Basic Stats for Omics

Contents

Basic Statistics for Omics: A Hands-On Coding Guide for Biology Students 1
- Introduction . . . 1
- About This Book . . . 1
- Module 1: Building Your Foundation . . . 2
  - Chapter 1: Programming Basics & Data Wrangling . . . 2
  - Chapter 2: Binomial Test . . . 2
  - Chapter 3: Fisher's Exact Test . . . 3
  - Chapter 4: T-test vs Mann-Whitney U . . . 3
  - Chapter 5: ANOVA vs Kruskal-Wallis . . . 3
  - Chapter 6: Pearson vs Spearman Correlation . . . 4
- Module 2: Handling High-Dimensional Data . . . 4
  - Chapter 7: PCA & Batch Effect Correction . . . 4
  - Chapter 8: Clustering & NMF (Unsupervised Learning) . . . 4
  - Chapter 9: Linear Models & GLM (Regression Analysis) . . . 5
- Module 3: Ensuring Your Results Are Real . . . 5
  - Chapter 10: Multiple Testing Correction (FDR, Bonferroni) . . . 5
  - Chapter 11: Power Analysis & Sample Size Calculation . . . 6
- How to Use This Book . . . 6

Basic Statistics for Omics: A Hands-On Coding Guide for Biology Students

Introduction

Welcome to your journey into statistical thinking for modern biology! This book bridges the gap between biological concepts and the computational skills you need to analyze real genomic data. Whether you're curious about gene expression patterns, protein abundance, or DNA sequences, you'll learn to extract meaningful insights from complex datasets.

Each chapter builds progressively, starting with fundamental data manipulation and advancing to sophisticated analysis techniques used in current research. By the end, you'll be equipped to critically evaluate methods sections in scientific papers and conduct your own exploratory analyses.

About This Book

This book is designed as a collection of interactive Google Colab tutorials, allowing you to learn by doing. You'll work through practical examples from genomics, transcriptomics, and proteomics—getting hands-on experience with the statistical approaches that drive modern biological discoveries.

Who This Book Is For: - Undergraduate biology students ready to add computational skills to their toolkit - Research interns beginning work in omics laboratories - Anyone curious about how biologists turn raw

data into biological insights

What You'll Need: – A Google account (for accessing Colab notebooks—no installation required!) – Basic familiarity with genetics concepts (genes, DNA, proteins) – Enthusiasm for learning (we'll teach you the rest)

Recommended Preparation: If you'd like to refresh your genetics and genomics background, consider reviewing Human Genetics before diving in.

A Note on Development: This book emerged from our laboratory's need to train incoming students and interns. While the core content has been tested with multiple student cohorts, it hasn't undergone formal peer review yet. If you spot errors or have suggestions for improvement, we welcome your feedback via GitHub Issues. Your input helps make this resource better for future learners!

☐ Language Choice: R or Python?

All chapters are available in **both R and Python**! Choose the language that best fits your needs:

- **R (Tidyverse)**: Traditional choice for bioinformatics and statistical genomics. Excellent visualization with ggplot2.
- **Python (Pandas/SciPy)**: Versatile across data science, machine learning, and bioinformatics. Great for integrating with other tools.

Our recommendation: Pick one and stick with it throughout the course. Both teach the same statistical concepts – the syntax is just different. If you're unsure, try Chapter 1 in both languages and see which feels more comfortable!

Module 1: Building Your Foundation

Chapter 1: Programming Basics & Data Wrangling

☐ R Version | Python Version

Why this matters: Before you can answer any biological question with data, you need to organize it. Real research involves messy spreadsheets – gene expression data in one file, patient information in another. This chapter teaches you how to bring them together cleanly.

What you'll learn: Think of this as learning the language. In R, you'll use `dplyr`; in Python, you'll use `pandas`. Both teach you to filter, sort, and join datasets – essential skills for every chapter that follows.

Biological motivation: Imagine you have RNA–seq data from 100 patients with different cancer types. You also have a separate file with their ages, treatments, and survival outcomes. How do you combine these to ask meaningful questions? That's what we solve here.

Chapter 2: Binomial Test

☐ R Version | Python Version

Why this matters: Many biological questions are about counting: How many mutations? How many cells respond to treatment? The binomial test helps you determine if what you're seeing is more than just chance.

What you'll learn: This is your first hypothesis test. You'll learn to ask "Is this surprisingly high or low compared to what we'd expect by chance?"

Biological motivation: Suppose you find 15 de novo mutations in a gene among 20 autism patients. Is that a lot? Should we investigate this gene further? The binomial test gives you a principled answer. You'll also learn why we need to account for confounders like paternal age – older fathers have more mutations anyway.

Connection to next chapter: You're testing one gene here. But what if you want to test whether an entire set of genes (like "synaptic genes") is enriched? That's Fisher's Exact Test.

Chapter 3: Fisher's Exact Test

☐ R Version | Python Version

Why this matters: Gene set enrichment is everywhere in omics papers. When you find 100 differentially expressed genes, you want to know: "Are these related? Do they belong to specific pathways?"

What you'll learn: How to properly test if a set of genes is over-represented in your results, and why choosing the right background gene set is crucial for getting the right answer.

Biological motivation: You identified 50 genes mutated in cancer patients. Among them, 10 are DNA repair genes. That sounds like a lot – but is it? There are 20,000 human genes, and maybe 500 are DNA repair genes. Fisher's test tells you if 10 out of 50 is truly enriched. More importantly, you'll see how using the wrong background (like "all human genes" when you only tested 5,000) completely changes your conclusion.

Connection to next chapter: So far we've dealt with counts. But what about measurements – like gene expression levels or protein concentrations? That's where t-tests come in.

Chapter 4: T-test vs Mann-Whitney U

☐ R Version | Python Version

Why this matters: When you measure something continuous (expression level, cell size, tumor volume), you need different tools. But not all data behaves nicely – some is skewed, some has outliers.

What you'll learn: How to check if your data meets the assumptions for a t-test using Q-Q plots, and when to switch to the non-parametric Mann-Whitney U test.

Biological motivation: You measured protein X in healthy vs. diseased patients. Is it higher in disease? A t-test seems obvious, but what if most diseased patients have normal levels and just a few have extremely high values? The data isn't bell-shaped anymore. Q-Q plots help you see this visually, and Mann-Whitney handles it properly.

Connection to next chapter: T-tests compare two groups. But diseases often have multiple subtypes (Stage I, II, III cancer). How do we compare them all at once without doing multiple t-tests and inflating our error rate? That's ANOVA.

Chapter 5: ANOVA vs Kruskal-Wallis

☐ R Version | Python Version

Why this matters: Comparing multiple groups simultaneously is incredibly common – cancer stages, different treatments, multiple cell types. Doing many t-tests leads to false discoveries.

What you'll learn: How ANOVA elegantly handles multiple groups by partitioning variance, and how to follow up with post-hoc tests (Tukey's HSD or Dunn's test) to find which specific groups differ.

Biological motivation: You're testing a new drug at three doses (low, medium, high) plus a control. You want to know if dose matters, and if so, which doses are actually different from control. ANOVA tells you "yes, there's an effect," and post-hoc tests tell you "high dose is different from control, but low isn't."

Connection to next chapter: All these tests look at differences between groups. But what about relationships? Does gene A's expression correlate with gene B's? That's correlation analysis.

Chapter 6: Pearson vs Spearman Correlation

☐ R Version | Python Version

Why this matters: Biology is full of relationships. Does enzyme activity increase with temperature? Does tumor size correlate with survival time? Correlation quantifies these relationships.

What you'll learn: The critical difference between linear relationships (Pearson) and monotonic relationships (Spearman), and why correlation ≠ causation. You'll always start with a scatter plot – never trust a correlation coefficient alone.

Biological motivation: You notice that as gene X expression goes up, gene Y expression goes up. Are they co-regulated? Maybe. But you'll learn to be cautious – both might just respond to the same underlying factor (like cell cycle stage). You'll also see why arbitrary cutoffs like "correlation > 0.7 is strong" are misleading – it completely depends on your context and sample size.

Connection to Module 2: You've now mastered univariate and bivariate analyses. But omics datasets have thousands of genes measured simultaneously. How do you even visualize that? Enter PCA.

Module 2: Handling High-Dimensional Data

Chapter 7: PCA & Batch Effect Correction

☐ R Version | Python Version

Why this matters: You can't visualize 20,000 dimensions. PCA reduces your data to 2–3 dimensions while preserving the most important patterns. It's also your first line of defense for quality control.

What you'll learn: How PCA works conceptually (finding directions of maximum variance), how to interpret PC plots, and critically – how to spot and fix batch effects that can completely ruin your analysis.

Biological motivation: You collected samples from cancer patients and healthy controls over two years. When you do PCA, you see two clusters – but they're not cancer vs. healthy, they're Year 1 vs. Year 2! The sequencing platform changed, creating a batch effect stronger than your biological signal. You'll learn to diagnose this problem and fix it with ComBat before any real analysis.

Connection to next chapter: PCA shows you overall structure, but what if there are hidden subgroups within your data? Clustering methods find them.

Chapter 8: Clustering & NMF (Unsupervised Learning)

☐ R Version | Python Version

Why this matters: Diseases aren't always simple categories. "Breast cancer" actually contains multiple molecular subtypes with different treatments and outcomes. Clustering helps discover these hidden groups.

What you'll learn: Hierarchical clustering (builds a tree of similarities), K-means (partitions into K groups), and NMF (finds interpretable biological signatures).

Biological motivation: You have gene expression data from 200 breast cancer tumors. Instead of assuming they're all the same, clustering might reveal 4 subtypes – some driven by hormone receptors, others by cell proliferation. NMF goes further and tells you which genes define each subtype, giving you biological insight, not just groups. This is how cancer subtypes like Luminal A, Luminal B, Her2+, and Basal were originally discovered.

Connection to next chapter: Clustering is unsupervised – you don't tell it what to look for. But what if you want to model how gene expression predicts a clinical outcome? That's regression.

Chapter 9: Linear Models & GLM (Regression Analysis)

☐ R Version | Python Version

Why this matters: Regression lets you model relationships while controlling for confounders. Does gene X predict survival after accounting for age, gender, and tumor stage?

What you'll learn: Linear models for continuous outcomes and generalized linear models for other types (counts, binary outcomes). You'll understand what regression coefficients mean and how to check if your model assumptions hold.

Biological motivation: You want to know if a gene's expression level predicts patient survival time. Simple correlation isn't enough because older patients and those with advanced-stage disease survive less time anyway. Multiple regression lets you ask: "Holding age and stage constant, does this gene still matter?" This is the foundation of nearly all modern omics analysis.

Connection to Module 3: You now have powerful analysis tools. But there's a catch – when you test thousands of genes, some will look significant purely by chance. Module 3 addresses this and helps you design robust studies.

Module 3: Ensuring Your Results Are Real

Chapter 10: Multiple Testing Correction (FDR, Bonferroni)

☐ R Version | Python Version

Why this matters: Test 20,000 genes at $p < 0.05$, and you'll get ~1,000 false positives even if nothing is real. This chapter prevents you from publishing noise.

What you'll learn: The multiple testing problem, why it's severe in omics, and two main solutions: Bonferroni (very strict) and FDR/Benjamini-Hochberg (more practical).

Biological motivation: You test every human gene for differential expression between cases and controls. You find 500 "significant" genes at $p < 0.05$. Exciting! But wait – 20,000 tests × 0.05 = 1,000 expected false positives by pure chance. Maybe your 500 "hits" are all noise. FDR correction ensures that if you call 100 genes significant, only ~5 are likely false positives (at FDR 5%). This is why every omics paper reports "adjusted p-values" or "q-values."

Connection to next chapter: Even after correcting for multiple testing, you might not find anything significant – not because there's no effect, but because your sample size is too small. Power analysis helps you plan better studies.

Chapter 11: Power Analysis & Sample Size Calculation

□ R Version | Python Version

Why this matters: Underpowered studies waste time and money. Overpowered studies are inefficient. This chapter helps you design studies that can actually detect real effects.

What you'll learn: The concepts of Type I and Type II errors, statistical power, effect size, and how to calculate the sample size you need before collecting data.

Biological motivation: You want to know if treatment increases protein levels by 20% on average. How many mice do you need? Too few and you'll miss the effect even if it's real. Too many and you're wasting animals and resources. Power analysis gives you the answer: "With 15 mice per group, you have 80% power to detect a 20% increase if it exists." This isn't just good science – grant reviewers and ethics committees expect this calculation.

The full circle: With proper power calculation (Chapter 11), you collect enough samples. You organize them correctly (Chapter 1), test hypotheses appropriately (Chapters 2–6), visualize high-dimensional patterns (Chapters 7–8), model relationships (Chapter 9), and correct for multiple testing (Chapter 10). You're now equipped to do real omics research.

How to Use This Book

Choose your language first: All 11 chapters are available in both R and Python. Pick one and stick with it for consistency. The statistical concepts are identical – only the syntax differs.

Linear path: If you're new to statistics, go chapter by chapter. Each builds on previous concepts.

Selective reading: If you have some background, jump to topics you need: – Need to analyze RNA-seq data? → Chapters 1, 7, 10 – Planning an experiment? → Chapter 11 – Found "significantly enriched" pathways but unsure about the methods? → Chapter 3

Before starting: All notebooks run in Google Colab – you just need a Google account and internet connection. No installation required.

Practice philosophy: Each chapter follows a 30% theory / 70% practice approach. You'll write code, visualize data, and interpret results – just like real research. Don't just read; run the code, change parameters, and see what happens.

Remember: Statistics isn't about memorizing formulas. It's about asking clear questions and having principled ways to answer them. These tools let you turn your biological curiosity into reproducible knowledge.