Analysis of High-Dimensional Biological Data

Lecture Script, Version 0.6

Christian L. Müller

2026-07-23

Table of contents

Analysis of High-Dimensional Biological Data **8**
Lecture Overview . . . 8
How To Use These Notes . . . 8
Course Roadmap . . . 9
Recurring Principles . . . 9

1 Course Overview, Biology Recap, and Statistical Workflows **11**
Chapter Aim . . . 11
Learning Goals . . . 11
1.1 Why This Course Exists . . . 12
1.2 From Biology To Data Objects . . . 13
1.3 The Central Data Object . . . 14
1.4 What Can Serve As X? . . . 15
1.5 What Are The Y Variables? . . . 16
1.6 What Are The Statistical Tasks? . . . 17
1.7 Distributional Starting Points . . . 19
1.8 Why High-Dimensional Biology Is Hard . . . 19
1.9 A Statistical Workflow As Objects . . . 20
Suggested Reading . . . 21
References . . . 21

2 Distributions and Generative Models for Biological Data **22**
Chapter Aim . . . 22
Learning Goals . . . 22
2.1 From A Data Matrix To A Random Variable . . . 23
2.2 What Makes A Distribution Appropriate? . . . 24
2.3 A Coherent Distribution Map . . . 24
2.4 Continuous Distributions . . . 25
2.5 Binary And Categorical Distributions . . . 27
2.6 Count Distributions: Poisson And Negative Binomial . . . 29
2.7 Discrete Compositions: Multinomial And Dirichlet-Multinomial . . . 30
2.8 Continuous Compositions: Log-Ratios And The Logistic-Normal Model . . . 33
2.9 The Overdispersion Pattern . . . 35
2.10 Mixtures And Zero Inflation . . . 35
2.11 Likelihood And Model Fitting . . . 36

2.12 Model Checking . . . 37
Summary . . . 37
References . . . 38

3 Variance Stabilization and Normalization 39
Chapter Aim . . . 39
Learning Goals . . . 40
3.1 What Normalization Is Trying To Estimate . . . 40
3.2 Library Size And Total-Count Scaling . . . 41
3.3 Common-Scale Counts And Rarefying . . . 42
3.4 Median-Of-Ratios Normalization . . . 42
3.5 Mean-Variance Relationships . . . 43
3.6 The Delta Method . . . 44
3.7 Important Variance-Stabilizing Transformations . . . 45
3.8 DESeq2 VST, rlog, And Model-Based Transformations . . . 46
3.9 Single-Cell RNA-seq: Simple Logs And Residuals . . . 47
3.10 Compositional Normalization And Log-Ratios . . . 48
3.11 How Transformation Choices Affect Downstream Analysis . . . 50
3.12 Practical Summary . . . 50
References . . . 51

4 Distances, Dimensionality Reduction, and (Non-)Linear Embeddings 52
Chapter Aim . . . 52
Learning Goals . . . 53
4.1 Geometry Starts With Preprocessing . . . 53
4.2 Distances And Dissimilarities . . . 54
4.3 From Distances To Similarities And Graphs . . . 56
4.4 PCA As Linear Dimension Reduction . . . 57
4.5 Classical MDS And PCoA . . . 59
4.6 Non-Metric MDS . . . 60
4.7 The High-Dimensional Problem . . . 61
4.8 Nonlinear Embeddings . . . 61
4.8.1 ISOMAP . . . 62
4.8.2 t-SNE . . . 63
4.8.3 UMAP . . . 64
4.9 Interpreting Embeddings . . . 65
Summary . . . 65
References . . . 66

5 Clustering 67
Chapter Aim . . . 67
Learning Goals . . . 68
5.1 What A Clustering Algorithm Returns . . . 68

5.2 Choosing The Object To Cluster . . . 69
5.3 Centroid-Based Clustering: k-Means . . . 69
5.4 Medoids And Robust Representatives . . . 70
5.5 Hierarchical Clustering . . . 71
5.6 Heatmaps As Clustered Views . . . 72
5.7 Choosing The Number Of Clusters . . . 73
5.8 Silhouette And Cluster Separation . . . 74
5.9 Stability As Validation . . . 74
5.10 Density-Based Clustering . . . 75
5.11 Model-Based Clustering . . . 76
5.12 Graph-Based Clustering In Single-Cell Workflows . . . 76
5.13 What To Report . . . 78
Summary . . . 78
References . . . 79

6 Hypothesis Testing **80**
Chapter Aim . . . 80
Learning Goals . . . 81
6.1 The Five-Step Logic Of Testing . . . 81
6.2 Errors, Power, And Effect Sizes . . . 82
6.3 The Two-Sample t-Test . . . 83
6.4 Rank-Based Testing . . . 84
6.5 Permutation Tests . . . 85
6.6 Borrowing Strength Across Features . . . 86
6.7 Count Models For Differential Expression . . . 86
6.8 Genome-Wide Association Testing . . . 87
6.9 Multiple Testing . . . 88
6.10 Benjamini-Hochberg FDR Control . . . 90
6.11 P-Value Histograms And Independent Filtering . . . 91
6.12 Design, Covariates, And Confounding . . . 91
6.13 Community-Level Testing With PERMANOVA . . . 92
6.14 What To Report . . . 93
Summary . . . 93
References . . . 94

7 Regression and Classification **95**
Chapter Aim . . . 95
Learning Goals . . . 96
7.1 From Testing To Regression . . . 96
7.2 Feature-Wise Linear Models . . . 97
7.3 Contrasts . . . 98
7.4 Wald Tests For Regression Coefficients . . . 99
7.5 Generalized Linear Models . . . 100

7.6 Negative-Binomial Regression For Count Data . . . 100
7.7 Regression For Prediction . . . 102
7.8 Classification And Logistic Regression . . . 103
7.9 High-Dimensional Regression And Regularization . . . 104
7.10 Model Selection For Penalized Regression . . . 104
7.11 Sparse Logistic Regression . . . 106
7.12 Sparse Log-Contrast Regression For Compositions . . . 107
7.13 What To Report . . . 108
Summary . . . 109
References . . . 109

8 Network Learning **111**
Chapter Aim . . . 111
Learning Goals . . . 112
8.1 What Does An Edge Mean? . . . 112
8.2 Gaussian Graphical Models . . . 113
8.3 Why Sparsity Is Needed . . . 114
8.4 Node-Wise Lasso Regression . . . 115
8.5 From Node-Wise Regression To Precision Matrices . . . 116
8.6 Graphical Lasso . . . 117
8.7 Model Selection With StARS . . . 117
8.8 Compositional Data And SPIEC-EASI . . . 118
8.9 What To Report . . . 119
Summary . . . 120
References . . . 120

9 Network Analysis **122**
Chapter Aim . . . 122
Learning Goals . . . 123
9.1 From Learned Matrices To Networks . . . 123
9.2 Node-Level Summaries . . . 125
9.3 Global Network Size And Density . . . 126
9.4 Paths, Diameter, And Clustering . . . 127
9.5 Centrality . . . 128
9.6 Communities And Modularity . . . 129
9.7 Small-World Structure And Null Models . . . 129
9.8 Generative Null Models For Networks . . . 130
9.8.1 Erdos-Renyi Graphs . . . 130
9.8.2 Random Geometric Graphs In Two Or Three Dimensions . . . 131
9.8.3 Watts-Strogatz Small-World Graphs . . . 132
9.8.4 Scale-Free Graphs By Preferential Attachment . . . 132
9.8.5 Sticky And Configuration Models . . . 133
9.9 Fitting A Null Model To An Observed Graph . . . 134

9.10 Comparing Networks . . . 136
9.11 Graphlets And Graphlet Correlation Distance . . . 137
9.12 Robustness And Attack Analysis . . . 138
9.13 What To Report . . . 139
Summary . . . 140
References . . . 140

10 Matrix Factorization and Autoencoders **142**
Chapter Aim . . . 142
Learning Goals . . . 143
10.1 The Factorization Object . . . 143
10.2 PCA And The Singular Value Decomposition . . . 144
10.3 Nonnegative Matrix Factorization . . . 145
10.4 Prior Structure And Network Component Analysis . . . 145
10.5 Multi-Omics Factor Models . . . 146
10.6 Generalized Matrix Factorization For Counts . . . 147
10.7 Autoencoders As Nonlinear Factorizations . . . 148
10.8 Count-Aware Autoencoders And DCA . . . 149
10.9 Validation And Interpretation . . . 150
Summary . . . 151
References . . . 151

11 Deep Learning and Foundation Models **153**
Chapter Aim . . . 153
Learning Goals . . . 154
11.1 From Hand-Crafted Features To End-To-End Learning . . . 154
11.2 Representations And Multimodal Biology . . . 155
11.3 Sequence-To-Function Models . . . 156
11.4 Count-Aware Deep Models For Single Cells . . . 157
11.5 The Pretrain And Fine-Tune Paradigm . . . 157
11.6 ESM As An Archetypical Foundation Model . . . 159
11.7 Protein Structure And Learned Geometry . . . 160
11.8 Evaluation Still Decides The Claim . . . 160
11.9 What Changes, And What Does Not . . . 161
References . . . 162

12 Notation and Dimensions **163**
12.1 Core Convention . . . 163
12.2 Data Matrices . . . 163
12.3 Indices . . . 164
12.4 Probability Operators . . . 164
12.5 Counts and Normalized Data . . . 165

12.6 Distributional View Of X . . . 165
12.6.1 Entry-wise View . . . 165
12.6.2 Row-wise Compositional View . . . 166
12.6.3 Column-wise Feature View . . . 166
12.6.4 Practical Rule . . . 167
12.7 Metadata and Design Matrices . . . 167
12.8 Dimension Reduction . . . 167
12.9 Clustering . . . 168
12.10 Testing . . . 168
12.11 Networks . . . 168
12.12 Deep Learning and Latent Models . . . 169
12.13 Dimension Checklist . . . 169

13 Resources **170**
13.1 Course Materials . . . 170
13.2 Hosting Plan . . . 170
13.3 Exercise Sheets . . . 171
13.4 Chapter Links . . . 171
13.5 How To Reuse Links . . . 171

14 Complete References **174**
References . . . 174

Analysis of High-Dimensional Biological Data

Lecture Overview

High-dimensional biological data arise when modern measurement technologies record many molecular, cellular, or spatial variables for each biological observation. RNA sequencing, single-cell sequencing, proteomics, metabolomics, methylation assays, microbiome profiling, and spatial omics can produce thousands to millions of measurements per experiment. The promise is systems-level resolution; the difficulty is that the number of measured features is often much larger than the number of independent observations.

These lecture notes are the mathematical companion to the highly visual lecture slides and the course book. The slides provide intuition, examples, and visual anchors. The notes keep notation, dimensions, modeling assumptions, optimization problems, and interpretation explicit. Modern Statistics for Modern Biology serves as the main external online reference for fuller treatment, examples, R code, and exercises.

> **Course materials**
>
> The notes should be read together with the visual lecture slides and the course book. The main external online reference is Modern Statistics for Modern Biology by Susan Holmes and Wolfgang Huber, especially the chapters on data visualization, multivariate analysis, high-throughput count data, clustering, testing, and supervised learning.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

How To Use These Notes

The notes are not meant to replace the slides or the book. They provide the mathematical spine of the course: notation, assumptions, objectives, models, and interpretation. Use the

linked book sections for deeper derivations and code, use the exercises for computation, and use the slides for the visual intuition that is intentionally kept lighter here.

The recurring workflow is simple but demanding: formulate a biological question, identify the observation and feature units, decide whether the measurements are counts, continuous values, binary variables, or compositions, choose a preprocessing and modeling scale, and validate whether the resulting pattern is stable and biologically plausible.

Course Roadmap

Chapter 01 introduces the biological context, matrix notation, and the object-oriented statistical workflow. Chapter 02 reviews probability distributions as generative models for continuous, binary, count, and compositional data. Chapter 03 connects normalization, variance stabilization, residuals, and log-ratio transformations to the models that motivate them. Chapters 04 and 05 turn transformed data into geometry: distances, embeddings, clustering, and cluster validation.

The second half of the course moves from exploratory structure to formal inference and modern high-dimensional modeling. Chapter 06 covers hypothesis testing and false discovery control. Chapter 07 develops regression, classification, regularization, sparse logistic regression, and log-contrast regression. Chapters 08 and 09 separate network learning from network analysis. Chapters 10 and 11 connect matrix factorization, autoencoders, deep learning, and biological foundation models.

Recurring Principles

Dimensions first. Every analysis begins by defining the rows, columns, and entry type of the data object. The most common analysis matrix is $X \in \mathbb{R}^{n \times p}$, but raw sequencing data are usually better represented as a count matrix $C \in \mathbb{N}_0^{n \times p}$.

Models and transformations belong together. Log transforms, variance-stabilizing transforms, Pearson residuals, and log-ratio coordinates are not arbitrary preprocessing tricks. They are tied to assumptions about how the data were generated and what variation should be treated as noise.

Exploration is not confirmation. PCA plots, UMAP embeddings, heatmaps, clusters, and networks can reveal structure, but they do not by themselves prove biological mechanisms. Testing, prediction, model selection, and validation each require their own assumptions.

High dimensionality changes evidence. When p is large relative to n, random correlations, overfitting, batch effects, and multiple testing become central rather than peripheral. A good

analysis therefore reports not only a result, but also the data scale, preprocessing, model, tuning choices, and validation strategy.

1 Course Overview, Biology Recap, and Statistical Workflows

Chapter Aim

This chapter sets up the course. We first establish the biological motivation: modern biology can now measure genomes, transcriptomes, proteomes, metabolomes, microbiomes, images, and other molecular or cellular signals at large scale. We then introduce the statistical object that will appear throughout the lecture:

$$X \in \mathbb{R}^{n \times p}. \tag{1.1}$$

Here rows are biological observations and columns are molecular features. The central difficulty is that p, the number of measured features, is often much larger than n, the number of observations. This high-dimensional structure makes visualization, normalization, modeling, testing, prediction, and interpretation tightly connected.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

> **Main external reference**
>
> For the statistical philosophy behind this chapter, read the MSMB Introduction. For the role of plots in statistical workflows, see MSMB Data Visualization. For experimental design, confounding, quality control, and reproducibility, see MSMB Experimental Design.

Learning Goals

After this chapter, students should be able to:

1. Explain why high-throughput biological measurements naturally lead to high-dimensional data.
2. Interpret $X \in \mathbb{R}^{n \times p}$ as a samples-by-features or cells-by-features data matrix.
3. Distinguish raw count matrices C from processed analysis matrices X.
4. Give examples of biological observations and molecular features across omics data types.
5. Describe common statistical tasks: visualization, prediction, association testing, clustering, network construction, and regression.
6. Explain why high dimensionality, compositionality, sparsity, noise, batch effects, and confounding make biological data analysis difficult.
7. Describe the course workflow: biological question, measurement technology, data matrix, preprocessing, exploratory analysis, modeling, validation, and interpretation.

1.1 Why This Course Exists

Modern biology is no longer limited to measuring one gene, one protein, or one phenotype at a time. A single experiment can now produce thousands to millions of measurements. Sequencing technologies can read DNA fragments, quantify RNA molecules, profile microbial communities, or measure genomic variation across many individuals. Other technologies measure proteins, metabolites, images, spatial locations, or combinations of these modalities.

This creates a new kind of statistical problem. The data are rich enough to reveal biological structure, but they are also noisy, heterogeneous, and high-dimensional. More measurements do not automatically mean more understanding. Without careful statistical workflows, high-throughput data can produce convincing but misleading patterns.

The course therefore has two linked goals. First, we need enough biological intuition to understand what was measured and what the entries of the data matrix mean. Second, we need enough statistical intuition to analyze those entries without losing track of dimensions, assumptions, and biological meaning. The main object of the course is not a single method, but the disciplined translation from a biological question to a mathematical object and then to a statistical operation.

The lecture script is a mathematical companion to the highly visual lecture slides and the course book. The slides provide intuition and examples, while the script keeps notation, assumptions, dimensions, and model statements explicit. A deeper external reference is Modern Statistics for Modern Biology, which provides worked examples, R and Bioconductor code, exercises, and more detail.

1.2 From Biology To Data Objects

The biological recap is not meant to replace a biology textbook. Its role in this course is to explain what the rows, columns, and entries of high-dimensional biological data matrices can represent.

At the level of notation, the main point is:

$$\text{biological system} \longrightarrow \text{measurement technology} \longrightarrow \text{data object} \longrightarrow \text{statistical model}. \tag{1.2}$$

Most chapters begin somewhere along this chain. For example, a sequencing experiment starts with DNA, RNA, or amplicons, but the statistical analysis usually begins once the measurement has been summarized as a table of counts or abundances.

The biological recap starts with the cell as the fundamental biological unit. Cells contain DNA, RNA, proteins, metabolites, organelles, membranes, and molecular machines. Many high-throughput measurements are attempts to observe one layer of this system at scale.

DNA, genes, and genomes. DNA stores hereditary information as a sequence over a four-letter alphabet. A genome is the full DNA content of an organism. Genes are DNA segments that can contribute to functional products, such as proteins or functional RNAs. The molecular basis of DNA structure is historically anchored by the double-helix model (WATSON and CRICK 1953).

Sequencing technologies read pieces of DNA or RNA. Historically, sequencing was expensive and slow. The Human Genome Project made whole-genome sequencing a landmark scientific project (Venter et al. 2001). Since then, sequencing costs have fallen dramatically, and next-generation sequencing has made large-scale genomic measurement routine.

For statistics, the important point is not only that we can read sequences. It is that sequencing creates structured data. Reads may be aligned to genomic locations, summarized as counts assigned to genes, transcripts, peaks, taxa, or variants, or converted into categorical variables such as mutation status. In microbiome experiments, sequencing reads often become relative abundances or discrete compositions. In addition, every molecular measurement is accompanied by metadata about samples, patients, tissues, batches, treatments, or experimental conditions.

These measurements become the raw material for C and X. In this script, the biological story is always translated into the dimension and type of a data object. Whenever genes, cells, tissues, or sequencing reads appear in an example, the first questions are: what is the row, what is the column, and what kind of entry is being measured?

This translation from biological sequence to statistical object is developed computationally in Exercise Sheet 01, where DNA sequences are summarized by nucleotide frequencies and

GC content before being connected to simple generative models and likelihood calculations (Exercise Sheet 01 2026).

Transcription and translation. The central molecular biology flow is often summarized as DNA to RNA to protein, following the central dogma of molecular biology (Crick 1970). Through transcription, a gene is copied into RNA. Through translation, ribosomes read messenger RNA and synthesize proteins.

This flow motivates several omics layers. Genomics studies DNA sequence and genomic variation. Transcriptomics studies RNA abundance and gene expression. Proteomics studies proteins and protein abundance. Metabolomics studies small molecules and metabolic states. Each layer has its own measurement technology and its own statistical complications. This is why the same broad matrix notation $X \in \mathbb{R}^{n \times p}$ is useful, but not sufficient by itself. We must also know what the entries mean.

Transcriptomics, proteomics, and metabolomics. Transcriptomic data measure RNA abundance. In modern workflows this often means RNA sequencing, so the raw entries are usually nonnegative counts. After filtering, normalization, and variance-stabilizing or logarithmic transformations, the entries of the analysis matrix X are often treated as approximately continuous values. Early genome-wide expression studies already used matrix displays and clustering to reveal structure across genes and conditions (Eisen et al. 1998).

Proteomic data measure proteins or peptides. Their entries are usually continuous or semi-continuous abundance estimates rather than sequencing counts. They can have missing values, dynamic range issues, platform-specific noise, and batch effects. Metabolomic data are similar in that they often produce continuous intensity or abundance measurements, but they are additionally affected by compound annotation, sample preparation, and instrument sensitivity.

The key lesson is that every omics layer can be put into a matrix, but the entries do not all mean the same thing. A genomic mutation matrix may be binary or categorical. A copy-number matrix may be integer-valued or continuous after preprocessing. A transcriptomic RNA-seq matrix begins as counts. A proteomic or metabolomic matrix is often continuous. A microbiome matrix usually begins as counts but is often interpreted compositionally. These distinctions determine which probability models, transformations, distances, and tests are appropriate.

1.3 The Central Data Object

Throughout the course, the central data object is:

$$X \in \mathbb{R}^{n \times p}. \tag{1.3}$$

The rows of X are observations. Depending on the experiment, an observation may be a patient, tissue sample, cell, spatial location, environmental sample, or experimental unit. The columns of X are molecular features. A feature may be a gene, transcript, protein, metabolite, taxon, genomic locus, chromatin peak, or image-derived measurement.

The entry x_{ij} is the value of feature j in observation i. The row vector $x_i^\top \in \mathbb{R}^{1\times p}$ describes one observation across all features. The column vector $x_{\cdot j} \in \mathbb{R}^n$ describes one feature across all observations.

For raw sequencing counts, we use:

$$C \in \mathbb{N}_0^{n\times p}. \tag{1.4}$$

The count c_{ij} is the raw count for feature j in observation i. After filtering, normalization, transformation, centering, or scaling, the analysis matrix is written as X. This distinction matters. A method that is reasonable for transformed values x_{ij} may be inappropriate for raw counts c_{ij}.

The following objects are used repeatedly:

Object	Dimension	Interpretation in Chapter 01
C	$n \times p$	raw nonnegative count matrix
X	$n \times p$	processed analysis matrix
$x_i^\top$	$1 \times p$	feature profile of observation i
$x_{\cdot j}$	n	feature j measured across observations
m_i	context-dependent	metadata for observation i
Z	$n \times r$	design or covariate matrix
y	n	response vector
Y	$n \times q$	response matrix with q outcomes

The row dimension must match whenever objects describe the same observations. For example, if X contains molecular measurements and Z contains metadata, both have n rows.

1.4 What Can Serve As X?

The same statistical layout can represent many biological data types.

Genomic data. Features may be genomic locations, variants, copy number regions, or mutation indicators. The statistical type depends on preprocessing: a single-nucleotide variant can be binary, categorical, or encoded as an allele count, while copy-number data may be integer-valued or continuous after log-ratio transformation and segmentation.

Transcriptomic data. Features are often genes or transcripts. For RNA-seq, the raw data matrix C contains nonnegative counts: c_{ij} is the number of reads or fragments assigned to feature j in observation i. After normalization and transformation, the analysis matrix X is typically treated as continuous. Unlike germline genomic variation, gene expression is dynamic across tissues, time, disease states, perturbations, and experimental conditions.

Proteomic and metabolomic data. Proteomic features are proteins or peptides; metabolomic features are small molecules or molecular peaks. Entries are usually continuous abundance or intensity estimates, although detection indicators, censoring, and missingness can introduce binary or semi-continuous components. These modalities are often affected by dynamic range, annotation uncertainty, instrument variation, and batch effects.

Microbiome data. Features may be taxa, operational taxonomic units, amplicon sequence variants, genes, or pathways. The raw entries are usually sequencing counts, so C is a nonnegative integer matrix. However, the observed row total depends on sequencing depth, and the data are often interpreted through relative abundances. This makes microbiome data both count data and discrete compositional data. They are also typically sparse and zero-inflated. Large-scale microbiome studies such as global ocean microbiome analyses illustrate how biological context, sequencing technology, and statistical structure meet (Sunagawa et al. 2015).

For example, if C is a microbiome count table, the row sum

$$N_i = \sum_{j=1}^{p} c_{ij}. \tag{1.5}$$

is the sequencing depth for observation i. The vector c_i can be viewed as a discrete composition constrained by N_i. This is one reason microbiome analysis requires care: changing one component's relative abundance changes the apparent proportions of others.

1.5 What Are The Y Variables?

Many statistical tasks relate X to an outcome or response Y. The type of Y again matters. Disease status is binary, disease subtype or tissue type is categorical, disease stage may be ordinal, body mass index and many molecular phenotypes are continuous, and survival outcomes combine an event indicator with a time variable. Treatment response may be binary, categorical, continuous, or time-to-event depending on the study design. Environmental conditions and experimental perturbations are often encoded as covariates in a design matrix rather than as a single response.

When there is a single response for each observation, we often write:

$$y \in \mathbb{R}^n. \tag{1.6}$$

For multiple responses, we may use:

$$Y \in \mathbb{R}^{n \times q}. \tag{1.7}$$

The row dimension must match X: the same n observations appear in both the molecular data and the response or metadata.

1.6 What Are The Statistical Tasks?

Once X, C, Z, and Y have been defined, the statistical task becomes a statement about one or more of these objects. The same biological data matrix can be used for prediction, association testing, clustering, dimension reduction, regression, or network construction. The task determines which mathematical object must be specified next: a prediction function, a hypothesis, a distance, a loss function, a graph, or a probability model.

Prediction. We may want to predict Y from X. Examples include predicting disease status from gene expression, treatment response from molecular profiles, or obesity status from microbiome composition.

The high-dimensional challenge is that $p \gg n$. There may be many more features than observations, so flexible models can overfit. Later chapters discuss regression, penalization, cross-validation, and prediction.

In the simplest supervised setting, the target is a vector $y \in \mathbb{R}^n$, and the goal is to learn a function

$$\hat{y} = f(X). \tag{1.8}$$

For a new observation $x_{\text{new}}^\top \in \mathbb{R}^{1 \times p}$, the model produces a prediction $\hat{y}_{\text{new}}$. The central statistical question is whether the learned function generalizes beyond the observed training data.

Association and biomarker discovery. We may want to test whether features are associated with a phenotype, condition, treatment, or outcome. In feature-wise analysis, we often examine one column $x_{\cdot j}$ or $c_{\cdot j}$ at a time.

This creates a multiple testing problem. If $p = 20{,}000$, then testing each feature separately means testing thousands of hypotheses. Chapter 06 returns to p-values, false discoveries, and multiple testing correction.

Written formally, feature-wise testing considers hypotheses

$$H_{0j} : \text{feature } j \text{ is not associated with the condition}, \qquad j = 1, \dots, p. \tag{1.9}$$

The result is a vector of p-values

$$(p_1, \dots, p_p), \tag{1.10}$$

which must be adjusted or interpreted with multiplicity in mind.

Structure discovery. We may want to discover structure within X without a response variable. This includes visualizing observations with PCA or nonlinear embeddings, clustering cells or samples, grouping features into modules, finding latent factors, and constructing networks between features. These tasks are often exploratory. They can suggest biological hypotheses, but they do not automatically prove them.

For example, dimension reduction constructs a lower-dimensional representation

$$U = f(X), \qquad U \in \mathbb{R}^{n \times d}, \tag{1.11}$$

where $d \ll p$. Clustering constructs labels

$$z_i \in \{1, \dots, K\}, \qquad i = 1, \dots, n. \tag{1.12}$$

Both U and z depend on preprocessing choices, distance choices, and modeling assumptions.

Network construction. We may want to examine associations between features, for example gene-gene relationships, microbial co-abundance patterns, or protein interaction patterns. Network construction asks which features appear connected, conditionally dependent, or functionally related.

If a network connects features, its adjacency matrix has dimension $p \times p$. If a graph connects observations, its adjacency matrix has dimension $n \times n$. This distinction will matter in the network chapters.

Formally, a network is written as

$$G = (V, E), \tag{1.13}$$

with adjacency matrix A. The dimension of A is determined by the node set V, not by the word "network." Always define the nodes.

1.7 Distributional Starting Points

Many high-throughput sequencing data sets begin as counts. A minimal entry-wise count model is:

$$c_{ij} \sim \text{Poisson}(\lambda_{ij}). \tag{1.14}$$

In practice, biological count data are often more variable than a Poisson model allows. A common feature-wise alternative is:

$$c_{ij} \sim \text{Negative Binomial}(\mu_{ij}, \phi_j), \tag{1.15}$$

where μ_{ij} is the expected count and ϕ_j is a feature-specific dispersion parameter.

Sequencing data also have a compositional aspect. For observation i, define the library size

$$N_i = \sum_{j=1}^{p} c_{ij}. \tag{1.16}$$

A row-wise model can treat the count vector c_i as a discrete composition:

$$c_i \sim \text{Multinomial}(N_i, \pi_i), \qquad \pi_i \in \Delta^{p-1}. \tag{1.17}$$

This row-wise view is especially important for microbiome data, where the observed counts are constrained by sequencing depth and often interpreted through relative abundance.

1.8 Why High-Dimensional Biology Is Hard

High-dimensional biological data are difficult for several reasons.

Large p, small n. The number of features often greatly exceeds the number of observations. A typical biological study may have tens or hundreds of samples but thousands or tens of thousands of features. This makes ordinary low-dimensional intuition unreliable.

Measurement noise. High-throughput technologies are powerful but imperfect. Sequencing depth, amplification bias, alignment, batch effects, instrument variation, sample quality, and preprocessing choices can all affect the data.

Heterogeneity. Biological systems are heterogeneous. Samples may differ by cell type composition, disease stage, genetic background, tissue origin, treatment history, and environment. Some heterogeneity is the biological signal; some is unwanted variation.

Compositionality. Sequencing often produces counts under a finite total. In microbiome data, amplicon sequencing measures relative abundance. In RNA-seq, library size differences also matter. This means raw counts cannot be compared naively across observations.

Sparsity and zeros. Many biological count matrices contain many zeros. A zero may mean absence, low abundance below detection, insufficient sequencing depth, dropout, or technical failure. The statistical interpretation depends on the technology.

Multiple testing. Testing thousands of features creates many opportunities for false positives. A method that seems reasonable for one feature can become misleading when repeated across all features.

Interpretation. Even when a model predicts well or a pattern looks clear, the biological interpretation may be uncertain. A predictive feature is not necessarily causal. A cluster is not automatically a cell type. A network edge is not automatically a direct interaction.

1.9 A Statistical Workflow As Objects

A useful analysis workflow starts with a biological question and turns it into a target quantity. If the technology produces counts, the first mathematical object is often the raw count matrix $C \in \mathbb{N}_0^{n \times p}$. The next objects are the metadata and covariates, usually represented by a design matrix $Z \in \mathbb{R}^{n \times r}$, and the analysis matrix $X \in \mathbb{R}^{n \times p}$ produced by filtering, normalization, transformation, centering, scaling, or embedding. At every stage, the analyst should be able to say what the rows are, what the columns are, and what statistical type the entries have.

After these objects are defined, the analysis can move to the statistical task. Prediction requires a function from X to y or Y. Testing requires null hypotheses and a multiplicity strategy. Clustering and dimension reduction require a representation and usually a distance or loss. Network analysis requires a node set and an edge definition. Regression requires a response, a design matrix, and assumptions about the error or count distribution. Validation then asks whether the result is stable under reasonable changes, whether it generalizes to new data, and whether it is biologically plausible.

This workflow is iterative. A first plot may reveal a batch effect. A clustering result may reveal mislabeled samples. A regression model may expose a confounder. Statistical analysis often loops back rather than moving in a straight line.

This interplay between exploratory data analysis and confirmatory analysis is a recurring theme in the main reference. The relevant distinction is whether a step is being used to generate hypotheses, validate a model, or support a confirmatory claim.

The script will use this principle throughout: first define the object and its dimensions, then define the statistical operation on that object.

Suggested Reading

Read the MSMB Introduction after this chapter. Pay particular attention to the discussion of heterogeneous biological data, the large p, small n problem, exploratory data analysis, and computational reproducibility.

For students who want to connect this chapter immediately to practice, MSMB Data Visualization shows how graphics support statistical thinking, MSMB Experimental Design explains why design and metadata matter before modeling, and MSMB High-Throughput Count Data gives an early view of RNA-seq count data.

The corresponding Exercise Sheet 01 moves from sequence data to count summaries, maximum likelihood, and mixture models, providing a practical bridge from the biological recap to the probability models of Chapter 02 (Exercise Sheet 01 2026).

References

Crick, Francis. 1970. "Central Dogma of Molecular Biology." *Nature* 227 (5258): 561–63. https://doi.org/10.1038/227561a0.

Eisen, Michael B., Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. "Cluster Analysis and Display of Genome-Wide Expression Patterns." *Proceedings of the National Academy of Sciences* 95 (25): 14863–68. https://doi.org/10.1073/pnas.95.25.14863.

Exercise Sheet 01. 2026. *Exercise Sheet 01 Solutions: Generative Models, Nucleotide Frequencies, Maximum Likelihood, and Mixture Models*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise01_Solution.html.

Sunagawa, Shinichi, Luis Pedro Coelho, Samuel Chaffron, et al. 2015. "Structure and Function of the Global Ocean Microbiome." *Science* 348 (6237). https://doi.org/10.1126/science.1261359.

Venter, J. Craig, Mark D. Adams, Eugene W. Myers, et al. 2001. "The Sequence of the Human Genome." *Science* 291 (5507): 1304–51. https://doi.org/10.1126/science.1058040.

WATSON, J. D., and F. H. C. CRICK. 1953. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid." *Nature* 171 (4356): 737–38. https://doi.org/10.1038/171737a0.

2 Distributions and Generative Models for Biological Data

Chapter Aim

Chapter 02 asks a basic modeling question: what kind of random mechanism could have generated the entries, rows, or columns of a high-throughput biological data matrix?

This chapter reviews probability distributions as generative models for biological data. The aim is to understand how the support, parameters, expectation, variance, and dependence structure of a distribution encode assumptions about the data. For high-throughput biological data, the most important distinction is between continuous measurements, binary or categorical measurements, count measurements, and discrete compositions.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

> **Main external reference**
>
> The main reference is MSMB Generative Models, especially the discussion of Bernoulli, binomial, multinomial, and Poisson models. For likelihood and maximum likelihood, see MSMB Statistical Modeling. For overdispersed count models in RNA-seq, see MSMB High-Throughput Count Data. For mixture models, including Gamma-Poisson ideas, see MSMB Mixture Models.

The corresponding computations are developed in Exercise Sheet 01, which starts from sequence summaries and then works through multinomial likelihoods, Poisson count models, Poisson mixtures, expectation-maximization, and Gaussian mixture models (Exercise Sheet 01 2026).

Learning Goals

After this chapter, students should be able to:

1. Identify whether a biological measurement is continuous, binary, categorical, count-valued, or compositional.
2. Choose a simple probability distribution whose support matches the data type.
3. Explain what it means to use a probability distribution as a generative model.
4. Distinguish entry-wise, row-wise, and column-wise models for C and X.
5. Define overdispersion and recognize why Poisson and multinomial models are often too restrictive for biological count data.
6. Explain how a Gamma-Poisson mixture leads to a negative binomial distribution.
7. Explain how a Dirichlet-multinomial model extends the multinomial model for overdispersed discrete compositions.
8. Describe Aitchison's logistic-normal model as a generative model for continuous compositions.
9. Connect generative assumptions to likelihood, estimation, model checking, and later normalization choices.

2.1 From A Data Matrix To A Random Variable

A generative model treats an observed entry c_{ij}, a row vector c_i, or a column vector $c_{\cdot j}$ as a realization of a random object. This changes the question from "what number did we observe?" to "what random mechanism could plausibly produce numbers like this?"

For an entry-wise model we might write

$$C_{ij} \sim P_{\theta_{ij}}, \tag{2.1}$$

where $P_{\theta_{ij}}$ is a probability distribution with parameter θ_{ij}. For a row-wise compositional model we might write

$$C_i \sim P_{\theta_i}, \tag{2.2}$$

where $C_i = (C_{i1}, \dots, C_{ip})^\top$ is the random count vector for observation i. For a feature-wise model we might isolate one column and write

$$Y_j = C_{\cdot j} \in \mathbb{N}_0^n. \tag{2.3}$$

The same data matrix can therefore support several distinct generative views. Entry-wise models focus on single observations of single features. Row-wise models focus on the composition of one observation across many features. Column-wise models focus on how one feature varies across observations.

2.2 What Makes A Distribution Appropriate?

The first requirement is support. A probability model should assign probability to values that the measurement can actually take. A binary mutation indicator needs a distribution on $\{0, 1\}$. A genotype count may need a distribution on $\{0, 1, 2\}$. A sequencing read count needs a distribution on nonnegative integers. A normalized protein abundance may need a continuous distribution on $\mathbb{R}$ or $\mathbb{R}_{>0}$. A relative abundance vector must live on a simplex.

The second requirement is moment structure. Many distributions imply a relationship between the mean and the variance, and this relationship is often decisive for biological data. A Poisson distribution has variance equal to its mean. A binomial distribution has variance smaller than or equal to a particular function of its mean. Biological count data often vary more than these simple models allow.

The third requirement is dependence structure. A model for a single entry does not automatically model dependencies between features. A multinomial model creates negative dependence among counts because the row total is fixed. A Dirichlet-multinomial model adds extra row-to-row variation in the underlying composition. Later network chapters will ask a different kind of dependence question: whether features are associated or conditionally dependent across observations.

2.3 A Coherent Distribution Map

The distributions in this chapter are best organized by the type of data they generate.

Data type	Support	Typical biological example	Basic distribution	Overdispersed or flexible extension
continuous real-valued	$\mathbb{R}$	transformed expression, regression residual	normal	mixture of normals, Laplace
continuous positive	$\mathbb{R}_{>0}$	rates, intensities, waiting times, positive abundances	gamma	gamma mixtures, log-normal
binary	$\{0, 1\}$	mutation present/absent, responder/non-responder	Bernoulli	beta-Bernoulli

Data type	Support	Typical biological example	Basic distribution	Overdispersed or flexible extension
bounded counts	$\{0, \dots, m\}$	number of successes among m trials	binomial	beta-binomial
categorical	$\{1, \dots, K\}$	class label, genotype category, nucleotide identity	categorical	mixture of categoricals
unbounded counts	$\mathbb{N}_0$	read counts, mutation counts, binding-site counts	Poisson	negative binomial, zero-inflated models
discrete composition	$\{c \in \mathbb{N}_0^p : \sum_j c_j = N\}$	amplicon or taxon count vector	multinomial	Dirichlet-multinomial
continuous composition	Δ^{p-1}	relative abundance vector	Dirichlet	logistic-normal

This table is not a decision tree. It is a map of assumptions. The model should be chosen because its support and variability match the scientific object, not because it is mathematically convenient.

2.4 Continuous Distributions

Continuous distributions are useful when measurements can be treated as real-valued after preprocessing. This is often appropriate for transformed gene expression, protein intensities, metabolite intensities, image-derived features, and residuals in regression models.

The normal distribution is the central reference model for real-valued data:

$$X \sim \mathcal{N}(\mu, \sigma^2), \qquad \mathbb{E}(X) = \mu, \qquad \mathrm{Var}(X) = \sigma^2. \tag{2.4}$$

Its density is

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \qquad x \in \mathbb{R}, \qquad \sigma^2 > 0. \tag{2.5}$$

The parameter $\mu \in \mathbb{R}$ is the mean and location parameter. The parameter $\sigma^2 > 0$ is the variance, and σ is the standard deviation. The normal distribution is symmetric and fully determined by these two parameters. It is often a useful model for transformed data or for residual variation, but it is not a good generative model for raw sequencing counts because counts are nonnegative integers and often show mean-variance relationships that are incompatible with constant variance.

For positive continuous quantities, the gamma distribution is useful. In this chapter it plays a special role because it will become a distribution over unobserved Poisson rates. We use the shape-rate parameterization:

$$\Lambda \sim \text{Gamma}(\alpha, \beta), \qquad \mathbb{E}(\Lambda) = \frac{\alpha}{\beta}, \qquad \text{Var}(\Lambda) = \frac{\alpha}{\beta^2}. \tag{2.6}$$

Its density is

$$f(\lambda \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda), \qquad \lambda > 0, \qquad \alpha > 0,\ \beta > 0. \tag{2.7}$$

Here α is the shape parameter and β is the rate parameter. Increasing α changes the shape and, together with β, controls the mean. Increasing β shrinks the distribution toward smaller values because this is a rate, not a scale. The gamma distribution has support on $\mathbb{R}_{>0}$. It is therefore natural for rates, intensities, and positive latent quantities. In a Gamma-Poisson model, it represents biological or technical variation in the expected count before a Poisson sampling step occurs.

The Laplace distribution is another continuous distribution that is useful in mixture examples. It is symmetric like the normal distribution, but more sharply peaked at its center and heavier-tailed:

$$f(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right), \qquad x \in \mathbb{R}, \qquad b > 0. \tag{2.8}$$

For this parameterization,

$$\mathbb{E}(X) = \mu, \qquad \text{Var}(X) = 2b^2. \tag{2.9}$$

The parameter $\mu \in \mathbb{R}$ is the location parameter and is also the mean and median. The parameter $b > 0$ is a scale parameter; larger b gives a wider distribution and heavier tails. For this course, the Laplace distribution is less important as a standalone biological model than as a reminder that the shape of a distribution matters. Two distributions can have similar centers but very different tails, and these differences affect estimation, robustness, and model checking.

2.5 Binary And Categorical Distributions

Binary data arise when each observation has only two possible outcomes. Examples include mutation present versus absent, responder versus non-responder, or a quality-control indicator. A Bernoulli random variable is written as

$$X \sim \text{Bernoulli}(\pi), \qquad \Pr(X = 1) = \pi, \qquad \Pr(X = 0) = 1 - \pi. \tag{2.10}$$

Equivalently, its probability mass function is

$$\Pr(X = x \mid \pi) = \pi^x (1 - \pi)^{1-x}, \qquad x \in \{0, 1\}, \qquad 0 \le \pi \le 1. \tag{2.11}$$

The expectation and variance are

$$\mathbb{E}(X) = \pi, \qquad \text{Var}(X) = \pi(1 - \pi). \tag{2.12}$$

The parameter π is the success probability and is also the mean. It is not a feature index; throughout the script the feature dimension is denoted by p, while probabilities for binary, categorical, and compositional models are denoted by π.

If m independent Bernoulli trials have the same success probability π, their sum has a binomial distribution:

$$X \sim \text{Binomial}(m, \pi), \qquad \Pr(X = x \mid m, \pi) = \binom{m}{x} \pi^x (1 - \pi)^{m-x}, \qquad x = 0, \dots, m. \tag{2.13}$$

The binomial moments are

$$\mathbb{E}(X) = m\pi, \qquad \text{Var}(X) = m\pi(1 - \pi). \tag{2.14}$$

Here $m \in \mathbb{N}$ is the fixed number of trials and π is the success probability in each trial. The random variable X counts successes, so its support is finite: $x = 0, \dots, m$. In biological data, m might be a known number of cells, reads, alleles, or trials, depending on the experiment.

The beta distribution is the two-category analogue of the Dirichlet distribution and is useful when the success probability itself varies across observations. We write

$$\Pi \sim \text{Beta}(a, b), \qquad \mathbb{E}(\Pi) = \frac{a}{a + b}, \qquad \text{Var}(\Pi) = \frac{ab}{(a + b)^2 (a + b + 1)}. \tag{2.15}$$

Its density is

$$f(\pi \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1}, \qquad 0 < \pi < 1, \qquad a > 0,\ b > 0. \tag{2.16}$$

The parameters a and b are positive concentration parameters. Their ratio controls the mean success probability, while their sum controls how tightly Π concentrates around that mean. If

$$\Pi_i \sim \text{Beta}(a, b), \qquad X_i \mid \Pi_i \sim \text{Binomial}(m, \Pi_i), \tag{2.17}$$

then the marginal distribution of X_i is beta-binomial, with probability mass function

$$\Pr(X_i = x \mid m, a, b) = \binom{m}{x}\frac{\Gamma(x+a)\Gamma(m-x+b)\Gamma(a+b)}{\Gamma(a)\Gamma(b)\Gamma(m+a+b)}, \qquad x = 0, \ldots, m. \tag{2.18}$$

This beta-binomial construction is the binary or bounded-count version of the same overdispersion idea used later for Gamma-Poisson and Dirichlet-multinomial models.

The categorical distribution generalizes Bernoulli data from two categories to K categories. If Y is a categorical random variable, then

$$\Pr(Y = k) = \pi_k, \qquad k = 1, \ldots, K, \qquad \sum_{k=1}^{K} \pi_k = 1. \tag{2.19}$$

The probability mass function can be written compactly as

$$\Pr(Y = y \mid \pi_1, \ldots, \pi_K) = \prod_{k=1}^{K} \pi_k^{\mathbb{1}\{y=k\}}, \qquad y \in \{1, \ldots, K\}. \tag{2.20}$$

The parameter vector $\pi = (\pi_1, \ldots, \pi_K)^\top$ contains category probabilities. Each π_k is nonnegative, and the probabilities sum to one. This distribution is useful for labels, classes, nucleotide identities, and other one-of-K outcomes. If we count the number of observations in each category across N independent trials, we arrive at the multinomial distribution.

2.6 Count Distributions: Poisson And Negative Binomial

The Poisson distribution is the simplest model for nonnegative unbounded counts. It is appropriate when events occur independently and at a constant rate over a fixed exposure. For virus mutation data, the observed quantity could be a mutation count along a sequence or over a time interval. In sequencing data, the observed quantity may be a read count assigned to a feature.

The Poisson model is

$$X \sim \text{Poisson}(\lambda), \qquad \Pr(X = x \mid \lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!}, \qquad x \in \mathbb{N}_0. \tag{2.21}$$

The parameter $\lambda > 0$ is both the rate and the mean number of events in the relevant exposure unit. The exposure unit must be part of the scientific definition: counts per gene, per genomic window, per cell, per sample, or per unit time are different random variables.

Its defining moment relationship is

$$\mathbb{E}(X) = \lambda, \qquad \text{Var}(X) = \lambda. \tag{2.22}$$

This equality between mean and variance is both elegant and restrictive. For biological count data, it is often too restrictive. Replicate counts with similar means may show much larger variability than the Poisson model allows. This phenomenon is called overdispersion.

Overdispersion means that the observed or modeled variance exceeds the variance implied by a simple reference model. For a Poisson reference model, overdispersion can be stated as

$$\text{Var}(X) > \mathbb{E}(X). \tag{2.23}$$

One generative explanation is that the Poisson rate itself varies across observations. Instead of assuming a fixed rate λ, we introduce a latent random rate Λ. To keep the notation consistent with feature-wise sequencing models, we use the negative-binomial mean-dispersion parameters $\mu > 0$ and $\phi \geq 0$. For $\phi > 0$, the Gamma-Poisson hierarchy is

$$X \mid \Lambda \sim \text{Poisson}(\Lambda), \qquad \Lambda \sim \text{Gamma}\left(\phi^{-1}, (\phi\mu)^{-1}\right), \tag{2.24}$$

where the gamma distribution uses the shape-rate parameterization from Equation 2.6. Under this parameterization,

$$\mathbb{E}(\Lambda) = \mu, \qquad \text{Var}(\Lambda) = \phi\mu^2. \tag{2.25}$$

This is the Gamma-Poisson model. It is generative in two stages: first draw an observation-specific rate, then draw a count conditional on that rate. Marginally, after integrating over Λ, the count follows a negative binomial distribution in mean-dispersion form:

$$X \sim \mathrm{NB}(\mu, \phi), \qquad \mathbb{E}(X) = \mu, \qquad \mathrm{Var}(X) = \mu + \phi\mu^2. \tag{2.26}$$

For $\phi > 0$, the corresponding probability mass function is

$$\Pr(X = x \mid \mu, \phi) = \frac{\Gamma(x + \phi^{-1})}{\Gamma(\phi^{-1})x!} \left(\frac{1}{1 + \phi\mu}\right)^{\phi^{-1}} \left(\frac{\phi\mu}{1 + \phi\mu}\right)^{x}, \qquad x \in \mathbb{N}_0. \tag{2.27}$$

The parameter $\mu > 0$ is the mean count. The parameter $\phi \geq 0$ is the dispersion parameter. If $\phi = 0$, the variance reduces to the Poisson variance μ and the gamma mixing distribution has collapsed to a fixed rate; the probability mass function in Equation 2.27 is then understood through its Poisson limit. As ϕ increases, the latent rates vary more strongly across observations, and the counts become more variable than the Poisson model permits. Some books and software describe the same negative-binomial family with size-probability parameters. In this script, however, p is reserved for the number of features, so the canonical notation is always $\mathrm{NB}(\mu, \phi)$. This mean-variance relationship is central in RNA-seq and other sequencing count models; see MSMB High-Throughput Count Data and reviews of count distributions for biological data (Townes 2020).

For feature-wise sequencing models, the expected count often includes an observation-specific size factor s_i:

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \mu_{ij} = s_i q_{ij}. \tag{2.28}$$

Here q_{ij} is an underlying abundance-like quantity and s_i accounts for sequencing depth or exposure. The dispersion ϕ_j may be feature-specific. This equation already points toward Chapter 03, where normalization and variance stabilization become necessary.

2.7 Discrete Compositions: Multinomial And Dirichlet-Multinomial

Sequencing data often have another structure beyond being counts: a row of counts is constrained by a total. For observation i, define the library size

$$N_i = \sum_{j=1}^{p} C_{ij}. \tag{2.29}$$

Conditional on this total, a simple model for the full count vector is multinomial:

$$C_i \mid N_i, \pi_i \sim \text{Multinomial}(N_i, \pi_i), \qquad \pi_i \in \Delta^{p-1}. \tag{2.30}$$

Here N_i is the fixed total count for observation i, and $\pi_i = (\pi_{i1}, \dots, \pi_{ip})^\top$ is the vector of feature probabilities for that observation. The entry π_{ij} is the probability that one draw from observation i is assigned to feature j.

The simplex is

$$\Delta^{p-1} = \left\{ \pi \in \mathbb{R}^p_{\geq 0} : \sum_{j=1}^{p} \pi_j = 1 \right\}. \tag{2.31}$$

The multinomial probability mass function is

$$\Pr(C_i = c_i \mid N_i, \pi_i) = \frac{N_i!}{\prod_{j=1}^{p} c_{ij}!} \prod_{j=1}^{p} \pi_{ij}^{c_{ij}}, \qquad \sum_{j=1}^{p} c_{ij} = N_i. \tag{2.32}$$

Under this model, the marginal expectation and variance of component j are

$$\mathbb{E}(C_{ij} \mid N_i, \pi_i) = N_i \pi_{ij}, \qquad \text{Var}(C_{ij} \mid N_i, \pi_i) = N_i \pi_{ij}(1 - \pi_{ij}). \tag{2.33}$$

The covariance between two different components is negative:

$$\text{Cov}(C_{ij}, C_{i\ell} \mid N_i, \pi_i) = -N_i \pi_{ij} \pi_{i\ell}, \qquad j \neq \ell. \tag{2.34}$$

This negative covariance is not necessarily biological competition. It is partly mathematical: when the total N_i is fixed, allocating more counts to one feature leaves fewer counts for the others.

The multinomial model is often too narrow for microbiome and other ecological count data because it assumes that the composition π_i is fixed. In practice, the composition itself varies across observations because of biological heterogeneity, sampling, environment, subject-to-subject variation, and technical effects. A generative extension is to make π_i random:

$$\pi_i \sim \text{Dirichlet}(\alpha), \qquad C_i \mid \pi_i, N_i \sim \text{Multinomial}(N_i, \pi_i). \tag{2.35}$$

The Dirichlet distribution is a distribution on the simplex. Its density is

$$f(\pi_i \mid \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{j=1}^{p} \Gamma(\alpha_j)} \prod_{j=1}^{p} \pi_{ij}^{\alpha_j - 1}, \qquad \pi_i \in \Delta^{p-1}, \qquad \alpha_j > 0. \tag{2.36}$$

The parameter vector $\alpha = (\alpha_1, \dots, \alpha_p)^\top$ contains positive concentration parameters, and

$$\alpha_0 = \sum_{j=1}^{p} \alpha_j \tag{2.37}$$

is the total concentration. The relative sizes of the α_j determine the mean composition. The magnitude of α_0 controls how strongly compositions concentrate around that mean: large α_0 produces less row-to-row variation in π_i, while small α_0 produces more variable and often more extreme compositions.

After integrating out π_i, the count vector has a Dirichlet-multinomial distribution:

$$C_i \sim \text{DirichletMultinomial}(N_i, \alpha). \tag{2.38}$$

Its probability mass function is

$$\Pr(C_i = c_i \mid N_i, \alpha) = \frac{N_i!}{\prod_{j=1}^{p} c_{ij}!} \frac{\Gamma(\alpha_0)}{\Gamma(N_i + \alpha_0)} \prod_{j=1}^{p} \frac{\Gamma(c_{ij} + \alpha_j)}{\Gamma(\alpha_j)}, \qquad \sum_{j=1}^{p} c_{ij} = N_i. \tag{2.39}$$

The mean composition is controlled by

$$\pi_{0j} = \frac{\alpha_j}{\alpha_0}, \tag{2.40}$$

Thus π_{0j} is the expected relative abundance of feature j under the random composition π_i.

The Dirichlet variance and covariance are

$$\text{Var}(\pi_{ij}) = \frac{\pi_{0j}(1 - \pi_{0j})}{\alpha_0 + 1}, \qquad \text{Cov}(\pi_{ij}, \pi_{i\ell}) = -\frac{\pi_{0j}\pi_{0\ell}}{\alpha_0 + 1}, \qquad j \neq \ell. \tag{2.41}$$

This equation makes the role of α_0 visible. Holding the mean composition π_0 fixed, increasing α_0 reduces variation in the latent composition π_i.

The variance of component j is inflated relative to the multinomial variance by a factor depending on α_0:

$$\text{Var}(C_{ij}) = N_i \pi_{0j}(1 - \pi_{0j}) \left(\frac{N_i + \alpha_0}{1 + \alpha_0} \right). \tag{2.42}$$

This is the compositional analogue of the Gamma-Poisson idea. The multinomial assumes a fixed composition; the Dirichlet-multinomial assumes a random composition. Randomness in π_i creates extra variation in the observed counts. Dirichlet-multinomial models are therefore

natural candidates for overdispersed ecological and microbiome count data (Harrison et al. 2020; Townes 2020).

2.8 Continuous Compositions: Log-Ratios And The Logistic-Normal Model

The Dirichlet and Dirichlet-multinomial models are natural first models for compositions, but they are not the only generative route. Aitchison's logistic-normal model treats a composition as the inverse image of an unconstrained Gaussian random vector under a log-ratio transformation (Aitchison and Shen 1980). This idea is especially important because many normalization and distance choices in Chapter 03 and Chapter 04 are built around log-ratio geometry rather than ordinary Euclidean geometry on raw proportions.

Let $\pi_i \in \Delta^{p-1}$ be the latent composition for observation i. The centered log-ratio transform compares each component with the geometric mean of all components:

$$\Delta_+^{p-1} = \left\{ \pi \in \mathbb{R}^p_{>0} : \sum_{j=1}^{p} \pi_j = 1 \right\} \tag{2.43}$$

is the positive interior of the simplex. Log-ratio transformations are defined on Δ_+^{p-1}, not on compositions with exact zeros.

$$g(\pi_i) = \left(\prod_{j=1}^{p} \pi_{ij} \right)^{1/p}. \tag{2.44}$$

$$\operatorname{clr}(\pi_i)_j = \log \frac{\pi_{ij}}{g(\pi_i)}, \qquad j = 1, \dots, p. \tag{2.45}$$

The vector $\operatorname{clr}(\pi_i)$ has length p, but its components sum to zero. It therefore lives in a $(p-1)$-dimensional subspace of $\mathbb{R}^p$. This is already a warning that compositions should not be treated as ordinary unconstrained vectors: only relative information is identifiable.

For a generative model it is convenient to use a log-ratio coordinate system with $p-1$ unconstrained coordinates. One simple choice is the additive log-ratio transform with the last component as reference:

$$\begin{aligned} U_i &= \operatorname{alr}(\pi_i) = \left(\log \frac{\pi_{i1}}{\pi_{ip}}, \dots, \log \frac{\pi_{i,p-1}}{\pi_{ip}} \right)^\top, \\ U_i &\in \mathbb{R}^{p-1}. \end{aligned} \tag{2.46}$$

The inverse transform maps $U_i = (u_{i1}, \dots, u_{i,p-1})^\top$ back to the simplex:

$$\begin{aligned} \pi_{ij} &= \frac{\exp(u_{ij})}{1 + \sum_{\ell=1}^{p-1} \exp(u_{i\ell})}, \qquad j = 1, \dots, p-1, \\ \pi_{ip} &= \frac{1}{1 + \sum_{\ell=1}^{p-1} \exp(u_{i\ell})}. \end{aligned} \tag{2.47}$$

A logistic-normal model is then defined by placing a multivariate normal distribution on these log-ratio coordinates:

$$\begin{aligned} U_i &\sim \mathcal{N}_{p-1}(m, \Sigma), \\ \pi_i &= \mathrm{alr}^{-1}(U_i). \end{aligned} \tag{2.48}$$

Here $m \in \mathbb{R}^{p-1}$ is the mean vector in additive log-ratio coordinates, and $\Sigma \in \mathbb{R}^{(p-1)\times(p-1)}$ is a symmetric positive-definite covariance matrix. The density of U_i is

$$\begin{aligned} q_U(u) &= (u-m)^\top \Sigma^{-1} (u-m), \\ f_U(u \mid m, \Sigma) &= \frac{1}{(2\pi)^{(p-1)/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} q_U(u) \right\}, \\ & \qquad u \in \mathbb{R}^{p-1}. \end{aligned} \tag{2.49}$$

The induced logistic-normal density on the simplex, written in the same additive log-ratio coordinates, is

$$\begin{aligned} q(\pi) &= \left(\mathrm{alr}(\pi) - m\right)^\top \Sigma^{-1} \left(\mathrm{alr}(\pi) - m\right), \\ f_\Pi(\pi \mid m, \Sigma) &= \frac{1}{(2\pi)^{(p-1)/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} q(\pi) \right\} \\ & \quad \times \prod_{j=1}^{p} \pi_j^{-1}, \qquad \pi \in \Delta_+^{p-1}. \end{aligned} \tag{2.50}$$

For sequencing counts, the composition can be coupled to a multinomial sampling layer:

$$\begin{aligned} C_i \mid N_i, \pi_i &\sim \mathrm{Multinomial}(N_i, \pi_i), \\ \pi_i &= \mathrm{alr}^{-1}(U_i), \\ U_i &\sim \mathcal{N}_{p-1}(m, \Sigma). \end{aligned} \tag{2.51}$$

This model is generative in the same sense as the earlier hierarchical models. First draw an unconstrained Gaussian vector U_i in log-ratio space, then map it to a composition π_i, then draw counts conditional on the library size N_i. Compared with the Dirichlet, the logistic-normal can

represent richer covariance structure among log-ratios. The price is that interpretation depends on the chosen log-ratio coordinates and zeros require special care before taking logarithms. Here we only need the basic model; Chapter 03 will return to log-ratio transformations, pseudocounts, and the distinction between count data and continuous compositional data. For a compositional-data overview, see the course notes by Egozcue and Tolosana-Delgado (Egozcue and Tolosana-Delgado 2007).

2.9 The Overdispersion Pattern

The two main overdispersion constructions in this chapter have the same mathematical logic. A simple sampling distribution is made more flexible by placing a distribution on one of its parameters.

Simple model	Latent parameter model	Marginal model	What becomes random?
Poisson	Gamma-Poisson	negative binomial	count rate Λ
multinomial	Dirichlet-multinomial	Dirichlet-multinomial	composition π_i
Bernoulli/binomial	beta-Bernoulli/beta-binomial	beta-binomial	success probability π

The generative principle can be written abstractly as

$$\theta_i \sim \mathcal{G}, \qquad Y_i \mid \theta_i \sim P_{\theta_i}. \tag{2.52}$$

The conditional distribution P_{θ_i} describes sampling variation after the latent parameter is known. The mixing distribution $\mathcal{G}$ describes variation in the latent parameter across observations. Marginally, the data are more variable than under a model with fixed θ.

This idea is one of the most important pieces of statistical intuition in the course. Overdispersion is not merely a nuisance parameter. It is often the signature that biological observations differ in ways the simplest sampling model does not represent.

2.10 Mixtures And Zero Inflation

Mixture models generalize the same idea. Instead of assuming that every observation comes from one homogeneous distribution, a mixture model assumes that each observation first chooses a latent component and then draws a value from that component. For a finite mixture with K components:

$$L_i \sim \text{Categorical}(\omega_1, \dots, \omega_K), \qquad Y_i \mid L_i = k \sim F_k. \tag{2.53}$$

The marginal density or mass function is

$$f(y) = \sum_{k=1}^{K} \omega_k f_k(y), \qquad \sum_{k=1}^{K} \omega_k = 1. \tag{2.54}$$

The parameters $\omega_1, \dots, \omega_K$ are mixture weights. Each $\omega_k \geq 0$, and the weights sum to one. The component distribution F_k has density or mass function f_k, and may have its own parameters, such as a component-specific mean or covariance matrix. Mixtures can describe heterogeneous populations, latent classes, heavy tails, and distributions with unusual shapes. A zero-inflated model is a particularly common count-data mixture:

$$Y_i \sim \begin{cases} 0, & \text{with probability } \omega, \\ F, & \text{with probability } 1 - \omega, \end{cases} \tag{2.55}$$

where F may be a Poisson, negative binomial, or another count distribution. If F has probability mass function f, the zero-inflated probability mass function is

$$\Pr(Y_i = y) = \omega \, \mathbb{1}\{y = 0\} + (1 - \omega) f(y), \qquad 0 \leq \omega \leq 1. \tag{2.56}$$

The parameter ω is the probability of the extra-zero component. Such models are useful when zeros arise from more than one mechanism, for example true absence and limited detection. In biological applications, however, zero inflation should be treated as a modeling hypothesis, not as an automatic conclusion.

2.11 Likelihood And Model Fitting

A generative model becomes a statistical model once its unknown parameters are estimated from data. Suppose $Y_1, \dots, Y_n$ are independent observations from a distribution with parameter θ. The likelihood is

$$L(\theta; y_1, \dots, y_n) = \prod_{i=1}^{n} f(y_i \mid \theta). \tag{2.57}$$

The log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{n} \log f(y_i \mid \theta). \tag{2.58}$$

Maximum likelihood estimation chooses the parameter value that makes the observed data most probable under the model:

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta} \ell(\theta). \tag{2.59}$$

Simple discrete-data examples show why this is a natural estimation principle (see MSMB Generative Models and MSMB Statistical Modeling). In Exercise Sheet 01, the same principle appears in maximum-likelihood calculations for nucleotide and count models and in the latent-component updates of mixture models (Exercise Sheet 01 2026). The key point for this course is that likelihood always depends on the chosen distribution. Estimating a Poisson rate, a negative binomial mean and dispersion, or a Dirichlet-multinomial concentration parameter are different statistical problems because they assume different generative mechanisms.

2.12 Model Checking

A distributional assumption should be checked against the data. The first check is support: does the model generate values of the right type? The second check is scale: are the model-implied means and variances plausible? The third check is shape: does the model capture skewness, tails, zeros, and heterogeneity?

For counts, a simple diagnostic is the relation between empirical means and variances across features. If the empirical variance is systematically larger than the empirical mean, a Poisson model is too restrictive. For compositions, the multinomial can be checked by asking whether replicate count vectors vary more than expected under a fixed composition. For continuous data, histograms, density estimates, Q-Q plots, and residual plots help assess symmetry, tail behavior, and transformations.

Model checking is not separate from generative thinking. A generative model makes predictions about what repeated data sets should look like. If simulated data from the fitted model do not resemble the observed data in the aspects that matter, then the model is not adequate for the analysis goal.

Summary

Chapter 02 provides the probability language used throughout the rest of the course. Continuous data suggest normal, gamma, Laplace, or mixture models depending on support and shape. Binary and categorical data suggest Bernoulli, binomial, categorical, and multinomial models. Sequencing counts often begin with Poisson reasoning but usually require negative binomial or related overdispersed models. Discrete compositions begin with the multinomial model but

often require Dirichlet-multinomial or logistic-normal extensions. The logistic-normal model is the bridge from this chapter to log-ratio transformations in Chapter 03.

The most important conceptual move is generative. We do not only ask which formula fits a histogram. We ask what data-generating mechanism the formula represents. A Poisson model assumes a fixed rate. A negative binomial model can be understood as a Poisson model with a Gamma-distributed latent rate. A multinomial model assumes a fixed composition. A Dirichlet-multinomial model assumes that the composition itself varies across observations. These hierarchical constructions explain overdispersion and prepare the ground for normalization, variance stabilization, testing, and regression.

References

Aitchison, J., and S. M. Shen. 1980. "Logistic-Normal Distributions: Some Properties and Uses." *Biometrika* 67 (2): 261. https://doi.org/10.2307/2335470.

Egozcue, Juan José, and Raimon Tolosana-Delgado. 2007. *Lecture Notes on Compositional Data Analysis*. Universitat de Girona.

Exercise Sheet 01. 2026. *Exercise Sheet 01 Solutions: Generative Models, Nucleotide Frequencies, Maximum Likelihood, and Mixture Models*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise01_Solution.html.

Harrison, Joshua G., W. John Calder, Vivaswat Shastry, and C. Alex Buerkle. 2020. "Dirichlet-Multinomial Modelling Outperforms Alternatives for Analysis of Microbiome and Other Ecological Count Data." *Molecular Ecology Resources* 20 (2): 481–97. https://doi.org/10.1111/1755-0998.13128.

Townes, F. William. 2020. *Review of Probability Distributions for Modeling Count Data*. https://arxiv.org/abs/2001.04343.

3 Variance Stabilization and Normalization

Chapter Aim

Chapter 02 treated distributions as generative models. Chapter 03 asks what these models imply for preprocessing. Raw high-throughput count data are not only noisy; their noise is structured. Library sizes differ across observations, rows of sequencing data often contain relative rather than absolute information, and the variance of a feature often changes with its mean. Normalization and transformation are therefore part of the statistical model, because they put observations and features on a scale where distances, PCA, clustering, heatmaps, and regression behave more meaningfully. The purpose of this chapter is to make the link between measured counts and transformed analysis scales explicit.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

> **Main external reference**
>
> The main book reference is MSMB VST Delta Method, where variance-stabilizing transformations are derived from the delta method. For RNA-seq size factors, count models, and DESeq2 transformations, see MSMB High-Throughput Count Data, MSMB Count Data Normalization, and MSMB RNA-seq VST. For the broader experimental-design view of mean-variance relationships, see MSMB Mean-Variance Relationships.

The empirical consequences of these choices are worked out in Exercise Sheet 02, where raw count matrices are compared with median-of-ratios normalization, variance-stabilizing transformations, PCA, and single-cell normalization strategies (Exercise Sheet 02 2026). The compositional continuation appears in Exercise Sheet 03 through 16S normalization, log-ratio geometry, and ordination (Exercise Sheet 03 2026).

Learning Goals

After this chapter, students should be able to:

1. Explain why raw sequencing counts cannot usually be compared directly across observations.
2. Distinguish normalization from variance-stabilizing transformation.
3. Interpret total-count scaling, common-scale normalization, rarefying, median-of-ratios normalization, and log-ratio transformations as different answers to different modeling assumptions.
4. Derive the variance-stabilizing transformation associated with a mean-variance relationship using the delta method.
5. Connect the Poisson, negative-binomial, multinomial, Dirichlet-multinomial, and logistic-normal models from Chapter 02 to concrete preprocessing choices.
6. Explain why transformed data may be useful for exploratory analysis even when formal testing is done with a count model.
7. Recognize that simple transformations can perform well in practice, but that their behavior should still be understood through the model they approximate.

3.1 What Normalization Is Trying To Estimate

For count data, a first model is that the observed count C_{ij} is generated from an expected count μ_{ij}. In sequencing data this expectation often factors into an observation-specific exposure and an abundance-like quantity:

$$\mu_{ij} = s_i q_{ij}. \tag{3.1}$$

Here $s_i > 0$ is a size factor or exposure for observation i, and $q_{ij} \geq 0$ is the quantity we would like to compare across observations after accounting for exposure. The size factor can represent sequencing depth, sampling efficiency, total captured molecules, or another multiplicative technical effect. The abundance q_{ij} is not necessarily an absolute molecular count in the biological system; in many sequencing assays it is still a relative or assay-specific quantity.

Normalization is the act of estimating or specifying the nuisance scale s_i and removing it from the count scale. A simple normalized count is

$$\tilde{x}_{ij} = \frac{c_{ij}}{\hat{s}_i}. \tag{3.2}$$

This equation encodes a model. It assumes that the dominant unwanted difference between observations is multiplicative and row-specific. It also assumes that the same $\hat{s}_i$ is appropriate

for all features in row i. In RNA-seq count models, size factors are used to model sampling-depth bias before dispersion estimation and testing (see MSMB Count Data Normalization).

Normalization and transformation should be kept conceptually separate. Normalization estimates and removes nuisance scale. Transformation changes the measurement scale, often to reduce heteroscedasticity or make distances more interpretable. In practice they are often chained:

$$x_{ij} = g\left(\frac{c_{ij}}{\hat{s}_i}\right), \tag{3.3}$$

where g might be a logarithm, a square-root transform, a variance-stabilizing transform, a residual transform, or a log-ratio transform.

3.2 Library Size And Total-Count Scaling

The simplest size factor is proportional to the row total

$$N_i = \sum_{j=1}^{p} C_{ij}. \tag{3.4}$$

If N_i is interpreted as sequencing depth, then total-count scaling uses

$$\hat{s}_i = \frac{N_i}{L}, \tag{3.5}$$

where $L > 0$ is a reference depth, such as the mean library size, the median library size, 10^4, or 10^6. The resulting scaled count is

$$\tilde{x}_{ij} = L\frac{c_{ij}}{N_i}. \tag{3.6}$$

If $L = 10^6$, this is counts per million. For single-cell workflows one often sees $L = 10^4$. The constant L does not change relative differences within an observation, but it changes the numerical scale on which pseudocounts and logarithms operate. This is why the scaling constant is not completely harmless when the next step is $\log(\tilde{x}_{ij} + \kappa)$.

Total-count scaling is appropriate when library size is the main nuisance effect and the row total does not itself carry biological signal. It can be misleading under strong compositional effects. If a small number of features dominate one observation, then all other scaled features are pushed downward even if their absolute molecule counts did not change. This is the same compositional constraint that appeared in Chapter 02 in the multinomial model.

3.3 Common-Scale Counts And Rarefying

In microbiome and amplicon sequencing, one tempting strategy is to force all observations to have the same library size. Let

$$N_{\min} = \min_{i=1,\dots,n} N_i. \tag{3.7}$$

A deterministic common-scale version of the counts would be

$$\tilde{x}_{ij} = N_{\min} \frac{c_{ij}}{N_i}. \tag{3.8}$$

Rarefying instead creates a random subsample of size $N_{\min}$ from each row. Conditional on the observed count vector $c_i = (c_{i1}, \dots, c_{ip})^\top$ and its total N_i, this can be written as

$$C_i^{\text{rare}} \mid c_i \sim \text{Multinomial}\left(N_{\min}, \frac{c_i}{N_i}\right). \tag{3.9}$$

This equation makes the problem visible. Rarefying adds another multinomial sampling step and discards observed counts from deeper samples. It may make library sizes equal, but it does so by injecting avoidable randomness. McMurdie and Holmes' "Waste Not, Want Not" argument is the main warning against treating rarefying as a default normalization method (McMurdie and Holmes 2014). Weiss and colleagues make the complementary point that normalization and differential-abundance strategies depend on the data characteristics, especially sparsity, library-size variation, and compositionality (Weiss et al. 2017).

3.4 Median-Of-Ratios Normalization

Bulk RNA-seq workflows often use a more robust size-factor estimate than the library total. DESeq2's median-of-ratios normalization constructs a pseudo-reference feature profile by taking the geometric mean of each feature across observations. For features that are positive in all observations, the geometric mean is

$$g_j = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \log c_{ij}\right) = \left(\prod_{i=1}^{n} c_{ij}\right)^{1/n}. \tag{3.10}$$

The logarithms in Equation 3.10 are not decoration. They make clear why features with zeros need special handling: if any $c_{ij} = 0$, then $\log c_{ij}$ is undefined and the product form gives $g_j = 0$. In the basic median-of-ratios construction, only features with $g_j > 0$ contribute to the ratio calculation. For those features, each observation is compared with the pseudo-reference:

$$r_{ij} = \frac{c_{ij}}{g_j}. \tag{3.11}$$

The size factor for observation i is then the median of these ratios:

$$\hat{s}_i = \text{median}_{j:g_j>0}\, r_{ij}. \tag{3.12}$$

Equivalently, this can be written on the log scale as

$$\log \hat{s}_i = \text{median}_{j:g_j>0} \left(\log c_{ij} - \frac{1}{n} \sum_{\ell=1}^{n} \log c_{\ell j} \right). \tag{3.13}$$

The key assumption is that most features are not systematically changing, or that changes are roughly balanced between up- and down-regulation. Under that assumption, the median ratio estimates the sample-specific multiplicative effect without being dominated by a small number of highly expressed or differentially expressed features. Exercise Sheet 02 works through this idea in the pasilla data, comparing raw counts, median-of-ratios normalization, heatmaps, VST-transformed data, and PCA (Exercise Sheet 02 2026).

In the notation of Chapter 02, a feature-wise negative-binomial model can be written as

$$C_{ij} \sim \text{NB}(\mu_{ij}, \phi_j), \qquad \mu_{ij} = s_i q_{ij}, \qquad \text{Var}(C_{ij}) = \mu_{ij} + \phi_j \mu_{ij}^2. \tag{3.14}$$

This model makes clear why size-factor estimation and dispersion estimation belong together. The size factor affects the mean, and the mean affects the variance. Normalization is therefore not independent of modeling; it is one part of the same mean-variance story.

3.5 Mean-Variance Relationships

A central reason for transforming high-throughput data is heteroscedasticity: features with larger means often have larger variances. Suppose that, after accounting for exposure, a random count Y has mean μ and variance

$$\mathbb{E}(Y) = \mu, \qquad \text{Var}(Y) = v(\mu). \tag{3.15}$$

Different models imply different variance functions. For a Poisson model,

$$v(\mu) = \mu. \tag{3.16}$$

For the mean-dispersion negative-binomial model used in Chapter 02,

$$v(\mu) = \mu + \phi\mu^2. \tag{3.17}$$

For a measurement with approximately constant coefficient of variation, one has

$$v(\mu) = a\mu^2, \qquad a > 0. \tag{3.18}$$

These equations are not just descriptive. They tell us what a useful transformation should look like. If the variance grows linearly with the mean, a square-root-like transformation is natural. If the variance grows quadratically with the mean, a log-like transformation is natural. If both terms are present, as in the negative-binomial model, a good transformation behaves like a square root for small counts and like a logarithm for large counts. This point is developed through the delta method in the main reference (see MSMB VST Delta Method and MSMB Mean-Variance Relationships).

3.6 The Delta Method

The delta method is the mathematical bridge between a model's variance function and a variance-stabilizing transformation. Let Y be a random variable with mean μ and variance $v(\mu)$, and let g be a differentiable transformation. A first-order Taylor expansion around μ gives

$$g(Y) \approx g(\mu) + g'(\mu)(Y - \mu). \tag{3.19}$$

Taking variances in this approximation yields

$$\operatorname{Var}(g(Y)) \approx [g'(\mu)]^2 \operatorname{Var}(Y) = [g'(\mu)]^2 v(\mu). \tag{3.20}$$

To make the transformed variance approximately constant, choose g so that

$$[g'(\mu)]^2 v(\mu) = c^2, \tag{3.21}$$

where $c > 0$ is an arbitrary constant. Equivalently,

$$g'(\mu) = \frac{c}{\sqrt{v(\mu)}}. \tag{3.22}$$

Thus a variance-stabilizing transformation is obtained by integration:

$$g(\mu) = \int^{\mu} \frac{c}{\sqrt{v(t)}}\, dt. \tag{3.23}$$

The arbitrary multiplicative and additive constants do not change the variance-stabilizing property. They only change the scale and origin of the transformed data. This is why different software implementations can use slightly different constants while representing the same underlying idea.

3.7 Important Variance-Stabilizing Transformations

For a Poisson model, $v(\mu) = \mu$. Taking $c = 1$ gives

$$g(\mu) = \int^{\mu} \frac{1}{\sqrt{t}}\, dt = 2\sqrt{\mu}. \tag{3.24}$$

Up to scale, the square-root transform is therefore the Poisson variance-stabilizing transform. Anscombe's classical correction refines this for small counts:

$$g(y) = 2\sqrt{y + \frac{3}{8}}. \tag{3.25}$$

For a quadratic variance function $v(\mu) = a\mu^2$, one obtains

$$g(\mu) = \int^{\mu} \frac{1}{\sqrt{a}t}\, dt = \frac{1}{\sqrt{a}} \log(\mu). \tag{3.26}$$

This explains why logarithms are so common: they stabilize variance when the standard deviation is proportional to the mean. In real count data, however, zeros make $\log(y)$ undefined and small counts make the log extremely steep. For a normalized count $\tilde{x}_{ij} = c_{ij}/\hat{s}_i$, the shifted logarithm is

$$x_{ij} = \log\left(\tilde{x}_{ij} + \kappa\right) = \log\left(\frac{c_{ij}}{\hat{s}_i} + \kappa\right), \qquad \kappa > 0, \tag{3.27}$$

and is therefore a pragmatic transformation after size-factor normalization. The pseudocount κ is not just a numerical trick. It determines how strongly low counts are compressed or separated.

For the negative-binomial variance function $v(\mu) = \mu + \phi\mu^2$, the delta-method integral gives

$$g_\phi(\mu) = \int^{\mu} \frac{1}{\sqrt{t + \phi t^2}}\, dt. \tag{3.28}$$

One convenient scaled form is

$$g_\phi(y) = \frac{1}{\sqrt{\phi}} \operatorname{acosh}(2\phi y + 1), \qquad \phi > 0. \tag{3.29}$$

The same transformation can be written with the inverse hyperbolic sine as

$$g_\phi(y) = \frac{2}{\sqrt{\phi}} \operatorname{asinh}\left(\sqrt{\phi y}\right), \qquad \phi > 0. \tag{3.30}$$

Equivalently,

$$g_\phi(y) = \frac{1}{\sqrt{\phi}} \log\left(2\phi y + 1 + \sqrt{(2\phi y + 1)^2 - 1}\right). \tag{3.31}$$

The identity $\operatorname{acosh}(2u + 1) = 2\operatorname{asinh}(\sqrt{u})$ shows that Equations 3.29 and 3.30 are the same. The logarithm is present because

$$\operatorname{acosh}(z) = \log\left(z + \sqrt{z^2 - 1}\right), \qquad z \geq 1. \tag{3.32}$$

The important qualitative behavior is more important than the exact constant: for small y, the transform is square-root-like, and for large y with $\phi > 0$, it is log-like. Guan gives a more detailed theoretical treatment of variance-stabilizing transformations for Poisson, binomial, and negative-binomial distributions (Yu 2009). The same negative-binomial/gamma-Poisson idea is derived with the delta method in MSMB VST Delta Method.

3.8 DESeq2 VST, rlog, And Model-Based Transformations

DESeq2 uses the negative-binomial model to estimate size factors and dispersions, and it provides transformations for exploratory analysis (Love et al. 2014). In the notation of this script, the conceptual model is

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \mu_{ij} = s_i q_{ij}. \tag{3.33}$$

The variance-stabilizing transformation uses the estimated mean-dispersion relationship to define a transformation whose variance is approximately stable across the range of expected

counts. This is why the VST should not be understood as a generic formula applied blindly to counts. It is a transformation derived from an estimated count model.

The regularized logarithm, or rlog, has a different motivation. It fits a log-scale model with shrinkage of sample effects, producing log-like transformed values that are less sensitive to very different size factors. The shifted logarithm, VST, and rlog are contrasted for RNA-seq data in MSMB RNA-seq VST. For this course, the practical distinction is:

Transformation	Model idea	Typical use
shifted log	approximate log-scale stabilization after size-factor normalization	fast exploratory analysis
VST	delta-method transform using estimated mean-variance trend	PCA, clustering, heatmaps
rlog	regularized log-scale model with shrinkage	exploratory analysis when size factors vary strongly
Pearson residuals	standardized residuals from a count model	single-cell normalization and feature selection

The table is only a guide. The right choice depends on the assay, library-size variation, sparsity, downstream method, and whether the goal is visualization, clustering, network analysis, or formal inference.

3.9 Single-Cell RNA-seq: Simple Logs And Residuals

Single-cell RNA-seq makes the normalization problem sharper. The count matrix is sparse, library sizes vary strongly across cells, and many downstream methods operate on Euclidean distances in a low-dimensional embedding. A common preprocessing step is

$$x_{ij} = \log\left(\frac{c_{ij}}{\widehat{s}_i} + 1\right). \tag{3.34}$$

Transformation choices for single-cell RNA-seq have been compared across delta-method transformations, model residuals, inferred latent expression states, and factor-analysis approaches (Ahlmann-Eltze and Huber 2023). In these benchmarks, a simple shifted logarithm followed by PCA performed as well as or better than many more elaborate alternatives. This does not mean that modeling is unnecessary. It means that preprocessing should be judged both by theoretical behavior and by downstream performance.

SCTransform takes a residual-based route. It fits regularized negative-binomial regression models with sequencing depth as a covariate and uses Pearson residuals as transformed values (Hafemeister and Satija 2019). In simplified notation, suppose

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \log \mu_{ij} = \beta_{0j} + \beta_{1j} \log N_i. \tag{3.35}$$

The Pearson residual is

$$r_{ij} = \frac{c_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\mu}_{ij} + \hat{\phi}_j \hat{\mu}_{ij}^2}}. \tag{3.36}$$

This is not a simple monotone transformation of each count. It subtracts a fitted mean and divides by a fitted standard deviation. The connection to variance stabilization is direct: both approaches use a mean-variance model to put features on a more comparable scale. The difference is that VST transforms the observed count scale, while Pearson residuals measure deviations from a fitted model.

3.10 Compositional Normalization And Log-Ratios

For microbiome and amplicon data, a row often carries primarily relative information. The multinomial model from Chapter 02 wrote

$$C_i \mid N_i, \pi_i \sim \mathrm{Multinomial}(N_i, \pi_i), \qquad \pi_i \in \Delta^{p-1}. \tag{3.37}$$

Total-count scaling estimates π_i by

$$\hat{\pi}_{ij} = \frac{c_{ij}}{N_i}. \tag{3.38}$$

But ordinary Euclidean distances between relative abundance vectors ignore the geometry of the simplex. In addition, Equation 3.38 can contain exact zeros, and log-ratio transformations require positive entries. A simple pseudocount-and-closure estimate is

$$\hat{\pi}_{ij}^{(\kappa)} = \frac{c_{ij} + \kappa}{N_i + p\kappa}, \qquad \kappa > 0. \tag{3.39}$$

The choice of κ affects low-count features and should be reported. As introduced in Chapter 02, the centered log-ratio transform of a positive composition is

$$\operatorname{clr}(\pi_i)_j = \log \frac{\pi_{ij}}{g(\pi_i)}, \qquad g(\pi_i) = \left(\prod_{j=1}^{p} \pi_{ij} \right)^{1/p}. \tag{3.40}$$

The CLR transform returns a p-dimensional vector whose components sum to zero. It is symmetric in the components and useful for defining compositional geometry, but the transformed coordinates are constrained rather than free coordinates in $\mathbb{R}^p$.

The additive log-ratio transform chooses one component as a reference. If component r is the reference, then for $j \neq r$,

$$\operatorname{alr}_r(\pi_i)_j = \log \left(\frac{\pi_{ij}}{\pi_{ir}} \right), \qquad \operatorname{alr}_r(\pi_i) \in \mathbb{R}^{p-1}. \tag{3.41}$$

ALR coordinates are unconstrained, but they depend on the chosen reference component. This can be useful when the reference has scientific meaning or is stable, but it also means that interpretation is relative to that choice.

The isometric log-ratio transform uses an orthonormal basis of the CLR subspace. Let $H \in \mathbb{R}^{p \times (p-1)}$ satisfy

$$H^\top H = I_{p-1}, \qquad H^\top \mathbf{1} = 0. \tag{3.42}$$

Then

$$\operatorname{ilr}(\pi_i) = H^\top \operatorname{clr}(\pi_i) \in \mathbb{R}^{p-1}. \tag{3.43}$$

ILR coordinates are orthonormal coordinates for the simplex geometry, but their individual coordinates depend on the chosen basis H. In practice, this basis may be chosen through balances, trees, or other scientifically meaningful contrasts. Distances induced by these log-ratio transformations are treated in Chapter 04, where distances and embeddings are the central topic.

Exact zeros require special handling before applying logarithms. This is not a minor technicality, because sparse microbiome and single-cell matrices can contain many zeros. Exercise Sheet 03 develops this direction by using normalization, log-ratio transformations, and MDS for 16S sequencing data (Exercise Sheet 03 2026). The main conceptual point is that compositional normalization is not simply another size-factor method. It changes the coordinate system of the data analysis problem.

3.11 How Transformation Choices Affect Downstream Analysis

Many downstream algorithms implicitly assume that Euclidean geometry is meaningful. PCA, k-means, hierarchical clustering with Euclidean distance, and many regression methods are sensitive to scale. If raw counts are used, high-count features and high-depth observations can dominate the analysis. If log-transformed counts are used, low counts and pseudocount choices can matter strongly. If VST or residuals are used, the transformed scale is tied to an estimated mean-variance model.

This is why Chapter 3 is placed before distances, embeddings, and clustering. A distance between rows of X is not just a property of the biological samples; it is a property of the preprocessing map

$$C \longmapsto X. \tag{3.44}$$

For exploratory analysis, transformed matrices are often appropriate:

$$X_{ij} = g\left(\frac{C_{ij}}{\hat{s}_i}\right). \tag{3.45}$$

For differential expression or differential abundance testing, it is often better to keep the count likelihood and include size factors in the model:

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \log \mu_{ij} = \log s_i + z_i^\top \beta_j. \tag{3.46}$$

Here $z_i^\top$ is row i of the design matrix $Z \in \mathbb{R}^{n \times r}$, and $\log s_i$ is an offset. This equation is the bridge to the testing and regression chapters: normalization can enter the model as an offset rather than as a preliminary division.

3.12 Practical Summary

Normalization answers the question: which part of the observed scale is nuisance exposure? Variance stabilization answers the question: which transformation makes the noise approximately comparable across the mean range? Compositional log-ratio analysis answers a different question: which geometry is appropriate when only relative information is observed?

The main take-home message is that modeling and transformations are tied together. The Poisson model leads to a square-root transform. A quadratic mean-variance relationship leads to a logarithm. The negative-binomial model leads to a square-root-to-log VST. A fitted negative-binomial regression model leads naturally to Pearson residuals. A multinomial or

logistic-normal view of compositions leads to log-ratio transformations. These are not unrelated tricks; they are different faces of the same generative thinking from Chapter 02.

References

Ahlmann-Eltze, Constantin, and Wolfgang Huber. 2023. "Comparison of Transformations for Single-Cell RNA-seq Data." *Nature Methods*, ahead of print. https://doi.org/10.1038/s41592-023-01814-1.

Exercise Sheet 02. 2026. *Exercise Sheet 02 Solutions: Normalization and Variance-Stabilizing Transformations of Sequencing Data*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise02_Solution.html.

Exercise Sheet 03. 2026. *Exercise Sheet 03 Solutions: Normalization and Low Dimensional Representation of 16S Sequencing*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise03_Solution.html.

Hafemeister, Christoph, and Rahul Satija. 2019. "Normalization and Variance Stabilization of Single-Cell RNA-seq Data Using Regularized Negative Binomial Regression." *Genome Biology* 20 (296). https://doi.org/10.1186/s13059-019-1874-1.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12). https://doi.org/10.1186/s13059-014-0550-8.

McMurdie, Paul J., and Susan Holmes. 2014. "Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible." *PLoS Computational Biology* 10 (4): e1003531. https://doi.org/10.1371/journal.pcbi.1003531.

Weiss, Sophie, Zhenjiang Zech Xu, Shyamal Peddada, et al. 2017. "Normalization and Microbial Differential Abundance Strategies Depend Upon Data Characteristics." *Microbiome* 5 (1). https://doi.org/10.1186/s40168-017-0237-y.

Yu, Guan. 2009. "Variance Stabilizing Transformations of Poisson, Binomial and Negative Binomial Distributions." *Statistics &Amp; Probability Letters* 79 (14): 1621–29. https://doi.org/10.1016/j.spl.2009.04.010.

4 Distances, Dimensionality Reduction, and (Non-)Linear Embeddings

Chapter Aim

Chapters 02 and 03 built the statistical foundation for preprocessing high-dimensional biological data: generative assumptions motivate normalization, variance stabilization, residuals, and log-ratio transformations. Chapter 04 takes the next step: after preprocessing, observations are treated as points in a high-dimensional space, and we ask whether their geometry reveals structure.

The term geometry is used deliberately. A PCA plot, an MDS plot, a t-SNE embedding, or a UMAP embedding is not a neutral picture of the data. It is the result of a sequence of modeling and preprocessing choices: a representation is chosen, distances or similarities are computed, and a lower-dimensional representation is optimized. The same biological samples can therefore look different under different transformations and distances. This variability is a statistical property of the analysis pipeline.

In this chapter, distances, PCA scores, and embeddings are computed from a chosen analysis representation, not directly from raw counts unless explicitly stated. Row i is written as $x_i^\top \in \mathbb{R}^{1\times p}$, or equivalently $x_i \in \mathbb{R}^p$ when it is used as a column vector in formulas.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

> **Main external reference**
>
> The main book references are MSMB Multivariate Analysis, for matrices, SVD, PCA, biplots, and weighted PCA, and MSMB Heterogeneous Data, for ordination, MDS, batch effects, and nonlinear local methods. For visual interpretation, see MSMB Data Visualization. The exercises most closely linked to this chapter are Exercise Sheet 03, which works through Aitchison distance, MDS, Bray-Curtis dissimilarity, and t-SNE/UMAP examples, and Exercise Sheet 04, which connects PCA, neighborhood graphs, UMAP,

t-SNE, and clustering workflows (Exercise Sheet 03 2026; Exercise Sheet 04 2026).

Learning Goals

After this chapter, students should be able to explain why distance choices encode statistical assumptions, distinguish a metric distance from a more general dissimilarity, choose distances appropriate for transformed continuous data, count data, binary data, and compositions, derive PCA from centering and the singular value decomposition, compute classical MDS from a distance matrix, interpret non-metric MDS as rank-preserving ordination, explain ISOMAP as graph-geodesic MDS, and describe what t-SNE and UMAP optimize without treating their plots as direct evidence of biological clusters.

4.1 Geometry Starts With Preprocessing

A common exploratory workflow is: compute distances between observations, produce a low-dimensional embedding, and inspect whether the resulting display suggests groups, gradients, outliers, or batch effects. The mathematical version is the map

$$C \longrightarrow X \longrightarrow D \longrightarrow H. \tag{4.1}$$

Here C is the raw count matrix, X is the normalized and transformed analysis matrix, $D \in \mathbb{R}^{n \times n}$ is a matrix of pairwise dissimilarities between observations, and $H \in \mathbb{R}^{n \times k}$ is a low-dimensional representation, usually with $k = 2$ or $k = 3$ for visualization. The row $h_i^\top$ contains the embedded coordinates of observation i.

This sequence is useful because it separates two different questions. Chapter 03 asked how the data should be transformed so that noise and scale are handled responsibly. Chapter 04 asks how similarity should be measured once the observations are on that scale. For RNA-seq counts, a log-like or variance-stabilized X often makes Euclidean geometry more meaningful. For microbiome compositions, a CLR-transformed X gives Euclidean distances the interpretation of Aitchison distances. For binary mutation profiles, a distance that treats shared zeros carefully may be more appropriate than Euclidean distance. Thus, the distance is not chosen after the statistics are finished; it is part of the statistical modeling chain.

4.2 Distances And Dissimilarities

A distance between two observations is a function $d(x_i, x_{i'})$ that assigns a nonnegative number to a pair of rows. Collecting all pairwise distances gives

$$D = (d_{ii'})_{i,i'=1}^{n}, \qquad d_{ii'} = d(x_i, x_{i'}). \tag{4.2}$$

If the columns rather than the rows are compared, we write the feature distance matrix as $D_f \in \mathbb{R}^{p \times p}$. This distinction matters: distances between samples answer a different question than distances between genes, taxa, proteins, or peaks.

Mathematically, a metric distance satisfies four properties:

$$\begin{aligned} &d(x_i, x_{i'}) \geq 0, \\ &d(x_i, x_{i'}) = 0 \iff x_i = x_{i'}, \\ &d(x_i, x_{i'}) = d(x_{i'}, x_i), \\ &d(x_i, x_{i''}) \leq d(x_i, x_{i'}) + d(x_{i'}, x_{i''}). \end{aligned} \tag{4.3}$$

These are nonnegativity, identity of indiscernibles, symmetry, and the triangle inequality. Many useful biological dissimilarities do not satisfy all four properties. We still use them, but we should call them dissimilarities rather than distances when the distinction matters. This becomes important for methods such as classical MDS, because non-Euclidean dissimilarities can produce negative eigenvalues and distorted low-dimensional coordinates.

For transformed continuous data, the default distance is often Euclidean:

$$d_{\mathrm{E}}(x_i, x_{i'}) = \|x_i - x_{i'}\|_2 = \left\{ \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}^{1/2}. \tag{4.4}$$

Euclidean distance is natural when differences along the feature axes are meaningful and comparable. It is also the geometry behind ordinary PCA. But if one gene has much larger variance than all others, Euclidean distance will be dominated by that gene. A common response is to center and scale columns before computing distances. If $\bar{x}_{\cdot j}$ and $\hat{\sigma}_j$ are the empirical mean and standard deviation of feature j, the standardized value is

$$x_{ij}^{\mathrm{std}} = \frac{x_{ij} - \bar{x}_{\cdot j}}{\hat{\sigma}_j}. \tag{4.5}$$

Euclidean distance on standardized features is equivalent to a weighted Euclidean distance on centered features:

$$d_{\mathrm{WE}}(x_i, x_{i'}) = \left\{(x_i - x_{i'})^\top \Omega (x_i - x_{i'})\right\}^{1/2}, \qquad \Omega = \mathrm{diag}\left(\hat{\sigma}_1^{-2}, \ldots, \hat{\sigma}_p^{-2}\right). \tag{4.6}$$

Here Ω is a feature-weight matrix. Standardization can be helpful when features are measured on comparable biological scales but have different variances. It can be harmful when high-variance features are precisely the features that carry the biological signal. This is why preprocessing and distance choice should be stated explicitly in every analysis.

Correlation and cosine dissimilarities emphasize direction rather than absolute magnitude. The row-centered correlation dissimilarity is

$$d_{\mathrm{cor}}(x_i, x_{i'}) = 1 - \frac{\sum_{j=1}^p (x_{ij} - \bar{x}_{i\cdot})(x_{i'j} - \bar{x}_{i'\cdot})}{\left\{\sum_{j=1}^p (x_{ij} - \bar{x}_{i\cdot})^2\right\}^{1/2} \left\{\sum_{j=1}^p (x_{i'j} - \bar{x}_{i'\cdot})^2\right\}^{1/2}}, \tag{4.7}$$

where $\bar{x}_{i\cdot} = p^{-1} \sum_{j=1}^p x_{ij}$. The cosine dissimilarity is

$$d_{\cos}(x_i, x_{i'}) = 1 - \frac{x_i^\top x_{i'}}{\|x_i\|_2 \|x_{i'}\|_2}. \tag{4.8}$$

Cosine dissimilarity is common when the angle between expression profiles matters more than the magnitude. In single-cell workflows it is sometimes used after normalization and PCA because residual library-size or total-expression effects can otherwise influence Euclidean neighborhoods. Correlation and cosine dissimilarities are useful, but they are not interchangeable with Euclidean distance: they change the question from "how far apart are these observations?" to "how similar are their profiles up to centering or scaling?"

For nonnegative abundance data, especially ecological and microbiome data, Bray-Curtis dissimilarity is widely used. For $x_{ij} \geq 0$,

$$d_{\mathrm{BC}}(x_i, x_{i'}) = \frac{\sum_{j=1}^p |x_{ij} - x_{i'j}|}{\sum_{j=1}^p (x_{ij} + x_{i'j})}. \tag{4.9}$$

Equivalently, because $|a - b| = a + b - 2\min(a, b)$ for $a, b \geq 0$,

$$d_{\mathrm{BC}}(x_i, x_{i'}) = 1 - \frac{2\sum_{j=1}^p \min(x_{ij}, x_{i'j})}{\sum_{j=1}^p x_{ij} + \sum_{j=1}^p x_{i'j}}. \tag{4.10}$$

This form makes the ecological interpretation clear: two observations are similar when they share many abundances. Bray-Curtis is a dissimilarity, not a fully metric Euclidean distance. It is often sensible for nonnegative count-like or relative-abundance data, but it does not by itself solve the compositional problem discussed in Chapters 02 and 03.

For compositions, the more principled geometry is Aitchison geometry. If $\pi_i \in \Delta^{p-1}$ is a strictly positive composition, its centered log-ratio transform is

$$\operatorname{clr}(\pi_i)_j = \log \pi_{ij} - \frac{1}{p}\sum_{\ell=1}^{p} \log \pi_{i\ell}, \qquad j = 1, \dots, p. \tag{4.11}$$

The Aitchison distance between two positive compositions is Euclidean distance after CLR transformation:

$$d_{\mathrm{A}}(\pi_i, \pi_{i'}) = \left\| \operatorname{clr}(\pi_i) - \operatorname{clr}(\pi_{i'}) \right\|_2 . \tag{4.12}$$

This is the distance used in Exercise Sheet 03 for 16S data after CLR transformation (Exercise Sheet 03 2026). It is also the natural continuation of the multinomial, Dirichlet-multinomial, and logistic-normal composition models from Chapter 02. Zeros must be handled before applying the logarithm, usually by a clearly stated pseudocount, zero-replacement method, or feature-filtering rule.

For binary data, such as presence-absence features or mutation indicators, the relevant geometry can be different again. If $x_{ij} \in \{0, 1\}$, the Hamming distance counts mismatches:

$$d_{\mathrm{Ham}}(x_i, x_{i'}) = \sum_{j=1}^{p} \mathbf{1}\{x_{ij} \neq x_{i'j}\}. \tag{4.13}$$

The Jaccard dissimilarity ignores shared absences and compares the number of mismatches among features present in at least one of the two observations:

$$d_{\mathrm{Jac}}(x_i, x_{i'}) = 1 - \frac{\sum_{j=1}^{p} \mathbf{1}\{x_{ij} = 1,\ x_{i'j} = 1\}}{\sum_{j=1}^{p} \mathbf{1}\{x_{ij} = 1 \text{ or } x_{i'j} = 1\}}. \tag{4.14}$$

This distinction is important for sparse binary biological data: if most genes are not mutated, most proteins are not detected, or most taxa are absent, shared zeros can dominate Hamming distance without carrying much biological information.

4.3 From Distances To Similarities And Graphs

Some methods work with distances, while others work with similarities, kernels, or neighborhood graphs. A common way to turn Euclidean distances into similarities is the Gaussian kernel

$$K_{ii'} = \exp\left(-\frac{d_{\mathrm{E}}(x_i, x_{i'})^2}{2\tau^2} \right), \qquad \tau > 0. \tag{4.15}$$

The parameter τ controls the scale at which observations are considered similar. A small τ emphasizes very local neighborhoods; a large τ makes the similarity more global.

Modern single-cell workflows often construct a q-nearest-neighbor graph rather than working with all pairwise distances directly. Let $\mathcal{N}_q(i)$ denote the set of the q nearest neighbors of observation i under a chosen distance. A simple directed adjacency matrix is

$$A_{ii'} = \mathbf{1}\{i' \in \mathcal{N}_q(i)\}, \qquad A \in \{0,1\}^{n \times n}. \tag{4.16}$$

The nodes of this graph are observations, usually cells or samples. This graph view is the bridge from Chapter 04 to Chapter 05 on clustering: graph-based clustering methods do not cluster the original matrix directly; they cluster a graph whose edges were determined by preprocessing, dimension reduction, and a distance choice.

4.4 PCA As Linear Dimension Reduction

Principal component analysis is the basic linear method for summarizing high-dimensional continuous data. It is most meaningful when X is already on a scale where Euclidean geometry is sensible. For count data, this usually means that X is log-transformed, variance-stabilized, residualized, or otherwise transformed as in Chapter 03.

PCA begins by centering each feature. Let

$$\bar{x} = \frac{1}{n} X^\top \mathbf{1}_n \in \mathbb{R}^p, \tag{4.17}$$

where $\mathbf{1}_n$ is the n-vector of ones. The centered matrix is

$$X_c = X - \mathbf{1}_n \bar{x}^\top \in \mathbb{R}^{n \times p}. \tag{4.18}$$

If the features should contribute on comparable variance scales, one may instead use $X_s = X_c S_x^{-1}$ with

$$S_x = \operatorname{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_p). \tag{4.19}$$

In this chapter, X_c denotes centered data and X_s denotes centered and scaled data. We write the PCA equations for X_c, but the same formulas apply to X_s if scaling is chosen.

The first principal loading is the unit vector $u \in \mathbb{R}^p$ that maximizes the variance of the projected observations:

$$p_1 = \underset{u\in\mathbb{R}^p:\ \|u\|_2=1}{\operatorname{argmax}} \|X_c u\|_2^2 = \underset{u\in\mathbb{R}^p:\ \|u\|_2=1}{\operatorname{argmax}} u^\top X_c^\top X_c u. \tag{4.20}$$

Subsequent principal loadings solve the same problem subject to orthogonality constraints. Computationally, PCA is obtained from the singular value decomposition

$$X_c = U\Sigma V^\top, \tag{4.21}$$

where $r = \operatorname{rank}(X_c)$, $U \in \mathbb{R}^{n\times r}$ has orthonormal columns, $V \in \mathbb{R}^{p\times r}$ has orthonormal columns, and $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_r)$ with $\sigma_1 \geq \cdots \geq \sigma_r > 0$.

Following the notation fixed for this lecture script, the rank-k PCA approximation is written as

$$X_c \approx T_k P_k^\top, \qquad T_k = U_k\Sigma_k \in \mathbb{R}^{n\times k}, \qquad P_k = V_k \in \mathbb{R}^{p\times k}. \tag{4.22}$$

The rows of T_k are the PCA scores or observation coordinates. The columns of P_k are the loadings or feature directions. This convention keeps observation coordinates and feature weights dimensionally distinct.

The variance explained by component k is proportional to σ_k^2. The fraction of total centered variance explained by component k is

$$\mathrm{PVE}_k = \frac{\sigma_k^2}{\sum_{\ell=1}^r \sigma_\ell^2}. \tag{4.23}$$

The cumulative fraction explained by the first k components is

$$\mathrm{CPVE}_k = \frac{\sum_{\ell=1}^k \sigma_\ell^2}{\sum_{\ell=1}^r \sigma_\ell^2}. \tag{4.24}$$

PCA is often introduced as finding directions of maximal variance, but an equivalent view is reconstruction. Among all rank-k matrices, the truncated SVD gives the best approximation to X_c in squared-error loss:

$$T_k P_k^\top = \underset{\operatorname{rank}(M)\leq k}{\operatorname{argmin}} \|X_c - M\|_F^2. \tag{4.25}$$

This is why PCA is so useful as a denoising and visualization tool. It summarizes the dominant linear variation in X_c. But it also explains the limitation: PCA is optimized for Euclidean squared-error reconstruction, not for preserving every biological distinction. If a biological

signal is nonlinear, sparse, weak relative to technical variation, or present only in rare cell types, it may not dominate the first two principal components.

For Euclidean distances, PCA and classical MDS are closely related. If one applies classical MDS to Euclidean distances computed from centered data, the resulting principal coordinates are equivalent to PCA scores up to rotation and sign changes. This equivalence is one reason that PCA, SVD, biplots, and related multivariate methods are treated together in MSMB Multivariate Analysis.

4.5 Classical MDS And PCoA

Multidimensional scaling starts from a distance or dissimilarity matrix rather than directly from the feature matrix. Given $D = (d_{ii'}) \in \mathbb{R}^{n \times n}$, the goal is to find coordinates $H \in \mathbb{R}^{n \times k}$ such that distances in the embedding approximate the original dissimilarities:

$$\|h_i - h_{i'}\|_2 \approx d_{ii'}, \qquad 1 \le i < i' \le n. \tag{4.26}$$

Classical MDS, also called principal coordinates analysis (PCoA), proceeds by converting squared distances into an estimated centered inner-product matrix. Let

$$J = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \tag{4.27}$$

be the centering matrix, and let $D^{\circ 2}$ denote elementwise squaring:

$$\left(D^{\circ 2}\right)_{ii'} = d_{ii'}^2. \tag{4.28}$$

The classical MDS matrix is

$$B = -\frac{1}{2} J D^{\circ 2} J. \tag{4.29}$$

If D is a Euclidean distance matrix, then B is the centered Gram matrix of the original points. We then compute the eigendecomposition

$$B = Q \Lambda Q^\top, \tag{4.30}$$

where the eigenvalues are ordered from largest to smallest. If the first k eigenvalues are positive, the k-dimensional classical MDS coordinates are

$$H_k = Q_k \Lambda_k^{1/2} \in \mathbb{R}^{n \times k}. \tag{4.31}$$

For Euclidean distances, this reconstructs the original centered geometry in enough dimensions. For non-Euclidean dissimilarities, negative eigenvalues can occur. They are not just numerical inconvenience; they indicate that the dissimilarities cannot be represented exactly as Euclidean distances. In practice, PCoA plots often keep the leading positive axes and report how much of the positive eigenspectrum they explain.

The proportion of positive eigenspectrum represented by coordinate k is

$$\text{PCoA-PVE}_k = \frac{\lambda_k}{\sum_{\ell:\lambda_\ell>0} \lambda_\ell}, \qquad \lambda_k > 0. \tag{4.32}$$

Exercise Sheet 03 applies this workflow to Aitchison distances between microbiome samples, where the first two PCoA axes explain only a moderate part of the total variation (Exercise Sheet 03 2026). A two-dimensional display is therefore not equivalent to a complete low-dimensional description.

4.6 Non-Metric MDS

Classical MDS tries to preserve numerical dissimilarities. Non-metric MDS (NMDS) relaxes this requirement. It assumes that the ranks of the dissimilarities are more trustworthy than their absolute values. In NMDS, one seeks coordinates $H \in \mathbb{R}^{n \times k}$ and a monotone function f such that

$$\|h_i - h_{i'}\|_2 \approx f(d_{ii'}). \tag{4.33}$$

A common stress criterion is

$$\text{Stress}(H) = \left[\frac{\sum_{i<i'} \left(\|h_i - h_{i'}\|_2 - f(d_{ii'}) \right)^2}{\sum_{i<i'} \|h_i - h_{i'}\|_2^2} \right]^{1/2}. \tag{4.34}$$

The function f is constrained to be monotone, so if $d_{ii'} < d_{\ell m}$, then $f(d_{ii'}) \leq f(d_{\ell m})$. NMDS is popular in community ecology because ecological dissimilarities often have meaningful orderings even when their numerical scale is hard to interpret. The optimization is non-convex, axes are arbitrary, and repeated runs may differ. Stress values and sensitivity checks are therefore part of the analysis.

The practical distinction between PCoA and NMDS is simple. PCoA asks whether the actual dissimilarities can be represented by Euclidean coordinates. NMDS asks whether the rank order

of dissimilarities can be represented by Euclidean coordinates. Exercise Sheet 03 explicitly contrasts these two views for Aitchison distances (Exercise Sheet 03 2026).

4.7 The High-Dimensional Problem

Distances in high dimensions can become less informative. One simple way to see this is to consider independent standardized noise features. Let

$$X_{ij} \sim \mathcal{N}(0,1) \qquad \text{independently for } j = 1, \dots, p. \tag{4.35}$$

For two independent observations X_i and $X_{i'}$, the squared Euclidean distance is

$$\|X_i - X_{i'}\|_2^2 = \sum_{j=1}^{p} (X_{ij} - X_{i'j})^2. \tag{4.36}$$

Since $X_{ij} - X_{i'j} \sim \mathcal{N}(0, 2)$, its expectation is

$$\mathbb{E}\left[\|X_i - X_{i'}\|_2^2\right] = 2p. \tag{4.37}$$

As p grows, distances can concentrate around their typical value unless there is structured signal. This does not mean high-dimensional distance analysis is useless. It means that feature selection, transformation, dimensionality reduction, denoising, and biological prior knowledge often determine whether the geometry is informative. This is why single-cell workflows commonly compute variable features, scale data, run PCA, and then construct neighbor graphs in PCA space rather than in the full raw feature space.

4.8 Nonlinear Embeddings

PCA and classical MDS are global linear or Euclidean methods. Nonlinear embedding methods aim to preserve local neighborhoods when the data may lie near a curved low-dimensional structure. A standard Swiss-roll example makes the issue visible: Euclidean distances through the ambient space may not reflect distances along the underlying manifold.

The generic nonlinear embedding problem can be written as

$$H = f_\theta(X), \qquad H \in \mathbb{R}^{n \times k}, \qquad k \ll p. \tag{4.38}$$

Unlike PCA, most nonlinear embeddings do not produce a simple loading matrix that tells us how each feature contributes to each axis. The coordinates are optimized to satisfy a

neighborhood-preservation objective. Consequently, axis labels such as "UMAP1" and "t-SNE2" should not be interpreted like principal components.

4.8.1 ISOMAP

ISOMAP is one of the clearest examples of manifold learning because it modifies classical MDS in only one essential way (Tenenbaum et al. 2000). Instead of applying MDS to direct Euclidean distances, it first approximates distances along the data manifold by shortest-path distances on a neighborhood graph. The method is motivated by situations where the ambient distance $\|x_i - x_{i'}\|_2$ cuts through empty space, while the meaningful distance follows a curved surface.

Let $\mathcal{N}_q(i)$ denote the q nearest neighbors of observation i under a chosen input distance d. ISOMAP constructs a weighted graph with edge weights

$$a_{ii'} = \begin{cases} d(x_i, x_{i'}), & i' \in \mathcal{N}_q(i) \text{ or } i \in \mathcal{N}_q(i'), \\ \infty, & \text{otherwise.} \end{cases} \tag{4.39}$$

The graph-geodesic distance between two observations is then the shortest-path distance

$$d_{\text{geo}}(x_i, x_{i'}) = \min_{\gamma: i \rightsquigarrow i'} \sum_{(u,v)\in\gamma} a_{uv}, \tag{4.40}$$

where $\gamma : i \rightsquigarrow i'$ ranges over all paths in the neighborhood graph from i to i'. Classical MDS is then applied to the matrix of squared graph-geodesic distances. With $D^{(2)}_{\text{geo}} = (d_{\text{geo}}(x_i, x_{i'})^2)_{i,i'=1}^n$ and $J = I_n - n^{-1}\mathbf{1}\mathbf{1}^\top$, the ISOMAP inner-product matrix is

$$B_{\text{ISO}} = -\frac{1}{2} J D^{(2)}_{\text{geo}} J. \tag{4.41}$$

The final coordinates are obtained from the leading positive eigenvalues and eigenvectors of B_{ISO}, exactly as in classical MDS. This makes ISOMAP conceptually useful: it separates the nonlinear part, which is the graph-geodesic distance construction, from the embedding part, which is ordinary spectral decomposition.

The method is sensitive to the neighborhood size q. If q is too small, the graph may become disconnected and geodesic distances are undefined between components. If q is too large, the graph contains shortcuts and the method collapses back toward ordinary Euclidean MDS. In high-dimensional biological data, ISOMAP is therefore best read as a geometric idea and diagnostic tool rather than as a default visualization method: preprocessing, denoising, and neighborhood construction strongly determine the result.

4.8.2 t-SNE

t-distributed stochastic neighbor embedding (t-SNE) converts distances into probability distributions over pairs of observations and then finds low-dimensional coordinates whose pairwise probabilities are similar (Maaten and Hinton 2008). For each observation i, high-dimensional similarities are often defined as conditional probabilities

$$p_{i'|i} = \frac{\exp\left(-\|x_i - x_{i'}\|_2^2/(2\tau_i^2)\right)}{\sum_{\ell \neq i} \exp\left(-\|x_i - x_\ell\|_2^2/(2\tau_i^2)\right)}, \qquad p_{i|i} = 0. \tag{4.42}$$

The bandwidth τ_i is chosen separately for each observation so that the effective neighborhood size matches a user-specified perplexity. If P_i denotes the conditional distribution over neighbors of observation i, the perplexity is

$$\mathrm{Perp}(P_i) = 2^{-\sum_{i' \neq i} p_{i'|i} \log_2 p_{i'|i}}. \tag{4.43}$$

Low perplexity emphasizes very local structure; high perplexity allows broader neighborhoods to influence the embedding. Exercise Sheet 03 varies this parameter on the Moignard single-cell data and shows how the apparent cluster structure changes (Exercise Sheet 03 2026).

The conditional probabilities are symmetrized:

$$p_{ii'} = \frac{p_{i'|i} + p_{i|i'}}{2n}. \tag{4.44}$$

In the low-dimensional embedding, t-SNE uses a heavy-tailed Student-type kernel:

$$q_{ii'} = \frac{\left(1 + \|h_i - h_{i'}\|_2^2\right)^{-1}}{\sum_{\ell \neq m} \left(1 + \|h_\ell - h_m\|_2^2\right)^{-1}}, \qquad q_{ii} = 0. \tag{4.45}$$

The objective is the Kullback-Leibler divergence from P to Q:

$$\mathrm{KL}(P\|Q) = \sum_{i \neq i'} p_{ii'} \log\left(\frac{p_{ii'}}{q_{ii'}}\right). \tag{4.46}$$

The asymmetry of this objective is important. If two points are close in the high-dimensional probability distribution, t-SNE strongly penalizes placing them far apart in the embedding. If two points are far apart in high dimensions, it is less concerned with their exact low-dimensional distance. This is why t-SNE is powerful for local cluster visualization but unreliable for interpreting global distances, cluster sizes, and empty space between clusters. Initialization and optimization details also matter; Kobak and Linderman emphasize that initialization can

strongly affect apparent global structure in both t-SNE and UMAP (Kobak and Linderman 2021).

4.8.3 UMAP

Uniform manifold approximation and projection (UMAP) also begins with local neighborhoods, but it represents the high-dimensional data as a weighted graph and then optimizes a low-dimensional graph representation (McInnes et al. 2018). For each observation i, let ρ_i be the distance to its nearest neighbor and let $\tau_i > 0$ be a local scale. For $i' \in \mathcal{N}_q(i)$, a simplified form of the directed high-dimensional edge weight from i to i' is

$$a_{ii'} = \exp\left(-\frac{\max\{0, d(x_i, x_{i'}) - \rho_i\}}{\tau_i}\right). \tag{4.47}$$

The local scale τ_i is chosen so that the effective number of neighbors is controlled by the parameter q, usually called `n_neighbors` in software. Directed weights are then combined into an undirected fuzzy graph, often using the probabilistic union

$$w_{ii'}^{\mathrm{sym}} = a_{ii'} + a_{i'i} - a_{ii'}a_{i'i}. \tag{4.48}$$

In the low-dimensional embedding, UMAP assigns a similarity to each pair of coordinates. A common parametric form is

$$\tilde{w}_{ii'} = (1 + a\|h_i - h_{i'}\|_2^{2b})^{-1}, \qquad a > 0,\ b > 0. \tag{4.49}$$

The embedding minimizes a cross-entropy between high-dimensional graph weights and low-dimensional graph weights:

$$\mathcal{L}_{\mathrm{UMAP}}(H) = \sum_{i<i'} \left[w_{ii'}^{\mathrm{sym}} \log\left(\frac{w_{ii'}^{\mathrm{sym}}}{\tilde{w}_{ii'}}\right) + (1 - w_{ii'}^{\mathrm{sym}}) \log\left(\frac{1 - w_{ii'}^{\mathrm{sym}}}{1 - \tilde{w}_{ii'}}\right) \right]. \tag{4.50}$$

This formula explains why UMAP is sensitive to the distance metric, the number of neighbors, the minimum-distance parameter, initialization, and random seed. The embedding is not a direct measurement. It is the outcome of graph construction and optimization. Becht and colleagues popularized UMAP for single-cell visualization, and later work has clarified how UMAP and t-SNE fit into a broader family of attraction-repulsion neighbor embedding methods (Becht et al. 2019; Böhm et al. 2023).

Exercise Sheet 03 compares UMAP distance metrics in a Seurat workflow, and Exercise Sheet 04 uses PCA and graph-based workflows that prepare the ground for clustering (Exercise Sheet 03 2026; Exercise Sheet 04 2026). The key lesson is not that one embedding is always better.

It is that changing the metric changes the nearest-neighbor graph, and changing the graph changes the embedding.

4.9 Interpreting Embeddings

A two-dimensional embedding should be read as an exploratory summary. It can suggest structure worth modeling, but it is not itself a model-based test. Several interpretation rules are worth making explicit.

First, local neighborhoods are usually more trustworthy than global distances in t-SNE and UMAP. ISOMAP tries to preserve larger-scale graph-geodesic distances, but these distances are only meaningful if the neighborhood graph is a reasonable approximation to the underlying manifold. If two nearby cells have similar neighbors, that may indicate related transcriptional states. But the distance between two far-away islands, or the relative sizes of two islands, is not generally meaningful.

Second, visible clusters are not automatically biological populations. Batch effects, sequencing technology, library size, cell-cycle state, mitochondrial content, feature selection, and preprocessing can all create apparent separation. Exercise Sheet 03 illustrates this with microbiome MDS plots in which sequencing technology explains a major axis of variation (Exercise Sheet 03 2026).

Third, the scale of the input matters. Applying Euclidean PCA to raw counts is usually a poor default because large counts and library-size effects dominate the geometry. Applying Euclidean distance after CLR transformation has a compositional interpretation through Aitchison distance. Applying UMAP to PCA scores means the neighborhood graph is built in a denoised linear subspace, not in the original gene space.

Finally, embedding coordinates should be linked back to the original features and metadata. PCA loadings, feature correlations with coordinates, marker genes, batch labels, covariates, and external annotations connect a low-dimensional display to statistical analysis. In the multivariate workflow, dimension reduction is an intermediate representation for interrogating the data (MSMB Multivariate Analysis; MSMB Heterogeneous Data).

Summary

The central object of this chapter is the geometry induced by an analysis matrix X. Distances and dissimilarities translate rows of X into pairwise comparisons. PCA summarizes dominant linear variation. Classical MDS and PCoA represent a distance matrix in Euclidean coordinates when possible. NMDS preserves ranks rather than raw dissimilarity values. ISOMAP applies classical MDS to shortest-path distances on a neighborhood graph. t-SNE and UMAP construct

local probability or graph representations and optimize low-dimensional coordinates to preserve neighborhood structure.

Modeling, transformation, distance choice, and embedding form a single analysis chain. Chapter 02 describes plausible data-generating assumptions. Chapter 03 shows how those assumptions motivate transformations. Chapter 04 shows how transformed data become geometry. If the geometry is chosen without regard to the data type and preprocessing scale, the resulting display may be statistically misleading. If the geometry is chosen deliberately, embeddings become useful exploratory tools for forming hypotheses, diagnosing confounding, and preparing for clustering.

References

Becht, Etienne, Leland McInnes, John Healy, et al. 2019. "Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP." *Nature Biotechnology* 37 (1): 38–44. https://doi.org/10.1038/nbt.4314.

Böhm, Jan Niklas, Philipp Berens, and Dmitry Kobak. 2023. *A Unifying Perspective on Neighbor Embeddings Along the Attraction-Repulsion Spectrum*. https://arxiv.org/abs/2007.08902.

Exercise Sheet 03. 2026. *Exercise Sheet 03 Solutions: Normalization and Low Dimensional Representation of 16S Sequencing*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise03_Solution.html.

Exercise Sheet 04. 2026. *Exercise Sheet 04 Solutions: Hierarchical Clustering and Graph-Based Clustering of High-Dimensional Biological Data*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise04_Solution.html.

Kobak, Dmitry, and George C. Linderman. 2021. "Initialization Is Critical for Preserving Global Data Structure in Both t-SNE and UMAP." *Nature Biotechnology* 39 (2): 156–57. https://doi.org/10.1038/s41587-020-00809-z.

Maaten, Laurens van der, and Geoffrey Hinton. 2008. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research* 9: 2579–605.

McInnes, Leland, John Healy, and James Melville. 2018. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. https://arxiv.org/abs/1802.03426.

Tenenbaum, Joshua B., Vin de Silva, and John C. Langford. 2000. "A Global Geometric Framework for Nonlinear Dimensionality Reduction." *Science* 290 (5500): 2319–23. https://doi.org/10.1126/science.290.5500.2319.

5 Clustering

Chapter Aim

Clustering is one of the most common exploratory tools in high-dimensional biology. We use it to propose cell types, microbial community states, patient subgroups, gene modules, protein families, and perturbation-response classes. The word "propose" is important. A clustering algorithm assigns labels, but the labels are not automatically biological truth. They are hypotheses produced by a specific representation, distance, algorithm, and tuning parameter.

Chapter 04 described how normalized and transformed data become geometry. Chapter 05 asks what happens when this geometry is turned into groups. The representation may consist of log-normalized counts, variance-stabilized values, CLR-transformed compositions, residuals, PCA scores, or another deliberately chosen scale. Clustering is then applied to this representation, to a distance matrix $D \in \mathbb{R}^{n \times n}$, or to a graph $A \in \mathbb{R}^{n \times n}$ whose nodes are observations.

Notation convention

Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external reference

The main book reference is MSMB Clustering, which covers hierarchical clustering, k-means, k-medoids, density-based clustering, cluster validation, and bootstrap ideas. For the preprocessing and geometry that feed clustering, see MSMB Multivariate Analysis and MSMB Heterogeneous Data. Exercise Sheet 04 is the main practical companion for this chapter: it works through hierarchical clustering, heatmaps, HMP microbiome clustering, WSS, the gap statistic, shared-nearest-neighbor graphs, Louvain clustering, resolution parameters, UMAP, and t-SNE (Exercise Sheet 04 2026).

Learning Goals

After this chapter, students should be able to describe a clustering result as a function of preprocessing, distance, algorithm, and tuning choices; distinguish centroid-based, medoid-based, hierarchical, density-based, model-based, and graph-based clustering; write down the objective optimized by k-means and the linkage rules used in hierarchical clustering; explain WSS, silhouette, gap statistic, and stability as cluster validation tools; and interpret single-cell graph clustering as community detection on a nearest-neighbor graph rather than as direct clustering of raw counts.

5.1 What A Clustering Algorithm Returns

At the most basic level, clustering assigns each observation to one of K groups. We write the estimated cluster label for observation i as

$$\hat{\eta}_i \in \{1, \dots, K\}, \qquad i = 1, \dots, n. \tag{5.1}$$

The vector $\hat{\eta} = (\hat{\eta}_1, \dots, \hat{\eta}_n)^\top$ defines the estimated partition. The set of observations assigned to cluster k is

$$\widehat{G_k} = \{i : \hat{\eta}_i = k\}, \qquad k = 1, \dots, K. \tag{5.2}$$

This notation deliberately uses $\hat{\eta}_i$ rather than L_i. In this lecture script, L_i is reserved for latent mixture-component labels when we write down generative models. The hat on $\hat{\eta}_i$ reminds us that algorithmic cluster labels are estimated from the data and may change when preprocessing, feature selection, distance, random seed, or resolution changes.

The clustering workflow can be summarized as

$$C \longrightarrow X \longrightarrow (D \text{ or } A) \longrightarrow \hat{\eta}. \tag{5.3}$$

This is the same chain as Chapter 04 with one additional step. A clustering result is therefore not only a property of the biological system. It is a property of the full workflow.

5.2 Choosing The Object To Cluster

Clustering rows of X groups observations: cells, samples, patients, tissues, perturbations, or time points. Clustering columns of X groups features: genes, taxa, proteins, peaks, or metabolites. These are different scientific questions. Observation clustering uses an observation distance matrix

$$D = (d_{ii'})_{i,i'=1}^{n}, \qquad d_{ii'} = d(x_i, x_{i'}), \tag{5.4}$$

whereas feature clustering uses

$$D_f = (d^f_{jj'})_{j,j'=1}^{p}, \qquad d^f_{jj'} = d_f(x_{\cdot j}, x_{\cdot j'}). \tag{5.5}$$

Heatmaps often cluster both rows and columns. This is visually useful, but it combines two different clustering problems. Rows may be clustered with a sample distance, while columns may be clustered with correlation, Euclidean distance on standardized expression, or another feature-wise dissimilarity. The dendrograms should therefore be interpreted as ordering devices unless the clustering choices have been justified biologically.

For transformed continuous data, Euclidean or correlation-based distances are common. For compositional microbiome data, Aitchison distance after CLR transformation is often more principled. For binary presence-absence or mutation data, Hamming or Jaccard distances may be more appropriate. Chapter 04 gave the formulas; Chapter 05 uses them as inputs.

5.3 Centroid-Based Clustering: k-Means

The most familiar clustering method is k-means. It assumes that the observations can be represented by K centroids in Euclidean space. For a proposed partition, the centroid of cluster k is

$$\hat{\mu}_k = \frac{1}{|\widehat{G}_k|} \sum_{i \in \widehat{G}_k} x_i \in \mathbb{R}^p. \tag{5.6}$$

The within-cluster sum of squares for K clusters is

$$W_K = \sum_{k=1}^{K} \sum_{i \in \widehat{G}_k} \|x_i - \hat{\mu}_k\|_2^2. \tag{5.7}$$

k-means chooses cluster labels and centroids to minimize this quantity:

$$(\hat{\eta}, \hat{\mu}_1, \dots, \hat{\mu}_K) = \underset{\eta, \mu_1, \dots, \mu_K}{\operatorname{argmin}} \sum_{i=1}^{n} \|x_i - \mu_{\eta_i}\|_2^2. \tag{5.8}$$

Given centroids, each observation is assigned to the nearest centroid:

$$\hat{\eta}_i = \underset{k \in \{1, \dots, K\}}{\operatorname{argmin}} \|x_i - \hat{\mu}_k\|_2^2. \tag{5.9}$$

Given labels, each centroid is updated by taking the mean in Equation 5.6. Alternating these two steps decreases the objective until convergence to a local optimum.

The assumptions hidden inside k-means are strong. It uses Euclidean squared distance, represents each cluster by a mean, favors roughly spherical clusters of comparable variance, and requires K to be chosen in advance. These assumptions can be reasonable for transformed continuous data in a low-dimensional representation, such as selected PCA scores. They are usually not appropriate for raw sequencing counts, sparse compositional data, or highly elongated trajectories.

5.4 Medoids And Robust Representatives

A centroid need not be an observed data point. In some settings, especially when the distance is not Euclidean or outliers are influential, it is preferable to represent each cluster by a medoid: an actual observation whose average dissimilarity to the other observations in its cluster is small. For cluster $\widehat{G}_k$, the medoid is

$$\hat{m}_k = \underset{i \in \widehat{G}_k}{\operatorname{argmin}} \sum_{i' \in \widehat{G}_k} d_{ii'}. \tag{5.10}$$

The K-medoids objective can be written as

$$(\hat{\eta}, \hat{m}_1, \dots, \hat{m}_K) = \underset{\eta, m_1, \dots, m_K}{\operatorname{argmin}} \sum_{i=1}^{n} d_{i, m_{\eta_i}}. \tag{5.11}$$

Here m_k indexes an observed row rather than an arbitrary point in $\mathbb{R}^p$. This makes medoid methods compatible with many dissimilarities, including ecological dissimilarities, and gives each cluster a real representative observation. The k-medoids and PAM alternatives to k-means are useful when a distance matrix is the natural input (see MSMB Clustering).

5.5 Hierarchical Clustering

Hierarchical clustering does not require choosing K before fitting. Agglomerative hierarchical clustering starts with each observation as its own cluster and repeatedly merges the two closest clusters. The initial partition is

$$\mathcal{P}^{(0)} = \{\{1\}, \{2\}, \dots, \{n\}\}. \tag{5.12}$$

At step t, two clusters A and B are merged according to a linkage function $\delta(A, B)$. The resulting sequence of merges is displayed as a dendrogram. The crucial choice is how to define the distance between sets of observations.

Single linkage uses the smallest pairwise distance:

$$\delta_{\text{single}}(A, B) = \min_{i \in A,\ i' \in B} d_{ii'}. \tag{5.13}$$

Complete linkage uses the largest pairwise distance:

$$\delta_{\text{complete}}(A, B) = \max_{i \in A,\ i' \in B} d_{ii'}. \tag{5.14}$$

Average linkage uses the mean pairwise distance:

$$\delta_{\text{average}}(A, B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{i' \in B} d_{ii'}. \tag{5.15}$$

Ward's linkage is defined for Euclidean data and merges clusters that cause the smallest increase in within-cluster sum of squares. If $\bar{x}_A$ and $\bar{x}_B$ are the centroids of clusters A and B, the increase caused by merging them is

$$\Delta_{\text{Ward}}(A, B) = \frac{|A||B|}{|A| + |B|} \|\bar{x}_A - \bar{x}_B\|_2^2. \tag{5.16}$$

These linkage choices can produce visibly different dendrograms from the same distance matrix. Single linkage can create chains because one close pair is enough to merge two groups. Complete linkage favors compact clusters because all cross-cluster distances must be small. Average linkage is a compromise. Ward's method is tied to Euclidean variance and is often useful for heatmaps of transformed continuous data.

To obtain a flat clustering from a dendrogram, one cuts the tree at a height h:

$$\hat{\eta}^{(h)} = \text{cut}\left(\mathcal{T}, h\right), \tag{5.17}$$

or chooses a desired number of clusters:

$$\hat{\eta}^{(K)} = \operatorname{cut}(\mathcal{T}, K). \tag{5.18}$$

Exercise Sheet 04 begins with this exact sensitivity: the same species distance matrix gives different dendrograms under single, complete, and average linkage (Exercise Sheet 04 2026). The exercise then uses hierarchical clustering and heatmaps for microbiome data, emphasizing that the clustering result depends on preprocessing, distance, linkage, and annotation.

5.6 Heatmaps As Clustered Views

A heatmap is a visualization of a matrix after rows and columns have been ordered. Let R be a permutation matrix that orders observations, and let Q be a permutation matrix that orders features. The displayed matrix is

$$X_{\text{heatmap}} = RXQ^{\top}. \tag{5.19}$$

When rows and columns are clustered independently, R is obtained from a row dendrogram and Q from a feature dendrogram. The colors in a heatmap can reveal blocks, gradients, and outliers, but the visual impression depends strongly on scaling. A common feature-wise standardized heatmap uses

$$z_{ij} = \frac{x_{ij} - \bar{x}_{\cdot j}}{\hat{\sigma}_j}. \tag{5.20}$$

This standardization changes the scientific question. It asks which observations are high or low relative to the feature's own distribution, not which observations have large absolute abundance. For count and compositional data, the transformation from C to X should already have been handled before heatmap scaling is applied.

The classic genome-wide expression heatmap of Eisen and colleagues helped establish clustering as a central exploratory tool in genomics (Eisen et al. 1998). The same caution still applies: clustered heatmaps are excellent for seeing structure, but the dendrogram is not a confirmatory model.

5.7 Choosing The Number Of Clusters

Many clustering methods require a choice of K, a tree-cut height, or a graph resolution parameter. This is model selection in an unsupervised setting, which is inherently difficult because there is no observed response variable that defines the correct answer.

For k-means, W_K always decreases as K increases. Indeed, allowing more clusters cannot make the optimal within-cluster sum of squares worse:

$$W_{K+1} \leq W_K. \tag{5.21}$$

The elbow heuristic looks for a value of K after which the decrease in W_K becomes small. This is useful as a diagnostic, but it is not a statistical guarantee.

The gap statistic compares the observed within-cluster dispersion to what would be expected under a reference distribution with no obvious cluster structure (Tibshirani et al. 2001). If $W_K^{*(b)}$ is the within-cluster dispersion for bootstrap reference dataset $b = 1, \ldots, B$, then the estimated gap is

$$\widehat{\mathrm{Gap}}(K) = \frac{1}{B} \sum_{b=1}^{B} \log W_K^{*(b)} - \log W_K. \tag{5.22}$$

The Monte Carlo standard error is often estimated as

$$\widehat{\mathrm{SE}}(K) = \left(1 + \frac{1}{B}\right)^{1/2} \mathrm{sd}_{b=1,\ldots,B}\left(\log W_K^{*(b)}\right). \tag{5.23}$$

One common rule chooses the smallest K such that the gap at K is within one standard error of the gap at $K+1$:

$$\widehat{K} = \min\left\{K : \widehat{\mathrm{Gap}}(K) \geq \widehat{\mathrm{Gap}}(K+1) - \widehat{\mathrm{SE}}(K+1)\right\}. \tag{5.24}$$

Exercise Sheet 04 uses WSS and the gap statistic for the HMP microbiome data (Exercise Sheet 04 2026). The important lesson is that these criteria are aids to judgment. Different criteria can prefer different values of K, especially when the data contain gradients rather than discrete groups.

5.8 Silhouette And Cluster Separation

The silhouette value of an observation compares its average dissimilarity to its own cluster with its average dissimilarity to the nearest other cluster (Rousseeuw 1987). For observation i, let

$$a_i = \frac{1}{|\widehat{G}_{\hat{\eta}_i}| - 1} \sum_{\substack{i' \in \widehat{G}_{\hat{\eta}_i} \\ i' \neq i}} d_{ii'} \tag{5.25}$$

be the average dissimilarity to observations in the same cluster. For another cluster $k \neq \hat{\eta}_i$, define

$$b_i(k) = \frac{1}{|\widehat{G}_k|} \sum_{i' \in \widehat{G}_k} d_{ii'}. \tag{5.26}$$

The nearest-other-cluster dissimilarity is

$$b_i = \min_{k \neq \hat{\eta}_i} b_i(k). \tag{5.27}$$

The silhouette value is

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{5.28}$$

Values near 1 indicate that observation i is much closer to its own cluster than to the nearest other cluster. Values near 0 indicate boundary points. Negative values suggest possible misassignment. Average silhouette values can help compare clusterings, but they inherit the chosen distance and can favor compact, well-separated clusters over trajectories or nested biological structure.

5.9 Stability As Validation

Cluster validation should ask whether the result is reproducible under small perturbations of the data or workflow. Stability-based validation makes this idea explicit (Lange et al. 2004). Suppose we draw B subsamples of the observations, cluster each subsample, and compare whether pairs of observations are repeatedly assigned together. A co-clustering matrix can be written as

$$\widehat{\Pi}_{ii'} = \frac{\sum_{b=1}^{B} \mathbf{1}\{i, i' \in S_b\} \mathbf{1}\{\hat{\eta}_i^{(b)} = \hat{\eta}_{i'}^{(b)}\}}{\sum_{b=1}^{B} \mathbf{1}\{i, i' \in S_b\}}, \tag{5.29}$$

where S_b is the set of observations included in subsample b. If a cluster is stable, pairs of observations inside it should have high co-clustering probabilities across resamples. If the apparent clusters are artifacts of noise, feature selection, or initialization, the co-clustering matrix will be diffuse.

Stability can also be assessed across preprocessing choices: different feature sets, different distance metrics, different numbers of principal components, and different random seeds. In high-dimensional biological data this is often more informative than reporting a single clustering run.

5.10 Density-Based Clustering

Density-based methods define clusters as dense regions separated by lower-density regions. DBSCAN is the classical example (Ester et al. 1996). Given a radius $\varepsilon > 0$, the ε-neighborhood of observation i is

$$\mathcal{B}_\varepsilon(i) = \{i' : d_{ii'} \leq \varepsilon\}. \tag{5.30}$$

Observation i is a core point if

$$|\mathcal{B}_\varepsilon(i)| \geq M, \tag{5.31}$$

where M is the minimum number of neighbors. Clusters are formed by connecting core points whose neighborhoods overlap, and points not density-reachable from any cluster are treated as noise.

Density-based clustering can find non-spherical clusters and can leave outliers unlabeled. Its parameters are also interpretable: ε sets the neighborhood radius, and M sets the local density threshold. But in high-dimensional biological data, distances can concentrate, densities can vary across cell states, and preprocessing has a large effect. This is why density-based clustering is often applied after dimension reduction or manifold learning rather than directly to raw high-dimensional counts.

5.11 Model-Based Clustering

The mixture models from Chapter 02 give a probabilistic view of clustering. In a finite mixture model, each observation is generated from one of K components:

$$p(x_i) = \sum_{k=1}^{K} \pi_k f(x_i; \theta_k), \qquad \pi_k \geq 0, \qquad \sum_{k=1}^{K} \pi_k = 1. \tag{5.32}$$

The posterior probability that observation i belongs to component k is

$$\tau_{ik} = \Pr(L_i = k \mid x_i) = \frac{\pi_k f(x_i; \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell f(x_i; \theta_\ell)}. \tag{5.33}$$

A hard cluster assignment can then be obtained by maximum posterior probability:

$$\hat{\eta}_i = \operatorname*{argmax}_{k \in \{1,\dots,K\}} \tau_{ik}. \tag{5.34}$$

This probabilistic notation is useful because it separates latent component membership L_i from the estimated clustering label $\hat{\eta}_i$. It also gives uncertainty through the posterior probabilities τ_{ik}. The limitation is that the model family must be appropriate for the data. A Gaussian mixture on PCA scores and a Dirichlet-multinomial mixture on counts answer different biological and statistical questions.

5.12 Graph-Based Clustering In Single-Cell Workflows

Modern single-cell RNA-seq workflows usually do not cluster raw gene expression vectors directly. A typical workflow normalizes counts, selects variable genes, scales or residualizes features, computes PCA, constructs a nearest-neighbor graph in PCA space, and then runs community detection. Chapter 04 introduced the q-nearest-neighbor graph

$$A^{\mathrm{kNN}}_{ii'} = \mathbf{1}\{i' \in \mathcal{N}_q(i)\}. \tag{5.35}$$

Many single-cell methods use a shared-nearest-neighbor graph. The shared-neighbor weight between observations i and i' can be written as

$$S_{ii'} = |\mathcal{N}_q(i) \cap \mathcal{N}_q(i')|. \tag{5.36}$$

The graph nodes are cells. The weighted edges encode neighborhood similarity in the chosen representation, usually PCA space. Clustering then becomes community detection on this graph.

The Louvain algorithm optimizes modularity or a related community-detection objective (Blondel et al. 2008). For an undirected weighted graph with adjacency A, node degrees $d_i = \sum_{i'} A_{ii'}$, and total edge weight

$$m = \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} A_{ii'}, \tag{5.37}$$

a resolution-parameterized modularity objective is

$$Q_\gamma(\eta) = \frac{1}{2m} \sum_{i=1}^{n} \sum_{i'=1}^{n} \left(A_{ii'} - \gamma \frac{d_i d_{i'}}{2m} \right) \mathbf{1}\{\eta_i = \eta_{i'}\}. \tag{5.38}$$

The parameter $\gamma > 0$ controls resolution. Larger γ generally favors more, smaller communities; smaller γ favors fewer, larger communities. This is why a single-cell clustering result should always report the resolution parameter, the number of PCs used, the number of neighbors, and the graph construction method.

Leiden clustering modifies Louvain-style community detection to improve connectivity and local optimality guarantees (Traag et al. 2019). In practice, Leiden and Louvain often produce similar-looking single-cell annotations at broad resolution, but Leiden is preferred in many modern Python workflows because it avoids some disconnected-community pathologies of Louvain.

Some software uses the Constant Potts Model (CPM) objective. If e_k is the total weight of edges inside cluster k and $n_k = |\widehat{G}_k|$, a simplified CPM objective is

$$H_{\mathrm{CPM}}(\eta) = \sum_{k=1}^{K} \left[e_k - \gamma \frac{n_k(n_k - 1)}{2} \right]. \tag{5.39}$$

Again, γ is a resolution parameter. The exact objective depends on software and graph construction, but the interpretive point is stable: single-cell clusters are communities in a constructed graph, and the graph is downstream of preprocessing and dimension reduction.

Exercise Sheet 04 makes this concrete by constructing an SNN graph, running Louvain clustering, and varying the resolution parameter in a Seurat workflow (Exercise Sheet 04 2026). The resulting clusters should be annotated using marker genes, metadata, batch information, and biological knowledge rather than accepted solely from the algorithm.

5.13 What To Report

A clustering result is only interpretable if the workflow is reported. At minimum, one should state what matrix was clustered, how features were filtered or selected, what transformation and normalization were used, what distance or graph was used, what algorithm was used, what tuning parameters were chosen, and how the result was validated.

Mathematically, this means that the label vector should be viewed as a function

$$\hat{\eta} = \mathcal{C}_{\lambda}\left(X,\ d,\ A\right), \tag{5.40}$$

where $\mathcal{C}_{\lambda}$ denotes the clustering method and λ collects tuning parameters such as K, linkage, tree-cut height, ε, M, number of neighbors, number of PCs, random seed, or graph resolution. This notation is intentionally broad. It reminds us that changing λ or changing the representation can change the labels.

Biological validation comes after algorithmic clustering. Marker genes, known cell-type annotations, external phenotypes, perturbations, replicate structure, batch labels, and held-out data should be used to decide whether clusters are meaningful. A cluster that is reproducible but explained by sequencing technology is not a new biological state; it is a technical effect with a stable signature.

Summary

Clustering turns geometry into labels. k-means minimizes within-cluster squared Euclidean distances and is best understood as centroid-based clustering. k-medoids uses observed representatives and can work directly with dissimilarity matrices. Hierarchical clustering builds a dendrogram by repeatedly merging clusters according to linkage rules. Heatmaps use clustering as an ordering device for rows and columns. Gap statistic, silhouette, and stability analyses help assess clustering choices, but none of them removes the need for scientific judgment. Density-based methods look for dense regions and can treat some observations as noise. Model-based clustering treats groups as latent mixture components. Graph-based single-cell clustering detects communities in a nearest-neighbor graph built from a chosen representation.

The central message is the same as in Chapter 04: clusters are not discovered in a vacuum. They are produced by a chain from raw counts to transformed matrix, distance or graph, algorithm, tuning parameter, and validation. Good clustering analysis makes that chain explicit.

References

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. "Fast Unfolding of Communities in Large Networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10): P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008.

Eisen, Michael B., Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. "Cluster Analysis and Display of Genome-Wide Expression Patterns." *Proceedings of the National Academy of Sciences* 95 (25): 14863–68. https://doi.org/10.1073/pnas.95.25.14863.

Ester, Martin, Hans-Peter Kriegel, Joerg Sander, and Xiaowei Xu. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–31.

Exercise Sheet 04. 2026. *Exercise Sheet 04 Solutions: Hierarchical Clustering and Graph-Based Clustering of High-Dimensional Biological Data.* https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise04_Solution.html.

Lange, Tilman, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. 2004. "Stability-Based Validation of Clustering Solutions." *Neural Computation* 16 (6): 1299–323. https://doi.org/10.1162/089976604773717621.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20: 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.

Tibshirani, Robert, Guenther Walther, and Trevor Hastie. 2001. "Estimating the Number of Clusters in a Data Set via the Gap Statistic." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 63 (2): 411–23. https://doi.org/10.1111/1467-9868.00293.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. "From Louvain to Leiden: Guaranteeing Well-Connected Communities." *Scientific Reports* 9 (5233). https://doi.org/10.1038/s41598-019-41695-z.

6 Hypothesis Testing

Chapter Aim

The previous chapters were mainly exploratory. We transformed count data, defined distances, built embeddings, and proposed clusters. Chapter 06 introduces formal hypothesis testing: the language used when we want to quantify evidence against a specified null model. In high-dimensional biological data, this is the machinery behind differential expression, differential abundance, community-level testing, genome-wide association, and many feature-wise screening analyses.

Testing may be formulated on raw counts or on transformed feature values. For feature-wise testing, the response for feature j is written as

$$y_j = x_{\cdot j} \in \mathbb{R}^n \qquad \text{or} \qquad y_j = c_{\cdot j} \in \mathbb{N}_0^n, \tag{6.1}$$

depending on whether the test is applied to transformed continuous values or raw counts. Covariates and experimental design variables are collected in

$$Z \in \mathbb{R}^{n \times r}. \tag{6.2}$$

This notation matters. In high-throughput data, testing is rarely just "run a test for each gene." It is a model for a feature response, a design matrix, an effect of interest, and a multiplicity correction over many hypotheses.

Notation convention

Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external reference

The main book reference for this chapter is MSMB Testing. The most relevant subsections are the motivation for FDR in high-throughput settings (MSMB Section 6.1.3), the general

testing workflow (MSMB Testing Five Steps), type I and type II errors (MSMB Testing Types Of Error), the t-test (MSMB Testing t-Test), FWER (MSMB Testing FWER), FDR (MSMB Testing FDR), and the Benjamini-Hochberg procedure (MSMB Testing BH). For RNA-seq count testing, see MSMB High-Throughput Count Data. For design, confounding, and replicates, see MSMB Experimental Design. Exercise Sheet 05 is the main practical companion: it covers PERMANOVA, univariate differential abundance, multiple testing, DESeq2 on the pasilla data, and high-dimensional inference examples (Exercise Sheet 05 2026).

Learning Goals

After this chapter, students should be able to formulate null and alternative hypotheses for a feature-wise test, compute and interpret a p-value, distinguish effect size from statistical evidence, explain type I error, type II error, power, FWER, and FDR, apply Bonferroni and Benjamini-Hochberg corrections, understand why ordinary t-tests are often unstable in small-sample high-dimensional data, explain empirical-Bayes variance moderation, write down a negative-binomial GLM for differential expression, describe GWAS as large-scale marginal association testing, and interpret PERMANOVA as a distance-based community-level test.

6.1 The Five-Step Logic Of Testing

A hypothesis test begins by choosing an effect and a statistic before looking at the evidence: specify the null model, define a statistic, understand its null distribution, compute the observed evidence, and only then make a decision (see MSMB Testing Five Steps). For a feature-wise comparison, the effect might be the difference in mean expression between treated and untreated samples, the log fold change of a gene, the differential abundance of a taxon, or the association between a SNP and a disease indicator. The null hypothesis specifies what "no effect" means.

For feature j, a generic null and alternative can be written as

$$H_{0j} : \theta_j = 0, \qquad H_{1j} : \theta_j \neq 0. \tag{6.3}$$

Here θ_j is the feature-specific effect of interest. It might be a difference in means, a regression coefficient, a log fold change, or a group effect in a generalized linear model. A test statistic T_j summarizes the evidence against H_{0j}:

$$T_j = t_j(y_j, Z). \tag{6.4}$$

Under the null hypothesis, the statistic has a null distribution:

$$T_j \mid H_{0j} \sim F_{0j}. \tag{6.5}$$

The p-value is the probability, under the null model, of observing a statistic at least as extreme as the one obtained. For a two-sided test with large absolute values indicating evidence against the null,

$$p_j = \Pr_{H_{0j}}\left(|T_j^0| \geq |T_j^{\text{obs}}|\right), \tag{6.6}$$

where T_j^0 is a random statistic drawn from the null distribution and T_j^{obs} is the observed value. A small p-value says that the observed statistic is unusual under the null model. It does not say that the null hypothesis has probability p_j, and it does not measure the size or biological importance of the effect.

If the p-value is compared to a significance level α, the decision rule is

$$\text{reject } H_{0j} \quad \text{if} \quad p_j \leq \alpha. \tag{6.7}$$

For a valid continuous test under the null hypothesis, the p-value is uniformly distributed:

$$p_j \mid H_{0j} \sim \text{Uniform}(0, 1). \tag{6.8}$$

This property is the basis for multiple-testing corrections and p-value histograms. In high-throughput analyses, a roughly uniform bulk of p-values with an excess near zero is often interpreted as a mixture of null and non-null features.

6.2 Errors, Power, And Effect Sizes

Testing decisions have two possible error modes, and this distinction is the bridge from single tests to power and multiplicity (see MSMB Testing Types Of Error). If the null hypothesis is true but rejected, this is a type I error:

$$\Pr(\text{reject } H_{0j} \mid H_{0j} \text{ true}) = \alpha. \tag{6.9}$$

If the null hypothesis is false but not rejected, this is a type II error:

$$\Pr(\text{do not reject } H_{0j} \mid H_{1j} \text{ true}) = \beta_j. \tag{6.10}$$

The power is the probability of rejecting a false null:

$$\text{Power}_j = 1 - \beta_j. \tag{6.11}$$

Power depends on the effect size, sample size, noise level, model assumptions, and correction for multiple testing. This is why p-values should be reported alongside effect sizes. A tiny effect can be highly significant in a large study, and a large effect can be non-significant in a small noisy study.

For a two-group comparison on transformed data, a simple effect size is the difference in group means:

$$\hat{\theta}_j = \bar{x}_{1j} - \bar{x}_{0j}. \tag{6.12}$$

For count-based differential expression, the natural effect size is often a log fold change:

$$\theta_j = \log_2 \left(\frac{q_{1j}}{q_{0j}} \right). \tag{6.13}$$

Here q_{1j} and q_{0j} are abundance-like quantities after accounting for exposure or size factors. A log2 fold change of 1 means a two-fold increase; a log2 fold change of -1 means a two-fold decrease.

6.3 The Two-Sample t-Test

The two-sample t-test is the classical reference point for transformed continuous feature data (see MSMB Testing t-Test for the corresponding single-feature derivation). Suppose observations belong to two groups, $g_i \in \{0, 1\}$. Let n_0 and n_1 be the group sizes. For feature j, the group means are

$$\bar{x}_{gj} = \frac{1}{n_g} \sum_{i:g_i=g} x_{ij}, \qquad g \in \{0, 1\}. \tag{6.14}$$

The unbiased group variances are

$$s^2_{gj} = \frac{1}{n_g - 1} \sum_{i:g_i=g} (x_{ij} - \bar{x}_{gj})^2, \qquad g \in \{0, 1\}. \tag{6.15}$$

If the two groups are assumed to have equal variance, the pooled variance estimator is

$$s^2_{pj} = \frac{(n_0 - 1)s^2_{0j} + (n_1 - 1)s^2_{1j}}{n_0 + n_1 - 2}. \tag{6.16}$$

The classical two-sample t-statistic is

$$T_j = \frac{\bar{x}_{1j} - \bar{x}_{0j}}{s_{pj}\sqrt{1/n_0 + 1/n_1}}. \tag{6.17}$$

Under the null hypothesis of equal means, and under normality and equal-variance assumptions,

$$T_j \mid H_{0j} \sim t_{n_0+n_1-2}. \tag{6.18}$$

The two-sided p-value is

$$p_j = 2\Pr\left(t_{n_0+n_1-2} \geq |T_j|\right). \tag{6.19}$$

The t-test is simple and interpretable, but the assumptions are often fragile in high-throughput biology. Small sample sizes make variance estimates noisy. Count data are discrete and heteroscedastic. Compositional data have relative constraints. Zero inflation and outliers are common. Transformations from Chapter 03 can make t-tests more reasonable, but they do not guarantee normality or equal variances.

Welch's t-test relaxes the equal-variance assumption by using the standard error

$$\mathrm{SE}_{\mathrm{Welch},j} = \left(\frac{s_{0j}^2}{n_0} + \frac{s_{1j}^2}{n_1}\right)^{1/2}. \tag{6.20}$$

It is often preferable to the pooled t-test when group variances differ, but it still operates on transformed continuous values and does not solve count-specific modeling issues.

6.4 Rank-Based Testing

For skewed or outlier-prone transformed abundances, a rank-based test can be more robust. Let R_{ij} be the rank of observation i among the n values of feature j:

$$R_{ij} = \mathrm{rank}(x_{ij}). \tag{6.21}$$

For a two-group Wilcoxon rank-sum test, the rank-sum statistic for group 1 is

$$W_j = \sum_{i:g_i=1} R_{ij}. \tag{6.22}$$

Under the null hypothesis that the two groups have the same distribution, and ignoring ties for simplicity,

$$\mathbb{E}_0(W_j) = \frac{n_1(n+1)}{2}, \qquad \mathrm{Var}_0(W_j) = \frac{n_0 n_1 (n+1)}{12}. \tag{6.23}$$

Because the Wilcoxon test uses ranks, it is invariant under strictly increasing transformations:

$$\mathrm{rank}(x_{ij}) = \mathrm{rank}\{h(x_{ij})\} \quad \text{for strictly increasing } h. \tag{6.24}$$

Exercise Sheet 05 uses this property in the microbiome setting: the Wilcoxon test gives the same p-value for absolute abundances and for a monotone log-transformed version, while the t-test changes because it depends on the numerical scale (Exercise Sheet 05 2026).

6.5 Permutation Tests

Permutation tests construct the null distribution by randomizing labels in a way that is valid under the null hypothesis. For a two-group test, let π_b be a permutation of the group labels for permutation $b = 1, \dots, B$. The permuted statistic is

$$T_j^{(b)} = t_j(y_j, \pi_b(g)). \tag{6.25}$$

The permutation p-value can be estimated as

$$\hat{p}_j = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{|T_j^{(b)}| \geq |T_j^{\mathrm{obs}}|\}}{B+1}. \tag{6.26}$$

The plus one in numerator and denominator avoids zero p-values and reflects that the observed labeling is one of the possible labelings. Permutation tests are attractive because they require fewer parametric assumptions. Their validity still depends on exchangeability: under the null, the labels being permuted must be interchangeable. If there are batches, paired samples, repeated measures, or time structure, unrestricted permutation can be invalid.

6.6 Borrowing Strength Across Features

In high-dimensional biology, the number of features is large but the number of observations is often small. A feature-specific variance estimate s_j^2 can be highly unstable. Moderated testing methods stabilize variance estimates by borrowing information across features. This is the central idea of limma's empirical-Bayes moderated t-statistic (Smyth 2004).

A simplified moderated variance estimator has the form

$$\tilde{s}_j^2 = \frac{d_0 s_0^2 + d_j s_j^2}{d_0 + d_j}. \tag{6.27}$$

Here s_j^2 is the ordinary feature-specific variance estimate, d_j is its degrees of freedom, s_0^2 is a prior variance estimated from all features, and d_0 is a prior degrees-of-freedom parameter. The moderated t-statistic replaces s_j by $\tilde{s}_j$:

$$\tilde{T}_j = \frac{\hat{\theta}_j}{\tilde{s}_j a_j}. \tag{6.28}$$

The factor a_j depends on the design and contrast. The important idea is that noisy feature-wise variances are shrunk toward a common trend or prior value. This improves stability when n is small and p is large. The same statistical philosophy appears repeatedly in high-throughput biology: estimate many related quantities jointly rather than pretending each feature is an isolated experiment.

6.7 Count Models For Differential Expression

For RNA-seq and related count data, it is often better to test using a count model rather than a t-test on transformed values. A feature-wise negative-binomial model uses

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \mathrm{Var}(C_{ij}) = \mu_{ij} + \phi_j \mu_{ij}^2. \tag{6.29}$$

The mean is linked to size factors and covariates by

$$\log \mu_{ij} = \log s_i + z_i^\top \beta_j. \tag{6.30}$$

Here s_i is the size factor for observation i, $z_i^\top$ is row i of the design matrix Z, and $\beta_j \in \mathbb{R}^r$ contains feature-specific regression coefficients. If the coefficient of interest is the last component, the null hypothesis is

$$H_{0j} : \beta_{jr} = 0 \qquad \text{versus} \qquad H_{1j} : \beta_{jr} \neq 0. \tag{6.31}$$

A Wald statistic has the form

$$W_j = \frac{\hat{\beta}_{jr}}{\mathrm{SE}(\hat{\beta}_{jr})}. \tag{6.32}$$

For large samples, the null approximation is

$$W_j \mid H_{0j} \approx \mathcal{N}(0, 1). \tag{6.33}$$

DESeq2 estimates size factors, dispersions, coefficients, test statistics, p-values, and adjusted p-values in this modeling framework (Love et al. 2014). Exercise Sheet 05 applies this workflow to the pasilla data and emphasizes that exploratory transformations for PCA and heatmaps are not the same as the count model used for formal testing (Exercise Sheet 05 2026).

Likelihood-ratio tests compare a full model and a reduced model. If $\ell_{\mathrm{full},j}$ and $\ell_{\mathrm{red},j}$ are the maximized log-likelihoods, then

$$\Lambda_j = 2\left(\ell_{\mathrm{full},j} - \ell_{\mathrm{red},j}\right). \tag{6.34}$$

Under regularity conditions and the null hypothesis,

$$\Lambda_j \mid H_{0j} \approx \chi^2_{\Delta r}, \tag{6.35}$$

where Δr is the difference in the number of parameters. This is useful when testing multi-parameter effects, such as whether a factor with several levels contributes to expression.

6.8 Genome-Wide Association Testing

Genome-wide association studies test many genetic variants for association with a phenotype. They are therefore another large-scale feature-wise testing problem, but the features are genotypes rather than genes, taxa, or proteins. If X_{ij} is the genotype of observation i at variant j, and Y_i is a quantitative trait, a simple marginal model is

$$Y_i = \alpha_j + \gamma_j X_{ij} + u_i^\top \delta_j + \varepsilon_{ij}, \tag{6.36}$$

where u_i contains covariates such as population-structure components. The feature-wise null is

$$H_{0j} : \gamma_j = 0. \tag{6.37}$$

GWAS often controls FWER rather than FDR because individual variant discoveries can have high downstream cost and because millions of tests are performed. GWAS is a useful example of why stringent thresholds are common and why sample size matters; see also the GWAS overview in the course paper list (Tam et al. 2019).

Rare-variant tests combine information across variants in a region. In a kernel association test such as SKAT, a set of variants is represented by a genotype matrix $G \in \mathbb{R}^{n \times q}$ for one region, and a kernel matrix can be written as

$$K = G W_G G^\top. \tag{6.38}$$

Here $W_G \in \mathbb{R}^{q \times q}$ is a variant-weight matrix. A simplified score statistic has the quadratic form

$$Q = r^\top K r, \tag{6.39}$$

where r is a residual vector from a null model with covariates but without the variant set. This is a multivariate association test: it asks whether a region, rather than a single variant, is associated with the phenotype (Wu et al. 2011).

6.9 Multiple Testing

The defining feature of high-dimensional testing is multiplicity. In omics, the object of interest is usually a list of discoveries, not a single isolated decision; this is why FDR becomes a natural error criterion (see MSMB Section 6.1.3). If we test m null hypotheses at level α, even when all null hypotheses are true, the expected number of false positives is

$$\mathbb{E}(V) = m\alpha, \tag{6.40}$$

where V is the number of falsely rejected true null hypotheses. If the tests are independent, the probability of at least one false positive under the global null is

$$\Pr(V > 0) = 1 - (1 - \alpha)^m. \tag{6.41}$$

For $m = 20000$ tests and $\alpha = 0.05$, the expected number of false positives under the global null is

$$m\alpha = 20000 \cdot 0.05 = 1000. \tag{6.42}$$

This simple calculation is the reason one-at-a-time testing is not enough when thousands of features are tested.

To define the relevant error criteria, it is useful to keep explicit bookkeeping for the relationship between the testing decision and the unknown truth. The following table adapts the notation of MSMB Table 6.2 (see MSMB Testing FDR).

Table 6.1: Multiple-testing outcomes, adapted from MSMB Table 6.2. Here m is the number of tested hypotheses, m_0 is the number of true null hypotheses, R is the number of rejections, V is the number of false rejections, and S is the number of true discoveries.

Test decision	Null hypothesis is true	Null hypothesis is false	Total
Rejected	V	S	R
Not rejected	U	T	$m - R$
Total	m_0	$m - m_0$	m

The table implies the accounting identities

$$R = V + S, \qquad m - R = U + T. \tag{6.43}$$

The family-wise error rate is the probability of making at least one false rejection (see MSMB Testing FWER):

$$\mathrm{FWER} = \Pr(V > 0). \tag{6.44}$$

The false discovery proportion is

$$\mathrm{FDP} = \frac{V}{\max(R, 1)}. \tag{6.45}$$

The false discovery rate is the expectation of the false discovery proportion (see MSMB Testing FDR):

$$\mathrm{FDR} = \mathbb{E}\left[\frac{V}{\max(R, 1)}\right]. \tag{6.46}$$

FWER control is stringent: it tries to avoid any false positive. FDR control is often more useful in omics: it allows some false positives but controls their expected fraction among discoveries.

Bonferroni control rejects hypothesis j if

$$p_j \leq \frac{\alpha}{m}. \tag{6.47}$$

Equivalently, the Bonferroni-adjusted p-value is

$$p_j^{\text{Bonf}} = \min(mp_j, 1). \tag{6.48}$$

This controls FWER but can be conservative when m is large.

6.10 Benjamini-Hochberg FDR Control

The Benjamini-Hochberg procedure controls the FDR under independence and certain dependence conditions (Benjamini and Hochberg 1995); see also MSMB Testing BH. Sort the p-values:

$$p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}. \tag{6.49}$$

For target FDR level α, define

$$\hat{k} = \max\left\{k : p_{(k)} \leq \frac{k}{m}\alpha\right\}. \tag{6.50}$$

If such a $\hat{k}$ exists, reject the hypotheses corresponding to $p_{(1)}, \ldots, p_{(\hat{k})}$. If not, reject none.

BH-adjusted p-values can be written as

$$p_{(k)}^{\text{BH}} = \min_{\ell \geq k}\left\{\frac{m}{\ell}p_{(\ell)}\right\}, \tag{6.51}$$

with truncation at 1:

$$p_{(k)}^{\text{BH}} \leftarrow \min\left(p_{(k)}^{\text{BH}}, 1\right). \tag{6.52}$$

The adjusted p-value is the smallest FDR level at which the feature would be called significant by the BH procedure. In differential expression and differential abundance tables, this is often reported as `padj`, `qvalue`, or adjusted p-value.

6.11 P-Value Histograms And Independent Filtering

A p-value histogram is a diagnostic for whether the test calibration and signal structure look plausible (see MSMB Testing P-Value Histogram). If most null hypotheses are true and tests are well calibrated, null p-values should contribute an approximately uniform background. Non-null features create an enrichment near zero. This informal mixture view can be written as

$$f(p) = \pi_0 f_0(p) + (1 - \pi_0) f_1(p), \qquad f_0(p) = 1 \text{ for } 0 \leq p \leq 1. \tag{6.53}$$

Here π_0 is the fraction of true null hypotheses. This equation is not required for BH, but it explains the intuition behind p-value histograms and local FDR methods (see MSMB Testing Local FDR for a fuller two-groups treatment).

Filtering can improve power if it removes features that have little chance of being significant without using the test statistic in a biased way. Let q_j be a filtering statistic, such as mean normalized count or prevalence. Independent filtering keeps features satisfying

$$q_j \geq q_{\min}. \tag{6.54}$$

The filter must be independent of the test statistic under the null. Filtering low-count genes before DESeq2 testing is an example used to improve power while preserving error control when done properly. The same principle can be extended by using informative covariates to weight hypotheses while preserving FDR control under the required independence conditions (see MSMB Testing IHW).

6.12 Design, Covariates, And Confounding

Many flawed testing analyses come from using the wrong design. Suppose $z_i^\top$ contains treatment, batch, sequencing type, sex, age, donor, or other covariates. A feature-wise linear model on transformed data is

$$y_j = Z\beta_j + \varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}(0, \sigma_j^2 I_n). \tag{6.55}$$

If the treatment coefficient is β_{jr}, the null is again

$$H_{0j} : \beta_{jr} = 0. \tag{6.56}$$

This formulation makes the role of covariates explicit. A one-factor model and a two-factor model can answer different questions. Exercise Sheet 05 shows this with the pasilla data, where sequencing type enters the design before testing the condition effect (Exercise Sheet 05 2026).

Confounding occurs when the effect of interest is statistically entangled with another variable. In matrix terms, the design is problematic if columns of Z are nearly collinear. A simple diagnostic is the rank:

$$\operatorname{rank}(Z) < r. \tag{6.57}$$

If the design matrix is rank deficient, some coefficients are not identifiable. Even if Z has full rank, near-confounding can inflate standard errors and reduce interpretability. This is why experimental design belongs before data analysis, not after it.

6.13 Community-Level Testing With PERMANOVA

Feature-wise testing asks which features differ. Sometimes the scientific question is instead whether the whole community or multivariate profile differs between groups. PERMANOVA is a distance-based test for this purpose (Anderson 2001). It begins with a distance matrix

$$D = (d_{ii'})_{i,i'=1}^{n}. \tag{6.58}$$

For a grouping variable, PERMANOVA partitions distance-based variability into between-group and within-group components:

$$SS_{\text{total}} = SS_{\text{between}} + SS_{\text{within}}. \tag{6.59}$$

The effect size is often reported as

$$R^2 = \frac{SS_{\text{between}}}{SS_{\text{total}}}. \tag{6.60}$$

A pseudo-F statistic compares between-group and within-group variability:

$$F_{\text{PERM}} = \frac{SS_{\text{between}}/(G-1)}{SS_{\text{within}}/(n-G)}. \tag{6.61}$$

Here G is the number of groups. Significance is assessed by permuting the group labels and recomputing the statistic:

$$\hat{p}_{\mathrm{PERM}} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{F_{\mathrm{PERM}}^{(b)} \geq F_{\mathrm{PERM}}^{\mathrm{obs}}\}}{B+1}. \tag{6.62}$$

PERMANOVA is not a general replacement for careful modeling. It tests whether groups differ in the distance-based sense induced by D. If the groups have different dispersions, a significant result may reflect different spread rather than only different group centers. Exercise Sheet 05 explicitly checks homogeneity of multivariate dispersions after a significant PERMANOVA result (Exercise Sheet 05 2026).

6.14 What To Report

A testing result is interpretable only if the full testing workflow is reported. At minimum, one should state the data object tested, the normalization and transformation, the features retained after filtering, the null and alternative hypotheses, the test statistic, the design matrix, the multiplicity correction, the significance threshold, the effect sizes, and the diagnostic checks.

For feature-wise testing, a compact result table often contains

$$\left(\hat{\theta}_j, \mathrm{SE}(\hat{\theta}_j), T_j, p_j, p_j^{\mathrm{adj}}\right), \qquad j = 1, \ldots, m. \tag{6.63}$$

The adjusted p-value controls the error criterion; the effect size tells us whether the result is biologically meaningful. Both are needed.

Differential abundance analysis in microbiome data is especially sensitive to preprocessing, compositionality, sparsity, and method choice. Nearing and colleagues' comparison of differential abundance methods found that different tools can produce substantially different discoveries across datasets (Nearing et al. 2022). This is not a reason to avoid testing. It is a reason to report assumptions, perform sensitivity analyses, and avoid treating a single method as an oracle.

Summary

Hypothesis testing turns a biological question into a null model, a statistic, a p-value, and a decision rule. In high-dimensional biology, the main complications are small sample sizes, noisy variance estimates, count distributions, compositional constraints, confounding, and multiplicity. Ordinary t-tests and Wilcoxon tests are useful starting points for transformed feature-wise data. Moderated tests borrow strength across features. Negative-binomial GLMs test differential expression directly on counts, while GWAS applies the same feature-wise testing logic to genetic variants. Bonferroni controls the probability of any false positive, while Benjamini-Hochberg controls the expected fraction of false discoveries among rejected hypotheses. PERMANOVA

then extends the testing discussion from feature-wise hypotheses to community-level differences defined by a distance matrix.

The chapter's central message is that a significant result is not a standalone scientific conclusion. It is evidence against a specified null model under a specified workflow. The workflow, effect size, multiplicity correction, diagnostics, and biological validation determine what the result can mean.

References

Anderson, Marti J. 2001. "A New Method for Non-Parametric Multivariate Analysis of Variance." *Austral Ecology* 26 (1): 32–46. https://doi.org/10.1046/j.1442-9993.2001.01070.x.

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 57 (1): 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Exercise Sheet 05. 2026. *Exercise Sheet 05 Solutions: Hypothesis Testing, Differential Abundance, PERMANOVA, and High-Dimensional Inference*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise05_Solution.html.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12). https://doi.org/10.1186/s13059-014-0550-8.

Nearing, Jacob T., Gavin M. Douglas, Molly G. Hayes, et al. 2022. "Microbiome Differential Abundance Methods Produce Different Results Across 38 Datasets." *Nature Communications* 13 (1). https://doi.org/10.1038/s41467-022-28034-z.

Smyth, Gordon K. 2004. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology* 3 (1): 1–25. https://doi.org/10.2202/1544-6115.1027.

Tam, Vivian, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, and David Meyre. 2019. "Benefits and Limitations of Genome-Wide Association Studies." *Nature Reviews Genetics* 20 (8): 467–84. https://doi.org/10.1038/s41576-019-0127-1.

Wu, Michael C., Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. 2011. "Rare-Variant Association Testing for Sequencing Data with the Sequence Kernel Association Test." *The American Journal of Human Genetics* 89 (1): 82–93. https://doi.org/10.1016/j.ajhg.2011.05.029.

7 Regression and Classification

Chapter Aim

Chapter 06 treated testing as a decision problem: define a null model, compute a statistic, obtain a p-value, and correct for multiplicity. Chapter 07 keeps the same design-based thinking but changes the emphasis. Regression models describe how an outcome changes with covariates, how a feature changes with an experimental condition, and how a class label can be predicted from high-dimensional measurements.

There are two recurring directions. In feature-wise regression, each molecular feature is treated as a response and the experimental design is the predictor:

$$y_j = x_{\cdot j} \in \mathbb{R}^n \qquad \text{or} \qquad y_j = c_{\cdot j} \in \mathbb{N}_0^n. \tag{7.1}$$

The covariates, treatments, batches, and experimental factors are collected in the design matrix

$$Z \in \mathbb{R}^{n \times r}. \tag{7.2}$$

In supervised regression or classification, the high-dimensional feature matrix is the predictor and an external response is the outcome:

$$X \in \mathbb{R}^{n \times p}, \qquad y \in \mathbb{R}^n \quad \text{or} \quad a \in \{0,1\}^n. \tag{7.3}$$

Here y is a continuous response, such as drug sensitivity or disease score, and a is a binary class label, such as case/control status. The same matrix X can therefore appear in two roles. In differential expression, columns of X or C are responses and Z explains them. In prediction, rows of X are predictor vectors and y or a is what we want to predict.

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external references

For count regression, generalized linear models, DESeq2, contrasts, and multifactor designs, the main book reference is MSMB High-Throughput Count Data, especially the sections on linear models (MSMB Count Data Linear Models), generalized linear models (MSMB Count Data GLM), DESeq2 GLMs (MSMB Count Data GLM DESeq2), and two-factor designs (MSMB Count Data Two-Factor Analysis). For prediction, classification, objective functions, penalization, and cross-validation, see MSMB Supervised Learning, especially MSMB Supervised Data, MSMB Supervised Objective Functions, MSMB Supervised Penalization, and MSMB Supervised Cross-Validation. The corresponding computations for DESeq2-style testing, high-dimensional regression, lasso with `glmnet`, stability selection, multi-split inference, and ridge regression are practiced in Exercise Sheet 05 (Exercise Sheet 05 2026).

Learning Goals

After this chapter, students should be able to write a feature-wise linear model with a design matrix, interpret regression coefficients and contrasts, distinguish ordinary least squares from generalized linear models, formulate a negative-binomial regression model for count data, explain the offset in DESeq2-like models, derive Wald statistics for contrasts, distinguish regression from classification, write logistic regression as a Bernoulli GLM, explain why regularization is needed when p is large relative to n, formulate ridge and lasso objectives, understand sparse logistic regression, and explain why log-contrast regression is a natural model for compositional predictors.

7.1 From Testing To Regression

The two-sample t-test from Chapter 06 is already a regression model in disguise. Suppose observations belong to two groups, $g_i \in \{0, 1\}$, and $y_j = x_{\cdot j}$ is a transformed continuous feature. A two-group model can be written as

$$y_{ij} = \beta_{0j} + g_i \beta_{1j} + \varepsilon_{ij}, \qquad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma_j^2). \tag{7.4}$$

The coefficient β_{0j} is the expected value in the reference group, and β_{1j} is the group difference for feature j:

$$\mathbb{E}(y_{ij} \mid g_i = 0) = \beta_{0j}, \qquad \mathbb{E}(y_{ij} \mid g_i = 1) = \beta_{0j} + \beta_{1j}. \tag{7.5}$$

Testing no group effect is therefore the regression hypothesis

$$H_{0j} : \beta_{1j} = 0. \tag{7.6}$$

This view matters because it immediately generalizes. Once the problem is written as a regression model, adding a batch effect, a sex effect, a continuous age covariate, a pairing variable, or an interaction is no longer a different testing problem. It is a different design matrix.

7.2 Feature-Wise Linear Models

For transformed continuous values, a feature-wise linear model uses

$$y_j = Z\beta_j + \varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}(0, \sigma_j^2 I_n). \tag{7.7}$$

Here $y_j \in \mathbb{R}^n$ is feature j across observations, $Z \in \mathbb{R}^{n \times r}$ is the design matrix, $\beta_j \in \mathbb{R}^r$ contains feature-specific coefficients, and $\varepsilon_j \in \mathbb{R}^n$ is residual noise. The fitted mean vector is

$$\mu_j = \mathbb{E}(y_j \mid Z) = Z\beta_j. \tag{7.8}$$

Ordinary least squares estimates β_j by minimizing the residual sum of squares:

$$\hat{\beta}_j = \arg\min_{\beta \in \mathbb{R}^r} \left\| y_j - Z\beta \right\|_2^2. \tag{7.9}$$

If $Z^\top Z$ is invertible, the solution is

$$\hat{\beta}_j = (Z^\top Z)^{-1} Z^\top y_j. \tag{7.10}$$

This equation is not only a computational formula. It says that the coefficient estimates are determined by the geometry of the design matrix. If two columns of Z are nearly collinear, the corresponding effects are hard to separate. If a treatment and a batch are perfectly confounded, the design matrix loses rank:

$$\operatorname{rank}(Z) < r. \tag{7.11}$$

In that case, some coefficients are not identifiable. Regression does not repair a bad design; it makes the consequences of the design explicit.

The residual vector and residual variance estimate are

$$\hat{\varepsilon}_j = y_j - Z\hat{\beta}_j, \qquad \hat{\sigma}_j^2 = \frac{\|\hat{\varepsilon}_j\|_2^2}{n-r}. \tag{7.12}$$

Under the Gaussian linear model, the covariance matrix of the coefficient estimator is

$$\mathrm{Cov}(\hat{\beta}_j \mid Z) = \sigma_j^2 (Z^\top Z)^{-1}. \tag{7.13}$$

In high-throughput biology, Equation 7.13 also explains why small sample sizes are painful. With few observations and several design columns, the coefficient uncertainty can be large even when the fitted effect looks substantial.

7.3 Contrasts

A regression coefficient is interpreted relative to the design matrix. For example, suppose an experiment has two binary factors: siRNA treatment and drug treatment. Let

$$z_{i1} = \mathbf{1}\{\text{observation } i \text{ receives siRNA}\}, \qquad z_{i2} = \mathbf{1}\{\text{observation } i \text{ receives drug}\}. \tag{7.14}$$

With an intercept, two main effects, and an interaction, the design row is

$$z_i^\top = \begin{pmatrix} 1 & z_{i1} & z_{i2} & z_{i1}z_{i2} \end{pmatrix}, \qquad Z \in \mathbb{R}^{n \times 4}. \tag{7.15}$$

For feature j, write

$$\beta_j = \begin{pmatrix} \beta_{0j} \\ \beta_{\text{siRNA},j} \\ \beta_{\text{drug},j} \\ \beta_{\text{siRNA:drug},j} \end{pmatrix}. \tag{7.16}$$

The expected transformed value is then

$$\mathbb{E}(y_{ij} \mid z_i) = \beta_{0j} + z_{i1}\beta_{\text{siRNA},j} + z_{i2}\beta_{\text{drug},j} + z_{i1}z_{i2}\beta_{\text{siRNA:drug},j}. \tag{7.17}$$

The interaction coefficient measures departure from additivity on the model scale. It is not the combined treatment effect itself. The combined treatment versus control effect is

$$\beta_{\text{siRNA},j} + \beta_{\text{drug},j} + \beta_{\text{siRNA:drug},j}. \tag{7.18}$$

Such quantities are contrasts. A contrast vector $h \in \mathbb{R}^r$ defines the estimand

$$\theta_j = h^\top \beta_j. \tag{7.19}$$

For the two-factor model above, the drug effect among siRNA-treated observations is represented by

$$h^\top = \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix}, \qquad h^\top \beta_j = \beta_{\text{drug},j} + \beta_{\text{siRNA:drug},j}. \tag{7.20}$$

This is the practical reason to care about design matrices. The design formula determines the coefficient vector. The scientific comparison is then a linear combination of those coefficients.

7.4 Wald Tests For Regression Coefficients

The testing machinery from Chapter 06 reappears once a coefficient or contrast has been estimated. For a contrast $h^\top \beta_j$, the null hypothesis is

$$H_{0j} : h^\top \beta_j = \theta_0. \tag{7.21}$$

The estimated contrast is $h^\top \hat{\beta}_j$, and its standard error is

$$\operatorname{SE}(h^\top \hat{\beta}_j) = \left[h^\top \widehat{\operatorname{Cov}}(\hat{\beta}_j) h \right]^{1/2}. \tag{7.22}$$

The Wald statistic is

$$W_j = \frac{h^\top \hat{\beta}_j - \theta_0}{\left[h^\top \widehat{\operatorname{Cov}}(\hat{\beta}_j) h \right]^{1/2}}. \tag{7.23}$$

With a large-sample normal approximation,

$$W_j \mid H_{0j} \approx \mathcal{N}(0, 1). \tag{7.24}$$

The usual two-sided p-value is

$$p_j = 2 \left\{ 1 - \Phi(|W_j|) \right\}, \tag{7.25}$$

where Φ is the cumulative distribution function of the standard normal distribution. This is the same structure as in Chapter 06: a model, an effect, an uncertainty estimate, a test statistic, and then multiplicity correction over many tested features.

7.5 Generalized Linear Models

Linear regression assumes a continuous response with additive Gaussian noise. Generalized linear models keep the linear predictor but change the response distribution and the link function. For observation i, write the conditional mean as

$$\mu_i = \mathbb{E}(Y_i \mid z_i). \tag{7.26}$$

A GLM links the mean to the covariates through

$$g(\mu_i) = z_i^\top \beta, \tag{7.27}$$

where g is a link function. The inverse link gives

$$\mu_i = g^{-1}(z_i^\top \beta). \tag{7.28}$$

The same design matrix can therefore be used for continuous, binary, and count outcomes, but the sampling model changes. A Gaussian linear model uses an identity link. Logistic regression uses a logit link for binary data. Negative-binomial regression uses a log link for count data. This GLM perspective is the bridge between ordinary regression and RNA-seq differential expression (see MSMB Count Data GLM).

7.6 Negative-Binomial Regression For Count Data

For sequencing counts, the feature-wise response is $y_j = c_{\cdot j} \in \mathbb{N}_0^n$. A DESeq2-like model uses a negative-binomial GLM for each feature j:

$$C_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \mathrm{Var}(C_{ij} \mid Z) = \mu_{ij} + \phi_j \mu_{ij}^2. \tag{7.29}$$

The mean is decomposed into an observation-specific size factor and an abundance-like quantity:

$$\mu_{ij} = s_i q_{ij}, \qquad s_i > 0. \tag{7.30}$$

The regression model is placed on the log scale:

$$\log q_{ij} = z_i^\top \beta_j. \tag{7.31}$$

Equivalently,

$$\log \mu_{ij} = \log s_i + z_i^\top \beta_j. \tag{7.32}$$

Thus $\log s_i$ is an offset: it is part of the linear predictor, but it is known or estimated before the coefficient-fitting step and is not a coefficient of biological interest. The fitted mean can be written as

$$\hat{\mu}_{ij} = \hat{s}_i \exp(z_i^\top \hat{\beta}_j). \tag{7.33}$$

For fixed dispersion ϕ_j, the coefficient estimate maximizes the negative-binomial log-likelihood:

$$\hat{\beta}_j = \arg\max_{\beta \in \mathbb{R}^r} \sum_{i=1}^{n} \ell_{ij}(\beta; c_{ij}, \hat{s}_i, z_i, \hat{\phi}_j). \tag{7.34}$$

Equivalently, it minimizes the negative log-likelihood:

$$\hat{\beta}_j = \arg\min_{\beta \in \mathbb{R}^r} \left\{ -\sum_{i=1}^{n} \ell_{ij}(\beta; c_{ij}, \hat{s}_i, z_i, \hat{\phi}_j) \right\}. \tag{7.35}$$

This is the precise analogue of least squares. Ordinary least squares minimizes squared residuals because a Gaussian noise model implies a squared-error objective. Negative-binomial regression maximizes a count likelihood because the response is discrete, heteroscedastic, and overdispersed. DESeq2 additionally estimates and moderates feature-specific dispersions by borrowing information across genes (Love et al. 2014); see also MSMB Count Data GLM DESeq2 and MSMB Count Data Sharing Information.

For a fitted negative-binomial GLM, a large-sample approximation to the coefficient covariance has the form

$$\widehat{\mathrm{Cov}}(\hat{\beta}_j) \approx (Z^\top \widehat{W}_j Z)^{-1}, \tag{7.36}$$

where $\widehat{W}_j \in \mathbb{R}^{n \times n}$ is a diagonal weight matrix. Under the mean-dispersion parameterization used here, a typical working-weight form is

$$\widehat{W}_j = \mathrm{diag}\left(\frac{\hat{\mu}_{1j}}{1 + \hat{\phi}_j \hat{\mu}_{1j}}, \ldots, \frac{\hat{\mu}_{nj}}{1 + \hat{\phi}_j \hat{\mu}_{nj}} \right). \tag{7.37}$$

The same contrast machinery then applies. A condition effect, batch-adjusted treatment effect, interaction effect, or combined-treatment effect can all be written as $h^\top \beta_j$ and tested using Equation 7.23.

7.7 Regression For Prediction

Regression is not always used for inference about a coefficient. Sometimes the goal is prediction. Let $y_i \in \mathbb{R}$ be a continuous outcome and $x_i^\top \in \mathbb{R}^{1\times p}$ the high-dimensional feature vector for observation i. A linear prediction rule has the form

$$\hat{y}_i = \beta_0 + x_i^\top \beta, \tag{7.38}$$

where $\beta \in \mathbb{R}^p$ now contains predictor coefficients. The empirical squared prediction loss is

$$\mathcal{L}_{\text{sq}}(\beta_0, \beta) = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \beta_0 - x_i^\top \beta\right)^2. \tag{7.39}$$

The distinction is now made by dimension and indexing. In feature-wise regression, β_j belongs to response feature j and has length r, the number of design columns. In supervised prediction, β has length p, the number of measured predictor features. When p is large, β is high-dimensional and ordinary least squares can overfit badly or may not even be identifiable.

The training error is

$$\widehat{R}_{\text{train}}(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i)), \tag{7.40}$$

where L is a loss function. The scientific quantity we usually care about is the prediction error on new observations:

$$R(\hat{f}) = \mathbb{E}\left[L(Y_{\text{new}}, \hat{f}(X_{\text{new}}))\right]. \tag{7.41}$$

This expectation is over future data from the target population. In high-dimensional biology, training error is often much too optimistic, especially if feature selection, preprocessing, tuning, and model choice were performed using the same observations. This is the central supervised-learning warning in MSMB Supervised Learning and MSMB Supervised Cross-Validation.

7.8 Classification And Logistic Regression

For binary classification, the response is $a_i \in \{0, 1\}$. Logistic regression models the conditional probability of class 1:

$$\rho_i = \Pr(a_i = 1 \mid x_i). \tag{7.42}$$

The logit link is

$$\log\left(\frac{\rho_i}{1-\rho_i}\right) = \beta_0 + x_i^\top \beta. \tag{7.43}$$

Equivalently,

$$\rho_i = \frac{\exp(\beta_0 + x_i^\top \beta)}{1 + \exp(\beta_0 + x_i^\top \beta)}. \tag{7.44}$$

The Bernoulli likelihood contribution is

$$\Pr(a_i \mid x_i, \beta_0, \beta) = \rho_i^{a_i}(1-\rho_i)^{1-a_i}. \tag{7.45}$$

The negative log-likelihood, averaged over observations, is

$$\mathcal{L}_{\log}(\beta_0, \beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[a_i \log \rho_i + (1-a_i)\log(1-\rho_i)\right]. \tag{7.46}$$

A class prediction is obtained by thresholding the fitted probability:

$$\hat{a}_i = \mathbf{1}\{\hat{\rho}_i \geq t\}, \qquad 0 < t < 1. \tag{7.47}$$

The default $t = 1/2$ is not always optimal. In medical classification, false negatives and false positives may have different consequences, and the decision threshold should reflect the scientific or clinical cost structure. The fitted probabilities and the final decisions are therefore different objects.

7.9 High-Dimensional Regression And Regularization

When p is large relative to n, many coefficient vectors fit the training data almost equally well. Regularization makes the optimization problem better posed by penalizing model complexity. Ridge regression solves

$$(\hat{\beta}_{0,\lambda}^{\text{ridge}}, \hat{\beta}_{\lambda}^{\text{ridge}}) = \arg\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - x_i^\top \beta\right)^2 + \lambda \|\beta\|_2^2 \right\}. \tag{7.48}$$

The penalty shrinks coefficients toward zero but usually does not set them exactly to zero. The lasso instead uses an ℓ_1 penalty:

$$(\hat{\beta}_{0,\lambda}^{\text{lasso}}, \hat{\beta}_{\lambda}^{\text{lasso}}) = \arg\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - x_i^\top \beta\right)^2 + \lambda \|\beta\|_1 \right\}. \tag{7.49}$$

Here

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|. \tag{7.50}$$

The lasso can produce sparse estimates, meaning that many $\hat{\beta}_{\lambda j}$ are exactly zero. This makes it attractive when the scientific hope is that only a subset of features is predictive, but sparsity is a modeling assumption, not a guarantee of biological causality (Tibshirani 1996). Penalization and model complexity are discussed in MSMB Supervised Penalization.

7.10 Model Selection For Penalized Regression

The tuning parameter $\lambda \geq 0$ controls the strength of regularization. Choosing λ is therefore part of the statistical specification. In high-dimensional biology, there are three common questions that should be kept separate: which value predicts best, which value gives a parsimonious likelihood-based model, and which selected features are stable under perturbations of the data.

Cross-validation. Cross-validation estimates out-of-sample prediction error. Let $\mathcal{I}_1, \ldots, \mathcal{I}_K$ be a partition of the observation indices into folds. If $\hat{f}_{\lambda}^{(-k)}$ is trained without fold k, the cross-validation estimate of prediction error is

$$\widehat{R}_{\text{CV}}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} L\left(y_i, \hat{f}_{\lambda}^{(-k)}(x_i)\right). \tag{7.51}$$

The selected tuning parameter is often chosen as

$$\hat{\lambda} = \arg\min_{\lambda \in \Lambda} \widehat{R}_{\mathrm{CV}}(\lambda), \tag{7.52}$$

where Λ is a grid of candidate values. A more conservative variant, often called the one-standard-error rule, chooses the largest λ whose cross-validation error is within one estimated standard error of the minimum:

$$\hat{\lambda}_{\mathrm{1SE}} = \max\left\{\lambda \in \Lambda : \widehat{R}_{\mathrm{CV}}(\lambda) \le \widehat{R}_{\mathrm{CV}}(\hat{\lambda}) + \widehat{\mathrm{SE}}_{\mathrm{CV}}(\hat{\lambda})\right\}. \tag{7.53}$$

This rule often gives a sparser model with essentially indistinguishable predictive performance. All preprocessing steps that use the outcome, including feature screening and tuning decisions, must be inside the cross-validation loop. Otherwise the estimated prediction error is biased downward (see MSMB Supervised Cross-Validation).

Information criteria. If the goal is model selection rather than prediction-error estimation, an information criterion can be used along the regularization path. For a fitted model at λ, let $\ell(\hat{\beta}_{0,\lambda}, \hat{\beta}_{\lambda})$ be the unpenalized log-likelihood evaluated at the penalized estimate, and let $\mathrm{df}(\lambda)$ denote an effective number of fitted parameters. The Akaike information criterion is

$$\mathrm{AIC}(\lambda) = -2\ell(\hat{\beta}_{0,\lambda}, \hat{\beta}_{\lambda}) + 2\,\mathrm{df}(\lambda). \tag{7.54}$$

The Bayesian information criterion uses a stronger sample-size-dependent penalty:

$$\mathrm{BIC}(\lambda) = -2\ell(\hat{\beta}_{0,\lambda}, \hat{\beta}_{\lambda}) + \mathrm{df}(\lambda)\log n. \tag{7.55}$$

For lasso linear regression, a common approximation is to count the intercept plus the number of nonzero coefficients,

$$\mathrm{df}(\lambda) \approx 1 + \left|\left\{j : \hat{\beta}_{\lambda j} \neq 0\right\}\right|. \tag{7.56}$$

The selected tuning parameter is then, for example,

$$\hat{\lambda}_{\mathrm{AIC}} = \arg\min_{\lambda \in \Lambda} \mathrm{AIC}(\lambda), \tag{7.57}$$

or

$$\hat{\lambda}_{\mathrm{BIC}} = \arg\min_{\lambda \in \Lambda} \mathrm{BIC}(\lambda). \tag{7.58}$$

AIC is motivated by predictive information loss and often favors larger models, while BIC penalizes model size through $\log n$ and therefore tends to favor more parsimonious models (Akaike 1974; Schwarz 1978). In high-dimensional settings, this interpretation is approximate: the effective degrees of freedom, correlated features, and the use of the same data for tuning all matter. Information criteria are useful for comparing models along a path, but they should not be mistaken for biological validation.

Stability selection. Sparse models can have similar prediction error but different selected feature sets. Stability selection asks how often each feature is selected under subsampling or resampling (Meinshausen and Bühlmann 2010). Let B subsamples be drawn, and let $\hat{S}_\lambda^{(b)}$ be the active set selected at λ in subsample b:

$$\hat{S}_\lambda^{(b)} = \left\{ j : \hat{\beta}_{\lambda j}^{(b)} \neq 0 \right\}, \qquad b = 1, \dots, B. \tag{7.59}$$

The selection probability of feature j is estimated by

$$\hat{\pi}_j(\lambda) = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\left\{ j \in \hat{S}_\lambda^{(b)} \right\}. \tag{7.60}$$

For a threshold $\pi_{\text{thr}} \in (1/2, 1)$, the stable set is

$$\hat{S}_{\text{stable}}(\lambda, \pi_{\text{thr}}) = \left\{ j : \hat{\pi}_j(\lambda) \geq \pi_{\text{thr}} \right\}. \tag{7.61}$$

Stability selection changes the question from "which features were selected once?" to "which features are repeatedly selected under reasonable perturbations of the data?" This is particularly important for omics data, where correlated genes, proteins, metabolites, or taxa can substitute for one another. The practical lasso workflow is therefore: fit a regularization path, select λ by cross-validation or an information criterion, inspect the selected variables, and then assess stability under resampling. This sequence is carried out for high-dimensional regression in Exercise Sheet 05 (Exercise Sheet 05 2026), and it is the same computational logic that will reappear in Chapter 08 when sparse regression is used to learn graph neighborhoods.

7.11 Sparse Logistic Regression

Sparse logistic regression combines the logistic loss with an ℓ_1 penalty:

$$(\hat{\beta}_{0,\lambda}, \hat{\beta}_\lambda) = \arg\min_{\beta_0, \beta} \left\{ \mathcal{L}_{\log}(\beta_0, \beta) + \lambda \|\beta\|_1 \right\}. \tag{7.62}$$

The coefficient vector β has one entry per measured feature. In a gene-expression classifier, β_j is the contribution of feature j to the log-odds of class 1, conditional on the other selected features:

$$\log\left(\frac{\rho_i}{1-\rho_i}\right) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j. \tag{7.63}$$

For a fixed λ, features with $\hat{\beta}_{\lambda j} = 0$ do not enter the fitted log-odds. This can make the classifier easier to inspect, but the selected set can be unstable when features are highly correlated. In omics, many genes, proteins, or taxa carry redundant information. Sparse prediction should therefore be assessed by held-out performance and by stability, not only by the apparent interpretability of one selected feature list.

7.12 Sparse Log-Contrast Regression For Compositions

Microbiome and other sequencing-derived abundance data are often compositional: the observed row carries relative information. If $\pi_i \in \Delta^{p-1}$ is a continuous composition, a log-contrast model predicts an outcome from log components:

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j \log \pi_{ij}. \tag{7.64}$$

The defining constraint is

$$\sum_{j=1}^{p} \beta_j = 0. \tag{7.65}$$

This constraint makes the predictor scale-invariant. If all components are multiplied by the same positive constant before renormalization, the log-contrast does not change:

$$\sum_{j=1}^{p} \beta_j \log(a\pi_{ij}) = \sum_{j=1}^{p} \beta_j \log \pi_{ij} + \log(a) \sum_{j=1}^{p} \beta_j = \sum_{j=1}^{p} \beta_j \log \pi_{ij}. \tag{7.66}$$

This is the regression analogue of the log-ratio geometry introduced in Chapter 03. The model depends on ratios among components, not on their absolute scale. Equivalently, the same linear predictor can be written as a model in all pairwise log-ratios. If $\mathbf{1}^\top \beta = 0$, then

$$\sum_{j=1}^{p} \beta_j \log \pi_{ij} = \frac{1}{p} \sum_{1 \le j < \ell \le p} (\beta_j - \beta_\ell) \log\left(\frac{\pi_{ij}}{\pi_{i\ell}}\right). \tag{7.67}$$

Thus a log-contrast regression is a constrained linear model on the complete set of pairwise log-ratios. The pairwise coefficient for $\log(\pi_{ij}/\pi_{i\ell})$ is proportional to $\beta_j - \beta_\ell$, so the model is identifiable through relative effects between components rather than through absolute effects of individual proportions. For a continuous response, sparse log-contrast regression can be written as

$$(\hat{\beta}_{0,\lambda}, \hat{\beta}_\lambda) = \arg\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j \log \pi_{ij} \right)^2 + \lambda \|\beta\|_1 \right\} \quad \text{subject to} \quad \mathbf{1}^\top \beta = 0. \tag{7.68}$$

For binary classification, the same log-contrast predictor can be inserted into the logistic model:

$$\log\left(\frac{\rho_i}{1-\rho_i}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j \log \pi_{ij}, \qquad \mathbf{1}^\top \beta = 0. \tag{7.69}$$

Zeros require special care because $\log 0$ is undefined. In practice, one must specify a zero-handling rule, such as filtering, pseudocounts, model-based replacement, or a method designed for sparse compositions. The important statistical point is not the pseudocount itself but the model scale: compositional predictors should be interpreted through log-ratios rather than ordinary changes in raw proportions. For the log-ratio foundation, see the compositional material used earlier (Egozcue and Tolosana-Delgado 2007) and Aitchison's logistic-normal construction (Aitchison and Shen 1980).

7.13 What To Report

A regression analysis is interpretable only if the response, predictors, model family, link function, design matrix, contrasts, preprocessing, and validation procedure are stated. For feature-wise regression, one should report the tested contrast, the coefficient estimate, standard error, test statistic, p-value, adjusted p-value, and the model formula. For prediction, one should report the training data, the outcome, preprocessing, feature filtering, loss function, tuning procedure, whether λ was selected by cross-validation, AIC, BIC, or another criterion, any stability-selection threshold, the final model, and performance on held-out data.

For feature-wise models, the compact result object is

$$\left(h^\top \hat{\beta}_j, \mathrm{SE}(h^\top \hat{\beta}_j), W_j, p_j, p_j^{\mathrm{adj}}\right), \qquad j = 1, \ldots, m. \tag{7.70}$$

For prediction models, the compact result object is instead

$$\left(\hat{f}, \hat{\lambda}, \widehat{R}_{\mathrm{CV}}(\hat{\lambda}), \widehat{R}_{\mathrm{test}}(\hat{f})\right), \tag{7.71}$$

where $\widehat{R}_{\mathrm{test}}$ is computed on data not used to fit or tune the model. Regression for inference and regression for prediction can use similar formulas, but they answer different scientific questions.

Summary

Regression is the common language behind covariate adjustment, differential expression, contrast testing, and supervised prediction. In feature-wise models, each molecular feature is a response and Z encodes the design. In supervised learning, rows of X are predictor vectors and the outcome is a phenotype, trait, or class label. Linear models are appropriate for transformed continuous responses. Negative-binomial GLMs model counts directly and explain the role of size-factor offsets in DESeq2-like workflows. Logistic regression models binary outcomes through log-odds. High-dimensional regression requires regularization, and sparse methods trade variance reduction and interpretability against instability and tuning dependence. For compositional predictors, log-contrast regression preserves the central log-ratio principle: only ratios are interpretable.

The chapter's central message is that regression is not one method. It is a modeling grammar. The response type, design matrix, link function, likelihood, penalty, contrast, and validation scheme determine what the fitted coefficients and predictions mean.

References

Aitchison, J., and S. M. Shen. 1980. "Logistic-Normal Distributions: Some Properties and Uses." *Biometrika* 67 (2): 261. https://doi.org/10.2307/2335470.

Akaike, Hirotugu. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19 (6): 716–23. https://doi.org/10.1109/TAC.1974.1100705.

Egozcue, Juan José, and Raimon Tolosana-Delgado. 2007. *Lecture Notes on Compositional Data Analysis*. Universitat de Girona.

Exercise Sheet 05. 2026. *Exercise Sheet 05 Solutions: Hypothesis Testing, Differential Abundance, PERMANOVA, and High-Dimensional Inference*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise05_Solution.html.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12). https://doi.org/10.1186/s13059-014-0550-8.

Meinshausen, Nicolai, and Peter Bühlmann. 2010. "Stability Selection." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72 (4): 417–73. https://doi.org/10.1111/j.1467-9868.2010.00740.x.

Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6 (2): 461–64. https://doi.org/10.1214/aos/1176344136.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58 (1): 267–88. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

8 Network Learning

Chapter Aim

Chapter 08 is about learning a network from high-dimensional biological data. In Chapter 09, networks are treated as given objects for downstream analysis. Here the graph itself is estimated from a data matrix, and every edge is therefore a statistical claim.

The rows of the network-learning input are observations and the columns are features. In a gene network, the p nodes are genes. In a metabolite network, the nodes are metabolites. In a microbiome association network, the nodes may be taxa or sequence variants. We write the graph as

$$G = (V, E), \qquad V = \{1, \dots, p\}. \tag{8.1}$$

The adjacency matrix is

$$A \in \{0,1\}^{p \times p}, \qquad A_{j\ell} = 1 \quad \Longleftrightarrow \quad (j, \ell) \in E. \tag{8.2}$$

For the undirected graphs considered in this chapter,

$$A = A^\top, \qquad A_{jj} = 0. \tag{8.3}$$

The main question is how to estimate A from X. The key message is that simple pairwise correlations do not separate direct from indirect associations. Sparse regression and sparse inverse covariance estimation try to estimate conditional dependence structure instead. Node-wise lasso estimates each node from all other nodes. Graphical lasso estimates the whole precision matrix at once. StARS then gives a stability-based way to choose the regularization level. Methods such as SPIEC-EASI combine compositional preprocessing with sparse graphical-model estimation for microbiome data (Kurtz et al. 2015).

> **Notation convention**
>
> Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked

whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external references

For graph language and graph encodings, see MSMB Networks and Trees, especially the graph foundations (MSMB Networks Graphs). The statistical network-learning methods in this chapter are based mainly on node-wise lasso for high-dimensional graphs (Meinshausen and Bühlmann 2006), sparse inverse covariance estimation with graphical lasso (Friedman et al. 2008; Yuan and Lin 2007), stability selection for graphical models via StARS (Liu et al. 2010), and compositional microbial network inference via SPIEC-EASI (Kurtz et al. 2015). Practical computation of SPIEC-EASI with MB and graphical-lasso variants, StARS-style stability selection through `pulsar`, and SparCC comparison is included in Exercise Sheet 6, Task 1 (Exercise Sheet 06 2026). NetCoMi network construction and analysis is then treated as a separate follow-up task in Exercise Sheet 6, Task 2. The sparse-regression and stability-selection machinery behind node-wise lasso is prepared by the high-dimensional regression exercises in Exercise Sheet 05 (Exercise Sheet 05 2026).

Learning Goals

After this chapter, students should be able to distinguish a known biological network from an inferred statistical network, define an adjacency matrix for a feature graph, explain why correlation networks can contain indirect edges, write down the Gaussian graphical model, connect zeros in the precision matrix to conditional independence, compute partial correlations from the precision matrix, formulate node-wise lasso regression, symmetrize node-wise edge estimates, formulate the graphical lasso objective, explain how StARS selects a regularization parameter using subsampling stability, and explain why microbiome network inference needs normalization or log-ratio preprocessing before sparse graphical modeling.

8.1 What Does An Edge Mean?

A learned network is not automatically a mechanistic interaction map. An edge may mean correlation, conditional dependence, predictive association, co-occurrence after preprocessing, or agreement with a chosen model. These meanings are different.

For feature vectors $x_{\cdot j}$ and $x_{\cdot \ell}$, a pairwise association network might define an edge by thresholding a sample correlation:

$$A^{\mathrm{cor}}_{j\ell} = \mathbf{1}\left\{|\hat{\rho}_{j\ell}| \geq \tau\right\}, \qquad j \neq \ell. \tag{8.4}$$

This is easy to compute, but a correlation edge can be indirect. If gene 1 regulates gene 2 and gene 2 regulates gene 3, then genes 1 and 3 may be correlated even if there is no direct statistical dependence after conditioning on gene 2. The purpose of graphical models is to move from marginal association toward conditional association.

The preprocessing step matters before any network is learned. Raw count data

$$C \in \mathbb{N}_0^{n \times p} \tag{8.5}$$

are usually not appropriate inputs for Gaussian graphical modeling. One typically first constructs a transformed matrix X, for example using variance-stabilized values, residuals, or log-ratio coordinates:

$$C \longmapsto X. \tag{8.6}$$

The network is only interpretable relative to this map. A graph learned from CLR-transformed compositions is a graph in log-ratio geometry. A graph learned from residuals is a graph after removing the fitted mean structure. A graph learned from raw proportions is usually hard to interpret because sequencing depth and compositional constraints can induce artificial associations.

The distinction between an embedding and the underlying graph is important computationally as well. In graph-based single-cell clustering, the KNN or SNN adjacency matrix is the object being clustered; the two-dimensional layout is only a display of that object. The construction and inspection of such an SNN graph is worked through in Exercise Sheet 04 (Exercise Sheet 04 2026). The microbial-network counterpart, where the graph is an inferred association network rather than a clustering graph, is computed with SPIEC-EASI in Exercise Sheet 6, Task 1 (Exercise Sheet 06 2026).

8.2 Gaussian Graphical Models

The cleanest mathematical foundation for this chapter is the Gaussian graphical model. Assume that each row of X is an independent draw from a multivariate normal distribution:

$$x_i \sim \mathcal{N}_p(\mu, \Sigma), \qquad i = 1, \dots, n. \tag{8.7}$$

Here $x_i \in \mathbb{R}^p$ is the column-vector version of row i, $\mu \in \mathbb{R}^p$ is the mean vector, and $\Sigma \in \mathbb{R}^{p \times p}$ is the covariance matrix. The precision matrix is

$$\Omega = \Sigma^{-1}. \tag{8.8}$$

In a Gaussian graphical model, missing edges correspond to zeros in the precision matrix:

$$A_{j\ell} = 0 \quad \iff \quad \Omega_{j\ell} = 0, \qquad j \neq \ell. \tag{8.9}$$

The statistical meaning is conditional independence:

$$\Omega_{j\ell} = 0 \quad \iff \quad X_j \perp X_\ell \mid X_{-(j,\ell)}. \tag{8.10}$$

Here X_j and X_ℓ are random feature coordinates, and $X_{-(j,\ell)}$ denotes all remaining coordinates. This is why precision-based graphs are often more useful than correlation graphs: the edge asks whether two features remain associated after adjusting for the rest of the measured system.

The partial correlation between features j and ℓ is computed from the precision matrix as

$$\rho_{j\ell|-(j,\ell)} = -\frac{\Omega_{j\ell}}{\sqrt{\Omega_{jj}\Omega_{\ell\ell}}}. \tag{8.11}$$

Thus estimating a Gaussian graphical model can be seen either as estimating a sparse precision matrix or as estimating a sparse partial-correlation network.

8.3 Why Sparsity Is Needed

The sample covariance matrix is

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top, \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{8.12}$$

If $p > n$, then $\hat{\Sigma}$ is singular:

$$\operatorname{rank}(\hat{\Sigma}) \leq n - 1 < p. \tag{8.13}$$

The inverse covariance $\hat{\Sigma}^{-1}$ does not exist. Even when $p < n$, the inverse can be unstable if p is close to n or if features are strongly correlated. Network learning therefore needs regularization. The biological assumption is often that the conditional-dependence graph is sparse: each feature is directly connected to only a small subset of all possible features.

The number of possible undirected edges is

$$\binom{p}{2} = \frac{p(p-1)}{2}. \tag{8.14}$$

Without sparsity, a high-dimensional network can quickly become uninterpretable. With too much sparsity, real dependencies are missed. Network learning is therefore a model-selection problem over the regularization level.

8.4 Node-Wise Lasso Regression

The first sparse route is neighborhood selection by node-wise lasso regression, following the high-dimensional graph-estimation approach of Meinshausen and Bühlmann (Meinshausen and Bühlmann 2006). For node j, use feature j as a response and all remaining features as predictors:

$$x_{\cdot j} = X_{-j}\beta^{(j)} + \varepsilon^{(j)}. \tag{8.15}$$

Here $X_{-j} \in \mathbb{R}^{n\times(p-1)}$ is the analysis matrix with column j removed, and $\beta^{(j)} \in \mathbb{R}^{p-1}$ contains the regression coefficients for predicting node j. Node-wise lasso estimates

$$\widehat{\beta}^{(j)}_{\lambda} = \arg\min_{\beta\in\mathbb{R}^{p-1}} \left\{ \frac{1}{2n} \left\| x_{\cdot j} - X_{-j}\beta \right\|_2^2 + \lambda \|\beta\|_1 \right\}. \tag{8.16}$$

This is the lasso from Chapter 07, now used locally at each node. The estimated neighborhood of node j is

$$\widehat{N}_{\lambda}(j) = \left\{ \ell \neq j : \widehat{\beta}^{(j)}_{\lambda,\ell} \neq 0 \right\}. \tag{8.17}$$

Because the regression of j on ℓ and the regression of ℓ on j are fit separately, the two directions can disagree. An undirected graph requires symmetrization. The OR rule defines an edge if either regression selects it:

$$\widehat{A}^{\mathrm{OR}}_{j\ell}(\lambda) = \mathbf{1}\left\{ \ell \in \widehat{N}_{\lambda}(j) \text{ or } j \in \widehat{N}_{\lambda}(\ell) \right\}. \tag{8.18}$$

-nodewise-or-rule} \end{equation}

The AND rule defines an edge only if both regressions select it:

$$\widehat{A}^{\mathrm{AND}}_{j\ell}(\lambda) = \mathbf{1}\left\{ \ell \in \widehat{N}_{\lambda}(j) \text{ and } j \in \widehat{N}_{\lambda}(\ell) \right\}. \tag{8.19}$$

-nodewise-and-rule} \end{equation}

The OR rule is more sensitive and usually gives denser graphs. The AND rule is more conservative. The node-wise lasso approach is attractive because it decomposes a large graph-learning problem into p familiar regression problems (Meinshausen and Bühlmann 2006). Its

limitation is that the global object is assembled from local fits; it does not directly estimate one positive-definite precision matrix.

Computationally, each node-wise fit is an ordinary high-dimensional lasso problem with a different response column. The regularization-path and cross-validation mechanics are therefore the same as in sparse regression, while the stability question is now asked for edges rather than for a phenotype-prediction model; the relevant sparse-regression and stability-selection computations are practiced in Exercise Sheet 05, and their network-learning version is applied directly through SPIEC-EASI MB and graphical-lasso fits in Exercise Sheet 6, Task 1 (Exercise Sheet 05 2026; Exercise Sheet 06 2026).

8.5 From Node-Wise Regression To Precision Matrices

The connection between node-wise regression and precision matrices is exact under a Gaussian model. Partition the random vector into coordinate j and all other coordinates. The conditional mean of X_j given X_{-j} is linear:

$$\mathbb{E}(X_j \mid X_{-j}) = \mu_j + \Sigma_{j,-j}\Sigma_{-j,-j}^{-1}(X_{-j} - \mu_{-j}). \tag{8.20}$$

Equivalently, in terms of the precision matrix,

$$\mathbb{E}(X_j \mid X_{-j}) = \mu_j - \frac{1}{\Omega_{jj}} \sum_{\ell \neq j} \Omega_{j\ell}(X_\ell - \mu_\ell). \tag{8.21}$$

Thus the population regression coefficient of X_ℓ in the regression of X_j on all other features is

$$\beta_\ell^{(j)} = -\frac{\Omega_{j\ell}}{\Omega_{jj}}, \qquad \ell \neq j. \tag{8.22}$$

Therefore

$$\beta_\ell^{(j)} = 0 \quad \Longleftrightarrow \quad \Omega_{j\ell} = 0. \tag{8.23}$$

This is why node-wise variable selection can recover a conditional-dependence graph. The lasso is being used not because the response is a phenotype, but because selecting predictors in each node-wise regression selects candidate neighbors.

8.6 Graphical Lasso

The second sparse route estimates the precision matrix directly. If the data are centered, the Gaussian negative log-likelihood for a precision matrix Ω is, up to constants,

$$\operatorname{tr}(\hat{\Sigma}\Omega) - \log\det(\Omega), \qquad \Omega \succ 0. \tag{8.24}$$

The graphical lasso adds an ℓ_1 penalty to encourage off-diagonal sparsity:

$$\hat{\Omega}_\lambda = \arg\min_{\Omega \succ 0} \left\{ \operatorname{tr}(\hat{\Sigma}\Omega) - \log\det(\Omega) + \lambda \sum_{j \neq \ell} |\Omega_{j\ell}| \right\}. \tag{8.25}$$

Some implementations penalize all entries, and some penalize only off-diagonal entries. The statistical target for graph structure is the off-diagonal pattern. Once $\hat{\Omega}_\lambda$ has been estimated, the graph is

$$\hat{A}_{j\ell}(\lambda) = \mathbf{1}\left\{\hat{\Omega}_{\lambda,j\ell} \neq 0\right\}, \qquad j \neq \ell. \tag{8.26}$$

The fitted partial correlations are

$$\hat{\rho}_{j\ell|-(j,\ell)} = -\frac{\hat{\Omega}_{\lambda,j\ell}}{\sqrt{\hat{\Omega}_{\lambda,jj}\hat{\Omega}_{\lambda,\ell\ell}}}. \tag{8.27}$$

Graphical lasso estimates one coherent precision matrix and therefore one coherent Gaussian graphical model (Friedman et al. 2008; Yuan and Lin 2007). The regularization parameter λ controls sparsity. Large λ gives a sparse graph; small λ gives a dense graph. The difficult part is choosing λ in a way that does not merely produce a visually pleasing network.

8.7 Model Selection With StARS

StARS, the Stability Approach to Regularization Selection, chooses the regularization level by asking whether the selected edges are stable under subsampling (Liu et al. 2010). Let $\lambda \in \Lambda$ be a candidate regularization value. For subsample $b = 1, \ldots, B$, fit the graph-learning method on a subset of the observations and obtain an adjacency matrix

$$\hat{A}^{(b)}(\lambda) \in \{0,1\}^{p \times p}. \tag{8.28}$$

For edge (j, ℓ), the estimated selection probability is

$$\hat{\pi}_{j\ell}(\lambda) = \frac{1}{B}\sum_{b=1}^{B} \hat{A}^{(b)}_{j\ell}(\lambda). \tag{8.29}$$

An edge is unstable if it is selected in some subsamples but not others. The Bernoulli variability of edge (j, ℓ) is

$$\hat{\xi}_{j\ell}(\lambda) = 2\hat{\pi}_{j\ell}(\lambda)\left[1 - \hat{\pi}_{j\ell}(\lambda)\right]. \tag{8.30}$$

This quantity is near zero if the edge is almost always absent or almost always present, and it is largest when $\hat{\pi}_{j\ell}(\lambda)$ is near $1/2$. The average instability over all possible undirected edges is

$$\hat{D}(\lambda) = \frac{2}{p(p-1)} \sum_{1 \le j < \ell \le p} \hat{\xi}_{j\ell}(\lambda). \tag{8.31}$$

Given a threshold τ_{StARS}, often chosen as a small value such as 0.05, StARS selects a regularization level whose estimated graph instability is controlled:

$$\hat{D}(\hat{\lambda}) \le \tau_{\mathrm{StARS}}. \tag{8.32}$$

In practice, one evaluates a regularization path and chooses a graph that is as rich as possible while keeping instability below the threshold. StARS is not a proof that the selected edges are true biological interactions. It is a model-selection criterion: it asks whether the graph is reproducible under perturbations of the sample.

8.8 Compositional Data And SPIEC-EASI

Microbiome network learning is harder because sequencing counts are sparse, overdispersed, and compositional. If c_i is the count vector for observation i and

$$N_i = \sum_{j=1}^{p} c_{ij}, \tag{8.33}$$

then the observed proportions

$$\hat{\pi}_{ij} = \frac{c_{ij}}{N_i} \tag{8.34}$$

live on the simplex. Increasing the observed proportion of one taxon forces at least one other proportion to decrease. This can induce negative correlations even when the underlying absolute

abundances are independent. A naive correlation network on proportions is therefore not a reliable ecological interaction network (Friedman and Alm 2012).

Log-ratio transformations address the compositional scale. For positive compositions, the centered log-ratio transform is

$$\operatorname{clr}(\pi_i)_j = \log \pi_{ij} - \frac{1}{p} \sum_{\ell=1}^{p} \log \pi_{i\ell}. \tag{8.35}$$

Zeros must be handled before this transformation is applied. SPIEC-EASI combines compositional preprocessing ideas with sparse inverse covariance estimation. At a high level, its workflow is

$$C \longmapsto \tilde{X} \longmapsto \hat{A}_\lambda, \tag{8.36}$$

where $\tilde{X}$ is a transformed representation suitable for sparse graphical modeling, and $\hat{A}_\lambda$ is learned using either neighborhood selection or graphical lasso-type estimation. The regularization parameter is selected using stability ideas such as StARS. SPIEC-EASI combines normalization or compositional transformation with sparse graph estimation because the graph is only meaningful on the scale on which conditional associations are modeled (Kurtz et al. 2015).

The complete SPIEC-EASI workflow with American Gut data, MB and GLASSO options, StARS-style tuning through `pulsar`, SparCC comparison, graph plotting, and degree distributions is worked through in Exercise Sheet 6, Task 1 (Exercise Sheet 06 2026).

8.9 What To Report

A learned network should be reported as a statistical estimate, not as a discovered wiring diagram. At minimum, one should state the input data object, filtering rules, normalization or transformation, node definition, graph-learning method, regularization path, model-selection criterion, final number of nodes and edges, and any stability diagnostics.

A compact learned-network result can be written as

$$\left(\hat{A}_{\hat{\lambda}}, \hat{\lambda}, \hat{D}(\hat{\lambda}), |V|, |E|\right). \tag{8.37}$$

For Gaussian graphical models, it is also useful to report the estimated precision matrix or partial correlations:

$$\left(\hat{\Omega}_{\hat{\lambda}}, \hat{P}_{\hat{\lambda}}, \hat{A}_{\hat{\lambda}}\right), \tag{8.38}$$

where $\hat{P}_{\hat{\lambda}}$ contains partial correlations derived from Equation 8.27. Biological interpretation should be cautious: an edge is a conditional association under the chosen model and preprocessing, not automatically a direct physical interaction or causal effect.

Summary

Network learning turns a feature matrix into an estimated graph. Correlation networks are simple but often confound direct and indirect associations. Gaussian graphical models define edges through conditional dependence, represented by nonzero off-diagonal entries of the precision matrix. Node-wise lasso estimates each node's neighborhood by sparse regression. Graphical lasso estimates the whole sparse precision matrix directly. StARS chooses the regularization level by measuring graph instability under subsampling. For microbiome data and other compositional measurements, methods such as SPIEC-EASI combine normalization or log-ratio preprocessing with sparse graphical-model estimation.

The chapter's central message is that a learned network is a model-dependent statistical object. The edge set depends on preprocessing, node definition, distributional assumptions, regularization, and model selection. This is precisely why network learning must be separated from network analysis: before computing centralities or modules, we need to know what the edges mean and how they were estimated.

References

Exercise Sheet 04. 2026. *Exercise Sheet 04 Solutions: Hierarchical Clustering and Graph-Based Clustering of High-Dimensional Biological Data*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise04_Solution.html.

Exercise Sheet 05. 2026. *Exercise Sheet 05 Solutions: Hypothesis Testing, Differential Abundance, PERMANOVA, and High-Dimensional Inference*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise05_Solution.html.

Exercise Sheet 06. 2026. *Exercise Sheet 06 Solutions: Task 1 Sparse Microbial Network Inference with SPIEC-EASI, Task 2 Network Analysis with NetCoMi, and Deep Count Autoencoders*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise06_Solution.html.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2008. "Sparse Inverse Covariance Estimation with the Graphical Lasso." *Biostatistics* 9 (3): 432–41. https://doi.org/10.1093/biostatistics/kxm045.

Friedman, Jonathan, and Eric J. Alm. 2012. "Inferring Correlation Networks from Genomic Survey Data." *PLoS Computational Biology* 8 (9): e1002687. https://doi.org/10.1371/journal.pcbi.1002687.

Kurtz, Zachary D., Christian L. Müller, Emily R. Miraldi, Dan R. Littman, Martin J. Blaser, and Richard A. Bonneau. 2015. "Sparse and Compositionally Robust Inference of Microbial Ecological Networks." *PLOS Computational Biology* 11 (5): e1004226. https://doi.org/10.1371/journal.pcbi.1004226.

Liu, Han, Kathryn Roeder, and Larry Wasserman. 2010. "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models." *Advances in Neural Information Processing Systems* 23: 1432–40.

Meinshausen, Nicolai, and Peter Bühlmann. 2006. "High-Dimensional Graphs and Variable Selection with the Lasso." *The Annals of Statistics* 34 (3): 1436–62. https://doi.org/10.1214/009053606000000281.

Yuan, M., and Y. Lin. 2007. "Model Selection and Estimation in the Gaussian Graphical Model." *Biometrika* 94 (1): 19–35. https://doi.org/10.1093/biomet/asm018.

9 Network Analysis

Chapter Aim

Chapter 08 treated a network as an estimated statistical object: an adjacency matrix, a partial-correlation matrix, or a sparse precision matrix was learned from data. Chapter 09 starts after this estimation step. The question is no longer only whether an edge should be present. The question is what can be learned from the graph once its nodes and edges have been defined.

This distinction is essential. Network analysis can summarize a graph by degrees, paths, modules, centralities, robustness curves, or comparisons to reference graph models. None of these quantities has a biological meaning by itself. A high-degree taxon is only a high-degree taxon relative to the preprocessing, association measure, sparsification rule, and node definition used to construct the graph. A module is only a module in the graph that was actually analyzed. Chapter 09 therefore keeps the construction map visible while introducing the mathematical summaries used to characterize networks.

Notation convention

Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external references

For graph terminology, graph encodings, graphs constructed from data, and biological network workflows, see MSMB Networks and Trees, especially the sections on graph encodings (MSMB Networks Graph Encoding), graphs from data (MSMB Networks Graphs From Data), gene set enrichment as a bridge to network thinking (MSMB Networks Gene Sets), and high-scoring subgraphs (MSMB Networks Significant Subgraphs). The transition from association estimation to adjacency matrices follows the modular network-construction view used in NetCoMi (Peschel et al. 2021), with microbial network computation and comparison practiced in Exercise Sheet 6, Task 2 (Exercise Sheet 06 2026). General network summaries and null-model thinking are connected to small-world networks (Watts and Strogatz 1998; Humphries et al. 2006), modularity and communities

(Girvan and Newman 2002; Newman 2006), graphlet-based graph comparison (Hočevar and Demšar 2014; Yaveroğlu et al. 2014), and robustness under node removal (Albert et al. 2000; Iyer et al. 2013).

Learning Goals

After this chapter, students should be able to describe the map from a learned association or precision matrix to a graph, distinguish signed, unsigned, weighted, unweighted, and dissimilarity-based networks, compute degree, density, path length, diameter, and clustering coefficients, explain the difference between degree, closeness, betweenness, and eigenvector centrality, define modularity for a proposed community assignment, compare network summaries to null models, describe graphlet-based graph comparison at a high level, and interpret robustness curves under random and targeted node removal.

9.1 From Learned Matrices To Networks

Network analysis begins with a graph

$$G = (V, E), \qquad V = \{1, \dots, p\}, \tag{9.1}$$

where the nodes are features: genes, proteins, metabolites, taxa, peaks, or other molecular variables. In this chapter, p is the number of nodes, inherited from the number of columns of the data matrix. For an undirected unweighted graph, the adjacency matrix is

$$A \in \{0,1\}^{p \times p}, \qquad A = A^\top, \qquad A_{jj} = 0. \tag{9.2}$$

The set of edges is

$$E = \left\{ (j, \ell) : 1 \le j < \ell \le p, A_{j\ell} = 1 \right\}. \tag{9.3}$$

The number of edges is denoted by

$$q = |E| = \sum_{1 \le j < \ell \le p} A_{j\ell}. \tag{9.4}$$

Chapter 08 often produced more than a binary graph. A sparse Gaussian graphical model may produce an estimated precision matrix $\hat{\Omega}$ and a partial-correlation matrix $\hat{P}$ with entries

$$\hat{P}_{j\ell} = -\frac{\hat{\Omega}_{j\ell}}{\sqrt{\hat{\Omega}_{jj}\hat{\Omega}_{\ell\ell}}}, \qquad j \neq \ell. \tag{9.5}$$

This matrix is an association matrix. It can be analyzed directly as a weighted signed graph, or it can be transformed into an unweighted adjacency matrix. More generally, let

$$R = (r_{j\ell}) \in [-1, 1]^{p \times p} \tag{9.6}$$

denote a symmetric association matrix. The entries may be correlations, SparCC associations, proportionality scores transformed to an association scale, partial correlations, or conditional-dependence scores. The important point is that the mathematical object used for network analysis is not uniquely determined by R. One must still decide which graph to construct.

A simple unweighted signed-threshold graph uses

$$A^{\text{thr}}_{j\ell} = \mathbf{1}\left\{|r_{j\ell}| \geq \tau\right\}, \qquad j \neq \ell. \tag{9.7}$$

The corresponding weighted graph can keep the association values on selected edges:

$$W^{\text{thr}}_{j\ell} = r_{j\ell}\mathbf{1}\left\{|r_{j\ell}| \geq \tau\right\}. \tag{9.8}$$

If the sign matters, positive and negative edge sets can be recorded separately:

$$E^{+} = \left\{(j, \ell) : r_{j\ell} \geq \tau\right\}, \qquad E^{-} = \left\{(j, \ell) : r_{j\ell} \leq -\tau\right\}. \tag{9.9}$$

Some graph summaries, especially shortest-path summaries, require nonnegative dissimilarities rather than signed associations. Two common transformations emphasize different biological questions. An unsigned dissimilarity treats strong positive and strong negative associations as close:

$$d^{\text{unsigned}}_{j\ell} = 1 - r^2_{j\ell}. \tag{9.10}$$

A signed dissimilarity treats positive association as close and negative association as far:

$$d^{\text{signed}}_{j\ell} = \frac{1 - r_{j\ell}}{2}. \tag{9.11}$$

Thus the same partial-correlation matrix can give several network objects: a signed weighted association network, an unsigned co-association network, a binary threshold graph, or a

dissimilarity graph used for path-based summaries. NetCoMi makes this modular distinction explicit: preprocessing, zero handling, normalization, association estimation, sparsification, adjacency construction, network analysis, visualization, and comparison are separate choices in one workflow (Peschel et al. 2021). The companion computation is the NetCoMi exercise in Exercise Sheet 6, Task 2, where microbial association networks are constructed, analyzed, visualized, and compared. This is why a network result should report not only A, but also the matrix from which it was derived.

A compact analysis object is therefore

$$\mathcal{G} = (V, R, D, W, A)\,, \tag{9.12}$$

where R is an association matrix, D is a dissimilarity matrix if path lengths require one, W is a weighted adjacency matrix, and A is the binary adjacency matrix used for unweighted summaries. In practice not all components are needed, but the analysis should state which ones are used.

9.2 Node-Level Summaries

The simplest node-level property is degree. For an undirected unweighted graph, the degree of node j is

$$k_j = \sum_{\ell=1}^{p} A_{j\ell}. \tag{9.13}$$

The degree distribution is the empirical distribution of the k_j values:

$$\hat{F}_k(t) = \frac{1}{p} \sum_{j=1}^{p} \mathbf{1}\left\{k_j \le t\right\}. \tag{9.14}$$

Degree is often the first summary plotted because it immediately distinguishes dense, hub-dominated, banded, random-like, and modular graphs. In a microbial association network, a high-degree taxon is sometimes called a hub. This term should be used carefully. A hub is a graph-theoretic statement, not automatically a keystone species or causal driver. A taxon can have high degree because it is broadly associated, because it is abundant and easy to estimate, because the sparsification threshold is permissive, or because it sits in a compositional structure that induces many associations.

For a weighted graph with nonnegative edge weights $w_{j\ell} \ge 0$, the analogous node summary is strength:

$$s_j = \sum_{\ell=1}^{p} w_{j\ell}. \tag{9.15}$$

If W contains signed associations, it is usually better to separate positive and negative strength:

$$s_j^+ = \sum_{\ell=1}^{p} \max(w_{j\ell}, 0), \qquad s_j^- = \sum_{\ell=1}^{p} \max(-w_{j\ell}, 0). \tag{9.16}$$

This separation is useful because positive and negative microbial associations often have different interpretations. Positive associations can reflect shared habitat, shared response to unmeasured covariates, cross-feeding, or co-colonization. Negative associations can reflect exclusion, different environmental preferences, competition, or the closure effect of compositions. Network analysis does not decide between these mechanisms; it summarizes the graph produced by the chosen statistical construction.

9.3 Global Network Size And Density

The edge density of an undirected graph is the fraction of possible edges that are present:

$$\delta(G) = \frac{2q}{p(p-1)}. \tag{9.17}$$

Density is deceptively important. Many network summaries change strongly with density. A graph with more edges will usually have shorter paths, higher degrees, larger connected components, and often higher clustering. Therefore comparisons between biological groups should either control for the construction rule or compare against a null model that preserves relevant baseline quantities such as p, q, or the degree sequence.

The connected components of G are maximal node sets in which every pair of nodes is joined by some path. If $\mathcal{C}(G)$ is the collection of connected components and $C_{\max}$ is the largest component, then the relative largest connected component size is

$$L_{\max}(G) = \frac{|C_{\max}|}{p}. \tag{9.18}$$

This quantity is especially useful for sparse biological networks because many path-based summaries are unstable or undefined when a graph fragments into many components.

9.4 Paths, Diameter, And Clustering

A path between nodes j and ℓ is a sequence of adjacent nodes connecting them. For an unweighted graph, let

$$\operatorname{dist}_G(j,\ell) = \text{length of the shortest path from } j \text{ to } \ell \tag{9.19}$$

when j and ℓ are connected. If they are not connected, the distance is infinite. For disconnected biological graphs, one often averages only over connected pairs:

$$\bar{L}(G) = \frac{1}{|\mathcal{P}_c|} \sum_{(j,\ell)\in\mathcal{P}_c} \operatorname{dist}_G(j,\ell), \tag{9.20}$$

where

$$\mathcal{P}_c = \{(j,\ell) : 1 \le j < \ell \le p, \operatorname{dist}_G(j,\ell) < \infty\}. \tag{9.21}$$

The diameter is the longest finite shortest-path distance:

$$\operatorname{diam}(G) = \max_{(j,\ell)\in\mathcal{P}_c} \operatorname{dist}_G(j,\ell). \tag{9.22}$$

For weighted networks, shortest paths require edge lengths, not association strengths. If $s_{j\ell}$ is a similarity, an edge length might be $1/s_{j\ell}$ or a signed or unsigned dissimilarity. This choice changes path-based summaries. A strong negative association may be close in an unsigned network but far in a signed network, as in Equations 9.10 and 9.11.

The local clustering coefficient measures how close the neighbors of node j are to forming a clique. Let $N(j) = \{\ell : A_{j\ell} = 1\}$ and $k_j = |N(j)|$. The number of triangles touching node j is

$$T_j = \sum_{\ell<m} A_{j\ell} A_{jm} A_{\ell m}. \tag{9.23}$$

For $k_j \ge 2$, the local clustering coefficient is

$$C_j = \frac{2T_j}{k_j(k_j-1)}. \tag{9.24}$$

The average clustering coefficient is

$$\bar{C}(G) = \frac{1}{p}\sum_{j=1}^{p} C_j, \tag{9.25}$$

with C_j usually set to zero or treated as undefined when $k_j < 2$. High clustering means that neighbors of a node tend to be connected to each other. In biological association networks this can indicate modules, shared environmental drivers, taxonomic relatedness, pathway structure, or technical dependencies. The statistic is useful, but interpretation again depends on the graph construction.

9.5 Centrality

Centrality measures rank nodes by different graph-theoretic notions of importance. Degree centrality is local:

$$C_D(j) = \frac{k_j}{p-1}. \tag{9.26}$$

It asks how many neighbors node j has. Closeness centrality is path-based. A robust version for disconnected graphs is harmonic closeness:

$$C_H(j) = \frac{1}{p-1}\sum_{\ell \neq j} \frac{1}{\mathrm{dist}_G(j,\ell)}, \tag{9.27}$$

where $1/\infty = 0$. A node has high harmonic closeness if it is close, in shortest-path distance, to many other nodes.

Betweenness centrality asks whether a node lies on shortest paths between other nodes. Let σ_{ab} be the number of shortest paths from a to b, and let $\sigma_{ab}(j)$ be the number of those paths that pass through j. Then

$$C_B(j) = \sum_{\substack{a<b \\ a\neq j,\ b\neq j}} \frac{\sigma_{ab}(j)}{\sigma_{ab}}. \tag{9.28}$$

Betweenness is sensitive to small changes in sparse graphs because adding or removing a few edges can reroute many shortest paths. In inferred biological networks, this makes betweenness useful as a descriptive diagnostic but risky as a stand-alone claim about biological control.

Eigenvector centrality assigns high scores to nodes connected to other high-scoring nodes. For a nonnegative adjacency or weight matrix A, it is defined by

$$Au = \lambda_{\max} u, \qquad u_j \geq 0, \tag{9.29}$$

where $\lambda_{\max}$ is the largest eigenvalue. This measure is global: a node can score highly not only because it has many neighbors, but because it is connected to a central region of the graph.

Different centralities answer different questions. A node can have high degree but low betweenness if it lies inside a dense module. A node can have modest degree but high betweenness if it connects two otherwise separated modules. For microbiome networks, "keystone" language should therefore be linked to the specific centrality used, and ideally to robustness analyses or independent biological validation.

9.6 Communities And Modularity

Many biological graphs appear modular: nodes form groups with many within-group edges and fewer between-group edges. Let $g_j \in \{1, \ldots, K\}$ denote the community label assigned to node j. The modularity of this partition is

$$Q(g) = \frac{1}{2q} \sum_{j=1}^{p} \sum_{\ell=1}^{p} \left[A_{j\ell} - \frac{k_j k_\ell}{2q} \right] \mathbf{1}\left\{ g_j = g_\ell \right\}. \tag{9.30}$$

The term $A_{j\ell}$ counts observed within-community edges. The term $k_j k_\ell/(2q)$ is the expected number of such edges under a degree-preserving random-graph baseline. Modularity is therefore not merely a count of within-community edges; it asks whether there are more within-community edges than expected from the degrees.

Community detection methods often search for a partition with large modularity, but the result is not unique in general. Different algorithms can find different partitions with similar modularity. Biological interpretation should therefore focus on stable and interpretable modules: groups enriched for pathways, taxonomic groups, environmental niches, protein complexes, or coherent covariate associations. A complementary perspective is given by high-scoring subgraphs: when nodes carry scores from a statistical analysis, one can search for connected regions of a known network that concentrate signal (MSMB Networks Significant Subgraphs).

9.7 Small-World Structure And Null Models

Network summaries become more meaningful when compared to a reference model. Small-world analysis is a classic example. Informally, a small-world network has short path lengths, like a random graph, but high clustering, like a locally organized graph (Watts and Strogatz 1998). Let $\bar{C}(G)$ be the average clustering coefficient and $\bar{L}(G)$ the average shortest-path length. Let

$\bar{C}_0$ and $\bar{L}_0$ be corresponding averages under a null model. A common small-world coefficient is

$$\sigma_{\mathrm{SW}}(G) = \frac{\bar{C}(G)/\bar{C}_0}{\bar{L}(G)/\bar{L}_0}. \tag{9.31}$$

The null model matters. An Erdos-Renyi graph with the same number of nodes and edges preserves only coarse density. A configuration-model graph preserves the degree sequence more closely. If the observed graph has hubs, then comparing to a null model that ignores hubs can overstate how surprising its clustering or path length is. This is the broader lesson: a network property is not simply large or small; it is large or small relative to a reference distribution.

9.8 Generative Null Models For Networks

A generative graph model is a probability distribution on graphs. In network analysis, such models are often used as null models: they define what network structure would look like if only a limited set of graph properties were preserved. Let

$$G_0 \sim \mathcal{M}_0(\theta_0) \tag{9.32}$$

denote a graph drawn from a null model $\mathcal{M}_0$ with parameter θ_0. The model parameter should usually be estimated or chosen to match a property of the observed graph G, such as p, q, density, mean degree, or the degree sequence.

9.8.1 Erdos-Renyi Graphs

The Erdos-Renyi model is the simplest density-preserving random graph model. In the $G(p, \pi)$ version, the node set is fixed as $V = \{1, \dots, p\}$ and each possible undirected edge is included independently:

$$A_{j\ell} \sim \text{Bernoulli}(\pi), \qquad 1 \le j < \ell \le p. \tag{9.33}$$

The expected number of edges is

$$\mathbb{E}(|E|) = \binom{p}{2}\pi. \tag{9.34}$$

To match an observed graph with q edges, choose

$$\hat{\pi} = \frac{2q}{p(p-1)} = \delta(G). \tag{9.35}$$

The procedure is: fix the observed number of nodes p, estimate $\hat{\pi}$ from the observed density, sample independent Bernoulli edges for all $j < \ell$, symmetrize the adjacency matrix, and set the diagonal to zero. This null model preserves only the expected density. It does not preserve hubs, clustering, geometry, or community structure.

The $G(p, q)$ version preserves the number of edges exactly rather than in expectation. It samples uniformly from all graphs with p nodes and q edges:

$$\Pr(G_0 = g) = \binom{\binom{p}{2}}{q}^{-1} \tag{9.36}$$

for every graph g with exactly q undirected edges. The procedure is: list all $\binom{p}{2}$ possible edges, sample q of them uniformly without replacement, and form the adjacency matrix.

9.8.2 Random Geometric Graphs In Two Or Three Dimensions

A random geometric graph creates edges because nodes are close in an unobserved or observed geometric space. This is useful as a reference when locality matters, for example spatial proximity, habitat gradients, tissue neighborhoods, or latent ecological niches.

For dimension $d \in \{2, 3\}$, sample node positions

$$u_j \sim \text{Uniform}([0, 1]^d), \qquad j = 1, \dots, p. \tag{9.37}$$

Given a radius $r > 0$, connect nodes whose Euclidean distance is at most r:

$$A_{j\ell} = \mathbf{1}\left\{\|u_j - u_\ell\|_2 \le r\right\}, \qquad j \neq \ell. \tag{9.38}$$

For a two-dimensional random geometric graph, $d = 2$ and $u_j = (u_{j1}, u_{j2})^\top$. For a three-dimensional random geometric graph, $d = 3$ and $u_j = (u_{j1}, u_{j2}, u_{j3})^\top$. The procedure is: choose the dimension d, sample p positions in the unit square or cube, choose r, and connect all pairs within radius r.

To match the observed graph density, choose r so that the simulated expected edge density is close to $\delta(G)$. In practice this can be done by simulation or by solving

$$\mathbb{E}_{r,d}\left[\delta(G_0)\right] \approx \delta(G). \tag{9.39}$$

Random geometric graphs usually have high clustering because "nearby" is transitive: if node j is close to both ℓ and m, then ℓ and m are often close to each other. They are therefore useful null models when one wants to ask whether observed clustering is more than what would be expected from spatial or latent-space proximity alone.

9.8.3 Watts-Strogatz Small-World Graphs

The Watts-Strogatz construction, associated with Duncan Watts and Steven Strogatz, interpolates between a regular lattice and a random graph (Watts and Strogatz 1998). It is designed to produce graphs with high clustering and short path lengths.

Start with p nodes on a cycle. Choose an even integer K with $2 \leq K < p$. Each node is connected to $K/2$ nearest neighbors on each side:

$$A_{j\ell}^{(0)} = \mathbf{1}\left\{1 \leq \mathrm{dist}_{\mathrm{cycle}}(j,\ell) \leq \frac{K}{2}\right\}. \tag{9.40}$$

Here $\mathrm{dist}_{\mathrm{cycle}}(j,\ell)$ is the shortest circular distance between nodes j and ℓ. The initial graph has degree K at every node and strong local clustering.

Then rewire edges with probability ρ_{rewire}. For each edge (j,ℓ) in a systematic traversal, with probability ρ_{rewire} remove that edge and replace it by a new edge (j,m), where m is sampled uniformly from nodes not equal to j and not already adjacent to j:

$$(j,\ell) \longmapsto (j,m) \quad \text{with probability } \rho_{\mathrm{rewire}}. \tag{9.41}$$

The procedure is: construct the regular ring lattice, traverse its edges once, independently decide whether to rewire each edge, and avoid self-loops and duplicate edges. When $\rho_{\mathrm{rewire}} = 0$, the graph is a regular lattice. When ρ_{rewire} is large, the graph becomes more random. For small positive ρ_{rewire}, a few long-range shortcuts can sharply reduce path lengths while leaving much of the local clustering intact.

9.8.4 Scale-Free Graphs By Preferential Attachment

Scale-free graph models are designed to generate heavy-tailed degree distributions. A standard preferential-attachment construction grows the network one node at a time. Begin with a small connected graph G_{m_0} on m_0 nodes. At time $t = m_0 + 1, \ldots, p$, add a new node t and attach it to m existing nodes, where $m \leq m_0$. Existing node j is selected with probability proportional to its current degree:

$$\Pr(t \text{ connects to } j \mid G_{t-1}) = \frac{k_j(t-1)}{\sum_{\ell=1}^{t-1} k_\ell(t-1)}. \tag{9.42}$$

The procedure is: choose m_0 and m, initialize a connected seed graph, repeatedly add a new node, sample m distinct existing targets using degree-proportional probabilities, and add the corresponding edges. Nodes that already have high degree are more likely to receive new edges, producing a "rich get richer" mechanism.

A scale-free degree distribution is often summarized by a power law,

$$\Pr(K = k) \propto k^{-\alpha}, \qquad k \geq k_{\min}, \tag{9.43}$$

for some exponent $\alpha > 1$. In practice, a heavy-tailed degree distribution should not be asserted from a log-log plot alone. Biological networks are finite, inferred with error, and strongly affected by filtering and sparsification. Preferential-attachment graphs are best used as a mechanistic reference for hub-dominated structure, not as proof that the biological system literally grew by preferential attachment.

9.8.5 Sticky And Configuration Models

The configuration model is a degree-sequence-preserving null model. It starts from a target degree sequence

$$k_1, \ldots, k_p, \qquad \sum_{j=1}^{p} k_j = 2q. \tag{9.44}$$

The stub-matching construction gives node j exactly k_j half-edges, or stubs. All $2q$ stubs are then randomly paired to form edges. The procedure is: create k_j stubs for each node, sample a uniformly random perfect matching of all stubs, collapse each matched pair into an edge, and then decide how to handle self-loops and multi-edges. If one rejects self-loops and multi-edges until a simple graph is obtained, the result is a simple graph with the target degree sequence, although the rejection step can become inefficient for very heterogeneous degrees.

The sticky model is a related expected-degree model. Assign each node a nonnegative stickiness parameter θ_j. Conditional on these parameters, edges are sampled independently with probabilities proportional to the product of node stickiness values:

$$\Pr(A_{j\ell} = 1 \mid \theta) = \min\left\{1,\, c\theta_j\theta_\ell\right\}, \qquad j < \ell. \tag{9.45}$$

Here $c > 0$ controls the overall density. If θ_j is chosen to be related to the observed degree k_j, then high-degree nodes in the observed graph are also likely to have high expected degree in the null graph. The expected degree of node j is

$$\mathbb{E}(K_j \mid \theta) = \sum_{\ell \neq j} \Pr(A_{j\ell} = 1 \mid \theta). \tag{9.46}$$

The procedure is: estimate or choose θ_j from node-level propensities such as degree, choose c to match the observed density or expected edge count, and sample edges independently using Equation 9.45. Compared with Erdos-Renyi graphs, sticky and configuration models preserve hub structure much more strongly. They are therefore useful when the scientific question is whether clustering, modularity, robustness, or graphlet structure exceeds what would be expected from the degree sequence alone.

9.9 Fitting A Null Model To An Observed Graph

In practice, a null model is fitted with respect to an observed graph. Let $G_{\text{obs}} = (V, E_{\text{obs}})$ be the graph obtained from the network-construction step, with adjacency matrix A_{obs}. Its basic size summaries are

$$p_{\text{obs}} = |V|, \qquad q_{\text{obs}} = |E_{\text{obs}}|. \tag{9.47}$$

The general fitting workflow is:

1. choose the graph property to preserve;
2. estimate the null-model parameter from G_{obs};
3. simulate many graphs from the fitted null model;
4. compute the same statistic on the observed and simulated graphs;
5. compare the observed statistic to the simulated reference distribution.

Mathematically, the fitted null model is

$$\hat{\theta}_0 = \hat{\theta}_0(G_{\text{obs}}), \qquad G_0^{(b)} \sim \mathcal{M}_0(\hat{\theta}_0), \qquad b = 1, \dots, B. \tag{9.48}$$

For an Erdos-Renyi $G(p, \pi)$ null, the fitted parameter is

$$\hat{\theta}_0^{\text{ER}} = (p_{\text{obs}}, \hat{\pi}), \qquad \hat{\pi} = \frac{2q_{\text{obs}}}{p_{\text{obs}}(p_{\text{obs}} - 1)}. \tag{9.49}$$

For a random geometric graph in dimension d, the fitted parameter is

$$\hat{\theta}_0^{\text{RGG}} = (p_{\text{obs}}, d, \hat{r}), \qquad \mathbb{E}_{\hat{r},d}\left[\delta(G_0)\right] \approx \delta(G_{\text{obs}}). \tag{9.50}$$

For a Watts-Strogatz model, choose K to match the observed mean degree as closely as possible,

$$\hat{K} \approx \frac{2q_{\text{obs}}}{p_{\text{obs}}}, \tag{9.51}$$

usually rounded to a feasible even integer. The rewiring parameter can then be chosen by matching a target statistic, for example the observed average path length, clustering coefficient, or small-world coefficient:

$$\hat{\rho}_{\mathrm{rewire}} = \arg\min_{\rho \in [0,1]} \left| \mathbb{E}_{\rho}\left[T(G_0)\right] - T(G_{\mathrm{obs}}) \right|. \tag{9.52}$$

Here $T(G)$ is the statistic chosen for fitting. If the model is used strictly as a null model, one should avoid fitting the same statistic that will later be used as the test statistic; otherwise the comparison is circular. For example, one may match p and q and then ask whether clustering is unusually high, or preserve the degree sequence and ask whether modularity is unusually high.

For a preferential-attachment scale-free model, the fitted parameter is often chosen to match the observed size and approximate mean degree. Since each new node contributes m edges, the mean degree is approximately $2m$, so one can set

$$\hat{m} = \mathrm{round}\left(\frac{q_{\mathrm{obs}}}{p_{\mathrm{obs}}}\right), \tag{9.53}$$

with a small seed graph size $m_0 \geq \hat{m}$. More detailed fitting can compare the observed degree distribution to simulated degree distributions, but the basic use here is as a hub-generating reference model.

For a configuration model, the fitted parameter is the observed degree sequence itself:

$$\hat{\theta}_0^{\mathrm{conf}} = \left(k_1^{\mathrm{obs}}, \dots, k_{p_{\mathrm{obs}}}^{\mathrm{obs}}\right), \qquad k_j^{\mathrm{obs}} = \sum_{\ell=1}^{p_{\mathrm{obs}}} A_{\mathrm{obs},j\ell}. \tag{9.54}$$

For a sticky expected-degree model, a simple choice is to set θ_j proportional to the observed degree and then choose c to match the expected number of edges:

$$\hat{\theta}_j = k_j^{\mathrm{obs}}, \qquad \sum_{1 \leq j < \ell \leq p_{\mathrm{obs}}} \min\left\{1,\, \hat{c}\hat{\theta}_j\hat{\theta}_\ell\right\} \approx q_{\mathrm{obs}}. \tag{9.55}$$

Once the null graphs have been generated, compute the statistic of interest on each graph:

$$T_{\mathrm{obs}} = T(G_{\mathrm{obs}}), \qquad T_b = T(G_0^{(b)}), \qquad b = 1, \dots, B. \tag{9.56}$$

The resulting values $T_1, \dots, T_B$ are the reference distribution induced by the fitted null model. For small-worldness, a typical statistic is $\sigma_{\mathrm{SW}}(G)$ from Equation 9.31, or the pair $\left(\bar{C}(G), \bar{L}(G)\right)$. In ecological-network analysis, this workflow is often used to ask whether an observed microbial or ecological graph has more clustering and shorter paths than expected under reference graphs

with the same size or density. The accompanying ecological-network example applies exactly this logic to small-worldness: compute the observed summary, generate reference networks, recompute the summary, and compare the observed value to the null distribution.

If $T(G)$ is any graph statistic and $G_0^{(1)}, \ldots, G_0^{(B)}$ are null graphs, a simulation-based upper-tail p-value can be written as

$$\hat{p}_{\text{null}} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\left\{T(G_0^{(b)}) \geq T(G)\right\}}{B+1}. \tag{9.57}$$

This equation is useful beyond small-world analysis. The same logic applies to clustering, modularity, robustness, graphlet summaries, or any other statistic where the absolute value is hard to interpret. For biological networks, the null model should preserve the aspects of the construction that are not under scientific investigation, such as node count, edge count, degree distribution, or group-specific sparsity.

9.10 Comparing Networks

Network comparison appears in at least three forms. First, one may compare two networks from two biological groups. Second, one may compare an observed network to networks generated from a null model. Third, one may compare many networks using a summary-vector embedding.

Let $G^{(1)}$ and $G^{(2)}$ be two graphs with the same node set. At the edge level, a simple difference of association matrices is

$$\Delta R = R^{(1)} - R^{(2)}. \tag{9.58}$$

A differential association network can then be defined by thresholding this difference:

$$A_{j\ell}^{\Delta} = \mathbf{1}\left\{|\Delta R_{j\ell}| \geq \tau_{\Delta}\right\}. \tag{9.59}$$

Statistical tests for differential associations require care because many pairs are tested simultaneously and because the association estimates are themselves products of normalization and sparsification choices. This is one reason NetCoMi separates network construction, network analysis, and network comparison in its workflow (Peschel et al. 2021).

At the graph-summary level, define a vector of summaries

$$h(G) = \left(\delta(G), L_{\max}(G), \bar{C}(G), \bar{L}(G), Q(G), \ldots\right)^{\top}. \tag{9.60}$$

Two networks can then be compared by a distance between summaries:

$$d_{\text{summary}}(G^{(a)}, G^{(b)}) = \left\| h(G^{(a)}) - h(G^{(b)}) \right\|_2 . \tag{9.61}$$

This heuristic is useful when many graphs are available, for example networks constructed across body sites, time points, habitats, or perturbation conditions. After computing pairwise distances between graphs, one can use the dimensionality-reduction tools from Chapter 04 to visualize relationships among graphs.

9.11 Graphlets And Graphlet Correlation Distance

Graphlet-based methods provide a richer comparison by counting small induced subgraphs. A graphlet is a small connected induced subgraph, for example a two-node edge, a three-node path, a triangle, a four-node star, or a square. In contrast to a single global statistic such as density or average clustering, graphlets describe local wiring patterns. They are useful because two networks can have similar degree distributions and still be built from different local motifs.

Let $\mathcal{H}_m$ denote a collection of connected graphlets with at most m nodes. For a graphlet $H \in \mathcal{H}_m$, let

$$N_H(G) = \#\left\{U \subseteq V : G[U] \cong H\right\} \tag{9.62}$$

be the number of induced copies of H in G. Here $G[U]$ is the subgraph induced by the node set U, and $\cong$ denotes graph isomorphism. The vector of graphlet counts is

$$g_m(G) = \left(N_H(G)\right)_{H \in \mathcal{H}_m} . \tag{9.63}$$

Raw graphlet counts depend strongly on graph size and density. Graphlet correlation methods therefore move from graph-level counts to node-level graphlet-degree vectors. Graphlets can contain different node positions, called automorphism orbits. For example, in a three-node path, the middle node and the two endpoint nodes represent different orbits. Let $\mathcal{O}_m$ be the set of graphlet orbits for graphlets with at most m nodes. For node j and orbit $o \in \mathcal{O}_m$, define

$$M_{jo}(G) = \#\left\{\text{induced graphlet copies in which node } j \text{ touches orbit } o\right\}. \tag{9.64}$$

The matrix

$$M(G) = \left(M_{jo}(G)\right)_{\substack{j=1,\dots,p \\ o \in \mathcal{O}_m}} \in \mathbb{R}^{p \times |\mathcal{O}_m|} \tag{9.65}$$

is the graphlet-degree matrix. Each column describes how often nodes participate in one local topological role. The first orbit corresponds to ordinary degree, so graphlet degrees can be viewed as a systematic extension of the degree distribution.

The graphlet correlation matrix records correlations between graphlet-orbit counts across nodes:

$$\Gamma_{oo'}(G) = \mathrm{Cor}\left(M_{\cdot o}(G), M_{\cdot o'}(G)\right), \qquad o, o' \in \mathcal{O}_m. \tag{9.66}$$

This matrix summarizes how different local roles co-occur across nodes. A network in which hub nodes also frequently sit in triangles can have a different graphlet correlation matrix from a network in which hub nodes sit in tree-like stars, even if both networks have similar density.

The graphlet correlation distance compares these graphlet correlation matrices (Yaveroğlu et al. 2014; Hočevar and Demšar 2014). If vecl extracts the lower-triangular entries of a symmetric matrix, then one common schematic form is

$$d_{\mathrm{GCD}}(G^{(a)}, G^{(b)}) = \left\| \mathrm{vecl}\left[\Gamma(G^{(a)}) - \Gamma(G^{(b)})\right] \right\|_2, \tag{9.67}$$

The distance is small when two networks have similar local wiring organization and large when their graphlet-orbit correlation structures differ. In practice, graphlet correlation distance is a graph-level comparison tool: it does not say which biological interaction is true, but it can show whether networks from different habitats, treatments, or construction methods have similar local topology. For inferred microbiome networks, it should be interpreted after checking that node definitions, sparsity levels, and network-construction rules are comparable.

9.12 Robustness And Attack Analysis

Robustness analysis asks how network connectivity changes when nodes or edges are removed. This is natural in ecological and biological settings, where one might ask whether a microbial community remains connected after losing taxa, or whether a protein interaction network is vulnerable to removing central proteins (Montoya et al. 2006; Albert et al. 2000; Iyer et al. 2013).

Let $v_1, \ldots, v_p$ be a node-removal order. The order might be random, decreasing degree, decreasing betweenness, decreasing abundance, or decreasing external biological priority. After removing the first t nodes, the remaining node set is

$$V_t = V \setminus \{v_1, \ldots, v_t\}. \tag{9.68}$$

Let G_t be the induced subgraph on V_t. A simple robustness curve is the relative largest connected component size after removal:

$$R(t) = \frac{|C_{\max}(G_t)|}{p - t}, \qquad t = 0, \dots, p - 1. \tag{9.69}$$

The area under this curve gives a one-number robustness summary:

$$\mathrm{AUR}(G) = \frac{1}{p} \sum_{t=0}^{p-1} R(t). \tag{9.70}$$

Random removal and targeted removal answer different questions. Random removal measures generic tolerance to unstructured loss. Targeted removal measures vulnerability to a specific attack strategy, such as removing high-degree or high-betweenness nodes. In microbiome applications, robustness summaries have been used to connect network structure to perturbation, diet, antibiotic exposure, and host phenotypes (Mahana et al. 2016; Tipton et al. 2018). The interpretation should remain conditional: a network robustness curve describes the inferred association graph, not a direct intervention experiment on the biological system.

9.13 What To Report

A network-analysis result should make the construction and analysis layers explicit. At minimum, report the node definition, the input data matrix, preprocessing and filtering, association measure, transformation from association to dissimilarity or adjacency, sparsification rule, whether the graph is signed or unsigned, whether weights are used, the number of nodes, the number of edges, density, connected-component structure, centrality measures, community method, null model, and robustness strategy.

A compact reporting object is

$$\mathcal{R}_{\mathrm{net}} = (R, D, W, A, h(G), \mathcal{M}, \mathcal{N}), \tag{9.71}$$

where $h(G)$ is the vector of graph summaries, $\mathcal{M}$ contains modules or community assignments, and $\mathcal{N}$ describes the null model or graph-comparison reference. This object makes clear that biological claims arise from a chain of choices. Changing the association matrix, the sign convention, the threshold, or the null model can change the final interpretation.

Summary

Network analysis characterizes a graph after nodes and edges have been defined. The bridge from Chapter 08 is the map from a learned association, partial-correlation, or precision matrix to a concrete graph object. The same matrix can produce signed, unsigned, weighted, unweighted, or dissimilarity-based networks, and each choice changes the meaning of downstream summaries.

The main graph properties are node-level summaries such as degree and strength, global summaries such as density and connected components, path-based summaries such as average shortest path and diameter, clustering coefficients, centrality measures, community structure, graph-comparison distances, and robustness curves. These quantities turn a complex graph into interpretable mathematical summaries, but their interpretation depends on the construction process. A network statistic is always a statistic of a particular graph, and in this course that graph is usually the result of a high-dimensional statistical workflow.

References

Albert, Réka, Hawoong Jeong, and Albert-László Barabási. 2000. "Error and Attack Tolerance of Complex Networks." *Nature* 406 (6794): 378–82. https://doi.org/10.1038/35019019.

Exercise Sheet 06. 2026. *Exercise Sheet 06 Solutions: Task 1 Sparse Microbial Network Inference with SPIEC-EASI, Task 2 Network Analysis with NetCoMi, and Deep Count Autoencoders*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise06_Solution.html.

Girvan, M., and M. E. J. Newman. 2002. "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences* 99 (12): 7821–26. https://doi.org/10.1073/pnas.122653799.

Hočevar, Tomaž, and Janez Demšar. 2014. "A Combinatorial Approach to Graphlet Counting." *Bioinformatics* 30 (4): 559–65. https://doi.org/10.1093/bioinformatics/btt717.

Humphries, M. D, K Gurney, and T. J Prescott. 2006. "The Brainstem Reticular Formation Is a Small-World, Not Scale-Free, Network." *Proceedings of the Royal Society B: Biological Sciences* 273 (1585): 503–11. https://doi.org/10.1098/rspb.2005.3354.

Iyer, Swami, Timothy Killingback, Bala Sundaram, and Zhen Wang. 2013. "Attack Robustness and Centrality of Complex Networks." *PLoS ONE* 8 (4): e59613. https://doi.org/10.1371/journal.pone.0059613.

Mahana, Douglas, Chad M. Trent, Zachary D. Kurtz, et al. 2016. "Antibiotic Perturbation of the Murine Gut Microbiome Enhances the Adiposity, Insulin Resistance, and Liver Disease Associated with High-Fat Diet." *Genome Medicine* 8 (1). https://doi.org/10.1186/s13073-016-0297-9.

Montoya, José M., Stuart L. Pimm, and Ricard V. Solé. 2006. "Ecological Networks and Their Fragility." *Nature* 442 (7100): 259–64. https://doi.org/10.1038/nature04927.

Newman, M. E. J. 2006. "Modularity and Community Structure in Networks." *Proceedings of the National Academy of Sciences* 103 (23): 8577–82. https://doi.org/10.1073/pnas.0601602103.

Peschel, Stefanie, Christian L Müller, Erika von Mutius, Anne-Laure Boulesteix, and Martin Depner. 2021. "NetCoMi: Network Construction and Comparison for Microbiome Data in r." *Briefings in Bioinformatics* 22 (4). https://doi.org/10.1093/bib/bbaa290.

Tipton, Laura, Christian L. Müller, Zachary D. Kurtz, et al. 2018. "Fungi Stabilize Connectivity in the Lung and Skin Microbial Ecosystems." *Microbiome* 6 (1). https://doi.org/10.1186/s40168-017-0393-0.

Watts, Duncan J., and Steven H. Strogatz. 1998. "Collective Dynamics of Small-World Networks." *Nature* 393 (6684): 440–42. https://doi.org/10.1038/30918.

Yaveroğlu, Ömer Nebil, Noël Malod-Dognin, Darren Davis, et al. 2014. "Revealing the Hidden Language of Complex Networks." *Scientific Reports* 4 (1). https://doi.org/10.1038/srep04547.

10 Matrix Factorization and Autoencoders

Chapter Aim

This chapter asks whether the high-dimensional measurement matrix can be explained through a small number of latent factors. A factor may represent a biological gradient, a cell state, a pathway activity, a hidden batch effect, a regulatory program, or an axis of multi-omics variation. The key idea is to replace p observed features by k latent coordinates, with $k \ll p$, while keeping enough structure to reconstruct, denoise, or interpret the original measurements.

Matrix factorization and autoencoders are two versions of the same broad idea. Classical matrix factorization writes the data approximately as a product of two low-rank matrices. Autoencoders replace this linear product by two learned functions: an encoder that maps observations into a bottleneck representation and a decoder that reconstructs the data. For biological count data, the reconstruction loss should often be distributional rather than purely squared-error based. This connects the chapter back to Chapter 02 and Chapter 03: transformations, likelihoods, and latent representations are not separate topics.

Notation convention

Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external references

For the classical foundation, see MSMB Multivariate Analysis, especially PCA, SVD, biplots, and rank-reduced views of high-dimensional data. For heterogeneous and multi-table data, see MSMB Heterogeneous Data. Nonnegative matrix factorization is a useful parts-based baseline (Lee and Seung 1999). Network component analysis illustrates how prior graph structure can constrain a factorization (Liao et al. 2003). Multi-omics factor analysis is a probabilistic factor model for shared and view-specific latent structure across molecular modalities (Argelaguet et al. 2018; Nguyen and Wang 2020). Count-aware autoencoders for single-cell RNA-seq are exemplified by the deep count autoencoder (Eraslan et al. 2019). The corresponding computations for negative-binomial matrix factorization, `glmpca`, MSE versus NB autoencoder losses, nonlinear DCA, and PBMC3k

are in Exercise Sheet 6, Exercise 03 (Exercise Sheet 06 2026).

Learning Goals

After this chapter, students should be able to write down a low-rank factorization with correct dimensions, connect PCA and SVD to reconstruction error, explain why NMF imposes an additive nonnegative representation, formulate generalized matrix factorization for count data using a negative-binomial likelihood, describe how multi-omics factor models share latent factors across modalities, define an autoencoder through encoder, bottleneck, and decoder maps, and explain why count-aware losses can matter for single-cell denoising.

10.1 The Factorization Object

The basic factorization model is

$$X \approx HW^\top, \qquad X \in \mathbb{R}^{n\times p}, \qquad H \in \mathbb{R}^{n\times k}, \qquad W \in \mathbb{R}^{p\times k}, \qquad k \ll \min(n,p). \tag{10.1}$$

The matrix H contains latent coordinates or scores for the observations. Row $h_i^\top$ describes observation i in k latent dimensions. The matrix W contains feature loadings. Row $w_j^\top$ describes how feature j relates to the latent factors. The reconstructed value for entry (i,j) is

$$\hat{x}_{ij} = h_i^\top w_j. \tag{10.2}$$

Thus a factorization replaces np entries by approximately

$$nk + pk = k(n+p) \tag{10.3}$$

parameters. This is useful only when the low-rank structure captures real signal. If k is too small, important biological variation is lost. If k is too large, the factorization starts to memorize noise.

The reconstruction error for a squared-error factorization is

$$\mathcal{L}_{\text{sq}}(H,W) = \frac{1}{2}\left\|X - HW^\top\right\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - h_i^\top w_j\right)^2. \tag{10.4}$$

Many methods in this chapter can be understood by changing one of three ingredients: the loss function, the constraints on H and W, or the way the latent representation is parameterized.

10.2 PCA And The Singular Value Decomposition

For PCA-like factorizations, the data matrix should first be centered:

$$X_c = X - \mathbf{1}_n \bar{x}^\top, \qquad \bar{x} = \frac{1}{n} X^\top \mathbf{1}_n. \tag{10.5}$$

The singular value decomposition writes

$$X_c = U \Lambda V^\top, \tag{10.6}$$

where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{p \times r}$ have orthonormal columns, $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_r)$ contains singular values with $\lambda_1 \geq \cdots \geq \lambda_r > 0$, and $r = \operatorname{rank}(X_c)$. The rank-$k$ truncated SVD is

$$X_{c,k} = U_k \Lambda_k V_k^\top. \tag{10.7}$$

It solves the best rank-k approximation problem under squared error:

$$X_{c,k} = \arg \min_{\operatorname{rank}(M) \leq k} \|X_c - M\|_F^2 . \tag{10.8}$$

In the notation of this chapter, the PCA factorization can be written as

$$X_c \approx H W^\top, \qquad H = U_k \Lambda_k, \qquad W = V_k. \tag{10.9}$$

The columns of H are the principal component scores, and the columns of W are loading vectors. The amount of centered variation explained by the first k components is

$$\operatorname{PVE}(k) = \frac{\sum_{\ell=1}^{k} \lambda_\ell^2}{\sum_{\ell=1}^{r} \lambda_\ell^2}. \tag{10.10}$$

PCA is powerful because it gives the optimal linear rank-k reconstruction under squared error. It is also restrictive. The factors are orthogonal, entries of H and W can be positive or negative, and the Gaussian squared-error loss is most natural for continuous, approximately homoscedastic data. For normalized expression values, metabolomics intensities, or transformed measurements, this can be appropriate. For raw sequencing counts $C \in \mathbb{N}_0^{n \times p}$, it usually is not.

10.3 Nonnegative Matrix Factorization

Nonnegative matrix factorization is used when the data and the intended representation should be additive. Suppose

$$X \in \mathbb{R}_{\geq 0}^{n \times p}. \tag{10.11}$$

NMF estimates

$$X \approx HW^{\top}, \qquad H \in \mathbb{R}_{\geq 0}^{n \times k}, \qquad W \in \mathbb{R}_{\geq 0}^{p \times k}. \tag{10.12}$$

A common squared-error NMF objective is

$$(\hat{H}, \hat{W}) = \arg \min_{H \geq 0,\ W \geq 0} \left\| X - HW^{\top} \right\|_F^2 . \tag{10.13}$$

The nonnegativity constraint changes the interpretation. PCA can reconstruct an observation through cancellations between positive and negative loadings. NMF allows only additive combinations of nonnegative parts (Lee and Seung 1999). If features are gene expression levels, taxa abundances, or peak intensities, each factor can be interpreted as a nonnegative pattern of features, and each observation receives a nonnegative exposure to that pattern.

The factorization is not unique. For any positive diagonal matrix $D \in \mathbb{R}^{k \times k}$,

$$HW^{\top} = (HD)(WD^{-1})^{\top}. \tag{10.14}$$

Therefore factor scales must be normalized or interpreted with care. NMF is also nonconvex in (H, W) jointly. Different initializations can lead to different local optima, so stability of factors should be checked before assigning biological meaning.

10.4 Prior Structure And Network Component Analysis

Sometimes prior biological knowledge constrains which features can load on which factors. Network component analysis is an example in which the factorization represents hidden regulatory activities and their effects on measured outputs (Liao et al. 2003). A simplified model is

$$X \approx HW^{\top}, \tag{10.15}$$

where $H \in \mathbb{R}^{n\times k}$ contains latent regulator activities across observations and $W \in \mathbb{R}^{p\times k}$ contains regulatory effects on features. If feature j is not believed to be regulated by factor ℓ, the loading is constrained to zero:

$$W_{j\ell} = 0 \qquad \text{whenever prior knowledge excludes edge } \ell \to j. \tag{10.16}$$

Let $B \in \{0,1\}^{p\times k}$ encode this prior regulatory graph. The constraint can be written as

$$W = B \odot W, \tag{10.17}$$

where $\odot$ denotes entrywise multiplication. This form makes clear how network information enters the factorization. It does not estimate an arbitrary low-rank representation. It estimates a low-rank representation compatible with a hypothesized feature-factor graph.

The corresponding constrained least-squares fitting problem is

$$(\hat{H}, \hat{W}) = \arg \min_{H\in\mathbb{R}^{n\times k},\ W\in\mathbb{R}^{p\times k}} \left\|X - HW^\top\right\|_F^2 \quad \text{subject to} \quad W = B \odot W. \tag{10.18}$$

Equivalently, one can parameterize only the allowed entries of W. If $\mathcal{E}_B = \{(j,\ell) : B_{j\ell} = 1\}$ is the set of allowed feature-factor edges, then

$$(\hat{H}, \hat{W}_{\mathcal{E}_B}) = \arg \min_{H,\ W_{\mathcal{E}_B}} \sum_{i=1}^{n}\sum_{j=1}^{p} \left[x_{ij} - \sum_{\ell:(j,\ell)\in\mathcal{E}_B} h_{i\ell} w_{j\ell}\right]^2. \tag{10.19}$$

The optimization is still nonconvex in H and W jointly, and the scale of a regulator activity can be traded against the scale of its outgoing effects. NCA therefore relies not only on minimizing reconstruction error but also on identifiability conditions for the prior network and on a normalization convention for the recovered activities and loadings (Liao et al. 2003).

10.5 Multi-Omics Factor Models

In multi-omics studies, the same n observations may be measured in several modalities. Let

$$X^{(m)} \in \mathbb{R}^{n\times p_m}, \qquad m = 1, \dots, M, \tag{10.20}$$

where m indexes modalities such as transcriptomics, methylation, chromatin accessibility, proteomics, metabolomics, drug response, or mutation data. A shared factor model writes

$$X^{(m)} \approx H \left(W^{(m)}\right)^\top, \qquad H \in \mathbb{R}^{n \times k}, \qquad W^{(m)} \in \mathbb{R}^{p_m \times k}. \tag{10.21}$$

The shared score matrix H places all observations in one latent space. The loading matrix $W^{(m)}$ describes how modality m relates to those latent factors. A factor with nonzero loadings in several modalities captures shared variation; a factor with loadings concentrated in one modality captures view-specific variation.

A probabilistic multi-omics factor model can be written as

$$x_{ij}^{(m)} \sim p_m \left(x \mid \eta_{ij}^{(m)}, \psi_m\right), \qquad \eta_{ij}^{(m)} = h_i^\top w_j^{(m)}. \tag{10.22}$$

Here p_m is the observation model for modality m, and ψ_m contains modality-specific noise or dispersion parameters. This is the core idea behind methods such as MOFA: infer hidden factors that explain sources of variation across heterogeneous molecular views (Argelaguet et al. 2018). The biological payoff is that one factor can be linked simultaneously to several molecular layers. The statistical risk is that factors can also capture batch effects, library size, sample quality, or other technical axes. Factor interpretation must therefore use loadings, covariates, variance explained by view, and external biological knowledge.

10.6 Generalized Matrix Factorization For Counts

For sequencing data, the raw matrix is

$$C \in \mathbb{N}_0^{n \times p}. \tag{10.23}$$

A squared-error loss on raw counts is usually poorly matched to the data. Counts are nonnegative, heteroscedastic, and often overdispersed. A generalized matrix factorization keeps the low-rank structure but places it inside a distributional model. For a negative-binomial count model,

$$c_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j), \qquad \mathrm{Var}(c_{ij}) = \mu_{ij} + \phi_j \mu_{ij}^2. \tag{10.24}$$

The mean is positive, so the low-rank predictor is passed through a link function. With size factors $s_i > 0$,

$$\log \mu_{ij} = \log s_i + h_i^\top w_j, \tag{10.25}$$

or equivalently

$$\mu_{ij} = s_i \exp\left(h_i^\top w_j\right). \tag{10.26}$$

The negative log-likelihood objective is

$$\mathcal{L}_{\mathrm{NB}}(H, W, \phi) = -\sum_{i=1}^{n}\sum_{j=1}^{p} \log \Pr\left(c_{ij} \mid \mu_{ij}, \phi_j\right). \tag{10.27}$$

This is the same factorization idea as Equation 10.1, but the reconstruction target is now the mean of a count distribution rather than a real-valued matrix under Gaussian noise. The practical consequence is important: the loss function encodes the mean-variance relationship. A negative-binomial factorization can use the bottleneck to model biological structure without treating every large count deviation as equally surprising. This is the logic behind methods such as GLM-PCA for count data and the negative-binomial factorization computations in Exercise Sheet 6, Exercise 03 (Exercise Sheet 06 2026).

10.7 Autoencoders As Nonlinear Factorizations

An autoencoder replaces the linear map $x_i \mapsto h_i$ and the linear reconstruction $h_i \mapsto \hat{x}_i$ by learned functions. Let $x_i \in \mathbb{R}^p$ denote the column-vector version of row i of X. The encoder is

$$h_i = f_\theta(x_i), \qquad h_i \in \mathbb{R}^k, \tag{10.28}$$

and the decoder is

$$\hat{x}_i = g_\omega(h_i), \qquad \hat{x}_i \in \mathbb{R}^p. \tag{10.29}$$

The bottleneck matrix is

$$H = \begin{bmatrix} h_1^\top \\ \vdots \\ h_n^\top \end{bmatrix} \in \mathbb{R}^{n \times k}. \tag{10.30}$$

With squared reconstruction loss, the autoencoder objective is

$$\mathcal{L}_{\mathrm{AE}}(\theta, \omega) = \frac{1}{2}\sum_{i=1}^{n} \left\| x_i - g_\omega\left(f_\theta(x_i)\right)\right\|_2^2. \tag{10.31}$$

If both f_θ and g_ω are linear maps and the bottleneck has dimension k, the autoencoder is closely related to PCA: it learns a rank-k linear reconstruction. If nonlinear activation functions are included, the model becomes a nonlinear factorization. The representation h_i can describe curved manifolds, nonlinear gene-gene relationships, or complex denoising rules that a linear factorization cannot capture.

This flexibility is also the main danger. A sufficiently flexible autoencoder can reconstruct technical noise or cell-specific artifacts. The bottleneck dimension, regularization, training time, and validation metric become part of the statistical model.

10.8 Count-Aware Autoencoders And DCA

For single-cell RNA-seq counts, the input is again a count matrix $C \in \mathbb{N}_0^{n \times p}$. A count-aware autoencoder decodes each bottleneck representation into distributional parameters rather than directly into a real-valued reconstruction. In a negative-binomial autoencoder,

$$h_i = f_\theta(c_i), \qquad \mu_i = g_\omega^{(\mu)}(h_i), \tag{10.32}$$

with $\mu_i \in \mathbb{R}_{>0}^p$. Entrywise,

$$c_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j). \tag{10.33}$$

The negative-binomial autoencoder loss is

$$\mathcal{L}_{\mathrm{NB-AE}}(\theta, \omega, \phi) = -\sum_{i=1}^{n} \sum_{j=1}^{p} \log \Pr\left(c_{ij} \mid \mu_{ij}, \phi_j\right). \tag{10.34}$$

The deep count autoencoder extends this idea to account for overdispersion and sparsity in single-cell RNA-seq, with negative-binomial or zero-inflated negative-binomial observation models (Eraslan et al. 2019). In a zero-inflated version, an additional dropout probability $\pi_{ij} \in [0, 1]$ is decoded:

$$\pi_i = g_\omega^{(\pi)}(h_i), \qquad \pi_i \in [0,1]^p. \tag{10.35}$$

The zero-inflated negative-binomial probability mass function is

$$\Pr(c_{ij} = y) = \pi_{ij} \mathbf{1}\{y = 0\} + (1 - \pi_{ij}) \Pr_{\mathrm{NB}}\left(Y = y \mid \mu_{ij}, \phi_j\right). \tag{10.36}$$

The denoised expression value is usually taken to be the fitted mean or a transformation of it:

$$\hat{x}_{ij}^{\text{denoise}} = \hat{\mu}_{ij}. \tag{10.37}$$

The bottleneck H can also be used as a low-dimensional representation for clustering, visualization, or trajectory analysis. The key point is that the autoencoder is not just a neural-network trick. It is a generalized factorization in which the decoder defines a mean-variance-aware reconstruction model.

10.9 Validation And Interpretation

Latent factors are tempting because they give names to hidden axes of variation. In biological data analysis, that temptation should be handled carefully. A factor can correspond to cell type, disease state, differentiation, pathway activity, sequencing depth, batch, donor, or a mixture of these. The interpretation should therefore use several checks.

First, factor scores should be related to known covariates. If z_i is a known covariate and $h_{i\ell}$ is factor ℓ, one can examine

$$h_{i\ell} = \alpha_0 + \alpha_1 z_i + \varepsilon_i. \tag{10.38}$$

Second, loadings should be inspected at the feature level. Large absolute loadings indicate features that define a factor:

$$\mathcal{J}_\ell(m) = \left\{ j : |w_{j\ell}| \text{ is among the } m \text{ largest values} \right\}. \tag{10.39}$$

Third, downstream performance should be evaluated on the representation that is actually used. If H is used for clustering, clustering quality should be measured on H, not only on a two-dimensional visualization. The Exercise Sheet 6 autoencoder analysis makes this point directly: the native bottleneck representation and a two-dimensional t-SNE view can give different impressions of separability (Exercise Sheet 06 2026).

Finally, reconstruction quality alone is not enough. A model can reconstruct noise. A useful representation should reconstruct relevant signal, improve or preserve downstream structure, and remain stable under reasonable changes in initialization, regularization, and bottleneck dimension.

Summary

Matrix factorization represents a high-dimensional biological matrix through latent scores and feature loadings. PCA is the canonical linear squared-error factorization. NMF adds nonnegativity and yields additive parts-based representations. Prior-structured factorizations, such as network component analysis, constrain loadings using biological knowledge. Multi-omics factor models share latent scores across several molecular views and use modality-specific loadings to identify shared and view-specific axes of variation.

Generalized matrix factorization changes the loss from squared error to a distributional likelihood. For count data, a negative-binomial factorization respects nonnegativity, overdispersion, and size factors. Autoencoders extend the same idea by replacing linear scores and loadings with learned encoder and decoder functions. Count-aware autoencoders such as DCA combine nonlinear representation learning with a negative-binomial or zero-inflated negative-binomial observation model. The central message is that the latent representation is only meaningful relative to the data scale, likelihood, constraints, and validation strategy used to learn it.

References

Argelaguet, Ricard, Britta Velten, Damien Arnol, et al. 2018. "Multi-omics Factor Analysis—a Framework for Unsupervised Integration of Multi-omics Data Sets." *Molecular Systems Biology* 14 (6). https://doi.org/10.15252/msb.20178124.

Eraslan, Gökcen, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. 2019. "Single-Cell RNA-Seq Denoising Using a Deep Count Autoencoder." *Nature Communications* 10 (1). https://doi.org/10.1038/s41467-018-07931-2.

Exercise Sheet 06. 2026. *Exercise Sheet 06 Solutions: Task 1 Sparse Microbial Network Inference with SPIEC-EASI, Task 2 Network Analysis with NetCoMi, and Deep Count Autoencoders*. https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise06_Solution.html.

Lee, Daniel D., and H. Sebastian Seung. 1999. "Learning the Parts of Objects by Non-Negative Matrix Factorization." *Nature* 401 (6755): 788–91. https://doi.org/10.1038/44565.

Liao, James C., Riccardo Boscolo, Young-Lyeol Yang, Linh My Tran, Chiara Sabatti, and Vwani P. Roychowdhury. 2003. "Network Component Analysis: Reconstruction of Regulatory Signals in Biological Systems." *Proceedings of the National Academy of Sciences* 100 (26): 15522–27. https://doi.org/10.1073/pnas.2136632100.

Nguyen, Nam D., and Daifeng Wang. 2020. "Multiview Learning for Understanding Functional Multiomics." *PLOS Computational Biology* 16 (4): e1007677. https://doi.org/10.1371/journal.pcbi.1007677.

11 Deep Learning and Foundation Models

Chapter Aim

This final chapter is an outlook. The previous chapters built a statistical language for high-dimensional biological data: distributions, normalization, embeddings, clustering, testing, regression, graphical models, networks, and latent factors. Deep learning and foundation models do not remove that language. They change where some of the modeling work happens. Instead of constructing a small number of features by hand and then fitting a simple model, we often learn a representation directly from raw or lightly processed biological data. In the most ambitious cases, the representation is pretrained once on a very large biological corpus and then adapted to many downstream tasks.

The central question is therefore not whether deep learning is a replacement for high-dimensional statistics. It is how high-dimensional statistics reappears inside larger learned systems. The final prediction layer may be a linear or logistic regression. The validation problem is still a sampling problem. Batch effects, leakage, confounding, calibration, and distribution shift remain statistical problems. What changes is that the covariates may now be learned embeddings rather than measured genes, peaks, taxa, or variants.

Notation convention

Notation follows the shared convention in the notation chapter. Vectors are columns by default, row vectors are written with transposes, and dimensions should be checked whenever matrices are multiplied, transformed, modeled, or decomposed.

Main external references

For the statistical background, see MSMB Supervised Learning, especially cross-validation, bias-variance trade-offs, penalization, and method hacking. See also MSMB Experimental Design for reproducibility, data quality, and leaky workflows, and MSMB Image Data for a concrete setting where feature extraction and representation learning meet. Deep learning in biology is reviewed in broad terms (Ching et al. 2018). Sequence-based regulatory prediction is exemplified by DeepSEA (Zhou and Troyanskaya 2015). Count-aware single-cell autoencoding connects directly to Chapter 10 (Eraslan et al. 2019). Protein structure prediction and protein foundation models provide the clearest archetypical case for this

chapter (Senior et al. 2020; Jumper et al. 2021; Rives et al. 2021; Hayes et al. 2025). Low-rank adaptation is one modern way to fine-tune large pretrained models with relatively few trainable parameters (Hu et al. 2021).

Learning Goals

After this chapter, students should be able to explain the difference between hand-crafted features and end-to-end representation learning, describe the pretrain/fine-tune paradigm, recognize why fine-tuning often reduces to classical high-dimensional prediction on top of learned embeddings, use ESM as an archetypical biological foundation model, and evaluate deep biological models with the same statistical caution used throughout the course.

11.1 From Hand-Crafted Features To End-To-End Learning

In a classical high-dimensional workflow, raw measurements are converted into an analysis representation and then into features used by a statistical model. Written schematically,

$$R \longrightarrow \widetilde{X} \longrightarrow h(\widetilde{X}) \longrightarrow \hat{y}, \tag{11.1}$$

where R denotes the raw measurement object, $\widetilde{X}$ denotes the chosen analysis representation, and $h(\widetilde{X})$ denotes hand-designed or separately estimated features such as principal components, clusters, pathway scores, graph summaries, or selected genes. The statistical model acts only after this feature construction has happened.

End-to-end learning changes the pipeline by making the feature map part of the fitted model. A neural network is a parameterized function f_θ with many layers and many parameters. It may take normalized continuous features, raw count-like inputs, one-hot encoded sequences, images, protein sequences, molecular graphs, or multimodal data as input. The predictor is fitted by minimizing an empirical loss,

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell \left\{ y_i, f_\theta(x_i) \right\}, \tag{11.2}$$

where x_i denotes the input object for observation i, y_i is the target, and ℓ is a task-specific loss. For a continuous response, ℓ may be squared error. For a binary response, it is often cross-entropy. For count data, it may be the negative log-likelihood of a Poisson or negative-binomial distribution. This is the same modeling principle as before: one specifies a data object, a prediction target, a loss, and a validation strategy. The difference is that f_θ may learn both the representation and the prediction rule.

For binary classification with predicted probability $\hat{\pi}_i = f_\theta(x_i) \in (0,1)$, the empirical cross-entropy loss is

$$\mathcal{L}_{\text{bin}}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left\{y_i \log \hat{\pi}_i + (1-y_i)\log(1-\hat{\pi}_i)\right\}. \tag{11.3}$$

For a multiclass problem with K classes, one typically uses a softmax output. If $o_i \in \mathbb{R}^K$ are the output scores and $\hat{\pi}_{ik}$ is the predicted probability of class k, then

$$\hat{\pi}_{ik} = \frac{\exp(o_{ik})}{\sum_{\ell=1}^{K}\exp(o_{i\ell})}. \tag{11.4}$$

Thus deep learning does not evade likelihood thinking. It often hides a familiar generalized linear model at the output layer, while replacing the input covariates by learned representations.

11.2 Representations And Multimodal Biology

The representation view is the cleanest way to connect this chapter to matrix factorization. In Chapter 10, an autoencoder learned a low-dimensional representation

$$H = f_\theta(X), \qquad \hat{X} = g_\phi(H), \tag{11.5}$$

where $H \in \mathbb{R}^{n\times k}$ contains one learned vector $h_i^\top$ per observation. Deep supervised models use the same idea, but the representation is optimized for a prediction task rather than only for reconstruction.

Modern biological datasets often contain several modalities. For observation i, let

$$x_i^{(1)}, \ldots, x_i^{(M)} \tag{11.6}$$

denote measurements from M modalities, for example RNA counts, chromatin accessibility, protein abundance, histology images, perturbation labels, or clinical covariates. Modality-specific encoders map these objects into latent vectors,

$$h_i^{(m)} = f_{\theta_m}^{(m)}\left(x_i^{(m)}\right), \qquad m = 1, \ldots, M. \tag{11.7}$$

A fusion map then constructs a shared representation,

$$h_i = a_\psi\left(h_i^{(1)}, \ldots, h_i^{(M)}\right), \tag{11.8}$$

which may be used for classification, regression, imputation, matching, or generation. The statistical challenge is that modalities have different noise models and missingness patterns. RNA-seq and ATAC-seq are count-like and sparse. Protein abundance may be continuous after transformation. Mutation status may be binary. Images are high-dimensional continuous arrays. A good model must respect these differences, even if the final representation is a shared vector.

This is why the modeling lessons from Chapters 02 and 03 remain relevant. A model that treats raw sparse counts as ordinary Gaussian measurements may learn technical variation. A multimodal model that ignores missingness mechanisms may learn which assay was performed rather than the biology of interest.

11.3 Sequence-To-Function Models

Genomic sequence models provide a clean example of end-to-end learning because the input is not an expression matrix but a string. For a DNA window of length L, define a one-hot encoded sequence

$$S_i \in \{0,1\}^{L\times 4}, \tag{11.9}$$

where the four columns correspond to the nucleotides A, C, G, and T. A sequence model maps S_i to one or many functional readouts, such as transcription-factor binding, DNase hypersensitivity, histone marks, or variant effects. In a multitask setting with K binary chromatin features, the model outputs

$$\hat{\pi}_{ik} = f_{\theta,k}(S_i), \qquad k = 1, \dots, K. \tag{11.10}$$

The loss sums over both genomic windows and tasks:

$$\mathcal{L}_{\text{seq}}(\theta) = -\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\left\{y_{ik}\log\hat{\pi}_{ik} + (1-y_{ik})\log(1-\hat{\pi}_{ik})\right\}. \tag{11.11}$$

The important point is conceptual. The model is not first given motif counts, manually selected windows, or a small list of engineered sequence features. Instead, filters and higher-level sequence patterns are learned from the prediction task. DeepSEA is an early influential example of this idea for noncoding variant effect prediction (Zhou and Troyanskaya 2015).

11.4 Count-Aware Deep Models For Single Cells

Single-cell RNA-seq is a second example, but here the input is not a sequence. It is a sparse count vector. For cell i, let

$$c_i = \left(c_{i1}, \ldots, c_{ip}\right)^\top \in \mathbb{N}_0^p \tag{11.12}$$

denote raw gene counts. A count-aware autoencoder maps this high-dimensional vector into a latent representation and decodes it into distributional parameters, for example a negative-binomial mean and dispersion:

$$h_i = f_\theta(c_i), \qquad \mu_i = g_{\phi,\mu}(h_i), \qquad \alpha_i = g_{\phi,\alpha}(h_i). \tag{11.13}$$

The fitting objective is then a negative log-likelihood rather than an ordinary squared reconstruction error:

$$\mathcal{L}_{\mathrm{NB-AE}}(\theta, \phi) = -\sum_{i=1}^{n}\sum_{j=1}^{p} \log p_{\mathrm{NB}}\left(c_{ij}; \mu_{ij}, \alpha_{ij}\right). \tag{11.14}$$

This example is a useful warning against the phrase "deep learning" as if it were a single model. A good deep model for count data should still encode an appropriate observation model. DCA is a concrete example of such a count-aware approach for single-cell denoising (Eraslan et al. 2019), and it is the natural continuation of the count-factorization ideas in Chapter 10.

11.5 The Pretrain And Fine-Tune Paradigm

The conceptual shift from deep learning to foundation models is the move from task-specific supervised training to large-scale pretraining. In supervised training, one fits θ directly for the task of interest. In pretraining, one first learns parameters from a broad data source, often without manual labels:

$$\hat{\theta}_{\mathrm{pre}} = \operatorname*{argmin}_\theta \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\mathrm{pre}}\left(u_i; \theta\right), \tag{11.15}$$

where u_i are unlabeled or weakly labeled biological objects and N is typically very large. The pretrained encoder $f_{\hat{\theta}_{\mathrm{pre}}}$ maps a new object into an embedding. A downstream task can then be fitted on top of these embeddings:

$$z_i = f_{\hat{\theta}_{\mathrm{pre}}}(x_i) \in \mathbb{R}^d. \tag{11.16}$$

At this point, much of the downstream analysis becomes classical high-dimensional statistics again. If the pretrained representation is frozen, a regression or classification head estimates only a small set of downstream parameters. For a binary phenotype, a logistic fine-tuning head is

$$\Pr(Y_i = 1 \mid z_i) = \sigma\left(\beta_0 + z_i^\top \beta\right), \qquad \sigma(t) = \frac{1}{1 + \exp(-t)}. \tag{11.17}$$

Here β_0 is the intercept and $\beta \in \mathbb{R}^d$ contains regression coefficients. If d is large relative to the number of labeled examples, regularization is again necessary:

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \left[-\sum_{i=1}^{n} \left\{ y_i \log \widehat{\pi}_i + (1 - y_i) \log(1 - \widehat{\pi}_i) \right\} + \lambda \|\beta\|_1 \right]. \tag{11.18}$$

This equation is deliberately familiar. It is the sparse logistic regression problem from Chapter 07, now applied to pretrained embeddings instead of hand-defined molecular features. The foundation model changes the covariates; it does not remove the need for regularization, validation, or interpretation.

Fine-tuning can also update part of the pretrained model. Low-rank adaptation is one parameter-efficient strategy. For a pretrained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, one fits a low-rank update

$$W = W_0 + \Delta W, \qquad \Delta W = BA, \tag{11.19}$$

where $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$. The number of trainable parameters is therefore

$$r d_{\text{in}} + d_{\text{out}} r = r\left(d_{\text{in}} + d_{\text{out}}\right), \tag{11.20}$$

which can be much smaller than training all entries of W_0. This is a modern version of a familiar idea: restrict the effective dimension of the fitted model so that limited labeled data are not asked to estimate too many free parameters.

11.6 ESM As An Archetypical Foundation Model

Protein language models give perhaps the cleanest biological example of the foundation-model paradigm. A protein sequence is a string

$$a_{1:L} = (a_1, \dots, a_L), \qquad a_t \in \mathcal{A}, \tag{11.21}$$

where $\mathcal{A}$ is the amino-acid alphabet. In ESM-style masked language modeling, a subset $M \subset \{1, \dots, L\}$ of positions is masked, and the model is trained to predict the missing amino acids from the unmasked context:

$$\mathcal{L}_{\mathrm{MLM}}(\theta) = -\mathbb{E}_{a_{1:L}} \mathbb{E}_M \sum_{t \in M} \log p_\theta \left(a_t \mid a_{\{1,\dots,L\} \setminus M} \right). \tag{11.22}$$

This is not a supervised phenotype-prediction problem. The training signal comes from the sequence itself. Yet the resulting representation can encode biological structure and function because the distribution of natural protein sequences reflects evolutionary constraints (Rives et al. 2021). After pretraining, each residue and each full sequence has an embedding:

$$E = f_{\hat{\theta}_{\mathrm{pre}}}(a_{1:L}) \in \mathbb{R}^{L \times d}, \tag{11.23}$$

where row $e_t^\top$ represents residue position t. A sequence-level embedding can be obtained by pooling,

$$\bar{e} = \frac{1}{L} \sum_{t=1}^{L} e_t \in \mathbb{R}^d. \tag{11.24}$$

Downstream tasks can then use E or $\bar{e}$ for contact prediction, structure prediction, mutational effect prediction, annotation, or design. In its simplest form, contact prediction is again a fitted prediction problem on representations:

$$\Pr(A_{st} = 1 \mid e_s, e_t) = \sigma\left(\gamma_0 + \eta^\top \psi(e_s, e_t)\right), \tag{11.25}$$

where A_{st} indicates whether residues s and t are in contact, $\psi(e_s, e_t)$ is a pair representation constructed from the two residue embeddings, γ_0 is an intercept, and η contains regression coefficients. This is the point of the archetype: the impressive biological behavior comes from large-scale sequence pretraining, but many downstream uses still involve familiar supervised heads, calibration, test splits, and regularization.

ESM also illustrates the path from predictive to generative biological modeling. Later ESM models treat sequence, structure, and function as coupled token tracks and can be prompted or

sampled to generate proteins (Hayes et al. 2025). Conceptually, this extends the language-model idea from "predict masked residues" to "model a joint biological object" whose components may include amino-acid identity, structural tokens, functional annotations, and geometric information.

11.7 Protein Structure And Learned Geometry

Protein structure prediction is another example of end-to-end representation learning, but with a geometric output. Given a sequence $a_{1:L}$, the model predicts three-dimensional coordinates

$$R = (r_1, \dots, r_L)^\top \in \mathbb{R}^{L\times 3}, \tag{11.26}$$

where r_t is a coordinate vector for residue t or for a representative atom. Earlier systems combined learned pairwise potentials with geometric optimization (Senior et al. 2020). AlphaFold-style models make a larger part of the pipeline differentiable and learned, while still using strong geometric inductive structure (Jumper et al. 2021). From the perspective of this course, the key lesson is that high-dimensional biological prediction often benefits from combining large learned representations with domain constraints: sequences are discrete strings, structures live in three-dimensional geometry, and biological outputs have symmetries and physical restrictions.

11.8 Evaluation Still Decides The Claim

Deep learning makes leakage easier because the preprocessing and representation learning can involve enormous datasets. A model may perform well because homologous sequences, related donors, repeated images, shared batches, or near-duplicate perturbations appear across train and test splits. Therefore the statistical object of interest is not the training loss but the expected prediction risk

$$R(f) = \mathbb{E}_{(X,Y)\sim P_{\text{test}}}\left[\ell\{Y, f(X)\}\right]. \tag{11.27}$$

The empirical test risk estimates this quantity on held-out data:

$$\hat{R}_{\text{test}}(f) = \frac{1}{n_{\text{test}}} \sum_{i\in\mathcal{J}_{\text{test}}} \ell\left\{y_i, f(x_i)\right\}. \tag{11.28}$$

For biological applications, the split must match the intended generalization claim. If the claim is generalization to new patients, split by patient. If the claim is generalization to new studies, split by study. If the claim is generalization to remote protein families, split by sequence

similarity or family. If the claim is perturbation response prediction, hold out perturbations, cell types, or combinations according to the desired use case.

Calibration is equally important. For a binary predictor that outputs $\hat{\pi}(X)$, ideal calibration means

$$\Pr(Y = 1 \mid \hat{\pi}(X) = s) = s \qquad \text{for all } s \in (0, 1). \tag{11.29}$$

A model can have high accuracy and poor calibration. In biology and medicine, this distinction matters because predicted probabilities are often used for prioritization, screening, or decision support. The final warning is distribution shift:

$$P_{\text{train}}(X, Y) \neq P_{\text{test}}(X, Y). \tag{11.30}$$

This inequality is common rather than exceptional. Assay protocols change, batches differ, populations differ, sequence databases are unevenly sampled, and images or omics profiles may carry site-specific signatures. Foundation models can sometimes improve robustness because pretraining exposes the model to broader variation. They can also amplify hidden biases from the pretraining corpus.

11.9 What Changes, And What Does Not

Deep learning changes high-dimensional biological statistics in three ways. First, representation learning becomes central: the data matrix X may be replaced by an embedding matrix H learned from images, sequences, counts, graphs, or multiple modalities. Second, prediction and preprocessing become more tightly coupled: normalization, denoising, imputation, and classification may be trained inside a single model. Third, foundation models introduce a separation between large-scale unsupervised or self-supervised pretraining and smaller supervised downstream analyses.

What does not change is just as important. Biological data still have dimensions, distributions, nuisance variables, and sampling designs. Count data remain count data. Compositional data remain compositional. Binary labels remain binary. A pretrained embedding is not automatically causal, interpretable, calibrated, or free of batch effects. The last layer of a foundation-model workflow is often a classical regression or classification problem, and the credibility of the result still depends on careful validation.

The course therefore ends where it began: with the relationship between biological measurement, statistical modeling, and scientific claims. Deep models provide powerful new function classes and pretrained representations. High-dimensional statistics provides the language for asking what was measured, what was learned, what was validated, and what can reasonably be concluded.

References

Ching, Travers, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, et al. 2018. "Opportunities and Obstacles for Deep Learning in Biology and Medicine." *Journal of The Royal Society Interface* 15 (141). https://doi.org/10.1098/rsif.2017.0387.

Eraslan, Gökcen, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. 2019. "Single-Cell RNA-Seq Denoising Using a Deep Count Autoencoder." *Nature Communications* 10 (1). https://doi.org/10.1038/s41467-018-07931-2.

Hayes, Thomas, Roshan Rao, Halil Akin, et al. 2025. "Simulating 500 Million Years of Evolution with a Language Model." *Science* 387 (6736): 850–58. https://doi.org/10.1126/science.ads0018.

Hu, Edward J., Yelong Shen, Phillip Wallis, et al. 2021. *LoRA: Low-Rank Adaptation of Large Language Models*. https://arxiv.org/abs/2106.09685.

Jumper, John, Richard Evans, Alexander Pritzel, et al. 2021. "Highly Accurate Protein Structure Prediction with AlphaFold." *Nature* 596 (7873): 583–89. https://doi.org/10.1038/s41586-021-03819-2.

Rives, Alexander, Joshua Meier, Tom Sercu, et al. 2021. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences." *Proceedings of the National Academy of Sciences* 118 (15): e2016239118. https://doi.org/10.1073/pnas.2016239118.

Senior, Andrew W., Richard Evans, John Jumper, et al. 2020. "Improved Protein Structure Prediction Using Potentials from Deep Learning." *Nature* 577 (7792): 706–10. https://doi.org/10.1038/s41586-019-1923-7.

Zhou, Jian, and Olga G Troyanskaya. 2015. "Predicting Effects of Noncoding Variants with Deep Learning–Based Sequence Model." *Nature Methods* 12 (10): 931–34. https://doi.org/10.1038/nmeth.3547.

12 Notation and Dimensions

12.1 Core Convention

The working agreement for notation is stored in `assets/references/notation-working-agreement.md`. Treat that file as the source of truth when expanding chapters.

The central object in the lecture is the high-throughput sequencing data matrix:

$$X \in \mathbb{R}^{n \times p}. \tag{12.1}$$

where rows are observational units and columns are measured features.

- n: number of observations, such as samples, patients, cells, images, or experimental units
- p: number of features, such as genes, proteins, metabolites, genomic loci, peaks, or extracted image features
- $x_i^\top \in \mathbb{R}^{1 \times p}$: row vector for observation i
- $x_{\cdot j} \in \mathbb{R}^n$: column vector for feature j
- x_{ij}: value of feature j in observation i

Vectors are column vectors by default. A row vector is written with a transpose, for example $x_i^\top$.

For sequencing data, the entries of X may represent raw counts, normalized abundances, transformed counts, or scaled values. The lecture must always state which version is being used. When the distinction matters, reserve C for raw counts and use X for the analysis matrix after the relevant preprocessing step.

12.2 Data Matrices

Symbol	Dimension	Meaning
X	$n \times p$	high-throughput sequencing data matrix used for analysis

Symbol	Dimension	Meaning
C	$n \times p$	raw sequencing count matrix
Y	$n \times q$	response matrix or second data table
y	n	response vector
Z	$n \times r$	covariate, design, or metadata matrix
W	$p \times k$	feature loadings, weights, or decoder matrix, depending on context
H	$n \times k$	latent scores or low-dimensional representation
D	$n \times n$	distance or dissimilarity matrix between observations
A	$p \times p$ or $n \times n$	adjacency matrix for a feature network or observation network

The meaning of A must always be stated because biological networks can connect genes, proteins, cells, samples, pathways, or phenotypes.

12.3 Indices

Index	Range	Meaning
i	$1, \dots, n$	observation index
j	$1, \dots, p$	feature index
g	$1, \dots, G$	group, class, or cluster index
k	$1, \dots, K$	latent dimension, component, or factor index
b	$1, \dots, B$	batch, block, bootstrap draw, or resample index

12.4 Probability Operators

Use a single TeX style for probability operators throughout the script. Expectations are written as $\mathbb{E}(Y)$ for simple arguments and as $\mathbb{E}\left[g(Y)\right]$ for longer expressions. Probabilities are

written with Pr, variances with Var, covariances with Cov, and correlations with Cor. Thus the preferred forms are

$$\mathbb{E}(Y), \qquad \Pr(Y = y \mid \theta), \qquad \operatorname{Var}(Y), \qquad \operatorname{Cov}(Y_1, Y_2). \tag{12.2}$$

Avoid switching to alternative spellings such as E, E, $\mathbb{P}$, Var, or Cov in chapter text.

12.5 Counts and Normalized Data

For count-based omics, use:

- $C \in \mathbb{N}_0^{n \times p}$: raw count matrix
- c_{ij}: raw count for feature j in observation i
- $s_i > 0$: size factor, library size factor, or exposure for observation i
- $X \in \mathbb{R}^{n \times p}$: analysis matrix after normalization, transformation, filtering, or scaling, depending on context
- $\tilde{X}$: explicitly normalized or transformed intermediate data matrix, when needed
- $\log_2(\tilde{x}_{ij} + a)$: log-transformed value with pseudocount a, when needed

Do not use X for both raw counts and transformed values in the same section. State whether X is raw, normalized, scaled, or embedded.

12.6 Distributional View Of X

Many analyses model the entries, rows, or columns of X using distributional assumptions. The most common starting point in this course is that high-throughput sequencing data are count data or discrete compositions.

12.6.1 Entry-wise View

An entry-wise model treats each c_{ij} as a count:

$$c_{ij} \sim \text{Poisson}(\lambda_{ij}). \tag{12.3}$$

or, when overdispersion matters,

$$c_{ij} \sim \text{Negative Binomial}(\mu_{ij}, \phi_j). \tag{12.4}$$

Here λ_{ij} or μ_{ij} is the expected count, and ϕ_j is a feature-specific dispersion parameter.

12.6.2 Row-wise Compositional View

Sequencing often measures relative abundance under a finite library size. For observation i, the row vector of counts is:

$$c_i^\top = (c_{i1}, \dots, c_{ip}), \qquad N_i = \sum_{j=1}^{p} c_{ij}. \tag{12.5}$$

A simple row-wise compositional model is:

$$c_i \sim \text{Multinomial}(N_i, \pi_i). \tag{12.6}$$

where $\pi_i \in \Delta^{p-1}$ is a vector of feature probabilities that sums to one. The simplex notation means:

$$\Delta^{p-1} = \left\{ \pi \in \mathbb{R}^p_{\geq 0} : \sum_{j=1}^{p} \pi_j = 1 \right\}. \tag{12.7}$$

This emphasizes that sequencing data often carry relative information: increasing one component's share can decrease another component's share even if the underlying biology is unchanged.

12.6.3 Column-wise Feature View

For feature-wise testing or regression, we often isolate one column:

$$y_j = c_{\cdot j} \in \mathbb{N}_0^n. \tag{12.8}$$

and model it across observations using covariates:

$$y_j \sim \text{Negative Binomial}(\mu_j, \phi_j), \qquad \log \mu_j = \log s + Z\beta_j. \tag{12.9}$$

Here $s \in \mathbb{R}^n_{>0}$ is a vector of size factors or exposures, $Z \in \mathbb{R}^{n \times r}$ is the design matrix, and $\beta_j \in \mathbb{R}^r$ contains feature-specific regression coefficients.

12.6.4 Practical Rule

When introducing a method, state which object is being modeled:

- entries c_{ij}
- rows c_i
- columns $c_{\cdot j}$
- transformed values x_{ij}
- distances between rows or columns
- latent representations derived from X

12.7 Metadata and Design Matrices

Use:

- m_i: metadata for observation i, usually a row in a metadata table
- $Z \in \mathbb{R}^{n \times r}$: design or covariate matrix
- $\beta \in \mathbb{R}^r$: regression coefficients for one response
- $B \in \mathbb{R}^{r \times p}$: coefficient matrix when modeling many features

For feature-wise regression:

$$y_j = Z\beta_j + \varepsilon_j. \tag{12.10}$$

where $y_j = x_{\cdot j} \in \mathbb{R}^n$, $Z \in \mathbb{R}^{n \times r}$, $\beta_j \in \mathbb{R}^r$, and $\varepsilon_j \in \mathbb{R}^n$.

12.8 Dimension Reduction

For PCA-like decompositions, use centered or scaled data X_c:

$$X_c \approx T_k P_k^\top. \tag{12.11}$$

with:

- $X_c \in \mathbb{R}^{n \times p}$
- $T_k \in \mathbb{R}^{n \times k}$: scores or coordinates of observations
- $P_k \in \mathbb{R}^{p \times k}$: loadings or feature directions
- $k \ll \min(n, p)$

For embeddings:

$$U = f(X), \qquad U \in \mathbb{R}^{n \times d}. \tag{12.12}$$

where d is usually 2 or 3 for visualization, or larger for learned representations.

12.9 Clustering

Use:

- $z_i \in \{1, \dots, K\}$: cluster label for observation i
- $z \in \{1, \dots, K\}^n$: vector of cluster assignments
- $\mu_k \in \mathbb{R}^p$: centroid for cluster k
- $D \in \mathbb{R}^{n \times n}$: distance matrix used by the clustering algorithm

If clustering features instead of observations, state that explicitly and use a feature distance matrix $D_f \in \mathbb{R}^{p \times p}$.

12.10 Testing

Use:

- H_{0j}: null hypothesis for feature j
- H_{1j}: alternative hypothesis for feature j
- p_j: p-value for feature j
- q_j: adjusted p-value or false discovery rate estimate for feature j
- α: chosen significance level
- m: number of tested hypotheses

Often $m = p$, but not always. For pathway, interaction, or network tests, m can be different from the number of measured features.

12.11 Networks

Use:

- $G = (V, E)$: graph with nodes V and edges E
- $|V| = v$: number of nodes
- $|E|$: number of edges
- $A \in \mathbb{R}^{v \times v}$: adjacency matrix

- L: graph Laplacian, if used

Always state what nodes represent. A gene network, sample-similarity network, cell-cell graph, and pathway graph are different objects even if all are written as G.

12.12 Deep Learning and Latent Models

Use:

- $h_i \in \mathbb{R}^k$: latent representation for observation i
- $H \in \mathbb{R}^{n \times k}$: latent representation matrix
- f_θ: encoder, predictor, or learned function with parameters θ
- g_ϕ: decoder or generative function with parameters ϕ
- $\hat{X}$: reconstruction of X

For an autoencoder:

$$H = f_\theta(X), \qquad \hat{X} = g_\phi(H). \tag{12.13}$$

In implementations, rows are typically processed in mini-batches even when the function is written mathematically as acting on the full matrix.

12.13 Dimension Checklist

Before finalizing any formula, check:

1. What are the rows?
2. What are the columns?
3. Is the object raw, normalized, transformed, scaled, or embedded?
4. Are vectors columns or rows?
5. Does every matrix multiplication have matching inner dimensions?
6. Does the result have the dimension claimed in the text?
7. Are we clustering observations, features, or another object?
8. Does a network connect features, observations, or external biological entities?

13 Resources

13.1 Course Materials

These lecture notes are designed to be read in concert with the highly visual lecture slides and the course book. The notes provide the mathematical spine of the course; the slides provide visual intuition and examples; the book provides the broader narrative and supporting material.

Modern Statistics for Modern Biology by Susan Holmes and Wolfgang Huber is the main external online reference and is linked repeatedly from the lecture script.

Use this page as the stable hub for links to MSMB chapters that will be referenced repeatedly from the lecture script.

13.2 Hosting Plan

At present, the lecture links point to the online EMBL version. When a group-hosted mirror becomes available, the URLs can be updated centrally in `_book-links.qmd`. The lecture text should continue to use reference labels such as `[MSMB]`, `[MSMB Testing]`, and `[MSMB Supervised Learning]`.

This keeps the lecture script portable: the content can stay stable while the book location changes.

The local structure map for lecture planning is in `assets/references/msmb-structure.md`.

The slide inventory and source-to-chapter map is in `assets/references/slide-inventory.md`.

The notation working agreement is in `assets/references/notation-working-agreement.md`.

The six exercise sheets are available as HTML artifacts in `assets/exercises/` and are linked from the relevant chapters. These links allow the online lecture script to connect directly to the corresponding computational material.

13.3 Exercise Sheets

- Exercise Sheet 01: Generative models, nucleotide frequencies, maximum likelihood, and mixture models
- Exercise Sheet 02: Normalization and variance-stabilizing transformations of sequencing data
- Exercise Sheet 03: Normalization and low-dimensional representation of 16S sequencing
- Exercise Sheet 04: Hierarchical clustering and graph-based clustering
- Exercise Sheet 05: Hypothesis testing, differential abundance, PERMANOVA, and high-dimensional inference
- Exercise Sheet 06: Networks, NetCoMi, SPIEC-EASI, and deep count autoencoders

The course paper bibliography is in `assets/references/course-papers.bib`, with a readable inventory in `assets/references/course-papers-inventory.md`.

13.4 Chapter Links

- Home
- Introduction
- Generative Models for Discrete Data
- Statistical Modeling
- Data visualization
- Mixture Models
- Clustering
- Testing
- Multivariate Analysis
- High-Throughput Count Data
- Multivariate Methods for Heterogeneous Data
- Networks and Trees
- Image Data
- Supervised Learning
- Design of High Throughput Experiments and their Analyses
- Statistical Concordance
- References

13.5 How To Reuse Links

In any `.qmd` page, add this line near the bottom:

```
<!--
Main book link registry for the lecture.

When the group-hosted copy of Modern Statistics for Modern Biology is ready,
replace the base URL in the link definitions below. The lecture pages should
continue using labels such as [MSMB], [MSMB Clustering], and
[MSMB Multivariate Analysis] instead of hard-coded URLs.
-->

[MSMB]: https://www.huber.embl.de/msmb/
[MSMB Introduction]: https://www.huber.embl.de/msmb/00-chap.html
[MSMB Generative Models]: https://www.huber.embl.de/msmb/01-chap.html
[MSMB Statistical Modeling]: https://www.huber.embl.de/msmb/02-chap.html
[MSMB Data Visualization]: https://www.huber.embl.de/msmb/03-chap.html
[MSMB Mixture Models]: https://www.huber.embl.de/msmb/04-chap.html
[MSMB VST Delta Method]: https://www.huber.embl.de/msmb/04-chap.html#sec-mixtures-vst
[MSMB Clustering]: https://www.huber.embl.de/msmb/05-chap.html
[MSMB Testing]: https://www.huber.embl.de/msmb/06-chap.html
[MSMB Testing FDR Intuition]: https://www.huber.embl.de/msmb/06-chap.html#false-discovery-rat
[MSMB Testing Five Steps]: https://www.huber.embl.de/msmb/06-chap.html#sec-fivesteps
[MSMB Testing Types Of Error]: https://www.huber.embl.de/msmb/06-chap.html#sec-testing-typesc
[MSMB Testing t-Test]: https://www.huber.embl.de/msmb/06-chap.html#sec-testing-ttest
[MSMB Testing FWER]: https://www.huber.embl.de/msmb/06-chap.html#sec-testing-FWER
[MSMB Testing FDR]: https://www.huber.embl.de/msmb/06-chap.html#the-false-discovery-rate
[MSMB Testing P-Value Histogram]: https://www.huber.embl.de/msmb/06-chap.html#the-p-value-his
[MSMB Testing BH]: https://www.huber.embl.de/msmb/06-chap.html#the-benjamini-hochberg-algorit
[MSMB Testing Local FDR]: https://www.huber.embl.de/msmb/06-chap.html#sec-testing-localfdr
[MSMB Testing IHW]: https://www.huber.embl.de/msmb/06-chap.html#independent-hypothesis-weight
[MSMB Multivariate Analysis]: https://www.huber.embl.de/msmb/07-chap.html
[MSMB High-Throughput Count Data]: https://www.huber.embl.de/msmb/08-chap.html
[MSMB Count Data Table]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-counttable
[MSMB Count Data Normalization]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-no
[MSMB DESeq2 Basic Analysis]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-desec
[MSMB Count Data Multifactor Designs]: https://www.huber.embl.de/msmb/08-chap.html#sec-counto
[MSMB Count Data Linear Models]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-lm
[MSMB Count Data GLM]: https://www.huber.embl.de/msmb/08-chap.html#sec:glm
[MSMB Count Data GLM DESeq2]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-glmde
[MSMB Count Data Two-Factor Analysis]: https://www.huber.embl.de/msmb/08-chap.html#sec-counto
[MSMB Count Data Sharing Information]: https://www.huber.embl.de/msmb/08-chap.html#sec-counto
[MSMB RNA-seq VST]: https://www.huber.embl.de/msmb/08-chap.html#sec-countdata-varstab
[MSMB Heterogeneous Data]: https://www.huber.embl.de/msmb/09-chap.html
[MSMB Networks and Trees]: https://www.huber.embl.de/msmb/10-chap.html
```

```
[MSMB Networks Graphs]: https://www.huber.embl.de/msmb/10-chap.html#graphs
[MSMB Networks Graph Encoding]: https://www.huber.embl.de/msmb/10-chap.html#what-is-a-graph-a
[MSMB Networks Graphs From Data]: https://www.huber.embl.de/msmb/10-chap.html#graphs-from-dat
[MSMB Networks Gene Sets]: https://www.huber.embl.de/msmb/10-chap.html#from-gene-set-enrichme
[MSMB Networks Significant Subgraphs]: https://www.huber.embl.de/msmb/10-chap.html#significan
[MSMB Image Data]: https://www.huber.embl.de/msmb/11-chap.html
[MSMB Supervised Learning]: https://www.huber.embl.de/msmb/12-chap.html
[MSMB Supervised Data]: https://www.huber.embl.de/msmb/12-chap.html#what-are-the-data
[MSMB Supervised Linear Discrimination]: https://www.huber.embl.de/msmb/12-chap.html#linear-d
[MSMB Supervised Cross-Validation]: https://www.huber.embl.de/msmb/12-chap.html#sec-supervise
[MSMB Supervised Curse]: https://www.huber.embl.de/msmb/12-chap.html#sec-supervised-curse
[MSMB Supervised Objective Functions]: https://www.huber.embl.de/msmb/12-chap.html#sec-superv
[MSMB Supervised Penalization]: https://www.huber.embl.de/msmb/12-chap.html#penalization
[MSMB Supervised Microbiome Classification]: https://www.huber.embl.de/msmb/12-chap.html#exam
[MSMB Supervised Mouse Cells]: https://www.huber.embl.de/msmb/12-chap.html#sec-supervised-cla
[MSMB Experimental Design]: https://www.huber.embl.de/msmb/13-chap.html
[MSMB Mean-Variance Relationships]: https://www.huber.embl.de/msmb/13-chap.html#mean-variance
[MSMB Concordance]: https://www.huber.embl.de/msmb/14-chap.html
[MSMB References]: https://www.huber.embl.de/msmb/references.html
```

Then use reference-style links in the text:

```
For a deeper treatment, see [MSMB Multivariate Analysis].
```

14 Complete References

This final bibliography collects the unique references cited across the lecture script in one place. Chapter-specific reference lists remain attached to the chapters where the references are used.

References

Ahlmann-Eltze, Constantin, and Wolfgang Huber. 2023. "Comparison of Transformations for Single-Cell RNA-seq Data." *Nature Methods*. https://doi.org/10.1038/s41592-023-01814-1.

Aitchison, J., and S. M. Shen. 1980. "Logistic-Normal Distributions: Some Properties and Uses." *Biometrika* 67 (2): 261. https://doi.org/10.2307/2335470.

Akaike, Hirotugu. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19 (6): 716–23. https://doi.org/10.1109/TAC.1974.1100705.

Albert, Réka, Hawoong Jeong, and Albert-László Barabási. 2000. "Error and Attack Tolerance of Complex Networks." *Nature* 406 (6794): 378–82. https://doi.org/10.1038/35019019.

Anderson, Marti J. 2001. "A New Method for Non-Parametric Multivariate Analysis of Variance." *Austral Ecology* 26 (1): 32–46. https://doi.org/10.1046/j.1442-9993.2001.01070.x.

Argelaguet, Ricard, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. 2018. "Multi-omics Factor Analysis—a Framework for Unsupervised Integration of Multi-omics Data Sets." *Molecular Systems Biology* 14 (6). https://doi.org/10.15252/msb.20178124.

Becht, Etienne, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W H Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. 2019. "Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP." *Nature Biotechnology* 37 (1): 38–44. https://doi.org/10.1038/nbt.4314.

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 57 (1): 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. "Fast Unfolding of Communities in Large Networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008 (10): P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008.

Böhm, Jan Niklas, Philipp Berens, and Dmitry Kobak. 2023. "A Unifying Perspective on Neighbor Embeddings Along the Attraction-Repulsion Spectrum." https://arxiv.org/abs/2007.08902.

Ching, Travers, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, et al. 2018. "Opportunities and Obstacles for Deep Learning in Biology and Medicine." *Journal of The Royal Society Interface* 15 (141). https://doi.org/10.1098/rsif.2017.0387.

Crick, Francis. 1970. "Central Dogma of Molecular Biology." *Nature* 227 (5258): 561–63. https://doi.org/10.1038/227561a0.

Egozcue, Juan José, and Raimon Tolosana-Delgado. 2007. "Lecture Notes on Compositional Data Analysis." Universitat de Girona.

Eisen, Michael B., Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. "Cluster Analysis and Display of Genome-Wide Expression Patterns." *Proceedings of the National Academy of Sciences* 95 (25): 14863–68. https://doi.org/10.1073/pnas.95.25.14863.

Eraslan, Gökcen, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. 2019. "Single-Cell RNA-Seq Denoising Using a Deep Count Autoencoder." *Nature Communications* 10 (1). https://doi.org/10.1038/s41467-018-07931-2.

Ester, Martin, Hans-Peter Kriegel, Joerg Sander, and Xiaowei Xu. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–31. AAAI Press.

Exercise Sheet 01. 2026. "Exercise Sheet 01 Solutions: Generative Models, Nucleotide Frequencies, Maximum Likelihood, and Mixture Models." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise01_Solution.html.

Exercise Sheet 02. 2026. "Exercise Sheet 02 Solutions: Normalization and Variance-Stabilizing Transformations of Sequencing Data." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise02_Solution.html.

Exercise Sheet 03. 2026. "Exercise Sheet 03 Solutions: Normalization and Low Dimensional Representation of 16S Sequencing." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise03_Solution.html.

Exercise Sheet 04. 2026. "Exercise Sheet 04 Solutions: Hierarchical Clustering and Graph-Based Clustering of High-Dimensional Biological Data." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise04_Solution.html.

Exercise Sheet 05. 2026. "Exercise Sheet 05 Solutions: Hypothesis Testing, Differential Abundance, PERMANOVA, and High-Dimensional Inference." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise05_Solution.html.

Exercise Sheet 06. 2026. "Exercise Sheet 06 Solutions: Task 1 Sparse Microbial Network Inference with SPIEC-EASI, Task 2 Network Analysis with NetCoMi, and Deep Count Autoencoders." https://bio-datascience.github.io/ahbd/assets/exercises/AHDBD_SS26_Exercise06_Solution.html.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2008. "Sparse Inverse Covariance Estimation with the Graphical Lasso." *Biostatistics* 9 (3): 432–41. https://doi.org/10.1093/biostatistics/kxm045.

Friedman, Jonathan, and Eric J. Alm. 2012. "Inferring Correlation Networks from Genomic Survey Data." *PLoS Computational Biology* 8 (9): e1002687. https://doi.org/10.1371/journal.pcbi.1002687.

Girvan, M., and M. E. J. Newman. 2002. "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences* 99 (12): 7821–26. https://doi.org/10.1073/pnas.122653799.

Hafemeister, Christoph, and Rahul Satija. 2019. "Normalization and Variance Stabilization of Single-Cell RNA-seq Data Using Regularized Negative Binomial Regression." *Genome Biology* 20 (296). https://doi.org/10.1186/s13059-019-1874-1.

Harrison, Joshua G., W. John Calder, Vivaswat Shastry, and C. Alex Buerkle. 2020. "Dirichlet-Multinomial Modelling Outperforms Alternatives for Analysis of Microbiome and Other Ecological Count Data." *Molecular Ecology Resources* 20 (2): 481–97. https://doi.org/10.1111/1755-0998.13128.

Hayes, Thomas, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, et al. 2025. "Simulating 500 Million Years of Evolution with a Language Model." *Science* 387 (6736): 850–58. https://doi.org/10.1126/science.ads0018.

Hočevar, Tomaž, and Janez Demšar. 2014. "A Combinatorial Approach to Graphlet Counting." *Bioinformatics* 30 (4): 559–65. https://doi.org/10.1093/bioinformatics/btt717.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. "LoRA: Low-Rank Adaptation of Large Language Models." https://arxiv.org/abs/2106.09685.

Humphries, M. D, K Gurney, and T. J Prescott. 2006. "The Brainstem Reticular Formation Is a Small-World, Not Scale-Free, Network." *Proceedings of the Royal Society B: Biological Sciences* 273 (1585): 503–11. https://doi.org/10.1098/rspb.2005.3354.

Iyer, Swami, Timothy Killingback, Bala Sundaram, and Zhen Wang. 2013. "Attack Robustness and Centrality of Complex Networks." *PLoS ONE* 8 (4): e59613. https://doi.org/10.1371/journal.pone.0059613.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, et al. 2021. "Highly Accurate Protein Structure Prediction with AlphaFold." *Nature* 596 (7873): 583–89. https://doi.org/10.1038/s41586-021-03819-2.

Kobak, Dmitry, and George C. Linderman. 2021. "Initialization Is Critical for Preserving Global Data Structure in Both t-SNE and UMAP." *Nature Biotechnology* 39 (2): 156–57. https://doi.org/10.1038/s41587-020-00809-z.

Kurtz, Zachary D., Christian L. Müller, Emily R. Miraldi, Dan R. Littman, Martin J. Blaser, and Richard A. Bonneau. 2015. "Sparse and Compositionally Robust Inference of Microbial Ecological Networks." *PLOS Computational Biology* 11 (5): e1004226. https://doi.org/10.1371/journal.pcbi.1004226.

Lange, Tilman, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. 2004. "Stability-Based Validation of Clustering Solutions." *Neural Computation* 16 (6): 1299–1323. https://doi.org/10.1162/089976604773717621.

Lee, Daniel D., and H. Sebastian Seung. 1999. "Learning the Parts of Objects by Non-Negative Matrix Factorization." *Nature* 401 (6755): 788–91. https://doi.org/10.1038/44565.

Liao, James C., Riccardo Boscolo, Young-Lyeol Yang, Linh My Tran, Chiara Sabatti, and Vwani P. Roychowdhury. 2003. "Network Component Analysis: Reconstruction of Regulatory Signals in Biological Systems." *Proceedings of the National Academy of Sciences* 100 (26): 15522–27. https://doi.org/10.1073/pnas.2136632100.

Liu, Han, Kathryn Roeder, and Larry Wasserman. 2010. "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models." In *Advances in Neural Information Processing Systems*, 23:1432–40.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12). https://doi.org/10.1186/s13059-014-0550-8.

Maaten, Laurens van der, and Geoffrey Hinton. 2008. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research* 9: 2579–2605.

Mahana, Douglas, Chad M. Trent, Zachary D. Kurtz, Nicholas A. Bokulich, Thomas Battaglia, Jennifer Chung, Christian L. Müller, Huilin Li, Richard A. Bonneau, and Martin J. Blaser. 2016. "Antibiotic Perturbation of the Murine Gut Microbiome Enhances the Adiposity, Insulin Resistance, and Liver Disease Associated with High-Fat Diet." *Genome Medicine* 8 (1). https://doi.org/10.1186/s13073-016-0297-9.

McInnes, Leland, John Healy, and James Melville. 2018. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." https://arxiv.org/abs/1802.03426.

McMurdie, Paul J., and Susan Holmes. 2014. "Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible." *PLoS Computational Biology* 10 (4): e1003531. https://doi.org/10.1371/journal.pcbi.1003531.

Meinshausen, Nicolai, and Peter Bühlmann. 2006. "High-Dimensional Graphs and Variable Selection with the Lasso." *The Annals of Statistics* 34 (3): 1436–62. https://doi.org/10.1214/009053606000000281.

———. 2010. "Stability Selection." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72 (4): 417–73. https://doi.org/10.1111/j.1467-9868.2010.00740.x.

Montoya, José M., Stuart L. Pimm, and Ricard V. Solé. 2006. "Ecological Networks and Their Fragility." *Nature* 442 (7100): 259–64. https://doi.org/10.1038/nature04927.

Nearing, Jacob T., Gavin M. Douglas, Molly G. Hayes, Jocelyn MacDonald, Dhwani K. Desai, Nicole Allward, Casey M. A. Jones, et al. 2022. "Microbiome Differential Abundance Methods Produce Different Results Across 38 Datasets." *Nature Communications* 13 (1). https://doi.org/10.1038/s41467-022-28034-z.

Newman, M. E. J. 2006. "Modularity and Community Structure in Networks." *Proceedings of the National Academy of Sciences* 103 (23): 8577–82. https://doi.org/10.1073/pnas.0601602103.

Nguyen, Nam D., and Daifeng Wang. 2020. "Multiview Learning for Understanding Functional Multiomics." *PLOS Computational Biology* 16 (4): e1007677. https://doi.org/10.1371/journal.pcbi.1007677.

Peschel, Stefanie, Christian L Müller, Erika von Mutius, Anne-Laure Boulesteix, and Martin Depner. 2021. "NetCoMi: Network Construction and Comparison for Microbiome Data in r." *Briefings in Bioinformatics* 22 (4). https://doi.org/10.1093/bib/bbaa290.

Rives, Alexander, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, et al. 2021. "Biological Structure and Function Emerge from Scaling Unsupervised

Learning to 250 Million Protein Sequences." *Proceedings of the National Academy of Sciences* 118 (15): e2016239118. https://doi.org/10.1073/pnas.2016239118.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20: 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.

Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6 (2): 461–64. https://doi.org/10.1214/aos/1176344136.

Senior, Andrew W., Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, et al. 2020. "Improved Protein Structure Prediction Using Potentials from Deep Learning." *Nature* 577 (7792): 706–10. https://doi.org/10.1038/s41586-019-1923-7.

Smyth, Gordon K. 2004. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology* 3 (1): 1–25. https://doi.org/10.2202/1544-6115.1027.

Sunagawa, Shinichi, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, et al. 2015. "Structure and Function of the Global Ocean Microbiome." *Science* 348 (6237). https://doi.org/10.1126/science.1261359.

Tam, Vivian, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, and David Meyre. 2019. "Benefits and Limitations of Genome-Wide Association Studies." *Nature Reviews Genetics* 20 (8): 467–84. https://doi.org/10.1038/s41576-019-0127-1.

Tenenbaum, Joshua B., Vin de Silva, and John C. Langford. 2000. "A Global Geometric Framework for Nonlinear Dimensionality Reduction." *Science* 290 (5500): 2319–23. https://doi.org/10.1126/science.290.5500.2319.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58 (1): 267–88. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Tibshirani, Robert, Guenther Walther, and Trevor Hastie. 2001. "Estimating the Number of Clusters in a Data Set via the Gap Statistic." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 63 (2): 411–23. https://doi.org/10.1111/1467-9868.00293.

Tipton, Laura, Christian L. Müller, Zachary D. Kurtz, Laurence Huang, Eric Kleerup, Alison Morris, Richard Bonneau, and Elodie Ghedin. 2018. "Fungi Stabilize Connectivity in the Lung and Skin Microbial Ecosystems." *Microbiome* 6 (1). https://doi.org/10.1186/s40168-017-0393-0.

Townes, F. William. 2020. "Review of Probability Distributions for Modeling Count Data." https://arxiv.org/abs/2001.04343.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. "From Louvain to Leiden: Guaranteeing Well-Connected Communities." *Scientific Reports* 9 (5233). https://doi.org/10.1038/s41598-019-41695-z.

Venter, J. Craig, Mark D. Adams, Eugene W. Myers, Peter W. Li, Richard J. Mural, Granger G. Sutton, Hamilton O. Smith, et al. 2001. "The Sequence of the Human Genome." *Science* 291 (5507): 1304–51. https://doi.org/10.1126/science.1058040.

WATSON, J. D., and F. H. C. CRICK. 1953. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid." *Nature* 171 (4356): 737–38. https://doi.org/10.1038/171737a0.

Watts, Duncan J., and Steven H. Strogatz. 1998. "Collective Dynamics of Small-World Networks." *Nature* 393 (6684): 440–42. https://doi.org/10.1038/30918.

Weiss, Sophie, Zhenjiang Zech Xu, Shyamal Peddada, Amnon Amir, Kyle Bittinger, Antonio Gonzalez, Catherine Lozupone, et al. 2017. "Normalization and Microbial Differential Abundance Strategies Depend Upon Data Characteristics." *Microbiome* 5 (1). https://doi.org/10.1186/s40168-017-0237-y.

Wu, Michael C., Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. 2011. "Rare-Variant Association Testing for Sequencing Data with the Sequence Kernel Association Test." *The American Journal of Human Genetics* 89 (1): 82–93. https://doi.org/10.1016/j.ajhg.2011.05.029.

Yaveroğlu, Ömer Nebil, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. 2014. "Revealing the Hidden Language of Complex Networks." *Scientific Reports* 4 (1). https://doi.org/10.1038/srep04547.

Yu, Guan. 2009. "Variance Stabilizing Transformations of Poisson, Binomial and Negative Binomial Distributions." *Statistics &Amp; Probability Letters* 79 (14): 1621–29. https://doi.org/10.1016/j.spl.2009.04.010.

Yuan, M., and Y. Lin. 2007. "Model Selection and Estimation in the Gaussian Graphical Model." *Biometrika* 94 (1): 19–35. https://doi.org/10.1093/biomet/asm018.

Zhou, Jian, and Olga G Troyanskaya. 2015. "Predicting Effects of Noncoding Variants with Deep Learning–Based Sequence Model." *Nature Methods* 12 (10): 931–34. https://doi.org/10.1038/nmeth.3547.