目录

Chapter 1

项目介绍与学习目标

1.1　项目背景

二十一世纪以来，高通量测序技术（Next-Generation Sequencing, NGS）的飞速发展与测序成本的指数级下降，使生物学研究全面迈入”大数据”时代。如今，一台桌面级测序仪（如 Illumina MiSeq）即可在数小时内产出上千万条、总量达数亿碱基的 DNA 序列数据。面对如此规模的数据，传统的”手工比对、肉眼检查”早已无能为力，**生物信息学（Bioinformatics）**应运而生，成为连接测序数据与生物学发现之间不可或缺的桥梁。

生物信息学是一门交叉学科，它综合运用计算机科学、数学与统计学的方法，对生物数据进行存储、处理、分析与解读。对一名生命科学或数据科学方向的学生而言，掌握生物信息学的核心分析流程，已从”加分项”逐渐演变为”基本功”。

然而，入门生物信息学常常面临两大困难：其一，概念抽象、工具繁多，初学者容易在命令行的汪洋中迷失方向；其二，缺乏一条**完整、可复现、数据量适中的主线**来串联零散的知识点。本实验项目正是为解决这一痛点而设计。

1.1.1　为什么选择”大肠杆菌重测序”作为入门项目

本项目以**大肠杆菌（*Escherichia coli*）全基因组重测序与变异检测**为主线。选择这一主题，是基于以下考量：

- **基因组小、计算友好**：大肠杆菌 K-12 MG1655 的参考基因组仅约 4.6 Mb（兆碱基），不到人类基因组的千分之一点五。所有分析步骤（质控、比对、变异检测）都可在普通个人电脑上数分钟至数十分钟内完成，适合课堂环境与入门练习。
- **数据公开、真实可复现**：本项目所用的测序数据与参考基因组均来自国际公共数据库（ENA、NCBI），任何学生都可免费下载、复现全部结果。
- **流程完整、代表性强**：从原始测序数据到变异位点，覆盖了生物信息学最核心的一条”黄金流程”——**数据获取 → 质量控制 → 序列比对 → 比对后处理 → 变异检测 → 变异注释**。这条流程的每个环节都是通用技能，可直接迁移到人类、动植物、微生物等其他物种的分析中。
- **结果可解读、有科学内涵**：通过对比”同源对照样本”与”野生分离株样本”，学

生既能掌握变异检测的技术细节，又能深入理解测序深度、质量控制与假阳性之间的本质关系。

1.2　项目目标

1.2.1　知识目标

完成本项目后，学生应能：

1. 理解高通量测序（Illumina）的基本原理与数据特征；
2. 掌握 FASTA、FASTQ、SAM/BAM、VCF 等核心生物信息学文件格式的结构与含义；
3. 理解序列比对（BWA-MEM）与变异检测（bcftools）背后的基本算法思想；
4. 理解 Phred 质量分数、测序深度、覆盖度、假阳性等关键概念；
5. 掌握变异注释（SnpEff）的意义与变异类型的分类方法。

1.2.2　技能目标

完成本项目后，学生应能够：

1. 熟练使用 Linux 命令行完成文件的查看、压缩、统计等基本操作；
2. 使用 conda/mamba 搭建可复现的生物信息学分析环境；
3. 从公共数据库下载参考基因组与测序数据，并校验数据完整性；
4. 使用 FastQC 与 fastp 进行测序数据质量评估与过滤；
5. 使用 BWA-MEM 将测序读段比对到参考基因组，并使用 samtools 完成排序、去重、统计；
6. 使用 bcftools 进行变异位点检测与过滤，并解读 VCF 结果；
7. 使用 SnpEff 对变异进行功能注释与分类统计；
8. 撰写规范的分析报告，以图表与文字呈现分析结果。

1.2.3　能力目标（可迁移的通用能力）

- **工程化思维**：学会通过目录组织、脚本化、记录命令等方式管理一个可复现的分析项目；
- **排错能力**：学会阅读错误信息、查阅软件文档与社区资源（Biostars、SeqAnswers、GitHub Issues）独立解决问题；

- **数据素养**：学会对分析结果保持批判性审视，区分” 技术噪音” 与” 真实信号”。

1.3　项目总览

1.3.1　分析流程

本项目的主干流程如图 1.1 所示，共分为七个模块。整个流程是一个典型的” 变异检测流水线”（Variant Calling Pipeline）。

图 1.1：本项目主干分析流程

1.3.2　数据集方案

本项目提供**两套真实测序数据集**，形成” 主实验 + 延伸实验” 的递进设计：

项目	主实验	延伸实验
数据集编号	ERR15404827	DRR063436
样本菌株	大肠杆菌 K-12 MG1655	大肠杆菌野生分离株
测序平台	Illumina MiSeq	Illumina MiSeq
读长	250 bp 双端	75 bp 双端
测序数据量	约 124 Mb（约 49.7 万对读段）	约 18 Mb（约 12.4 万对读段）
估算覆盖深度	约 27×	约 4×
压缩数据大小	约 89 MB	约 13.6 MB
参考基因组	大肠杆菌 K-12 MG1655（RefSeq GCF_000005845.2）	
与参考的关系	**同源对照**（几乎无真实变异）	**异源**（存在真实 SNP）
教学重点	完整流程 + 假阳性过滤 + 阴性对照思维	真实变异检出 + 深度对比

表 1.1: 项目两套数据集的对比

数据方案说明

主实验使用**与参考基因组同源的 MG1655 样本**（深度 27×，读长 250 bp），目的是：让初学者在” 最干净、最可信” 的条件下完整走通流程，并学会**如何正确过滤假阳性**——因为理论上同源重测序应检测到极少量真实变异，凡是大规模检出的” 变异” 多半是测序错误或比对错误。

延伸实验改用**野生分离株**（深度 4×，读长 75 bp），与 MG1655 参考基因组之间真实存在单核苷酸多态性（SNP），学生可以观察到” 真实变异” 被检出的过程，并体会**测序深度对变异检测的影响**。

1.4　预备知识要求

本项目面向本科二、三年级以上或具有同等基础的初学者，建议具备以下先修知识：

- **生物学基础**：了解 DNA 双螺旋结构、碱基配对（A-T、C-G）、中心法则（DNA → RNA → 蛋白质）的基本概念；
- **计算机基础**：会使用 Linux 基本命令（`cd`、`ls`、`mkdir`、`cp`、`mv`），了解绝对路径与相对路径的概念；
- **英语基础**：能借助词典阅读软件帮助文档与数据库页面（本项目的命令与报错信息均为英文）。

零基础也不要紧

如果你从未接触过 Linux 命令行，建议在开始前花 2–3 小时完成一个 Linux 入门速成教程（如菜鸟教程 Linux 篇或软件嘉年华 Linux 基础），重点掌握文件与目录操作、通配符、管道（|）与重定向（>、>>）即可，本项目会在每个步骤给出

完整的、可直接复制的命令。

Chapter 2

实验环境与软件清单

2.1　硬件与系统要求

本项目的全部计算任务可在普通个人电脑上完成，推荐配置如下：

项目	推荐配置
操作系统	Linux（推荐 Ubuntu 20.04 及以上）或 macOS；Windows 建议使用 WSL2（Windows Subsystem for Linux）
CPU	4 核及以上（多线程工具可加速比对与质控）
内存	8 GB 及以上
硬盘	至少 20 GB 可用空间（软件环境 + 数据 + 中间文件）
网络	可访问 NCBI/ENA 等国际数据库（下载数据）

表 2.1：硬件与系统推荐配置

Windows 用户特别注意

生物信息学工具绝大多数为 Linux/macOS 原生命令行程序。Windows 用户强烈建议安装 **WSL2（Ubuntu）**后再进行本项目，不要直接在 Windows 的 CMD 或 PowerShell 中运行。WSL2 的安装与配置请参考微软官方文档或本项目附录 A.3。

2.2　所需软件清单

本项目所需的核心软件如表 2.2 所示。所有软件均通过 **conda**（或更快的 `mamba`）统一安装，以保证版本一致性、可复现性与环境隔离。

软件	验证版本	用途	许可证
Miniconda	24.x	包与环境管理器	BSD
mamba	1.5.x	更快的 conda 替代品	BSD
FastQC	0.12.1	测序数据质量评估	GPL
fastp	0.23.4	质量过滤与修剪	MIT
BWA	0.7.18	读段比对（BWA-MEM）	GPL
samtools	1.20	SAM/BAM 处理与统计	MIT/BSD
bcftools	1.20	变异检测与过滤	MIT/BSD
SnpEff	5.2	变异功能注释	LGPL
tabix	1.20	索引压缩的表格文件	MIT/BSD
htslib	1.20	底层库(samtools/bcftools 依赖)	MIT/BSD

表 2.2: 核心软件清单

关于版本号

表中" 验证版本" 为编写本教程时经过验证的稳定版本，用于确保命令行为一致。实际安装时，conda 通常会解析出更新的版本，只要**大版本号一致**（如 samtools 1.x、bcftools 1.x），命令即可通用。若需完全复现，可使用附录 A.2 提供的环境文件锁定版本。

2.3　安装与环境配置

2.3.1　第一步：安装 Miniconda

Miniconda 是一个轻量级的 conda 发行版，用于安装、运行和管理 Python 及各类命令行软件包。在 Linux 终端执行以下命令完成安装：

```
# 下载 Miniconda 安装脚本
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
# 运行安装脚本
bash Miniconda3-latest-Linux-x86_64.sh
# 按提示操作：回车阅读协议 -> 输入 yes 接受 -> 回车确认安装路径 -> 输入 yes 初始化
```

Listing 2.1: 安装 Miniconda（Linux x86_64）

安装完成后，关闭并重新打开终端（或执行 `source ~/.bashrc`），使 conda 生效。验证安装：

```
conda --version
# 输出形如：conda 24.11.0
```

Listing 2.2: 验证 conda 安装

2.3.2　第二步：配置 conda 源（可选但推荐）

为加快软件下载速度，可将 conda 的默认源切换为国内镜像（如清华大学 TUNA 镜像）。执行以下命令：

```
# 添加清华镜像
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
conda config --set show_channel_urls yes
```

Listing 2.3: 配置 conda 国内镜像源

为什么需要 conda-forge 与 bioconda

绝大多数生物信息学软件由社区维护在 **Bioconda** 频道中，而 Bioconda 又依赖 **conda-forge** 提供的基础库。因此，安装生信工具时必须同时配置这两个频道，并确保 `conda-forge` 与 `bioconda` 的优先级正确（通常 conda-forge 在前）。

2.3.3　第三步：创建独立的分析环境

最佳实践是为每个项目创建**独立的 conda 环境**，避免不同项目之间的依赖冲突。本项目创建一个名为 `bioinfo` 的环境：

```
# 创建环境（-y 自动确认，-c 指定频道）
conda create -n bioinfo -y -c conda-forge -c bioconda \
    mamba fastqc fastp bwa samtools bcftools snpeff tabix

# 激活环境
conda activate bioinfo

# 验证核心工具是否可用
bwa 2>&1 | head -1
samtools --version | head -1
```

```
bcftools --version | head -1
```

Listing 2.4: 创建并激活分析环境

用 mamba 加速安装

conda 的依赖解析速度较慢，建议在环境内先安装 `mamba`，之后用 `mamba install` 替代 `conda install`，速度可提升数倍。本项目上述命令中已包含 mamba。

2.3.4　第四步：下载 SnpEff 数据库

SnpEff 注释变异时需要物种对应的注释数据库。大肠杆菌 K-12 MG1655 的 SnpEff 数据库名为 `ASM584v2`（对应 RefSeq GCF_000005845.2）。在激活环境后执行：

```
# 查看可用的大肠杆菌数据库
snpEff databases | grep -i coli | head

# 下载 MG1655 (ASM584v2) 数据库
snpEff download -v ASM584v2
```

Listing 2.5: 下载 SnpEff 大肠杆菌数据库

SnpEff 数据库名可能变化

不同 SnpEff 版本中，MG1655 数据库的命名可能略有差异（如 `ASM584v2`、`GCF_000005845.2`）。执行 `snpEff databases | grep -i "K-12"` 确认准确名称。若下载失败，请参考附录 B 的排错方法。

2.4　项目目录结构

良好的目录组织是可复现分析的基础。请在开始实验前，按以下结构创建项目目录：

```
mkdir -p ecoli_reseq/{data,ref,results/{qc,align,variants,annotation},scripts,
    report}
cd ecoli_reseq
tree -L 2   # 若未安装 tree 可省略此命令
```

Listing 2.6: 创建项目目录结构

创建后的目录结构与用途如表 2.3 所示：

目录	用途
ecoli_reseq/	项目根目录
data/	存放测序数据（FASTQ 文件）
ref/	存放参考基因组及相关索引
results/qc/	质控结果（FastQC 报告、fastp 输出）
results/align/	比对结果（SAM/BAM 文件）
results/variants/	变异检测结果（VCF 文件）
results/annotation/	变异注释结果
scripts/	存放分析脚本
report/	存放最终分析报告

表 2.3: 项目目录结构说明

路径规范建议

始终使用相对路径或变量引用文件，避免在命令中写死绝对路径（如 /home/zhangsan/...），这样脚本才能在不同电脑上通用。本项目后续命令默认在 ecoli_reseq/ 目录下执行。

Chapter 3

测序技术基础

本章导读

在动手分析数据之前，理解” 数据从何而来” 至关重要。测序数据的质量特征、误差模式都直接决定了后续分析策略的选择。本章介绍 DNA 测序技术的发展脉络，并重点讲解本项目所使用的 Illumina 测序原理及其数据特征。

3.1　DNA 测序技术简史

3.1.1　第一代测序：Sanger 测序

1977 年，Frederick Sanger 发明了链终止法测序（Sanger sequencing），奠定了现代 DNA 测序的基础。其核心原理是” 双脱氧核苷酸终止”：在 DNA 合成反应中加入少量四种双脱氧核苷三磷酸（ddNTP），它们缺少 3′-羟基，一旦被掺入 DNA 链，链的延伸即告终止。通过凝胶电泳分离不同长度的终止片段，即可” 读出” 碱基序列。

Sanger 测序读长可达 800–1000 bp，准确率极高（>99.9%），至今仍是” 金标准”，广泛用于临床基因检测的验证。但其通量低、成本高，无法满足全基因组规模的测序需求。

3.1.2　第二代测序：高通量/大规模并行测序

2005 年前后，以 Roche 454、Illumina（Solexa）、ABI SOLiD 为代表的第二代测序（NGS）技术相继问世。它们的共同特征是**大规模并行化**——同时对数百万乃至数十亿条 DNA 片段进行测序，将单碱基测序成本降低了数个数量级。其中 Illumina 平台凭借高通量、低成本、低错误率的综合优势，成为目前应用最广泛的测序平台。

3.1.3　第三代测序：单分子实时测序

以 PacBio（单分子实时测序，SMRT）和 Oxford Nanopore（纳米孔测序）为代表的第三代测序，无需 PCR 扩增即可对单条 DNA 分子进行实时测序，读长可达数十 kb 甚至 Mb 级，但单碱基错误率相对较高（通常 5–15%）。第三代测序在基因组组装、结

构变异检测、甲基化分析等领域优势明显。

代际	代表平台	读长	主要特点
第一代	Sanger	800–1000 bp	准确率极高，通量低，成本高
第二代	Illumina	50–300 bp	高通量、低成本、低错误率（<1%）
第三代	PacBio / Nanopore	10 kb–Mb 级	超长读长，错误率较高，实时测序

表 3.1: 三代测序技术对比

3.2 Illumina 测序原理

本项目的数据由 Illumina MiSeq 平台产生，其测序过程可概括为" 桥式扩增 + 边合成边测序" 两大步骤。

3.2.1 文库制备

首先将基因组 DNA 随机打断为约几百 bp 的小片段（fragment），在片段两端连接上已知序列的**接头（adapter）**。这些接头含有与测序芯片（flow cell）表面互补的序列，是后续扩增与测序引物结合的锚点。

3.2.2 桥式 PCR 扩增（簇生成）

将文库加载到 flow cell 上，DNA 片段通过接头与芯片表面的互补寡核苷酸结合。随后进行" 桥式 PCR": 每条单链 DNA 分子在芯片表面反复折叠扩增，最终形成由同一模板扩增而来的 **DNA 簇（cluster）**。每个簇包含约 1000 条相同的 DNA 拷贝，为后续测序提供足够强的荧光信号。

3.2.3 边合成边测序（Sequencing by Synthesis, SBS）

测序时，依次加入带有**可逆终止子**和**荧光标记**的四种 dNTP。每一轮反应中:

1. 与模板互补的 dNTP 被 DNA 聚合酶掺入，其余 dNTP 被洗去;
2. 激发荧光，用相机记录每个簇发出的颜色，据此判断该位置掺入的碱基;
3. 切除荧光基团与终止子，进入下一轮合成。

如此循环，每次读出一个碱基，最终获得每个簇的序列，即一条**测序读段（read）**。

3.2.4　单端测序与双端测序

- **单端测序（Single-end, SE）**：只从 DNA 片段的一端开始测序，每条片段产生 1 条 read；
- **双端测序（Paired-end, PE）**：先从片段一端测序，再翻转到另一端测序，每条片段产生 2 条 read（称为 read1 和 read2，或 R1/R2），两者之间的物理距离（插入片段长度）已知。

为什么双端测序更好?

双端测序提供了额外的**位置信息**：两条 read 的相对位置与距离已知，这能显著提高比对准确性（尤其是重复区、结构变异区），也有助于跨越短的重复序列。本项目两个数据集均为双端测序。

3.3　测序质量与 Phred 分数

测序仪在读碱基时会根据荧光信号强度等指标，为每个碱基赋予一个**质量分数**，用于衡量”该碱基被读错”的概率。质量分数通常用 **Phred 分数**（记作 Q）表示，其定义为：

$$Q = -10 \log_{10} P \tag{3.1}$$

其中 P 为该碱基被读错的概率。例如，$Q = 30$ 表示错误概率为 $10^{-3} = 0.1\%$，即准确率 99.9%。常用的质量阈值：

Phred 分数	错误概率	准确率	常用标记
Q10	$10^{-1} = 10\%$	90%	低质量
Q20	$10^{-2} = 1\%$	99%	一般合格线
Q30	$10^{-3} = 0.1\%$	99.9%	高质量
Q40	$10^{-4} = 0.01\%$	99.99%	极高

表 3.2: Phred 质量分数对照表

在 FASTQ 文件中，质量分数以 ASCII 字符编码存储（详见第 4.2 节）。理解 Phred 分数是理解后续”质量控制（QC）”步骤的关键。

记忆技巧

$Q20$ 读作”Q twenty”，含义是”每 100 个碱基中约 1 个可能读错”。业界常用 $Q30$ 比例（即 $Q \geq 30$ 的碱基占比）作为评价测序质量的核心指标。

Chapter 4

核心生物信息学文件格式

本章导读

文件格式是生物信息学的”通用语言”。掌握 FASTA、FASTQ、SAM/BAM、VCF 四种核心格式的结构，是阅读数据、排查问题、编写脚本的前提。本章逐一详解这些格式。

4.1　FASTA 格式：序列的存储

FASTA 是最简单的序列存储格式，用于保存参考基因组、基因序列、蛋白质序列等。其规则：

- 每条序列由两行或多行组成：第一行以 > 开头，为**描述行（header）**，包含序列名称等信息；后续行为**序列本身**；
- 序列行可包含换行，但通常每行不超过一定长度（如 60、80 或整条染色体不换行）。

```
>NC_000913.3 Escherichia coli str. K-12 substr. MG1655, complete genome
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA
...
```

Listing 4.1: FASTA 格式示例

FASTA 中的序列名

参考基因组 FASTA 中，每个序列（染色体、质粒或 contig）都有一个唯一标识符（如 NCBI 的 `NC_000913.3`）。这个标识符必须与后续比对、变异检测、注释所使用的标识符**完全一致**，否则会因”染色体名不匹配”导致工具报错或结果异常。

4.2　FASTQ 格式：测序读段与质量

FASTQ 是存储测序读段及其质量分数的标准格式，每条 read 占据**四行**：

1. **第 1 行（标识行）**：以 `@` 开头，包含 read 的唯一标识符与描述信息；
2. **第 2 行（序列行）**：碱基序列（A/T/C/G/N，N 表示不确定碱基）；
3. **第 3 行（分隔行）**：以 `+` 开头，可重复标识信息或为空；
4. **第 4 行（质量行）**：与第 2 行等长的质量字符，每个字符编码对应碱基的 Phred 分数。

```
@ERR15404827.1 1/1
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATT
+
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

Listing 4.2: FASTQ 格式示例（一条 read）

4.2.1　质量字符的编码

质量行中的字符是 Phred 分数加上一个固定偏移量后转换成的 ASCII 字符。最常见的两种编码体系：

- **Phred+33（Sanger/Illumina 1.8+）**：质量字符 = ASCII($Q+33$)。例如 $Q=40$ 对应字符 `I`（ASCII 73）。这是目前绝大多数数据的编码方式。
- **Phred+64（旧版 Illumina 1.3–1.7）**：质量字符 = ASCII($Q+64$)，现已淘汰。

如何一眼判断编码

若质量行中出现了小写字母或 `!`、`#`、`$` 等字符，通常是 Phred+33；若最低字符为 `@`（ASCII 64）且无更低字符，则可能是 Phred+64。现代数据几乎都是 Phred+33，FastQC 也能自动检测。

4.3　SAM/BAM 格式：比对结果的存储

SAM（Sequence Alignment/Map）是存储”读段比对到参考基因组”结果的文本格式；BAM 是其二进制压缩版本，占用空间更小、支持随机访问（需配合索引）。两者内容等价，可用 samtools 相互转换。

4.3.1　SAM 的整体结构

SAM 文件由两部分组成：

- **头部区（header）**：以 `@` 开头的行，记录文件版本（`@HD`）、参考序列信息（`@SQ`，含序列名与长度）、比对程序（`@PG`）等元数据；
- **比对区（alignment records）**：每行一条 read 的比对结果，由 11 个必选字段 + 可选字段组成。

4.3.2　11 个必选字段

编号	字段名	含义
1	QNAME	read 名称（与 FASTQ 标识行对应）
2	FLAG	位标志（bitwise flag），记录比对状态，见下文
3	RNAME	比对到的参考序列名（染色体/contig）
4	POS	比对起始位置（1-based，最左端）
5	MAPQ	比对质量分数（Phred 尺度）
6	CIGAR	描述比对细节的紧凑字符串，见下文
7	RNEXT	mate（配对 read）所在的参考序列名
8	PNEXT	mate 的比对位置
9	TLEN	模板（插入片段）长度
10	SEQ	read 的碱基序列
11	QUAL	read 的碱基质量（与 FASTQ 质量行一致）

表 4.1: SAM 格式 11 个必选字段

4.3.3　FLAG 位标志

FLAG 字段用一个整数编码多种比对状态，采用二进制位运算。例如：

- `0x1`（十进制 1）：该 read 是双端测序中的一条；
- `0x2`（十进制 2）：该 read 与 mate 都正确比对（properly aligned）；
- `0x4`（十进制 4）：该 read 未比对到参考基因组；
- `0x8`（十进制 8）：该 read 的 mate 未比对；
- `0x10`（十进制 16）：该 read 比对到参考的负链（反向互补）；
- `0x40`（十进制 64）：该 read 是 read1（R1）；
- `0x80`（十进制 128）：该 read 是 read2（R2）。

FLAG 值是各状态对应数值的**和**。例如，FLAG=99（$1+2+32+64$）表示：双端、与 mate 均正确比对、read1。理解 FLAG 是使用 `samtools view -f/-F` 过滤 read 的基础。

4.3.4　CIGAR 字符串

CIGAR（Concise Idiosyncratic Gapped Alignment Report）紧凑地描述 read 相对参考序列的比对方式。常用操作符：

操作符	含义
M	匹配或错配（比对上的碱基）
I	插入（相对参考，read 中多出的碱基）
D	缺失（相对参考，read 中缺失的碱基）
S	软剪切（read 末端未比对的碱基，保留在 SEQ 中）
H	硬剪切（read 末端未比对的碱基，已从 SEQ 中移除）
N	跳过参考序列上的区域（如内含子，RNA 比对时常见）

表 4.2: CIGAR 常用操作符

例如，`150M` 表示 150 个碱基全部比对；`10S140M` 表示前 10 个碱基被软剪切、后 140 个碱基比对。

4.4　VCF/BCF 格式：变异位点的存储

VCF（Variant Call Format）是存储遗传变异（SNP、插入缺失等）的标准格式；BCF 是其二进制版本。VCF 也是由**头部区**（以 `##` 开头的元信息行 + 一行以 `#CHROM` 开头的列名行）和**数据区**（每个变异位点一行）组成。

4.4.1　数据区的前 8 个固定字段

字段	含义
CHROM	变异所在的参考序列名（染色体）
POS	变异位置（1-based）
ID	变异标识符（如 rs 号），无则为 .
REF	参考等位基因（参考基因组上的碱基）
ALT	变异等位基因（逗号分隔多个可能）
QUAL	变异质量分数（Phred 尺度）
FILTER	过滤标记（`PASS` 表示通过所有过滤）
INFO	附加信息（分号分隔的键值对，如 `DP=25`）

表 4.3: VCF 数据区固定字段

4.4.2　FORMAT 与样本列

从第 9 列开始，第 9 列是 FORMAT，定义样本列中各字段的顺序；第 10 列起为每个样本的具体基因型信息。常用 FORMAT 字段：

- GT：基因型（如 0/1 表示杂合，1/1 表示纯合变异，0/0 表示与参考一致）；
- DP：该位点的测序深度（覆盖的 read 数）；
- GQ：基因型质量分数；
- AD：各等位基因的 read 深度（支持 REF 和 ALT 的 read 数）。

```
##fileformat=VCFv4.2
#CHROM  POS     ID  REF  ALT  QUAL  FILTER  INFO            FORMAT   SAMPLE
NC_000913.3  100000  .  A  G  225  PASS  DP=27;MQ=60  GT:DP:AD:GQ
    1/1:27:0,27:99
```

Listing 4.3: VCF 数据行示例（示意）

重点理解：GT 与 DP

变异检测的核心输出是每个位点的**基因型（GT）**。以细菌（单倍体）为例，1/1 表示该位点为纯合变异（与参考不同），0/0 表示与参考一致。而 DP（深度）与 AD（各等位基因支持数）是判断变异是否可靠的关键证据，本项目第 10.3.3 节将深入讨论。

Chapter 5

核心算法原理

本章导读

”知其然，更要知其所以然。”理解工具背后的算法思想，才能正确选择参数、合理解读结果、在出问题时做出正确判断。本章以通俗语言介绍本项目涉及的三个核心算法：序列比对、变异检测与覆盖度统计。

5.1　序列比对算法

5.1.1　问题定义

序列比对（read alignment / mapping）的目标是：给定一条短 read 和一个参考基因组，找到该 read 在参考基因组上**最可能的来源位置**，并允许一定程度的错配（测序错误）与插入缺失（真实变异或错误）。

对人类基因组（约 30 亿 bp）而言，用朴素的逐位置比对（动态规划）比对数百万条 read，计算量是天文数字。因此，现代比对工具（BWA、Bowtie2、STAR 等）都采用**索引（index）**技术加速。

5.1.2　BWT 与 FM-index

BWA（Burrows-Wheeler Aligner）的核心是基于 **Burrows-Wheeler 变换（BWT）**和 **FM-index** 构建的全文索引。其思想可以这样直观理解：

1. 将参考基因组的所有”循环移位”按字典序排序，取每行最后一列，得到 BWT 字符串。这个变换是**可逆的**，且具有一个关键性质——相同碱基在变换后倾向于聚集成连续区段；
2. FM-index 在 BWT 上支持一种称为”**反向搜索（backward search）**”的操作，可以在 $O(m)$ 时间内（m 为 read 长度）快速定位某个短序列在参考基因组中的所有出现位置，而无需逐位置扫描；
3. 比对一条 read 时，BWA 从 read 末端开始逐碱基”反向搜索”，一旦发现不匹配，立即回溯，尝试其他位置。

这种”先建索引、再查询”的策略，使 BWA 能在数分钟内将数百万条 read 比对到人类基因组。

5.1.3　seed-and-extend 策略

BWA-MEM（本项目使用的比对算法）采用”种子-扩展”（seed-and-extend）策略：

1. **种子查找**：从 read 中提取若干短的”种子”（seed，如 19 bp 的精确匹配片段），利用 FM-index 快速找到它们在参考基因组上的候选位置；
2. **链化（chaining）**：将同一条 read 的多个种子候选位置”串联”成可能的一致比对（collinear chain）；
3. **扩展（extension）**：对每个候选位置，用 Smith-Waterman 局部比对算法向两端扩展，计算最优的完整比对，并给出比对质量（MAPQ）。

5.1.4　比对质量 MAPQ

MAPQ（mapping quality）是 BWA 为每条 read 赋予的”比对可信度”（Phred 尺度），计算公式可近似理解为：

$$\mathrm{MAPQ} = -10 \log_{10} P(\text{该比对位置错误}) \tag{5.1}$$

当一条 read 能比对到多个位置（如重复区）时，MAPQ 会很低（如 0），表示”无法确定唯一位置”。MAPQ 是后续变异检测中过滤低质量比对的重要依据。

5.2　变异检测算法

5.2.1　从比对到变异位点

变异检测（variant calling）的目标是：综合所有覆盖到某个基因组位置的 read 的比对信息，判断该位置是否存在变异（相对参考基因组的差异）。

本项目使用的 `bcftools mpileup + call` 采用”堆积（pileup）→ 基因型似然 → 变异判定”三步策略：

5.2.2　第一步：pileup 堆积

`mpileup` 将 BAM 文件中覆盖同一基因组位置的所有 read 的碱基”堆积”在一起，形成一个**位点深度矩阵**，记录每个位置 A/C/G/T 各有多少条 read 支持，以及它们的

碱基质量与比对质量。

5.2.3　第二步：计算基因型似然

对每个候选位点，`call` 基于观测到的碱基堆积，计算不同基因型假设下的**似然（likelihood）**。以单倍体（细菌）为例，只需比较两种假设：该位点与参考一致（REF）或为某种变异（ALT）。

直觉上：若某位置 20 条 read 中有 18 条是 G（与参考 A 不同），且这些 read 的碱基质量都很高，则” 该位点为 G” 的似然远高于” 该位点为 A（18 条 read 都是测序错误）” 的似然，工具就会判定此处存在变异。

5.2.4　第三步：贝叶斯后验与输出

实际算法会结合**先验概率**（如 SNP 的先验概率约 10^{-3}，InDel 约 10^{-6}）与似然，用贝叶斯公式计算**后验概率**，只有当后验概率超过阈值时才判定为变异，并输出相应的 QUAL（变异质量分数）。

$$\mathrm{QUAL} = -10 \log_{10} P(\text{该位置不是变异}) \tag{5.2}$$

例如，QUAL=225 表示” 该位置不是变异” 的概率为 $10^{-22.5}$，即变异判定的置信度极高。

为什么需要先验概率?

先验概率的作用是**抑制假阳性**。测序错误是随机发生的（每 100–1000 个碱基约 1 个错误），如果某个位点只有 1–2 条低质量 read 支持一个” 变异”，那么这更可能是测序错误而非真实变异。引入先验后，这类情况的后验概率很低，不会被判定为变异。这就是” 为什么不能简单地数 read 支持数来判定变异” 的原因。

5.3　测序深度与覆盖度

5.3.1　基本概念

- **测序深度（sequencing depth）**：每个基因组位置平均被多少条 read 覆盖。计算公式：

$$\text{深度} = \frac{\text{测序数据总量（总碱基数）}}{\text{基因组大小}} \tag{5.3}$$

例如，124 Mb 的测序数据比对到 4.6 Mb 的基因组，平均深度约 27×。

- **覆盖度（coverage / breadth）**：基因组中被至少 1 条 read 覆盖的碱基占比。理想情况下，深度足够时覆盖度接近 100%。

5.3.2　深度与变异检测的可靠性

测序深度直接影响变异检测的灵敏度与准确性：

深度	对变异检测的影响
<5×	大量位点未被覆盖或深度极低，漏检严重（假阴性多），检出的变异也不可靠
5–10×	可检出纯合变异，但杂合变异与低深度位点可靠性差
10–30×	常见研究标准，纯合与杂合变异均可较可靠检出
>30×	高深度，可检出低频变异（如亚克隆、肿瘤异质性），但计算成本升高

表 5.1：测序深度与变异检测可靠性的关系

本项目如何体现深度的影响

主实验（27×）与延伸实验（4×）形成天然对照：前者变异检测结果可信、假阳性少；后者将观察到大量低深度位点与更高的假阳性率。这一对比是本项目最重要的科学收获之一。

5.3.3　覆盖深度的不均匀性

实际测序中，基因组各位置的深度并非均匀——GC 含量极端区域、重复序列、PCR 扩增偏好等都会导致某些区域深度偏高或偏低。这就是为什么变异检测不能只依赖”平均深度”，而要关注每个位点的实际深度（VCF 中的 DP 值）。

Chapter 6

模块 1：数据获取

模块目标

本模块将完成两项任务：(1) **下载参考基因组**；(2) **下载测序数据**，并对下载的文件进行完整性校验。这是整个分析流程的起点，也是理解” 数据从何而来” 的实践环节。

6.1　学习目标

完成本模块后，学生应能：

- 从 NCBI RefSeq 数据库下载指定物种的参考基因组（FASTA）与注释文件（GFF）；
- 从 ENA（欧洲核苷酸档案）数据库下载测序数据（FASTQ）；
- 使用 `wget/curl` 下载网络文件；
- 使用 `md5sum` 校验文件完整性，理解校验和（checksum）的作用。

6.2　背景知识：公共生物信息学数据库

6.2.1　三大国际核酸数据库

全球三大核酸序列数据库——美国的 **GenBank**（NCBI）、欧洲的 **ENA**（EBI）、日本的 **DDBJ**——共同组成国际核苷酸序列数据库协作联盟（INSDC）。三方每日交换数据，因此同一份数据在三个数据库中都能找到，只是访问号（accession）前缀不同：

数据库	所属机构	访问号前缀示例
GenBank / SRA	美国 NCBI	SRR（测序 run）、SAMN（样本）
ENA	欧洲 EBI	ERR（测序 run）、SAMEA（样本）
DDBJ SRA	日本 DDBJ	DRR（测序 run）、SAMD（样本）

表 6.1：三大核酸数据库及其访问号前缀

本项目数据来自哪里?

主实验数据 ERR15404827 的访问号以 ERR 开头，属于 ENA；延伸实验数据 DRR063436 以 DRR 开头，属于 DDBJ。但二者都已同步到 ENA，因此**统一从 ENA 下载**即可，无需分别访问多个网站。

6.2.2　参考基因组从何而来?

RefSeq（Reference Sequence，参考序列）是 NCBI 维护的、经过人工审核的权威参考序列数据库。本项目使用的大肠杆菌 K-12 MG1655 参考基因组来自 RefSeq，其访问号为 GCF_000005845.2（GCF 前缀表示 RefSeq 基因组）。

一个 RefSeq 基因组目录通常包含以下文件：

- *_genomic.fna.gz：基因组序列（FASTA 格式），用于比对；
- *_genomic.gff.gz：基因组注释（GFF 格式），记录基因位置与功能；
- *_genomic.gbff.gz：GenBank 格式的完整记录（含注释与来源）；
- *_cds_from_genomic.fna.gz：所有编码区（CDS）序列。

本项目只需要前两个文件（fna 用于比对，gff 用于注释参考）。

6.3　实验步骤

6.3.1　步骤 1：进入项目目录并创建数据目录

```
mkdir -p ~/ecoli_reseq/{data,ref,results/{qc,align,variants,annotation},scripts,
    report}
cd ~/ecoli_reseq
```

Listing 6.1: 创建目录结构

6.3.2　步骤 2：下载参考基因组与注释文件

使用 wget 从 NCBI FTP 下载 MG1655 的参考基因组（FASTA）与注释文件（GFF）：

```
cd ref

# 下载基因组序列（FASTA）
wget -c https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2
    _ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz
```

```
# 下载基因组注释（GFF）
wget -c https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.gff.gz

# 解压基因组序列（BWA 索引需要未压缩的 FASTA）
gunzip GCF_000005845.2_ASM584v2_genomic.fna.gz

ls -lh
```

Listing 6.2: 下载参考基因组与注释

-c 参数的作用

wget -c（--continue）支持**断点续传**：若下载中断，再次执行同一命令会从断点继续，而非重新开始。下载大文件时强烈建议加 -c。

6.3.3　步骤 3：查看参考基因组

```
# 查看序列条数与名称
grep "^>" GCF_000005845.2_ASM584v2_genomic.fna

# 统计基因组总长度
grep -v "^>" GCF_000005845.2_ASM584v2_genomic.fna | tr -d '\n' | wc -c
```

Listing 6.3: 查看参考基因组信息

预期输出（本书编写时）：

```
>NC_000913.3 Escherichia coli str. K-12 substr. MG1655, complete genome
4641652
```

Listing 6.4: 预期输出

解读输出

MG1655 参考基因组包含 1 条染色体序列，RefSeq 编号为 NC_000913.3，总长 4,641,652 bp（约 4.64 Mb）。请记住这个序列名——后续比对、变异检测、注释的所有命令都会用到它，若序列名不一致会导致工具报错。

6.3.4　步骤 4：下载测序数据

测序数据从 ENA 的 FTP 站点下载。主实验数据 ERR15404827 为双端测序，包含两个 FASTQ 文件（R1 与 R2）：

```
cd ../data

# 下载 R1（read1）
wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/
    ERR15404827_1.fastq.gz

# 下载 R2（read2）
wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/
    ERR15404827_2.fastq.gz

ls -lh
```

Listing 6.5: 下载测序数据（主实验）

如何构造 ENA 下载链接?

ENA 的 FASTQ 下载链接遵循固定规则：`https://ftp.sra.ebi.ac.uk/vol1/fastq/` + 访问号前 6 位 + / + 完整访问号 + / + 访问号 + 后缀。例如 ERR15404827 的前 6 位是 ERR154，路径为 ERR154/027/ERR15404827/。掌握这一规则后，可对任意 ENA 访问号自行构造下载链接。

快速预览数据

在下载完成前，可用 `zcat 文件名 | head -8` 预览 FASTQ 文件的前 8 行（即前 2 条 read），确认文件内容正确、读长符合预期（主实验应为 250 bp）。

6.3.5　步骤 5：数据完整性校验

网络下载的文件可能因传输错误而损坏。校验和（checksum）通过算法（如 MD5）为文件生成一个” 指纹”，用于验证文件是否完整无损。

```
# 计算本地文件的 MD5 值
md5sum ERR15404827_1.fastq.gz ERR15404827_2.fastq.gz
```

Listing 6.6: 计算本地文件的 MD5 校验和

```
# 用 ENA API 查询官方 MD5（也可在浏览器直接打开该 URL）
```

```
curl -s "https://www.ebi.ac.uk/ena/portal/api/filereport?accession=ERR15404827&
    result=read_run&fields=run_accession,fastq_md5&format=tsv"
```

Listing 6.7: 获取 ENA 官方 MD5 校验值

将**本地计算的 MD5** 与 **ENA 官方提供的 MD5** 逐字符比对，完全一致即表示文件完整无损。

如何核对 MD5?

ENA 查询结果中，`fastq_md5` 字段会按顺序给出 R1、R2 两个文件的官方 MD5（以分号分隔），与 `md5sum` 输出的本地值一一对应。二者字母大小写不敏感，只要字符串内容一致即可。

MD5 不一致怎么办?

若本地 MD5 与官方不一致，说明文件损坏或下载不完整。请删除该文件并用 `wget -c` 重新下载（断点续传），再次校验。

6.4　思考题

思考题 1

为什么参考基因组的 FASTA 文件在解压前是 `.gz` 压缩格式，而下载测序数据的 FASTQ 文件也是 `.gz` 格式？这种压缩对生物信息学数据分析有什么意义？

思考题 2

`ERR15404827` 的 R1 与 R2 两个文件分别代表什么？为什么双端测序会产生两个 FASTQ 文件？如果只下载了 R1 而遗漏 R2，对后续分析会有什么影响？

思考题 3

MD5 校验和的作用是什么？设想一个场景：你下载的文件 MD5 与官方不一致，可能的原因有哪些？

Chapter 7

模块 2：测序数据质量控制

模块目标

本模块将对原始测序数据进行质量评估（FastQC）与过滤修剪（fastp），去除低质量碱基、接头污染与过短读段，为后续比对提供高质量的输入数据。质量控制是任何生信分析中**不可或缺、不可省略**的一步。

7.1　学习目标

完成本模块后，学生应能：

- 使用 FastQC 生成质量报告，并解读报告中的关键指标（碱基质量、GC 含量、接头污染、重复水平等）；
- 使用 fastp 进行质量过滤与修剪，并理解各过滤参数的含义；
- 对比质控前后的数据差异，理解”垃圾进、垃圾出”（Garbage In, Garbage Out）的原则。

7.2　背景知识：为什么要做质量控制？

原始测序数据中存在多种”噪音”，若不处理会直接影响下游分析：

- **测序错误**：Illumina 的碱基错误率约 0.1–1%，且随测序循环推进，read 末端的错误率会升高（”相位漂移”导致信号衰减）；
- **接头污染**：当插入片段短于测序读长时，测序会”读穿”插入片段，读到接头序列，这些接头序列并非基因组来源，必须去除；
- **低质量碱基**：个别碱基质量极低（如 $Q < 20$），会在比对与变异检测中引入错误；
- **PCR 重复**：文库扩增产生的重复 read（这部分通常在比对后处理，见模块 4）。

核心原则：垃圾进，垃圾出

下游分析的结论质量永远不可能超过输入数据的质量。质量差的读段会导致比对错误、变异假阳性甚至完全错误的结果。因此，**质控是生信分析中投入产出比最高的一步**，务必认真对待。

7.3　实验步骤

7.3.1　步骤 1：使用 FastQC 评估原始数据质量

FastQC 是最常用的测序数据质量评估工具，它会为每个 FASTQ 文件生成一份包含十余项指标的 HTML 报告。

```
cd ~/ecoli_reseq

fastqc -t 4 -o results/qc/ \
  data/ERR15404827_1.fastq.gz \
  data/ERR15404827_2.fastq.gz

ls results/qc/
# 输出：ERR15404827_1_fastqc.html  ERR15404827_1_fastqc.zip
#       ERR15404827_2_fastqc.html  ERR15404827_2_fastqc.zip
```

Listing 7.1: 运行 FastQC

参数说明：`-t 4` 使用 4 个线程并行；`-o` 指定输出目录。生成的 `.html` 文件可用浏览器打开查看，`.zip` 内含各指标的原始数据。

7.3.2　步骤 2：解读 FastQC 报告

打开 `ERR15404827_1_fastqc.html`，重点查看以下指标：

指标	正常表现	异常信号
Per base sequence quality	中位数质量 $Q > 30$，末端略降	大量 $Q < 20$，末端骤降
Per sequence quality scores	峰值位于高分段	峰值偏左（低质量）
Per base GC content	平直，接近物种 GC 比例	明显波动或双峰
Adapter content	接近 0	末端出现接头序列
Overrepresented sequences	无显著富集	某序列占比异常高

表 7.1: FastQC 报告关键指标解读

FastQC 用 绿色（通过）、黄色（警告）、红色（失败） 标记各项指标。

大肠杆菌的 GC 含量

大肠杆菌基因组的 GC 含量约 50.8%。FastQC 的 GC 含量图应呈现一个以 50% 为中心的较窄分布峰。若出现明显的双峰或异常偏移，可能提示污染或文库制备问题。

7.3.3　步骤 3：使用 fastp 过滤与修剪

fastp 是一个集质量过滤、接头去除、长度过滤于一体的高效工具，并自动生成过滤前后的对比报告。

```
fastp \
  -i data/ERR15404827_1.fastq.gz \
  -I data/ERR15404827_2.fastq.gz \
  -o results/qc/ERR15404827_1.clean.fastq.gz \
  -O results/qc/ERR15404827_2.clean.fastq.gz \
  --html results/qc/fastp.html \
  --json results/qc/fastp.json \
  --thread 4 \
  --qualified_quality_phred 20 \
  --unqualified_percent_limit 40 \
  --length_required 50 \
  --cut_front --cut_tail
```

Listing 7.2: 运行 fastp 质控

各参数含义：

参数	含义
`-i / -I`	输入的 R1 / R2 文件
`-o / -O`	输出的 R1 / R2 文件
`--html / --json`	输出 HTML / JSON 格式的质控报告
`--thread 4`	使用 4 线程
`--qualified_quality_phred 20`	合格碱基的质量阈值（$Q \geq 20$）
`--unqualified_percent_limit 40`	一条 read 中不合格碱基占比超过 40% 即丢弃
`--length_required 50`	过滤后长度小于 50 bp 的 read 丢弃
`--cut_front / --cut_tail`	从 read 前端/末端滑动窗口修剪低质量碱基

表 7.2: fastp 关键参数说明

如何选择质控参数?

质控参数应根据 FastQC 报告与下游需求调整，并非越严格越好。对变异检测而言，**过度修剪会损失有效信息、降低有效深度**。入门阶段建议采用上述相对温和的参数，理解每项参数后再根据数据情况微调。

7.3.4　步骤 4：查看 fastp 报告与质控统计

```
# 用 python 快速读取 JSON 报告中的关键统计
python3 -c "
import json
d = json.load(open('results/qc/fastp.json'))
s = d['summary']
for k in ['before_filtering','after_filtering']:
    x = s[k]
    print(k, '-> total_reads:', x['total_reads'],
          '| q30_rate: %.2f%%' % (x['q30_rate']*100),
          '| gc: %.2f%%' % (x['gc_content']*100))
"
```

Listing 7.3: 查看质控统计摘要

同时可用浏览器打开 `results/qc/fastp.html`，查看过滤前后的读长分布、质量分布、碱基组成等可视化对比图。

理解过滤统计

fastp 报告的 `q30_rate` 表示 $Q \geq 30$ 的碱基占比，是评价数据质量的核心指标。优质 Illumina 数据的 Q30 率通常在 85–95% 之间。对比 `before_filtering` 与 `after_filtering`，可直观看到过滤提升了多少数据质量，以及损失了多少读段（应在合理范围内，一般损失 <10%）。

7.3.5　步骤 5：质控后再用 FastQC 验证（可选）

为确认过滤效果，可对 clean 数据再次运行 FastQC：

```
fastqc -t 4 -o results/qc/ \
  results/qc/ERR15404827_1.clean.fastq.gz \
  results/qc/ERR15404827_2.clean.fastq.gz
```

Listing 7.4: 质控后复检

对比质控前后 FastQC 报告，重点关注：碱基质量曲线是否更平直、末端低质量区域是否被修剪、接头污染是否消失。

7.4　思考题

思考题 1

FastQC 报告显示某数据”Per base sequence quality” 在 read 末端（第 200–250 位）质量骤降到 Q10 以下，这说明什么？fastp 的哪个参数可以处理这一问题？

思考题 2

为什么过滤后（`after_filtering`）的读段总数通常略少于过滤前？如果过滤损失了超过 30% 的读段，可能意味着什么？你会如何排查？

思考题 3

” 接头污染”（adapter contamination）是如何产生的？为什么它主要出现在 read 的末端而不是开头？

Chapter 8

模块 3：序列比对

模块目标

本模块将使用 BWA-MEM 把经过质控的测序读段比对（mapping）到参考基因组上，回答” 每条 read 来自基因组的哪个位置” 这一核心问题。比对结果是后续变异检测的基础。

8.1　学习目标

完成本模块后，学生应能：

- 理解参考基因组索引（BWA index）的作用与原理；
- 使用 BWA-MEM 对双端测序数据进行比对，并理解 SAM 输出的结构；
- 理解比对率、比对质量（MAPQ）等关键指标的含义；
- 掌握” 先索引、再比对” 的通用分析范式。

8.2　背景知识：什么是序列比对

序列比对（read alignment / mapping）是生物信息学最核心的操作之一。它的任务是：给定一条短测序读段和一个参考基因组，确定该读段**最可能来自参考基因组的哪个位置**，并给出比对的质量评估。

这一过程类似于” 在一本已知的书中查找某个句子的出处”——参考基因组是那本” 书”，read 是要查找的” 句子片段”。但二者有本质区别：

- read 可能含有测序错误（句子可能有错别字）；
- read 可能与参考存在真实的序列差异（即我们要找的变异）；
- 基因组中存在大量重复序列（同一” 句子” 可能出现在书中多处）。

因此，比对算法必须在**速度**与**准确性**之间取得平衡，这正是 BWA 等工具的精妙之处（其算法原理见第 5.1 节）。

8.3　实验步骤

8.3.1　步骤 1：为参考基因组建立索引

BWA 采用” 先建索引、再查询” 的策略。索引可以理解为参考基因组的” 字典”，能极大加速后续比对。对同一参考基因组只需建立一次索引，可反复使用。

```
cd ~/ecoli_reseq

bwa index ref/GCF_000005845.2_ASM584v2_genomic.fna

ls -lh ref/
```

Listing 8.1: BWA 建立参考基因组索引

执行后，`ref/` 目录下会生成 5 个索引文件，后缀分别为 `.amb`、`.ann`、`.bwt`、`.pac`、`.sa`。

同时，还需用 samtools 建立另一个索引（供后续 `mpileup` 等工具使用）：

```
samtools faidx ref/GCF_000005845.2_ASM584v2_genomic.fna

ls ref/
# 额外生成：GCF_000005845.2_ASM584v2_genomic.fna.fai
```

Listing 8.2: samtools 建立 FASTA 索引

BWA 索引 vs. samtools 索引

二者不可混用：**BWA 索引**（`.bwt` 等）基于 BWT/FM-index，供 BWA 比对使用；**samtools 的 FASTA 索引**（`.fai`）记录每条序列的名称、长度与偏移量，供 samtools/bcftools 等工具快速定位序列使用。两者都要建，用途不同。

8.3.2　步骤 2：使用 BWA-MEM 进行比对

BWA-MEM 是 BWA 中针对 70 bp–1 Mb 读长优化的比对算法，也是当前最通用的短读比对方法。

```
bwa mem -t 4 \
  ref/GCF_000005845.2_ASM584v2_genomic.fna \
  results/qc/ERR15404827_1.clean.fastq.gz \
  results/qc/ERR15404827_2.clean.fastq.gz \
```

```
  > results/align/ERR15404827.sam
```

Listing 8.3: BWA-MEM 双端比对

命令解析：

参数/参数	含义
`bwa mem`	调用 BWA-MEM 比对算法
`-t 4`	使用 4 个线程并行
参考基因组路径	已建索引的 FASTA（自动调用对应索引）
R1、R2 文件路径	双端测序的成对读段
`> xxx.sam`	将比对结果重定向到 SAM 文件

表 8.1: BWA-MEM 命令解析

为什么用重定向而非 -o?

BWA 默认将 SAM 结果输出到**标准输出（stdout）**，因此用 > 重定向到文件。这是许多 Unix 工具的通用设计——便于在管道（|）中衔接后续命令。

8.3.3　步骤 3：查看 SAM 文件

```
# 查看 SAM 头部（以 @ 开头）
samtools view -H results/align/ERR15404827.sam | head -10

# 查看前 5 条比对记录（跳过头部）
samtools view results/align/ERR15404827.sam | head -5
```

Listing 8.4: 查看 SAM 文件头部与内容

理解 SAM 输出

SAM 头部（`@SQ` 行）记录了参考序列的名称与长度；比对区每行一条 read，包含 11 个字段（详见第 4.3 节）。重点观察第 2 列 FLAG、第 3 列 RNAME、第 4 列 POS、第 5 列 MAPQ 与第 6 列 CIGAR，它们共同描述了” 这条 read 比对到了哪里、比对得如何”。

8.3.4　步骤 4：评估比对结果（初步）

在转换为 BAM 并排序前，可先用 `samtools flagstat` 快速评估比对率：

```
samtools flagstat results/align/ERR15404827.sam
```

Listing 8.5: 初步评估比对率

解读 flagstat 输出

重点关注几个数字：**总 read 数**、**比对上的 read 占比**（mapped）与**正确配对比对占比**（properly paired）。对同源重测序（主实验），比对率通常应 >99%，properly paired 比例也应 >95%。若比对率异常低，需检查：数据是否与参考基因组匹配、是否有严重污染、参数是否错误。

8.4　思考题

思考题 1

为什么 BWA 要先” 建索引” 再比对？索引文件比参考基因组本身还大（.bwt 等文件），这样做” 划算” 吗？请从计算复杂度角度思考。

思考题 2

SAM 文件的 FLAG 字段值为 99，请拆解它表示的含义（提示：参考第 4.3 节的位标志表）。

思考题 3

MAPQ（比对质量）为 0 意味着什么？在变异检测中，为什么通常要过滤掉 MAPQ 低的 read？

Chapter 9

模块 4：比对后处理

模块目标

本模块对 SAM 比对结果进行一系列标准后处理：格式转换、排序、标记与去除 PCR 重复、建立索引，并统计比对质量、测序深度与覆盖度。这些处理是变异检测前的必要准备。

9.1　学习目标

完成本模块后，学生应能：

- 理解 SAM 与 BAM 格式的区别，熟练使用 `samtools view` 进行格式转换与筛选；
- 理解” 按坐标排序” 的意义；
- 理解 PCR 重复的来源与去除方法（`markdup`）；
- 使用 `flagstat`、`coverage`、`depth` 统计比对质量与覆盖度；
- 理解平均深度、覆盖度等概念的计算与含义。

9.2　背景知识：为什么要做比对后处理

BWA 输出的 SAM 文件存在几个问题，使其无法直接用于变异检测：

1. **文件巨大且低效**：SAM 是纯文本，占用空间大、读取慢；二进制 BAM 更紧凑且支持索引；
2. **无序**：BWA 按 read 在输入文件中的顺序输出，而非按基因组坐标排序；而变异检测等下游操作需要按坐标顺序访问（便于” 堆积” 同一位置的 read）；
3. **含 PCR 重复**：文库扩增产生的重复 read 会人为抬高某些位点的深度，导致变异检测偏差。

9.2.1　PCR 重复从何而来?

测序文库在制备过程中需经过 PCR 扩增以增加 DNA 量。如果两条 read 的**起始坐标与终止坐标完全相同**（即来自同一个原始 DNA 分子的多个 PCR 拷贝），它们就构成 PCR 重复。PCR 重复提供的是**冗余而非独立**的信息——若一个 PCR 重复簇中恰好有测序错误，该错误会被错误放大，产生假变异。

9.3　实验步骤

9.3.1　步骤 1：SAM 转换为 BAM

```
cd ~/ecoli_reseq

samtools view -bS results/align/ERR15404827.sam \
  -o results/align/ERR15404827.bam
```

Listing 9.1: SAM 转 BAM

参数说明：`-b` 输出 BAM 格式；`-S` 声明输入为 SAM 格式；`-o` 指定输出文件。

顺手清理大文件

SAM 文件通常比 BAM 大数倍且后续不再需要，转换后可删除以节省磁盘空间：`rm results/align/ERR15404827.sam`。

9.3.2　步骤 2：按坐标排序

```
samtools sort -@ 4 results/align/ERR15404827.bam \
  -o results/align/ERR15404827.sorted.bam
```

Listing 9.2: 按坐标排序 BAM

`-@ 4` 使用 4 线程加速排序。排序后，read 按参考基因组坐标从小到大排列，这是下游所有操作（去重、索引、变异检测）的前提。

9.3.3　步骤 3：标记 mate 信息（fixmate）

```
samtools fixmate -m results/align/ERR15404827.sorted.bam \
  results/align/ERR15404827.fixmate.bam
```

Listing 9.3: 补充 mate 信息

为什么要 fixmate?

`fixmate -m` 会核对并补齐双端 read 之间的 mate 坐标、插入片段长度等信息，这是 `markdup` 准确识别 PCR 重复的前提（markdup 需要比对两条 read 的坐标是否与另一对完全一致）。

9.3.4　步骤 4：再次排序

```
samtools sort -@ 4 results/align/ERR15404827.fixmate.bam \
  -o results/align/ERR15404827.fixmate.sorted.bam
```

Listing 9.4: fixmate 后再次排序

9.3.5　步骤 5：去除 PCR 重复

```
samtools markdup results/align/ERR15404827.fixmate.sorted.bam \
  results/align/ERR15404827.dedup.bam
```

Listing 9.5: 标记并去除 PCR 重复

`markdup` 会识别坐标完全相同的 read 对，将其中的重复标记并去除，只保留一条代表 read。

markdup 与 rmdup 的区别

旧教程中常见 `samtools rmdup`，现已废弃。现代版本请使用 `samtools markdup`。二者都是去除 PCR 重复，但 markdup 更准确（保留最高质量的代表 read）。

9.3.6　步骤 6：建立 BAM 索引

```
samtools index results/align/ERR15404827.dedup.bam
# 生成 ERR15404827.dedup.bam.bai
```

Listing 9.6: 建立 BAM 索引

BAM 索引（`.bai`）使工具能够随机访问基因组任意区域的比对结果，是 IGV 可视化与 mpileup 变异检测的必要文件。

9.3.7　步骤 7：比对统计

```
# 1. 基本比对统计
samtools flagstat results/align/ERR15404827.dedup.bam

# 2. 详细统计（插入片段、质量分布等）
samtools stats results/align/ERR15404827.dedup.bam > results/align/stats.txt
head -30 results/align/stats.txt

# 3. 覆盖度概览（每染色体一条汇总）
samtools coverage results/align/ERR15404827.dedup.bam
```

Listing 9.7: 统计比对情况

解读 flagstat 关键数字

以主实验为例，预期结果大致为：

- total reads：约 99 万条（97.3 万条 read 对 × 2）；
- mapped（比对率）：>99%；
- properly paired（正确配对）：>95%；
- duplicates（重复率）：通常 <5%，markdup 后应显著降低。

9.3.8　步骤 8：计算深度分布

```
samtools depth -a results/align/ERR15404827.dedup.bam \
  > results/align/depth.txt

# 查看平均深度与深度分布
awk '{s+=$3; if($3>0) c++} END{print "平均深度(含0):", s/NR; print "覆盖位点平均深度:", s/c; print "覆盖度:", c/NR*100"%"}' \
  results/align/depth.txt
```

Listing 9.8: 输出每个位点的深度

理解深度统计

`samtools depth` 输出三列：参考序列名、位置、该位点深度。`samtools coverage` 则直接给出每序列的覆盖度与平均深度汇总。对主实验（27×），平均深度应接近 27；覆盖度（≥ 1× 的位点占比）通常 >99.5%。若平均深度显著低于预期，说明

数据量或比对出了问题。

9.4　思考题

思考题 1

为什么要对 BAM 按坐标排序？如果不排序，`markdup` 和变异检测会怎样？请从算法访问模式的角度分析。

思考题 2

PCR 重复与” 生物学重复” 有何本质区别？为什么变异检测中要去除 PCR 重复，而不是保留它们来” 增加深度”？

思考题 3

主实验的预期平均深度约 27×。请用” 测序数据量 / 基因组大小” 的公式手动估算一次，并与 `samtools coverage` 的实际输出对比。二者若有差异，可能的原因是什么？

Chapter 10

模块 5：变异检测

模块目标

本模块是整个项目的核心。我们将综合所有覆盖到每个基因组位置的 read 信息，使用 bcftools 检测单核苷酸多态性（SNP）与插入缺失（InDel），并对变异结果进行质量过滤与统计解读。

10.1　学习目标

完成本模块后，学生应能：

- 使用 `bcftools mpileup + call` 完成变异检测，理解” 堆积 + 基因型判定” 的两步流程；
- 理解 VCF 文件中 QUAL、DP、GT、AD 等关键字段的含义；
- 使用 `bcftools filter` 对变异进行质量过滤，理解各过滤条件的意义；
- 使用 `bcftools stats` 与 `bcftools query` 统计和查看变异结果；
- 深入理解” 假阳性” 的来源及其与测序深度、过滤参数的关系。

10.2　背景知识：变异检测的两步流程

本项目使用的 `bcftools` 将变异检测拆分为两个可组合的步骤（其算法原理见第 5.2 节）：

1. **mpileup（堆积）**：读取 BAM 文件，将覆盖同一基因组位置的所有 read 的碱基” 堆积” 在一起，统计每个位置 A/C/G/T 的支持数与碱基质量；
2. **call（判定）**：对每个位点，基于堆积信息计算不同基因型假设的似然，结合先验概率做贝叶斯判定，输出变异位点。

这种” 分步 + 管道” 的设计体现了 Unix 哲学——每个工具做好一件事，通过管道（|）组合。

10.3　实验步骤

10.3.1　步骤 1：变异检测

```
cd ~/ecoli_reseq

bcftools mpileup --ploidy 1 \
  -f ref/GCF_000005845.2_ASM584v2_genomic.fna \
  results/align/ERR15404827.dedup.bam | \
bcftools call -mv -Ov \
  -o results/variants/ERR15404827.raw.vcf
```

Listing 10.1: bcftools 变异检测（单倍体模式）

参数解析：

参数	含义
`mpileup --ploidy 1`	按单倍体（细菌）处理，每个位点只判定一个等位基因
`mpileup -f <fasta>`	指定参考基因组
`call -m`	多等位基因 calling（允许一个位点多个 ALT）
`call -v`	仅输出变异位点（跳过与参考一致的位置）
`call -Ov`	输出未压缩的文本 VCF（`-Oz` 为压缩版）
`call -o <file>`	指定输出文件

表 10.1: 变异检测命令参数解析

为什么要指定 ploidy?

大肠杆菌是**单倍体**（haploid）——每个基因只有一个拷贝，不存在”杂合”（0/1）的概念。因此指定 `--ploidy 1` 使基因型判定更符合生物学实际：每个位点要么与参考一致（0），要么为纯合变异（1）。若用默认的二倍体模式，工具会试图判定”杂合位点”，对单倍体数据反而不合适。

处理多线程与大数据

`mpileup` 本身较耗时，未来分析更大基因组时可加 `--threads` 参数加速。本项目数据量小，默认单线程数分钟即可完成。

10.3.2　步骤 2：查看原始 VCF 结果

```
# 统计原始变异总数（排除 # 注释行）
grep -vc "^#" results/variants/ERR15404827.raw.vcf

# 查看前 10 个变异位点
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%QUAL\t%DP\n' \
  results/variants/ERR15404827.raw.vcf | head -10
```

Listing 10.2: 查看 VCF 内容

解读 query 输出

bcftools query 用 -f 指定输出格式，%CHROM、%POS 等为字段占位符。输出各列依次为：染色体、位置、参考碱基、变异碱基、变异质量（QUAL）、测序深度（DP）。观察原始结果中的变异数量与 QUAL、DP 分布——你会看到大量低质量、低深度的候选变异，它们大多是假阳性。

10.3.3　步骤 3：变异质量过滤

原始变异结果中混杂着大量假阳性，必须过滤。最核心的两个过滤维度是**变异质量（QUAL）**与**测序深度（DP）**：

```
bcftools filter \
  -i 'QUAL >= 30 && DP >= 10' \
  results/variants/ERR15404827.raw.vcf \
  -o results/variants/ERR15404827.filtered.vcf

# 统计过滤后的变异数
grep -vc "^#" results/variants/ERR15404827.filtered.vcf
```

Listing 10.3: 过滤低质量变异

过滤条件 QUAL >= 30 && DP >= 10 的含义：

- QUAL >= 30：变异质量分数不低于 30（即” 该位点不是变异” 的概率 $\leq 10^{-3}$）；
- DP >= 10：该位点至少被 10 条 read 覆盖（保证有足够证据支持判定）。

过滤参数是” 艺术” 也是” 科学”

过滤参数需根据数据特征与研究目的调整，不存在放之四海皆准的” 标准答案”。过严会漏掉真实变异（假阴性），过松会混入假阳性。本项目鼓励你**尝试多组过滤参数**（如 DP≥5/10/20、QUAL≥20/30/60），观察变异数量的变化，理解每个参

数的生物学意义——这比死记一个” 正确参数” 更有价值。

10.3.4　步骤 4：变异统计

```
# 生成完整统计
bcftools stats results/variants/ERR15404827.filtered.vcf \
  > results/variants/stats.txt

# 查看摘要统计（SNP/InDel 数量、转换颠换比等）
grep -E "SN|number of" results/variants/stats.txt | head -30
```

Listing 10.4: 生成变异统计报告

转换/颠换比（Ts/Tv）

碱基替换分为两类：**转换（transition, Ts）**指嘌呤之间（A↔G）或嘧啶之间（C↔T）的替换；**颠换（transversion, Tv）**指嘌呤与嘧啶之间的替换。由于分子机制的原因，真实生物变异中转换多于颠换，Ts/Tv 比值通常在 2–3 左右。若过滤后 Ts/Tv 异常（如 <1），往往提示仍有大量假阳性。这是评估变异集质量的一个经典指标。

10.3.5　步骤 5：解读主实验的” 近乎零变异” 结果

最重要的科学收获

主实验的样本与参考基因组**同为 MG1655**，理论上二者应完全一致。因此过滤后你很可能得到**极少量甚至 0 个变异**——这并非” 实验失败”，恰恰是**正确的科学结论**！它验证了：

1. 分析流程本身没有系统性引入大量假阳性；
2. 过滤策略有效地去除了测序/比对错误产生的噪音；
3. 变异检测的意义在于” 区分真实信号与背景噪音”，而不是” 找到越多越好”。

建议你对比**过滤前**（可能有数百上千个候选）与**过滤后**（接近零）的数量差异，直观感受” 质量过滤” 的力量。剩余的那极少量” 变异” 很可能是：低复杂度区域、重复序列附近的顽固假阳性，或样本在长期培养中真实积累的少量突变。

10.4　思考题

思考题 1

VCF 中 QUAL 字段与 FASTQ 中碱基质量（Phred 分数）有什么联系与区别？二者分别衡量” 什么出错” 的概率？

思考题 2

为什么过滤条件要同时考虑 QUAL 和 DP？只用一个条件（如只设 QUAL≥30）会有什么问题？

思考题 3

若把过滤条件改为 `DP >= 20 && QUAL >= 60`，变异数量如何变化？结合” 假阳性 vs 假阴性” 的权衡，解释这一现象。

Chapter 11

模块 6：变异注释与功能解读

模块目标

本模块使用 SnpEff 对过滤后的变异进行功能注释，判断每个变异落在基因组的什么位置（基因间区、编码区、非编码区等），以及是否改变蛋白质序列（同义、错义、无义等），从而将” 变异列表” 转化为” 生物学结论”。

11.1　学习目标

完成本模块后，学生应能：

- 理解变异注释的意义与基本流程；
- 使用 SnpEff 对 VCF 进行功能注释，并解读 ANN 字段；
- 掌握变异类型（同义、错义、无义、移码等）的分类与含义；
- 统计变异在基因组功能区域的分布。

11.2　背景知识：为什么要注释变异

变异检测得到的 VCF 只是一个” 位置 + 碱基变化” 的列表，例如”NC_000913.3 的第 100000 位 A→G”。这个信息本身不回答：这个变异重要吗？它落在基因里吗？会改变蛋白质吗？

变异注释（variant annotation）就是为每个变异补充其功能信息的过程。SnpEff 根据参考基因组的注释文件（GFF），判断每个变异位于哪个基因、哪个区域（外显子、内含子、基因间区等），并预测其对蛋白质的影响。

11.3　实验步骤

11.3.1　步骤 1：运行 SnpEff 注释

```
cd ~/ecoli_reseq
```

```

snpEff -v ASM584v2 \
  results/variants/ERR15404827.filtered.vcf \
  > results/annotation/ERR15404827.ann.vcf
```

Listing 11.1: SnpEff 变异注释

参数说明：`-v` 输出详细信息；`ASM584v2` 为 SnpEff 数据库中 MG1655 对应的数据库名（已在第 2 章下载）。

数据库名不匹配会怎样?

若 SnpEff 报错"database not found"，说明数据库名或路径不对。请回到第 2 章确认数据库已正确下载，或用 `snpEff databases | grep -i "K-12"` 查询准确的数据库名后替换命令中的 `ASM584v2`。

11.3.2　步骤 2：查看注释结果

注释后的 VCF 在 INFO 列新增了 `ANN` 字段，包含每个变异的功能注释。

```
# 查看 ANN 字段内容（前几个变异）
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%INFO/ANN\n' \
  results/annotation/ERR15404827.ann.vcf | head -10
```

Listing 11.2: 提取注释信息

SnpEff 还会在输出目录生成 `snpEff_genes.txt` 与 `snpEff_summary.html` 两个汇总文件，可用浏览器打开 HTML 查看可视化统计。

11.3.3　步骤 3：理解 ANN 字段

`ANN` 字段用 `|` 分隔多个子字段，其中最重要的几个：

子字段（位置）	含义
Allele（第 1 个）	被注释的变异等位基因
Annotation（第 2 个）	变异类型（如 missense_variant）
Gene Name（第 4 个）	基因名（如 lacZ）
Gene ID（第 5 个）	基因编号

表 11.1: ANN 字段关键子字段

11.3.4　步骤 4：变异类型分类统计

SnpEff 将变异按功能影响分类，核心类型如表 11.2 所示。

变异类型	含义
同义突变（synonymous）	密码子改变但编码的氨基酸不变（简并性）
错义突变（missense）	密码子改变导致氨基酸改变，可能影响蛋白功能
无义突变（nonsense）	密码子变为终止密码子，蛋白提前截断
移码突变（frameshift）	插入/缺失的碱基数非 3 的倍数，导致读框移位
基因间区（intergenic）	位于基因之间的非编码区
上游/下游（upstream/-downstream）	位于基因转录起始/终止位点附近

表 11.2： 主要变异类型及其含义

```
# 从 ANN 字段提取变异类型并计数
bcftools query -f '%INFO/ANN\n' \
  results/annotation/ERR15404827.ann.vcf \
  | cut -d'|' -f2 | sort | uniq -c | sort -rn
```

Listing 11.3： 统计各类型变异数量

解读主实验的注释结果

主实验过滤后变异极少（甚至为 0），因此注释结果也会很”干净”。这再次印证了”同源对照应接近零变异”的结论。真正丰富的注释结果将在延伸实验（野生株）中出现——届时你将看到大量错义突变、同义突变乃至基因间区变异，并可通过 Ts/Tv、变异类型分布评估变异集质量。

11.4　思考题

思考题 1

同义突变为什么不改变氨基酸？这体现了遗传密码的什么特性？为什么同义突变通常被认为是”中性”的？

思考题 2

错义突变与无义突变哪个对蛋白质功能的影响通常更大？为什么？

思考题 3

基因间区（intergenic）的变异为什么通常比编码区的变异更” 容忍”？这对理解进化有什么启发？

Chapter 12

模块 7：结果汇总与延伸实验

模块目标

本模块将前六个模块的成果汇总为一份规范的分析报告，并通过一个” 野生分离株” 的延伸实验，对比不同菌株、不同深度条件下的变异检测结果，深化对变异检测原理与局限性的理解。

12.1　学习目标

完成本模块后，学生应能：

- 将零散的中间结果汇总为一份结构化的分析报告；
- 独立完成一轮完整的” 野生株” 分析（举一反三、融会贯通）；
- 对比同源对照与异源样本的变异检测结果，理解测序深度、菌株差异对结果的影响；
- 学会用图表与文字呈现、解读分析结论。

12.2　步骤 1：汇总主实验关键结果

请整理并记录主实验（ERR15404827）的以下关键指标，形成分析报告的” 结果” 部分：

指标	数值	来源命令
原始 read 对数	约 49.7 万	fastp 报告
质控后 read 对数	记录	fastp 报告
Q30 比例	记录	fastp 报告
比对率	记录	`samtools flagstat`
正确配对比率	记录	`samtools flagstat`
平均测序深度	记录	`samtools coverage`
覆盖度	记录	`samtools coverage`
原始变异数	记录	`grep -vc "#" raw.vcf`
过滤后变异数	记录	`grep -vc "#" filtered.vcf`
SNP / InDel 数	记录	`bcftools stats`

表 12.1：分析报告关键指标汇总表

12.3　步骤 2：撰写分析报告

一份规范的生信分析报告通常包含以下结构，请据此撰写（可参考附录 F.2 的模板）：

1. **标题与摘要**：一句话说明分析目的、数据、方法与主要结论；
2. **数据与方法**：数据集信息（来源、平台、读长、深度）、软件版本、关键参数；
3. **结果**：质控统计、比对统计、变异检测与注释结果（配图表）；
4. **讨论**：解读结果、说明局限、回应”同源对照接近零变异”的科学意义；
5. **结论**：概括本项目最重要的发现与收获。

可复现性原则

报告中务必**记录所有软件版本**（`samtools --version`、`bcftools --version` 等）与**关键参数**，这是分析可复现、结果可信赖的基础。建议把所有命令保存到 `scripts/` 目录下的脚本文件中。

12.4　步骤 3：延伸实验——野生分离株分析

现在，请**独立**完成延伸实验：分析野生大肠杆菌分离株 `DRR063436`（数据信息见第 1 章表 1.1），并与 MG1655 参考基因组比对、检测变异。

12.4.1　延伸实验数据下载

```
cd ~/ecoli_reseq/data

wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/DRR063/DRR063436/DRR063436_1.fastq.gz
wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/DRR063/DRR063436/DRR063436_2.fastq.gz

md5sum DRR063436_1.fastq.gz DRR063436_2.fastq.gz
# 与 ENA 官方 MD5 比对
```

Listing 12.1: 下载延伸实验数据

12.4.2　延伸实验完整流程

延伸实验的流程与主实验**完全一致**,只需将文件名 ERR15404827 替换为 DRR063436。请按以下顺序独立完成（提示：读长为 75 bp，深度约 4×）：

1. **质控**：FastQC 评估 + fastp 过滤（注意读长仅 75 bp，`--length_required` 可适当下调，如 40）；
2. **比对**：BWA-MEM 比对到同一个 MG1655 参考基因组；
3. **后处理**：排序、fixmate、markdup、索引、统计；
4. **变异检测**：bcftools mpileup（`--ploidy 1`）+ call；
5. **过滤**：尝试与主实验相同的过滤条件，并对比；
6. **注释**：SnpEff 注释，统计变异类型。

```
fastp \
  -i data/DRR063436_1.fastq.gz \
  -I data/DRR063436_2.fastq.gz \
  -o results/qc/DRR063436_1.clean.fastq.gz \
  -O results/qc/DRR063436_2.clean.fastq.gz \
  --html results/qc/fastp_DRR063436.html \
  --json results/qc/fastp_DRR063436.json \
  --thread 4 \
  --qualified_quality_phred 20 \
  --unqualified_percent_limit 40 \
  --length_required 40
```

Listing 12.2: 延伸实验质控（读长 75 bp 的适配）

延伸实验的关键差异

野生株 DRR063436 与参考 MG1655 之间**真实存在大量 SNP**。你将检测到数千乃至上万个变异——这与主实验的”接近零变异”形成鲜明对比。请重点观察：在仅 4× 的低深度下，变异结果的质量如何？QUAL 与 DP 的分布与主实验有何不同？假阳性是否显著增多？

12.5　步骤 4：对比分析与讨论

完成两个数据集的分析后，请填写对比表并展开讨论：

指标	主实验（MG1655，27×）	延伸实验（野生株，4×）
过滤前变异数	记录	记录
过滤后变异数	记录	记录
Ts/Tv 比值	记录	记录
错义突变数	记录	记录
同义突变数	记录	记录

表 12.2：两数据集变异检测结果对比表

讨论要点

请围绕以下问题展开讨论：

1. 为什么延伸实验检测到大量变异，而主实验几乎为零？这说明了什么？
2. 两个数据集的 Ts/Tv 比值有何差异？低深度数据的 Ts/Tv 是否更偏离理论值（2–3）？为什么？
3. 4× 深度下，过滤参数（DP、QUAL）应如何调整？”高深度 + 严过滤”与”低深度 + 松过滤”各自的利弊是什么？
4. 结合本项目，谈谈你对”测序深度是变异检测质量的基石”这句话的理解。

12.6　项目总结

恭喜你完成了完整的生物信息学入门项目！回顾整个过程，你已独立走通了：

数据获取 → 质量控制 → 序列比对 → 比对后处理 → 变异检测 → 变异注释 → 结果解读

这条”黄金流程”是绝大多数基因组学分析的共同骨架——无论是人类遗传病研究、肿瘤基因组学，还是病原微生物溯源，都遵循同样的逻辑。你在此项目中学到的**命**

令行技能、格式理解、算法思想、质量控制意识与批判性思维，将是你继续深入生物信息学的坚实基础。

进一步学习建议

- 学习 `GATK`（Genome Analysis Toolkit）——人类/二倍体基因组变异检测的行业标准，理解其 BQSR、VQSR 等高级步骤；
- 学习 RNA-seq 分析流程（比对、定量、差异表达），探索转录组领域；
- 学习 Snakemake / Nextflow 等工作流语言，将本项目流程自动化、规模化；
- 学习 Python/R 数据可视化，提升结果呈现能力。

附录 A

软件与环境清单汇总

A.1　完整软件清单

本项目所需的全部软件及其推荐安装方式汇总如下：

软件	版本	用途	所属频道
Miniconda	24.x	环境管理器	官方安装脚本
mamba	1.5.x	快速包管理	conda-forge
FastQC	0.12.1	质量评估	bioconda
fastp	0.23.4	质量过滤	bioconda
BWA	0.7.18	序列比对	bioconda
samtools	1.20	BAM 处理	bioconda
bcftools	1.20	变异检测	bioconda
SnpEff	5.2	变异注释	bioconda
tabix	1.20	表格索引	bioconda

表 A.1: 软件与环境清单汇总

A.2　一键式环境配置

A.2.1　方式一：逐条命令安装

```
# 1. 安装 Miniconda（略，见第 2 章）
# 2. 配置镜像源（可选）
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
conda config --set show_channel_urls yes

# 3. 创建环境并安装全部软件
conda create -n bioinfo -y -c conda-forge -c bioconda \
    mamba fastqc fastp bwa samtools bcftools snpeff tabix
```

```

# 4. 激活环境
conda activate bioinfo

# 5. 下载 SnpEff 数据库
snpEff download -v ASM584v2
```

Listing A.1: 逐条命令安装（推荐入门使用）

A.2.2　方式二：使用环境文件（可复现）

将以下内容保存为 `environment.yml`,然后执行 `conda env create -f environment.yml`:

```
name: bioinfo
channels:
  - conda-forge
  - bioconda
dependencies:
  - mamba
  - fastqc=0.12.1
  - fastp=0.23.4
  - bwa=0.7.18
  - samtools=1.20
  - bcftools=1.20
  - snpeff=5.2
  - tabix=1.20
```

Listing A.2: environment.yml 环境文件

```
conda env create -f environment.yml
conda activate bioinfo
snpEff download -v ASM584v2
```

Listing A.3: 使用环境文件创建环境

两种方式如何选择?

方式一简单直观，适合首次入门；方式二通过 `environment.yml` 锁定版本，适合团队协作与长期复现，是科研工作的推荐做法。二者产出的环境完全等价。

A.3　Windows 用户：安装 WSL2

Windows 10/11 用户可通过 WSL2 获得完整的 Linux 环境：

```
# 安装 WSL 与 Ubuntu
wsl --install -d Ubuntu-22.04

# 重启电脑后，Ubuntu 会自动启动，设置用户名与密码
# 之后在 Ubuntu 终端中按本教程第 2 章继续配置
```

Listing A.4: 安装 WSL2（在 Windows PowerShell 中以管理员身份运行）

安装完成后，在 Ubuntu 中执行 `conda` 相关命令即可。Windows 的 C、D 盘等文件可在 WSL 中通过 `/mnt/c`、`/mnt/d` 路径访问。

不要在 Windows 原生环境直接运行

不要在 Windows 的 CMD 或 PowerShell 中直接运行 BWA、samtools 等 Linux 工具（除非使用专门的 Windows 版或通过 WSL）。绝大多数生信工具没有 Windows 原生版本，强行运行会遇到各种兼容问题。

附录 B

常见问题与排错

问题现象	原因与解决方法
`conda: command not found`	conda 未加入 PATH。重新打开终端，或执行 `source ~/.bashrc`；仍失败则检查安装时是否选择了初始化
`conda install` 速度极慢	网络问题。配置国内镜像源（见附录 A.2），或用 `mamba install` 加速
`bwa index` 报错 `[E::...]`	参考基因组文件损坏或格式错误。确认 FASTA 已正确解压、路径正确
`samtools: failed to open index`	缺少 BAM 索引（.bai）。先执行 `samtools index xxx.bam`
`mpileup` 报 `reference is not indexed`	参考基因组缺少.fai 索引。执行 `samtools faidx ref.fna`
`[mpileup] fail to load index`	BAM 未按坐标排序或索引过期。重新排序并重建索引
变异数为 0 或异常多	检查参考序列名是否一致、pliyody 是否设为 1、过滤参数是否合理
`SnpEff: database not found`	SnpEff 数据库未下载或名称错误。用 `snpEff databases` 查询准确名称
比对率异常低（<50%）	数据与参考不匹配、污染严重或参数错误。检查物种、FastQC 报告
`wget` 下载失败/超时	网络波动。加 `-c` 断点续传重试，或更换镜像（如 NCBI/ENA 互换）
`Permission denied`	文件/目录权限不足。用 `ls -l` 检查，必要时 `chmod +x` 或切换目录
`No space left on device`	磁盘空间不足。清理中间文件（如 SAM），或扩充磁盘

表 B.1: 常见问题与解决方法

排错的通用方法论

遇到报错时，请按以下顺序排查：

1. **完整阅读报错信息**——错误信息通常直接指出了问题所在（缺少文件、参数错误、版本不兼容等）；
2. **检查路径与文件名**——生信报错中相当一部分是拼写或路径错误；

3. **检查输入文件**——文件是否损坏？索引是否过期？格式是否正确？
4. **查阅官方文档与社区**——**工具名** `--help`、官方手册、Biostars、SeqAnswers、GitHub Issues；
5. **最小化复现**——用一个最小的示例定位问题。

附录 C

术语表

术语	解释
生物信息学（Bioinformatics）	运用计算机与统计方法处理、分析、解读生物数据的交叉学科
高通量测序（NGS）	可并行测定数百万条 DNA 片段序列的技术
读段（read）	测序仪输出的一条短序列
单端/双端测序（SE/PE）	只测片段一端 / 两端都测
Phred 分数（Q）	碱基质量的度量，$Q = -10\log_{10} P$
FASTA	存储序列的纯文本格式
FASTQ	存储序列及其质量分数的格式
SAM/BAM	存储比对结果的格式（文本/二进制）
VCF/BCF	存储变异位点的格式（文本/二进制）
比对（mapping/alignment）	将 read 定位到参考基因组的过程
MAPQ	比对质量分数
CIGAR	描述比对细节的紧凑字符串
测序深度（depth）	每个基因组位置平均被覆盖的 read 数
覆盖度（coverage）	被至少 1 条 read 覆盖的碱基占比
PCR 重复（duplicate）	文库扩增产生的坐标相同的冗余 read
变异检测（variant calling）	从比对结果判定变异位点的过程
SNP	单核苷酸多态性（单个碱基的变异）
InDel	插入/缺失变异
基因型（genotype）	某位点的等位基因组成（如 0/1、1/1）
转换/颠换（Ts/Tv）	嘌呤-嘌呤或嘧啶-嘧啶替换 / 嘌呤-嘧啶替换
同义突变（synonymous）	密码子改变但不改变氨基酸
错义突变（missense）	密码子改变导致氨基酸改变
无义突变（nonsense）	密码子变为终止密码子
移码突变（frameshift）	插入/缺失导致读框移位

表 C.1: 核心术语表

附录 D

全部命令速查

以下按模块汇总本项目使用的核心命令，便于快速查阅（假设已在 `ecoli_reseq/` 目录下，环境已激活）。

D.1　模块 1：数据获取

```
# 参考基因组
cd ref
wget -c https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz
wget -c https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.gff.gz
gunzip GCF_000005845.2_ASM584v2_genomic.fna.gz

# 测序数据（主实验）
cd ../data
wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/ERR15404827_1.fastq.gz
wget -c https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/ERR15404827_2.fastq.gz

# 校验
md5sum *.fastq.gz
```

Listing D.1: 模块 1 命令速查

D.2　模块 2：质量控制

```
fastqc -t 4 -o results/qc/ data/*.fastq.gz
fastp -i data/ERR15404827_1.fastq.gz -I data/ERR15404827_2.fastq.gz \
  -o results/qc/ERR15404827_1.clean.fastq.gz -O results/qc/ERR15404827_2.clean.fastq.gz \
```

```
  --html results/qc/fastp.html --json results/qc/fastp.json \
  --thread 4 --qualified_quality_phred 20 --unqualified_percent_limit 40 \
  --length_required 50 --cut_front --cut_tail
```

Listing D.2: 模块 2 命令速查

D.3　模块 3：序列比对

```
bwa index ref/GCF_000005845.2_ASM584v2_genomic.fna
samtools faidx ref/GCF_000005845.2_ASM584v2_genomic.fna
bwa mem -t 4 ref/GCF_000005845.2_ASM584v2_genomic.fna \
  results/qc/ERR15404827_1.clean.fastq.gz results/qc/ERR15404827_2.clean.fastq.gz \
  > results/align/ERR15404827.sam
```

Listing D.3: 模块 3 命令速查

D.4　模块 4：比对后处理

```
samtools view -bS results/align/ERR15404827.sam -o results/align/ERR15404827.bam
samtools sort -@ 4 results/align/ERR15404827.bam -o results/align/ERR15404827.sorted.bam
samtools fixmate -m results/align/ERR15404827.sorted.bam results/align/ERR15404827.fixmate.bam
samtools sort -@ 4 results/align/ERR15404827.fixmate.bam -o results/align/ERR15404827.fixmate.sorted.bam
samtools markdup results/align/ERR15404827.fixmate.sorted.bam results/align/ERR15404827.dedup.bam
samtools index results/align/ERR15404827.dedup.bam
samtools flagstat results/align/ERR15404827.dedup.bam
samtools coverage results/align/ERR15404827.dedup.bam
```

Listing D.4: 模块 4 命令速查

D.5　模块 5：变异检测

```
bcftools mpileup --ploidy 1 -f ref/GCF_000005845.2_ASM584v2_genomic.fna \
```

```
  results/align/ERR15404827.dedup.bam | \
bcftools call -mv -Ov -o results/variants/ERR15404827.raw.vcf
bcftools filter -i 'QUAL >= 30 && DP >= 10' \
  results/variants/ERR15404827.raw.vcf -o results/variants/ERR15404827.filtered.
    vcf
bcftools stats results/variants/ERR15404827.filtered.vcf > results/variants/
    stats.txt
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%QUAL\t%DP\n' results/variants/
    ERR15404827.filtered.vcf
```

Listing D.5: 模块 5 命令速查

D.6　模块 6：变异注释

```
snpEff -v ASM584v2 results/variants/ERR15404827.filtered.vcf > results/
    annotation/ERR15404827.ann.vcf
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\t%INFO/ANN\n' results/annotation/
    ERR15404827.ann.vcf
```

Listing D.6: 模块 6 命令速查

附录 E

思考题参考答案

关于参考答案

以下答案用于自检，建议先独立思考后再对照。部分开放性问题没有唯一答案，重在理解背后的原理。

模块 1 参考答案

1. **为什么用.gz 压缩?**：FASTQ/FASTA 文件体积庞大，gzip 压缩可节省 70–80% 存储与传输成本；且 gzip 是流式压缩，工具（如 fastp、bwa）可直接读取.gz 文件而无需先解压，兼顾了效率与便利。
2. **R1/R2 的含义**：双端测序从同一 DNA 片段的两端各测一条 read，分别存入 R1、R2 文件。二者按顺序一一配对。若遗漏 R2，则退化为单端数据，丢失了插入片段信息，比对准确性与变异检测可靠性都会下降。
3. **MD5 不一致的可能原因**：文件传输损坏（网络中断）、下载不完整（中断后未续传）、文件被篡改、或下载了错误版本的文件。

模块 2 参考答案

1. 说明测序循环后期信号衰减、错误率升高。fastp 的 `--cut_tail`（或 `--cut_right`）会从末端滑动窗口修剪低质量碱基。
2. 过滤会丢弃低质量/过短的 read，故总数减少。损失 >30% 说明数据整体质量差、或过滤参数过严（如 `--length_required` 过高、质量阈值过严），应检查 FastQC 报告并调整参数。
3. 当插入片段短于测序读长时，测序”读穿”了整个片段，继续读到另一端的接头序列，故接头污染出现在 read 末端。开头不会，因为测序从插入片段内部开始。

模块 3 参考答案

1. 索引（BWT/FM-index）把” 逐位置扫描比对”（代价随参考长度线性增长）转化为” 常数级快速查询”，使得数百万条 read 比对成为可能。索引虽大，但一次建立、反复使用，且换来数量级的加速，非常划算。
2. FLAG=99 = 1 + 2 + 32 + 64，即：双端测序（1）、与 mate 均正确比对（2）、mate 在负链（32）、是 read1（64）。
3. MAPQ=0 表示比对位置不唯一（可能比对到多个位置，如重复区）。这类 read 的来源不确定，若用于变异检测会引入错误，故通常需过滤掉。

模块 4 参考答案

1. 按坐标排序后，同一位置的 read 在文件中连续存放，变异检测（mpileup）可以顺序” 堆积” 同一位置的 read，避免随机访问带来的巨大开销。markdup 也需要坐标有序才能高效比较 read 对。
2. PCR 重复是同一 DNA 分子的多个扩增拷贝，信息冗余且会放大测序错误；生物学重复是独立来源的样本。变异检测需独立证据，PCR 重复只会放大错误、造成假阳性，故须去除。
3. 用” 数据量/基因组大小” 估算为约 124 Mb / 4.64 Mb ≈ 27×。实际值略低的可能原因：部分 read 被过滤/去重、部分 read 比对到参考基因组之外的序列（如质粒）、覆盖不均等。

模块 5 参考答案

1. 二者都是 Phred 尺度的质量分数，但对象不同：FASTQ 的碱基质量衡量” 该碱基被测序仪读错” 的概率；VCF 的 QUAL 衡量” 该位点不是变异（判定错误）” 的概率。
2. QUAL 反映变异判定的统计置信度，DP 反映证据量（覆盖 read 数）。只设 QUAL 会放过” 高置信但证据薄弱” 的位点（如 2 条 read 支持却置信高）；只设 DP 会放过” 证据足但置信低” 的位点。二者结合更稳健。
3. 更严的过滤（DP≥20、QUAL≥60）会进一步减少变异数，去掉更多疑似假阳性，但也可能漏掉真实变异（尤其低深度区域），体现” 灵敏度 vs 精确度” 的权衡。

模块 6 参考答案

1. 遗传密码具有简并性（多个密码子编码同一氨基酸），故密码子第三位常发生同义替换而不改变氨基酸。同义突变不改变蛋白序列，通常不受（或很弱地受）自然选择，因此被认为是” 中性” 的。
2. 无义突变影响通常更大：它导致蛋白提前截断，几乎必然破坏功能；错义突变仅改变一个氨基酸，其影响取决于位置与性质差异，可能是中性、有害或（极少）有利。
3. 基因间区不编码蛋白，变异通常不直接影响蛋白功能，因而更” 容忍”、更易保留。这提示非编码区是进化中重要的变异” 蓄水池”，也可能蕴含调控元件等功能信息。

附录 F

数据集信息卡与报告模板

F.1　数据集信息卡

F.1.1　主实验数据集：ERR15404827

字段	值
Run 访问号	ERR15404827
样本访问号	SAMEA118914020
研究项目	PRJEB94349
物种	Escherichia coli str. K-12 substr. MG1655
测序平台	Illumina MiSeq
文库类型	双端（PAIRED），WGS，随机打断
读长	250 bp
read 对数	约 49.7 万
数据量	约 124 Mb（估算深度约 27×）
R1 下载地址	https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/ERR15404827_1.fastq.gz
R2 下载地址	https://ftp.sra.ebi.ac.uk/vol1/fastq/ERR154/027/ERR15404827/ERR15404827_2.fastq.gz

表 F.1: 主实验数据集信息卡

F.1.2　延伸实验数据集：DRR063436

字段	值
Run 访问号	DRR063436
样本访问号	SAMD00053107
研究项目	PRJDB4884
物种	Escherichia coli（野生分离株）
测序平台	Illumina MiSeq
文库类型	双端（PAIRED），WGS，随机打断
读长	75 bp
read 对数	约 12.4 万
数据量	约 18 Mb（估算深度约 4×）
R1 下载地址	https://ftp.sra.ebi.ac.uk/vol1/fastq/DRR063/DRR063436/DRR063436_1.fastq.gz
R2 下载地址	https://ftp.sra.ebi.ac.uk/vol1/fastq/DRR063/DRR063436/DRR063436_2.fastq.gz

表 F.2: 延伸实验数据集信息卡

F.1.3　参考基因组信息卡

字段	值
物种	Escherichia coli str. K-12 substr. MG1655
RefSeq 访问号	GCF_000005845.2
染色体序列号	NC_000913.3
基因组大小	4,641,652 bp
基因组下载地址	https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz
注释下载地址	https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.gff.gz
SnpEff 数据库名	ASM584v2

表 F.3: 参考基因组信息卡

F.2　分析报告模板

以下为分析报告的结构模板，学生可据此撰写最终报告：

```
# 大肠杆菌重测序数据分析报告

## 1. 摘要
（一段话概括：目的、数据、方法、主要结论）
```

```
## 2. 数据与方法
- 数据集：ERR15404827（主）/ DRR063436（延伸）
- 参考基因组：GCF_000005845.2（NC_000913.3）
- 软件版本：FastQC 0.12.1、fastp 0.23.4、BWA 0.7.18、
  samtools 1.20、bcftools 1.20、SnpEff 5.2
- 关键参数：质量过滤 Q>=20；比对 BWA-MEM 默认；
  变异检测 ploidy=1；过滤 QUAL>=30 && DP>=10

## 3. 结果
- 3.1 质控统计（FastQC/fastp 关键指标）
- 3.2 比对统计（比对率、深度、覆盖度）
- 3.3 变异检测结果（数量、QUAL/DP 分布、Ts/Tv）
- 3.4 变异注释结果（类型分布）

## 4. 讨论
- 同源对照接近零变异的科学意义
- 深度对变异检测的影响（对比主/延伸实验）
- 局限性与改进方向

## 5. 结论
（概括最重要的发现与收获）
```

Listing F.1: 分析报告结构模板