TRUE,
          keysize=1.5,
          col=redgreen(75))
```

18.1.2 Disjoint clustering

Disjoint clustering techniques aim at separating the objects into individual, usually mutually exclusive, and in most cases, unconnected clusters. K-means clustering is one of the most typical algorithms where objects are assigned to k clusters using an iterative procedure that minimises the within-clusters sums of squares. Other available clustering methods include twinspan, self-organising maps, dbscan and Dirichlet multinomial mixtures (DMM). DMM were specifically developed to analyse MG data but can be equally useful for other sequencing-based omic datasets.

```r
# Load the dataset
data <- read.csv("mydata.csv")

# Perform K-means clustering
k <- 3  # number of clusters
km <- kmeans(data, k)

# View the cluster assignments
head(km$cluster)
```

```
# Load the package
library(DirichletMultinomial)

# Load the dataset
data <- read.csv("mydata.csv")

# Fit Dirichlet multinomial mixture model
model <- DMM(data, K=3, alpha=1, beta=1)

# View the cluster assignments
head(model$Z)
```

18.2 Dimension reduction and ordination

Ordination is a method complementary to data clustering, which enables displaying differences among samples graphically through reducing the dimensions of the original data set, so that similar objects are near and dissimilar objects are farther from each other.

18.2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is one of the most widely applied methods for ordination. PCA generates new synthetic variables (principal components) that are linear combinations of the original variables and capture as much variance of the original data as possible. The principal components are orthogonal to each other and correspond to the successive dimensions of maximum variance of the scatter of points. The distance preserved among objects is euclidean and the relationships among variables are linear, thus PCA should generally be applied after appropriate transformations.

```
# Load the dataset
data <- read.csv("mydata.csv")

# Perform PCA
pca <- prcomp(data, scale = TRUE)

# View the results
summary(pca)

# Plot the results
plot(pca, type = "l")
```

18.2.2 Principal Coordinate Analysis (PCoA)

Principal Coordinate Analysis (PCoA) is a multivariate analysis technique used to visualise and explore the patterns of variation in multivariate data. It is similar to Principal Component Analysis (PCA) but is specifically designed for distance-based data. PCoA transforms a distance matrix into a set of coordinates that can be plotted in two or three dimensions, allowing for visualisation of the relationships between samples based on their dissimilarity.

In multi-omics research, PCoA can be used to analyse and visualise the relationships between samples based on their similarity or dissimilarity in multiple omics data types, such as gene expression, metabolomics, or proteomics. By performing PCoA on these data types separately and then comparing the results, researchers can gain insight into how different omics layers contribute to the overall variation between samples. Additionally, PCoA can be used to identify groups or clusters of samples with similar omics profiles, which can provide insight into underlying biological processes or disease states. Overall, PCoA is a powerful tool for exploring and visualising the complex relationships between multiple omics data types in multi-omics research.

```r
# Load the distance matrix
dist_mat <- read.csv("mydistances.csv", row.names = 1)

# Perform PCoA
pcoa <- cmdscale(dist_mat, k = 2, eig = TRUE, add = TRUE)

# View the results
summary(pcoa)

# Plot the results
plot(pcoa$points, type = "n", xlab = "PCo1", ylab = "PCo2")
text(pcoa$points, labels = rownames(pcoa$points))
```

18.2.3 Non-metric Multidimensional Scaling (NMDS)

Non-metric Multidimensional Scaling (NMDS) is a multivariate analysis technique used to visualize and explore the patterns of variation in multivariate data. It is similar to Principal Coordinate Analysis (PCoA) but is more flexible in that it can handle non-linear relationships between variables. NMDS transforms a distance matrix into a set of coordinates that can be plotted in two or three dimensions, allowing for visualisation of the relationships between samples based on their dissimilarity. Unlike PCoA, NMDS does not assume a linear relationship between the distance matrix and the coordinates, making it a more powerful tool for analysing complex and non-linear relationships in multivariate data.

In multi-omics research, NMDS can be used to analyse and visualise the relationships between samples based on their similarity or dissimilarity in multiple omics data types, such as gene expression, metabolomics, or proteomics. By performing NMDS on these data types separately and then comparing the results, researchers can gain insight into how different omics layers contribute to the overall variation between samples. Additionally, NMDS can be used to identify groups or clusters of samples with similar omics profiles, which can provide insight into underlying biological processes or disease states. Overall, NMDS is a powerful tool for exploring and visualising the complex relationships between multiple omics data types in multi-omics research, particularly when the relationships between variables are non-linear.

```
# Load the dataset
data <- read.csv("mydata.csv", row.names = 1)

# Perform NMDS
library(vegan)
nmds <- metaMDS(data, distance = "bray")

# View the results
summary(nmds)

# Plot the results
plot(nmds$points, type = "n", xlab = "NMDS1", ylab = "NMDS2")
text(nmds$points, labels = rownames(nmds$points))
```

18.2.4 t-Distributed Stochastic Neighbour Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction technique used to visualise high-dimensional data in a low-dimensional space. t-SNE is particularly useful when exploring complex and nonlinear relationships between variables, and can be applied to various types of data including gene expression, proteomics, and metabolomics data. t-SNE works by first constructing a probability distribution over pairs of high-dimensional objects, such as genes or proteins, and then constructing a similar probability distribution over pairs of low-dimensional points. The technique then optimizes these probability distributions to minimise the divergence between them, resulting in a low-dimensional representation of the high-dimensional data.

In multi-omics research, t-SNE can be used to analyse and visualise the relationships between samples based on their omics profiles. By performing t-SNE on multiple omics data types separately and then comparing the results, researchers can gain insight into how different omics layers contribute to the overall variation between samples. Additionally, t-SNE can be used to identify clusters or groups of samples with similar omics profiles, which can provide insight into underlying biological processes or disease states. Overall, t-SNE is a powerful tool for visualising high-dimensional data in a low-dimensional space, allowing researchers to explore and analyse complex relationships in multi-omics research.

```
# Load the dataset
data <- read.csv("mydata.csv", row.names = 1)

# Perform t-SNE
```

```
library(Rtsne)
tsne <- Rtsne(data, dims = 2, perplexity = 30, verbose = TRUE)

# View the results
summary(tsne)

# Plot the results
plot(tsne$Y, col = "blue", pch = 19, xlab = "t-SNE1", ylab = "t-SNE2")
```

18.2.5 Uniform manifold approximation and projection (UMAP)

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimension reduction technique used to visualise high-dimensional data in a low-dimensional space. It is similar to t-Distributed Stochastic Neighbour Embedding (t-SNE) but is faster and more scalable, making it useful for larger datasets. UMAP works by constructing a fuzzy topological representation of the high-dimensional data and then optimising a low-dimensional representation that preserves the structure of this topological representation. This results in a low-dimensional representation of the high-dimensional data that preserves complex relationships between variables.

In multi-omics research, UMAP can be used to analyse and visualise the relationships between samples based on their omics profiles. By performing UMAP on multiple omics data types separately and then comparing the results, researchers can gain insight into how different omics layers contribute to the overall variation between samples. Additionally, UMAP can be used to identify clusters or groups of samples with similar omics profiles, which can provide insight into underlying biological processes or disease states. Overall, UMAP is a powerful tool for visualising high-dimensional data, particularly for large and complex datasets.

```
# Load the dataset
data <- read.csv("mydata.csv", row.names = 1)

# Perform UMAP
library(umap)
umap_result <- umap(data, n_components = 2, n_neighbors = 30)

# View the results
summary(umap_result)

# Plot the results
plot(umap_result$layout, col = "blue", pch = 19, xlab = "UMAP1", ylab = "UMAP2")
```

Chapter 19

Supervised analysis

The difference between supervised and unsupervised analyses in omic studies lies in the incorporation of experimental design information. Unlike unsupervised methods, supervised analyses incorporate prior information about the experimental design, making them useful for testing the effects of experimental factors and associating omic data with phenotypic features. Supervised analyses can be divided into two types: regression and classification.

Regression problems involve predicting a numeric variable or matrix based on the omic data and experimental factors, such as treatment or subject characteristics.

Classification problems involve classifying observations into groups based on their features across different omic layers.

19.1 Regression methods

Regression methods aim to model the relationship between the quantitative metrics of the omic features and the response variable. Regression methods are commonly used in supervised analysis of omic data to identify associations between the expression levels of different genes, metabolites, or other features and a particular outcome or response variable, such as a disease state or an experimental treatment.

19.1.1 PERMANOVA

PERMANOVA is a statistical method that tests for significant differences between groups in multivariate data by comparing the distribution of distance-based similarity matrices. This method can be used to identify biomarkers that differ significantly between groups, such as disease states or treatment groups, and determine whether there are significant differences in the overall pattern of gene expression, methylation, or other omic features between the groups.

```r
library(vegan)

# simulate gene expression data with 100 samples and 1000 genes
set.seed(123)
exprs <- matrix(rnorm(100*1000), nrow = 100, ncol = 1000)

# create a grouping variable with 2 groups
group <- rep(c("control", "treatment"), each = 50)

# calculate Euclidean distance matrix from gene expression data
dist_matrix <- vegdist(t(exprs), method = "euclidean")

# perform PERMANOVA analysis
permanova_results <- adonis(dist_matrix ~ group)

# view PERMANOVA results
permanova_results
```

19.1.2 ANOSIM

ANOSIM is a nonparametric statistical method that tests for significant differences in similarity between two or more groups of samples based on dissimilarity matrices. It is used to compare the dissimilarity between groups of samples based on their gene expression, metabolomics, or other omic data. The aim of ANOSIM is to determine if the differences in omic profiles between groups are significant and can be used to distinguish between groups.

```r
library(vegan)

# Load gene expression data
data <- read.csv("gene_expression_data.csv", row.names = 1)

# Define grouping variable
group <- c(rep("Group 1", 5), rep("Group 2", 5))

# Calculate dissimilarity matrix
dissimilarity <- vegdist(data)

# Perform ANOSIM
result <- anosim(dissimilarity, group)

# View ANOSIM results
result
```

19.1.3 Redundancy analysis (RDA)

Redundancy Analysis (RDA) is a multivariate statistical technique that identifies the linear relationships between a response variable and a set of explanatory variables. It is an extension of Principal Component Analysis (PCA) that can handle both continuous and categorical variables. RDA is used to identify the genes, metabolites, or other omic variables that are most strongly associated with a specific outcome or response variable, such as a disease state or drug response. It can also be used to visualise the relationship between the explanatory and response variables.

```r
library(vegan)

# Load gene expression data
data <- read.csv("gene_expression_data.csv", row.names = 1)

# Load disease state data
disease <- read.csv("disease_state.csv", row.names = 1)

# Perform RDA
result <- rda(data, disease)

# View RDA results
result

# Plot RDA biplot
plot(result, display = "biplot")
```

19.1.4 Canonical Correspondence Analysis (CCA)

Canonical Correspondence Analysis (CCA) is a multivariate statistical technique that explores the relationship between a set of explanatory variables and a set of response variables. It is an extension of Correspondence Analysis (CA) that can handle both continuous and categorical variables. CCA is used to identify the genes, metabolites, or other omic variables that are most strongly associated with a specific outcome or response variable, such as a disease state or drug response. It can also be used to visualise the relationship between the explanatory and response variables.

```r
library(vegan)

# Load gene expression data
data <- read.csv("gene_expression_data.csv", row.names = 1)

# Load disease state data
disease <- read.csv("disease_state.csv", row.names = 1)

# Perform CCA
result <- cca(data, disease)
```

```r
# View CCA results
result

# Plot CCA biplot
plot(result, display = "biplot")
```

19.1.5 Generalised linear modelling (GLM)

Generalised linear modelling (GLM) is a statistical framework that allows for the analysis of a wide range of response variables, including binary, count, and continuous data. In the context of supervised analysis of single omic layers, GLM can be used to identify associations between a specific response variable, such as disease status, and the omic data, such as gene expression or metabolite levels.

The first step in using GLM for supervised analysis of omic data is to select the appropriate distribution for the response variable. For example, if the response variable is binary (e.g., healthy vs. diseased), then a binomial distribution can be used. If the response variable is a count (e.g., the number of mutations), then a Poisson or negative binomial distribution can be used. If the response variable is continuous (e.g., expression levels), then a Gaussian distribution can be used. Once the appropriate distribution is selected, a GLM can be constructed by specifying a linear relationship between the response variable and the omic data, using one or more explanatory variables. The explanatory variables can be selected based on prior knowledge or through a variable selection process, such as forward or backward selection. The GLM model can then be fit to the data using maximum likelihood estimation or other methods. After fitting the GLM model, the significance of each explanatory variable can be assessed using hypothesis testing, such as Wald tests or likelihood ratio tests. The explanatory variables that are found to be significant can then be interpreted as predictors of the response variable.

Overall, GLM can be a useful tool for supervised analysis of single omic layers because it allows for the identification of specific predictors of a response variable, such as disease status, and can handle a wide range of response variable distributions. However, it is important to carefully select the appropriate distribution and explanatory variables to ensure the validity and accuracy of the model.

```r
# Load required packages
library(edgeR)   # for differential expression analysis
library(glmnet)  # for regularization and variable selection

# Load example data
data <- read.table("example_omic_data.txt", header=TRUE, sep="\t")

# Define the response variable
```

```r
response <- factor(data$disease_status)

# Define the explanatory variables
explanatory <- as.matrix(data[,2:ncol(data)])  # omic data

# Perform differential expression analysis to identify significant variables
dge <- DGEList(counts=explanatory, group=response)
dge <- calcNormFactors(dge)
design <- model.matrix(~response)
fit <- glmQLFit(dge, design)
qlf <- glmQLFTest(fit, coef=2)
significant_vars <- topTags(qlf, n=100)$table$ID  # select top 100 significant variables

# Fit a GLM model with regularization and variable selection
fit.glm <- cv.glmnet(x=explanatory[,significant_vars], y=response, family="binomial")
plot(fit.glm)

# Identify the most important variables
coef(fit.glm, s=fit.glm$lambda.min)
```

19.1.6 Generalised linear mixed modelling (GLMM)

Generalised linear mixed models (GLMMs) are an extension of GLMs that allow for the analysis of data with non-independent errors, such as clustered or longitudinal data. GLMMs incorporate random effects, which account for the correlation between observations within groups or over time, and fixed effects, which represent the relationships between the response and explanatory variables. GLMMs are widely used in omics data analysis, particularly for the analysis of longitudinal or repeated measures data, and for the integration of multiple omics data layers.

In GLMMs, the response variable y is related to the explanatory variables x through a link function g and a linear predictor:

```
g(E[y | x]) = x * beta + Z * b
```

where beta are the fixed effects coefficients, b are the random effects coefficients, Z is the design matrix for the random effects, and E[y | x] is the expected value of the response variable given the explanatory variables x. The random effects account for the correlation between observations within groups or over time, and are assumed to be normally distributed with mean 0 and covariance matrix D. The link function g specifies the relationship between the expected value of the response variable and the linear predictor. The choice of link function depends on the nature of the response variable and can be any member of the exponential family of distributions.

GLMMs are fitted using maximum likelihood estimation, which involves optimising the likelihood function with respect to the fixed and random effects coefficients, as well as the covariance matrix of the random effects. The likelihood function takes into account the correlation structure of the data and is typically evaluated using numerical methods such as the Laplace approximation or Monte Carlo Markov Chain (MCMC) methods.

GLMMs can be used for a wide range of applications in omics data analysis, including the analysis of longitudinal or repeated measures data, the integration of multiple omics data layers, the analysis of data with non-independent errors, and the modelling of gene-environment interactions. However, GLMMs can be computationally intensive and require careful consideration of the correlation structure of the data and the appropriate choice of link function and distribution.

```r
# Load the lme4 package
library(lme4)

# Load example data
data <- read.table("example_omic_data.txt", header=TRUE, sep="\t")

# Convert the Treatment treatment to a factor
data$Treatment <- as.factor(data$Treatment)

# Split the data into training and testing sets
set.seed(123)
train_indices <- sample(nrow(data), 0.7*nrow(data))
train_data <- data[train_indices,]
test_data <- data[-train_indices,]

# Fit a GLMM with random intercept and slope
model <- glmer(Treatment ~ . + (1 | Sepal.Length), data = train_data, family = binomial)

# Predict on the test data
test_data$predicted <- predict(model, newdata = test_data, type = "response")

# Calculate the accuracy of the predictions
accuracy <- sum(test_data$predicted > 0.5 & test_data$Treatment == "versicolor" |
                  test_data$predicted <= 0.5 & test_data$Treatment == "setosa" |
                  test_data$Treatment == "virginica") / nrow(test_data)

# Print the accuracy
cat(sprintf("Accuracy: %.2f%%\n", accuracy*100))
```

19.2 Classification methods

Classification methods aim to learn a model that can predict the class labels of new samples based on their omic profiles. Classification methods are commonly used in supervised analysis of omic data to classify samples into different categories based on their expression levels of genes, metabolites, or other features.

Overall, classification methods aim to learn a model that can accurately predict the class labels of new samples based on their omic profiles, and can be useful for identifying biomarkers that are predictive of disease or treatment response.

19.2.1 Random Forests (RF)

Random forests are an ensemble of decision trees that are trained on bootstrapped samples of the data and a random subset of the features, to reduce overfitting and improve accuracy.

```
library(randomForest)

# Load the dataset
data <- read.csv("mydata.csv")

# Split the data into training and test sets
trainIndex <- sample(1:nrow(data), 0.7*nrow(data))
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]

# Train the random forest classifier
rf <- randomForest(class ~ ., data=trainData, ntree=500, importance=TRUE)

# Make predictions on the test data
predictions <- predict(rf, testData)

# Evaluate the performance of the classifier
confusionMatrix(predictions, testData$class)
```

19.2.2 Support Vector Machines (SVM)

Support vector machines (SVMs) aim to find a hyperplane that maximally separates the samples belonging to different classes in the feature space, and can handle both linear and nonlinear relationships between the features and the response variable.

```
library(e1071)

# Load the dataset
data <- read.csv("mydata.csv")

# Split the data into training and test sets
trainIndex <- sample(1:nrow(data), 0.7*nrow(data))
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]

# Train the SVM classifier
svm <- svm(class ~ ., data=trainData, kernel="radial", cost=1)
```

```r
# Make predictions on the test data
predictions <- predict(svm, testData)

# Evaluate the performance of the classifier
confusionMatrix(predictions, testData$class)
```

Chapter 20

Multi-omic integration

Multi-omic data integration can be broadly classified into two categories: multi-staged analysis and meta-dimensional analysis. In the multi-staged approach, the data analysis is divided into multiple steps, where two omic layers are linked at each step, and the final step relates the relevant omic layers with the outcome of interest. This method takes advantage of the hierarchical nature of molecular biology's central dogma, assuming that variations in omic datasets occur in a hierarchical manner, with changes in DNA leading to changes in RNA, and so on. On the other hand, meta-dimensional analysis involves analyzing all omic datasets in a single analysis, which encompasses the entire range of features simultaneously, and enables the assessment of inter-omic interactions.

- **Multi-staged data integration**:
- **Meta-dimensional data integration**

Chapter 21

Multi-staged omics integration

Multi-stage omics integrations leverages the structure of biological organisation to analyse the data in multiple steps, relating two omic layers at a time, with the final step linking the relevant omic layers with the outcome of interest. In the past, the predominant method for integrated analysis of biological data was the multi-staged approach. This approach relied heavily on traditional statistical tools and hypothesis testing approximations. The multi-staged approach is advantageous in that it enables the systematic linking of multi-omic datasets in a stepwise manner, allowing for the development of knowledge that can be later used to test causally-oriented hypotheses. Furthermore, this approach is better suited to account for the biological asymmetries between different omic datasets.

One popular example of multi-staged integration is the three-stage or triangle method. In the first stage, SNPs are associated with the outcome of interest and filtered based on a genome-wide significance threshold. Then, SNPs significantly associated with the outcome in the first stage are tested for association with other omic layers: the SNPs associated with gene expression levels are called expression quantitative trait loci (eQTL); metabolite QTLs (mQTL) and protein QTLs (pQTL) can be similarly defined. Lastly, omic data retained in the second stage are used for association with the outcome in the third stage. Similar approaches could potentially be used to associate microbial MG data with MT, MP, ME data and outcomes of interest. Variations of this method where associations of other omic-layers are tested in stage one and the genomic associations are tested in later stages have also been proposed.

Contents of this section were created by Iñaki Odriozola and Antton Alberdi.

Chapter 22

Meta-dimensional omics integration

In meta-dimensional analysis all omic datasets are analysed in a single, simultaneous analysis. This kind of approach typically avoids using domain knowledge-based procedures to independently reduce features in single omic datasets, and aims at integrating multi-omic datasets in their whole complexity. Meta-dimensional integration methods can be grouped following several criteria but here we briefly summarise the classification first coined by Ritchie et al. (2015) [Ritchie et al., 2015] and recently reviewed by Reel et al. (2021) [Reel et al., 2021] (we refer interested readers to those publications for a more in depth treatment of the topic), which classifies the methods into concatenation-based, model-based and transformation-based integration methods. The three kinds of integration methods can be used for unsupervised and supervised analysis of multi-omic data, including classification and regression tasks.

- **Concatenation-based integration**
- **Transformation-based integration**
- **Model-based integration**

Contents of this section were created by Iñaki Odriozola and Antton Alberdi.

22.1 Concatenation-based integration

Concatenation-based integration combines multiple omic datasets, raw or preprocessed, into a single large matrix. One of the advantages of these approaches is their simplicity, since once the concatenation of multi-omic datasets is achieved, unsupervised and supervised analysis methods can be applied to the joint matrix, as in the case of the independent analysis of omic layers. Concatenation-based techniques offer a straightforward approach to utilising machine learning for the examination of both continuous and categorical data. Once the individual omics are concatenated, these methods can analyse all the combined features in an even-handed manner and pinpoint the most distinguishing features associated with a given phenotype. One of the main challenges of concatenation-based approaches is to ensure that the features of the different omic layers are comparable.

Several examples of unsupervised concatenation-based methods for multi-omic integration have been developed in recent years, most of them based on matrix-factorisation [Reel et al., 2021]. Joint non-negative matrix factorisation (Joint NMF) allowed integrating non-negative multi-omic data by decomposing the joint matrix into factors and loadings [Zhang et al., 2021]. Joint and Individual Variation Explained (JIVE) is an adaptation of NMF framework [Lock et al., 2013] which was later improved by Joint Bayes Factor (JBF) to handle the problems derived from the high sparsity of multi-omic datasets [Ray et al., 2014]. iCluster framework is based in similar principles to NMF but allows integration of datasets having negative values [Shen et al., 2009]. MoCluster [Meng et al., 2016], RLAcluster [Wu et al., 2015] and iClusterBayes [Mo et al., 2018] have further developed the framework and improved it in terms of diversity of handled data types, computation speed and clustering accuracy. Multi-Omics Factor Analysis (MOFA) is another recent development that allows discovering the principal sources of variability across different omic datasets [Argelaguet et al., 2018]. Regarding supervised analyses, any of the algorithms for supervised analysis of single omic layers can be used to analyse concatenated multi omic data. RF [Acharjee et al., 2016], SVM [Li et al., 2017], LASSO regression [Lee et al., 2017] or DL [Zhang et al., 2018] algorithms have been used, among others, for concatenation-based supervised analysis in multi-omic literature.

Contents of this section were created by Iñaki Odriozola and Antton Alberdi.

22.2 Transformation-based integration

In transformation-based integration, omic datasets are first transformed into an intermediate representation, typically a graph or a kernel matrix, and they are then merged before building the final model. This approach preserves the specific properties of each omic layer if they are transformed into appropriate intermediate representations, and a wide range of omic data can be combined as long as they share a unique identifier (i.e. a sample ID). Graph-based analyses have the advantage of easier interpretability and lower computational requirements whereas, overall, kernel-based methods provide higher predictive performance [Yan et al., 2017].

There are several methods available for transformation-based unsupervised analysis. Regularised Multiple Kernel Learning for Locality Preserving Projections (rMKL-LPP) [Speicher and Pfeifer, 2015] and PAMOGK [Tepeli et al., 2021] are examples of kernel- and graph-based methods that can be used for clustering. Meta-analytic SVM (Meta-SVM) [Kim et al., 2017] and NEighborhood based Multi-Omics clustering (NEMO) [Rappoport and Shamir, 2019] are other methods available for transformation-based unsupervised analysis. Most of the methods for transformation-based supervised analysis are kernel- or graph-based algorithms [Reel et al., 2021, Yan et al. [2017]]. The kernel-based integration approaches include Semi-Definite Programming SVM (SDP-SVM) [Lanckriet et al., 2004], Multiple Kernel Learning with Feature Selection (FSMKL) [Seoane et al., 2014], Relevance Vector Machine (RVM) [Tipping, 2001] and Ada-boost RVM [Wu et al., 2010]. The graph-based integration approaches include graph-based semi-supervised learning (included in supervised analyses following Reel et al. 2021 [Reel et al., 2021]) [Kim et al., 2015], graph sharpening [Shin et al., 2010] and composite network [Mostafavi and Morris, 2010]. Graph-based analyses have the advantage of easier interpretability and lower computational requirements whereas, overall, kernel-based methods provide higher predictive performance [Yan et al., 2017]. However, see Multi-Omics Graph Convolutional Networks (MOGONET) [Wang et al., 2021] for a high performing graph-based classification method.

Contents of this section were created by Iñaki Odriozola and Antton Alberdi.

22.3 Model-based integration

Model-based integration builds intermediate models from each omic layer and then builds a final model combining all intermediate models. An advantage of this approach is that it allows merging multiple omic types that have been collected in different sets of sampling units, if the outcome of interest is the same across datasets (e.g. specific disease). On the other hand, since the models are first built independently for different omic layers, these methods may fail to capture interactions between features belonging to different omic datasets, i.e. if there are two features belonging to different omic layers that affect the outcome, but only through their interaction and not when evaluated independently. Therefore, the model-based integration is particularly suitable when the different omic datasets are extremely heterogeneous (even collected from different samples), and concatenating or transforming them to a common intermediate form is not possible.

Model-based unsupervised integration methods include Format Concept Analysis (FCA) consensus clustering [Hristoskova et al., 2014], Bayesian consensus clustering (BCC) [Lock and Dunson, 2013] or Perturbation Clustering for Data Integration and Disease Subtyping (PINS+) [Nguyen et al., 2019]. Network-based methods such as Lemon Tree [Bonnet et al., 2015] or Similarity Network Fusion (SNF) [Wang et al., 2014] are also available for association analysis. Model-based supervised integration can use a variety of frameworks for model development, including majority-based voting [Drăghici and Potter, 2003], hierarchical classifiers [Bavafaye Haghighi et al., 2019], ensemble-based approaches such as XGBoost [Ma et al., 2020] or DL methods [Poirion et al., 2020]. Multi-omic data integration efforts such as ATHENA (Analysis Tool for Heritable and Environmental Network Associations) [Holzinger et al., 2014] or MOSAE (Multi-omics Supervised Autoencoder) [Tan et al., 2020] use model-based integration for disease prediction by combining a variety of modelling frameworks and algorithms.

Contents of this section were created by Iñaki Odriozola and Antton Alberdi.

Part V

RESOURCES

Chapter 23

Useful links

Data access

Host reference genomes

- **NCBI Genome (website):**
- **Ensembl (website):**
- **Vertebrates Genome Project (website):**

Metagenomic data

- **HoloFood Data Portal (website):**
- **MGnify (website):**
- **Earth Hologenome Initiative (website):**

Documentation

Genomics

- **Data Wrangling and Processing for Genomics (website):**
- **Vertebrate Genomes Project assembly pipeline tutorial (website):**

Shell command line usage

- **Introduction to the Command Line for Genomics (website):** general overview of basic command line usage.

R usage (General usage and programming)

- **Intro to R and RStudio for Genomics (website):**
- **Efficient R programming (website):** best practices for programming in R.

R usage (Graphics and visualisation)

- **Fundamentals of Data Visualization (website):** guide to making visualisations that accurately reflect the data, tell a story, and look professional.
- **R Graphics Cookbook (website):** a practical guide that provides more than 150 recipes to generate high-quality graphs using ggplot2.

Statistics

- **An Introduction to Statistical Learning (book):** freely available book about general statistical learning covering regression and classification problems through linear modelling and machine learning.
- **High dimensional statistics with R (website):** virtual lesson specialised in dealing with high dimensional data.

Chapter 24

References

Bibliography

Animesh Acharjee, Bjorn Kloosterman, Richard G F Visser, and Chris Maliepaard. Integration of multi-omics data for prediction of phenotypic traits using random forest. *BMC Bioinformatics*, 17 Suppl 5(Suppl 5):180, June 2016.

Antton Alberdi, Sandra B Andersen, Morten T Limborg, Robert R Dunn, and M Thomas P Gilbert. Disentangling host-microbiota complexity through hologenomics. *Nat. Rev. Genet.*, 23(5):281–297, May 2022.

Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-Omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Mol. Syst. Biol.*, 14(6):e8124, 2018.

Elham Bavafaye Haghighi, Michael Knudsen, Britt Elmedal Laursen, and Søren Besenbacher. Hierarchical classification of cancers of unknown primary using Multi-Omics data. *Cancer Inform.*, 18:1176935119872163, August 2019.

Eric Bonnet, Laurence Calzone, and Tom Michoel. Integrative multi-omics module network inference with Lemon-Tree. *PLoS Comput. Biol.*, 11(2):e1003983, February 2015.

Robert M Bowers, Nikos C Kyrpides, Ramunas Stepanauskas, Miranda Harmon-Smith, Devin Doud, T B K Reddy, Frederik Schulz, Jessica Jarett, Adam R Rivers, Emiley A Eloe-Fadrosh, Susannah G Tringe, Natalia N Ivanova, Alex Copeland, Alicia Clum, Eric D Becraft, Rex R Malmstrom, Bruce Birren, Mircea Podar, Peer Bork, George M Weinstock, George M Garrity, Jeremy A Dodsworth, Shibu Yooseph, Granger Sutton, Frank O Glöckner, Jack A Gilbert, William C Nelson, Steven J Hallam, Sean P Jungbluth, Thijs J G Ettema, Scott Tighe, Konstantinos T Konstantinidis, Wen-Tso Liu, Brett J Baker, Thomas Rattei, Jonathan A Eisen, Brian Hedlund, Katherine D McMahon, Noah Fierer, Rob Knight, Rob Finn, Guy Cochrane, Ilene Karsch-Mizrachi, Gene W Tyson, Christian Rinke, Genome Standards Consortium, Alla Lapidus, Folker Meyer, Pelin Yilmaz, Donovan H Parks, A M Eren, Lynn Schriml, Jillian F Banfield, Philip Hugenholtz, and Tanja Woyke. Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nat. Biotechnol.*, 35 (8):725–731, August 2017.

Pierre-Alain Chaumeil, Aaron J Mussig, Philip Hugenholtz, and Donovan H Parks. GTDB-Tk v2: memory friendly classification with the genome taxonomy database. *Bioinformatics*, 38(23):5315–5316, November 2022.

Haoyu Cheng, Gregory T Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat. Methods*, 18(2):170–175, February 2021.

Sorin Drăghici and R Brian Potter. Predicting HIV drug resistance with neural networks. *Bioinformatics*, 19(1):98–107, January 2003.

A Murat Eren, Özcan C Esen, Christopher Quince, Joseph H Vineis, Hilary G Morrison, Mitchell L Sogin, and Tom O Delmont. Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ*, 3:e1319, October 2015.

Jacob T Evans and Vincent J Denef. To dereplicate or not to dereplicate? *mSphere*, 5(3), May 2020.

Kristýna Fiedorová, Matěj Radvanský, Eva Němcová, Hana Grombiříková, Juraj Bosák, Michaela Černochová, Matej Lexa, David Šmajs, and Tomáš Freiberger. The impact of DNA extraction methods on stool bacterial and fungal microbiota community recovery. *Front. Microbiol.*, 10:821, April 2019.

Caleb N Fischer, Eric P Trautman, Jason M Crawford, Eric V Stabb, Jo Handelsman, and Nichole A Broderick. Metabolite exchange between microbiome members produces compounds that influence drosophila behavior. *Elife*, 6, January 2017.

W Gu, E D Crawford, B D O'Donovan, M R Wilson, E D Chow, H Retallack, and J L DeRisi. Depletion of abundant sequences by hybridization (DASH): using cas9 to remove unwanted high-abundance species in sequencing libraries and molecular counting applications. *Genome Biol.*, 17:41, March 2016.

Emily R Holzinger, Scott M Dudek, Alex T Frase, Sarah A Pendergrass, and Marylyn D Ritchie. ATHENA: the analysis tool for heritable and environmental network associations. *Bioinformatics*, 30(5):698–705, March 2014.

Anna Hristoskova, Veselka Boeva, and Elena Tsiporkova. A formal concept analysis approach to consensus clustering of multi-experiment expression data. *BMC Bioinformatics*, 15:151, May 2014.

Yiming Huang, Ravi U Sheth, Andrew Kaufman, and Harris H Wang. Scalable and cost-effective ribonuclease-based rRNA depletion for transcriptomics. *Nucleic Acids Res.*, 48(4):e20, February 2020.

Marcus B Jones, Sarah K Highlander, Ericka L Anderson, Weizhong Li, Mark Dayrit, Niels Klitgord, Martin M Fabani, Victor Seguritan, Jessica Green, David T Pride, Shibu Yooseph, William Biggs, Karen E Nelson, and J Craig

Venter. Library preparation methodology can influence genomic and functional predictions in human microbiome research. *Proc. Natl. Acad. Sci. U. S. A.*, 112(45):14024–14029, November 2015.

Dokyoon Kim, Je-Gun Joung, Kyung-Ah Sohn, Hyunjung Shin, Yu Rang Park, Marylyn D Ritchie, and Ju Han Kim. Knowledge boosting: a graph-based integration approach with multi-omics data and genomic knowledge for cancer clinical outcome prediction. *J. Am. Med. Inform. Assoc.*, 22(1):109–120, January 2015.

Sunghwan Kim, Jae-Hwan Jhong, Jungjun Lee, and Ja-Yong Koo. Erratum to: Meta-analytic support vector machine for integrating multiple omics data. *BioData Min.*, 10:8, February 2017.

Martin Kircher, Susanna Sawyer, and Matthias Meyer. Double indexing overcomes inaccuracies in multiplex sequencing on the illumina platform. *Nucleic Acids Res.*, 40(1):e3, January 2012.

Teemu Kivioja, Anna Vähärautio, Kasper Karlsson, Martin Bonke, Martin Enge, Sten Linnarsson, and Jussi Taipale. Counting absolute numbers of molecules using unique molecular identifiers. *Nat. Methods*, 9(1):72–74, November 2011.

Amelie J Kraus, Benedikt G Brink, and T Nicolai Siegel. Efficient and specific oligo-based depletion of rRNA. *Sci. Rep.*, 9(1):12281, August 2019.

Gert R G Lanckriet, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, November 2004.

Garam Lee, Lisa Bang, So Yeon Kim, Dokyoon Kim, and Kyung-Ah Sohn. Identifying subtype-specific associations between gene expression and DNA methylation profiles in breast cancer. *BMC Med. Genomics*, 10(Suppl 1):28, May 2017.

Sung-Gwon Lee, Dokyun Na, and Chungoo Park. Comparability of reference-based and reference-free transcriptome analysis approaches at the gene expression level. *BMC Bioinformatics*, 22(Suppl 11):310, October 2021.

Simin Li, Xiujie Chen, Xiangqiong Liu, Yang Yu, Hongying Pan, Rainer Haak, Jana Schmidt, Dirk Ziebolz, and Gerhard Schmalz. Complex integrated analysis of lncRNAs-miRNAs-mRNAs in oral squamous cell carcinoma. *Oral Oncol.*, 73:1–9, October 2017.

Morten T Limborg, Antton Alberdi, Miyako Kodama, Michael Roggenbuck, Karsten Kristiansen, and M Thomas P Gilbert. Applied hologenomics: Feasibility and potential in aquaculture. *Trends Biotechnol.*, 36(3):252–264, March 2018.

Eric F Lock and David B Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, October 2013.

Eric F Lock, Katherine A Hoadley, J S Marron, and Andrew B Nobel. JOINT AND INDIVIDUAL VARIATION EXPLAINED (JIVE) FOR INTEGRATED ANALYSIS OF MULTIPLE DATA TYPES. *Ann. Appl. Stat.*, 7 (1):523–542, March 2013.

Anjun Ma, Adam McDermaid, Jennifer Xu, Yuzhou Chang, and Qin Ma. Integrative methods and practical challenges for Single-Cell multi-omics. *Trends Biotechnol.*, 38(9):1007–1022, September 2020.

Chen Meng, Dominic Helm, Martin Frejno, and Bernhard Kuster. mocluster: Identifying joint patterns across multiple omics data sets. *J. Proteome Res.*, 15(3):755–765, March 2016.

Qianxing Mo, Ronglai Shen, Cui Guo, Marina Vannucci, Keith S Chan, and Susan G Hilsenbeck. A fully bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics*, 19(1):71–86, January 2018.

Sara Mostafavi and Quaid Morris. Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, 26(14):1759–1765, July 2010.

Alexandra A Mushegian, Roberto Arbore, Jean-Claude Walser, and Dieter Ebert. Environmental sources of bacteria and genetic variation in behavior influence Host-Associated microbiota. *Appl. Environ. Microbiol.*, 85(8), April 2019.

Hung Nguyen, Sangam Shrestha, Sorin Draghici, and Tin Nguyen. PINSPlus: a tool for tumor subtype discovery in integrated genomic data. *Bioinformatics*, 35(16):2843–2846, August 2019.

Lasse Nyholm, Adam Koziol, Sofia Marcos, Amanda Bolt Botnen, Ostaizka Aizpurua, Shyam Gopalakrishnan, Morten T Limborg, M Thomas P Gilbert, and Antton Alberdi. Holo-Omics: Integrated Host-Microbiota multi-omics for basic and applied biological research. *iScience*, 23(8):101414, August 2020.

Matthew R Olm, Christopher T Brown, Brandon Brooks, and Jillian F Banfield. drep: a tool for fast and accurate genomic comparisons that enables improved genome recovery from metagenomes through de-replication. *ISME J.*, 11(12): 2864–2868, December 2017.

Askarbek Orakov, Anthony Fullam, Luis Pedro Coelho, Supriya Khedkar, Damian Szklarczyk, Daniel R Mende, Thomas S B Schmidt, and Peer Bork. GUNC: detection of chimerism and contamination in prokaryotic genomes. *Genome Biol.*, 22(1):178, June 2021.

Donovan H Parks, Maria Chuvochina, Christian Rinke, Aaron J Mussig, Pierre-Alain Chaumeil, and Philip Hugenholtz. GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Res.*, 50(D1): D785–D794, January 2022.

O B Poirion, K Chaudhary, S Huang, and L X Garmire. Multi-omics-based pan-cancer prognosis prediction using an ensemble of deep-learning and machine-learning models. *medRxiv*, 2020.

Gianluca Prezza, Tobias Heckel, Sascha Dietrich, Christina Homberger, Alexander J Westermann, and Jörg Vogel. Improved bacterial RNA-seq by cas9-based depletion of ribosomal RNA reads. *RNA*, 26(8):1069–1078, August 2020.

R Development Core Team. *R: a language and environment for statistical computing*. Vienna, Austria, 2008.

T Rhyker Ranallo-Benavidez, Kamil S Jaron, and Michael C Schatz. GenomeScope 2.0 and smudgeplot for reference-free profiling of polyploid genomes. *Nat. Commun.*, 11(1):1432, March 2020.

Nimrod Rappoport and Ron Shamir. NEMO: cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, 35(18):3348–3356, September 2019.

Priyadip Ray, Lingling Zheng, Joseph Lucas, and Lawrence Carin. Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics*, 30(10):1370–1376, May 2014.

Parminder S Reel, Smarti Reel, Ewan Pearson, Emanuele Trucco, and Emily Jefferson. Using machine learning approaches for multi-omics data analysis: A review. *Biotechnol. Adv.*, 49:107739, July 2021.

Arang Rhie, Brian P Walenz, Sergey Koren, and Adam M Phillippy. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biol.*, 21(1):245, September 2020.

Arang Rhie, Shane A McCarthy, Olivier Fedrigo, Joana Damas, Giulio Formenti, Sergey Koren, Marcela Uliano-Silva, William Chow, Arkarachai Fungtammasan, Juwan Kim, Chul Lee, Byung June Ko, Mark Chaisson, Gregory L Gedman, Lindsey J Cantin, Francoise Thibaud-Nissen, Leanne Haggerty, Iliana Bista, Michelle Smith, Bettina Haase, Jacquelyn Mountcastle, Sylke Winkler, Sadye Paez, Jason Howard, Sonja C Vernes, Tanya M Lama, Frank Grutzner, Wesley C Warren, Christopher N Balakrishnan, Dave Burt, Julia M George, Matthew T Biegler, David Iorns, Andrew Digby, Daryl Eason, Bruce Robertson, Taylor Edwards, Mark Wilkinson, George Turner, Axel Meyer, Andreas F Kautt, Paolo Franchini, H William Detrich, 3rd, Hannes Svardal, Maximilian Wagner, Gavin J P Naylor, Martin Pippel, Milan Malinsky, Mark Mooney, Maria Simbirsky, Brett T Hannigan, Trevor Pesout, Marlys Houck, Ann Misuraca, Sarah B Kingan, Richard Hall, Zev Kronenberg, Ivan Sović, Christopher Dunn, Zemin Ning, Alex Hastie, Joyce Lee, Siddarth Selvaraj, Richard E Green, Nicholas H Putnam, Ivo Gut, Jay Ghurye, Erik Garrison, Ying Sims, Joanna Collins, Sarah Pelan, James Torrance, Alan Tracey, Jonathan Wood, Robel E Dagnew, Dengfeng Guan, Sarah E London, David F Clayton, Claudio V Mello, Samantha R Friedrich, Peter V Lovell, Ekaterina Osipova, Farooq O Al-Ajli, Simona Secomandi, Heebal Kim, Constantina

Theofanopoulou, Michael Hiller, Yang Zhou, Robert S Harris, Kateryna D Makova, Paul Medvedev, Jinna Hoffman, Patrick Masterson, Karen Clark, Fergal Martin, Kevin Howe, Paul Flicek, Brian P Walenz, Woori Kwak, Hiram Clawson, Mark Diekhans, Luis Nassar, Benedict Paten, Robert H S Kraus, Andrew J Crawford, M Thomas P Gilbert, Guojie Zhang, Byrappa Venkatesh, Robert W Murphy, Klaus-Peter Koepfli, Beth Shapiro, Warren E Johnson, Federica Di Palma, Tomas Marques-Bonet, Emma C Teeling, Tandy Warnow, Jennifer Marshall Graves, Oliver A Ryder, David Haussler, Stephen J O'Brien, Jonas Korlach, Harris A Lewin, Kerstin Howe, Eugene W Myers, Richard Durbin, Adam M Phillippy, and Erich D Jarvis. Towards complete and error-free genome assemblies of all vertebrate species. *Nature*, 592(7856):737–746, April 2021.

Marylyn D Ritchie, Emily R Holzinger, Ruowang Li, Sarah A Pendergrass, and Dokyoon Kim. Methods of integrating data to uncover genotype–phenotype interactions. *Nat. Rev. Genet.*, 16(2):85–97, January 2015.

Eugene Rosenberg and Ilana Zilber-Rosenberg. *The Hologenome Concept: Human, Animal and Plant Microbiota.* Springer, Cham, 2013.

Andreas Schroeder, Odilo Mueller, Susanne Stocker, Ruediger Salowsky, Michael Leiber, Marcus Gassmann, Samar Lightfoot, Wolfram Menzel, Martin Granzow, and Thomas Ragg. The RIN: an RNA integrity number for assigning integrity values to RNA measurements. *BMC Mol. Biol.*, 7:3, January 2006.

José A Seoane, Ian N M Day, Tom R Gaunt, and Colin Campbell. A pathway-based data integration framework for prediction of disease progression. *Bioinformatics*, 30(6):838–845, March 2014.

Michael Shaffer, Mikayla A Borton, Bridget B McGivern, Ahmed A Zayed, Sabina Leanti La Rosa, Lindsey M Solden, Pengfei Liu, Adrienne B Narrowe, Josué Rodríguez-Ramos, Benjamin Bolduc, M Consuelo Gazitúa, Rebecca A Daly, Garrett J Smith, Dean R Vik, Phil B Pope, Matthew B Sullivan, Simon Roux, and Kelly C Wrighton. DRAM for distilling microbial metabolism to automate the curation of microbiome function. *Nucleic Acids Res.*, 48(16):8883–8900, September 2020.

Ronglai Shen, Adam B Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, November 2009.

Hyunjung Shin, N Jeremy Hill, Andreas Martin Lisewski, and Joon-Sang Park. Graph sharpening. *Expert Syst. Appl.*, 37(12):7870–7879, December 2010.

Christian M K Sieber, Alexander J Probst, Allison Sharrar, Brian C Thomas, Matthias Hess, Susannah G Tringe, and Jillian F Banfield. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nat Microbiol*, 3(7):836–843, July 2018.

Felipe A Simão, Robert M Waterhouse, Panagiotis Ioannidis, Evgenia V Kriventseva, and Evgeny M Zdobnov. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19): 3210–3212, October 2015.

Nora K Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–75, June 2015.

Kaiwen Tan, Weixian Huang, Jinlong Hu, and Shoubin Dong. A multi-omics supervised autoencoder for pan-cancer clinical outcome endpoints prediction. *BMC Med. Inform. Decis. Mak.*, 20(Suppl 3):129, July 2020.

Yasuhiro Tanizawa, Takatomo Fujisawa, and Yasukazu Nakamura. DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication. *Bioinformatics*, 34(6):1037–1039, November 2017.

Yasin Ilkagan Tepeli, Ali Burak Ünal, Furkan Mustafa Akdemir, and Oznur Tastan. PAMOGK: a pathway graph kernel-based multiomics approach for patient clustering. *Bioinformatics*, 36(21):5237–5246, January 2021.

Kevin R Theis, Nolwenn M Dheilly, Jonathan L Klassen, Robert M Brucker, John F Baines, Thomas C G Bosch, John F Cryan, Scott F Gilbert, Charles J Goodnight, Elisabeth A Lloyd, Jan Sapp, Philippe Vandenkoornhuyse, Ilana Zilber-Rosenberg, Eugene Rosenberg, and Seth R Bordenstein. Getting the hologenome concept right: an Eco-Evolutionary framework for hosts and their microbiomes. *mSystems*, 1(2), March 2016.

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. https://www.jmlr.org/papers/volume1/tipping01a/tipping01a.pdf?ref=https://githubhelp.com, 2001. Accessed: 2023-4-18.

Gherman V Uritskiy, Jocelyne DiRuggiero, and James Taylor. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome*, 6(1):1–13, September 2018.

Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods*, 11(3): 333–337, March 2014.

Hengchao Wang, Bo Liu, Yan Zhang, Fan Jiang, Yuwei Ren, Lijuan Yin, Hangwei Liu, Sen Wang, and Wei Fan. Estimation of genome size using k-mer frequencies from corrected long reads. March 2020.

Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nat. Commun.*, 12(1):3445, June 2021.

Chia-Chin Wu, Shahab Asgharzadeh, Timothy J Triche, and David Z D'Argenio. Prediction of human functional genetic networks from heterogeneous data using RVM-based ensemble learning. *Bioinformatics*, 26(6):807–813, March 2010.

Dingming Wu, Dongfang Wang, Michael Q Zhang, and Jin Gu. Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification. *BMC Genomics*, 16:1022, December 2015.

Hsin-Jung Wu and Eric Wu. The role of gut microbiota in immune homeostasis and autoimmunity. *Gut Microbes*, 3(1):4–14, January 2012.

Kang K Yan, Hongyu Zhao, and Herbert Pang. A comparison of graph- and kernel-based –omics data integration algorithms for classifying complex traits. *BMC Bioinformatics*, 18(1), December 2017.

Bo Zhang, Matthew Brock, Carlos Arana, Chaitanya Dende, Nicolai Stanislas van Oers, Lora V Hooper, and Prithvi Raj. Impact of Bead-Beating intensity on the genus- and Species-Level characterization of the gut microbiome using amplicon and complete 16S rRNA gene sequencing. *Front. Cell. Infect. Microbiol.*, 11:678522, October 2021.

Li Zhang, Chenkai Lv, Yaqiong Jin, Ganqi Cheng, Yibao Fu, Dongsheng Yuan, Yiran Tao, Yongli Guo, Xin Ni, and Tieliu Shi. Deep Learning-Based Multi-Omics data integration reveals two prognostic subtypes in High-Risk neuroblastoma. *Front. Genet.*, 9:477, October 2018.