

# 基于SRA数据的全基因组变异检测流程构建与实现——BIO2503期末项目报告

作者：王誉凯

学号：5233111910158

小组：课程项目组2（个人）

课程：Linux操作系统与Shell开发的理论与实践

## 摘要

本项目构建了一个基于NCBI SRA数据库原始测序数据的基因组变异检测流程，流程包含了数据下载、质控、比对、变异识别和结果过滤。项目以大肠杆菌（E. coli K-12 MG1655）测序数据为例，使用Conda环境管理工具完成生信工具的依赖配置，采用Shell脚本实现自动化分析流程，使用FastQC、AdapterRemoval、BWA、Samtools、BCFtools生物信息学工具进行数据的处理与分析，最后输出标准VCF变异结果。脚本已发布至GitHub：[wangyukai585/bio2503\\_final\\_project](https://github.com/wangyukai585/bio2503_final_project)，便于复现与共享。

## 1 前言

随着高通量测序技术的广泛应用，基因组变异检测已成为基因组研究的重要组成部分。全基因组测序（WGS, Whole Genome Sequencing）是下一代测序技术，用于快速、低成本地确定生物体的完整基因组序列，其目的是准确检测出每个样本基因组中的变异集合。

通过构建一个标准化的分析流程，我们可以高效地处理原始测序数据，发现潜在的单核苷酸变异（SNP）与插入缺失变异（InDel），为后续基因的功能分析打下基础。

本项目以大肠杆菌（E. coli K-12 MG1655）为模型，使用公开测序数据SRR5168216完成全流程分析，内容包括：

- Conda环境创建

- 原始数据下载与解压
  - 质量控制 (FastQC)
  - 接头去除与低质量过滤 (AdapterRemoval)
  - 比对至参考基因组 (BWA)
  - BAM 文件处理与索引 (Samtools)
  - 变异识别与过滤 (BCFtools)
- 

## 2 数据集与方法

### 2.1 数据来源

- 原始数据: NCBI SRA 数据集 SRR5168216
- 参考基因组: E. coli K-12 MG1655 (NCBI FTP 提供)

### 2.2 主要工具与版本

- fastqc : 0.12.1
- adapterremoval : 2.3.4
- bwa : 0.7.19
- samtools : 1.21
- bcftools : 1.21
- conda (环境管理): 环境名 bioinfo\_course\_project

### 2.3 Conda 环境构建

运行脚本 `scripts/1_create_env.sh` 自动完成安装与依赖配置。

### 2.4 数据处理流程

每一步均有独立脚本:

步骤	脚本文件	功能
1	1_create_env.sh	创建 Conda 环境
2	2_download_data.sh	下载FASTQ文件
3	3_fastqc.sh	运行FastQC质控
4	4_adapter_removal.sh	剪切低质量和接头序列
5	5_alignment.sh	BWA比对并排序索引
6	6_variant_calling.sh	BCFtools调用变异
7	7_filter_variants.sh	筛选高质量VCF变异

## 2.5 详细分析流程

注意：开始执行分析流程前，系统应预装好Miniconda或Anaconda！！

### 1. 运行第一个脚本：创建环境

```
bash 1_create_env.sh
# 系统提示:使用conda activate bioinfo_course_project创建环境
conda activate bioinfo_course_project
```

### 2. 运行第二个脚本：下载数据

```
bash 2_download_data.sh
```

此时，项目结构应为：

```
bio2503_final_project/
└─ data/
    └─ genome/
        └─ Ecoli_K12_MG1655.fa # 参考基因组: E. coli K-12 MG1655
    └─ raw/
        └─ SRR5168216_1.fastq.gz # 数据集 SRR5168216
        └─ SRR5168216_2.fastq.gz # 数据集 SRR5168216
```

### 3. 运行第三个脚本：检查测序质量（使用FastQC）

```
bash 3_fastqc.sh
```

此时，输出目录结构如下：

```
results/
└─ fastqc/
    └─ SRR5168216_1_fastqc.html # 可以用浏览器打开 *.html 文件查看质量报告
    └─ SRR5168216_2_fastqc.html # 可以用浏览器打开 *.html 文件查看质量报告
    └─ SRR5168216_1_fastqc.zip
    └─ SRR5168216_2_fastqc.zip
```

经查看输出的质量报告，发现末端质量下降、接头污染。这意味着接头去除操作的必要性！

具体的质量报告，将于报告“结果”章节展示！

### 4. 运行第四个脚本：接头去除与质量修剪（使用AdapterRemoval）

```
bash 4_adapter_removal.sh
```

执行后将生成：

```
data/clean/
├─ SRR5168216_trimmed_1.fastq # 保留的高质量 paired reads
├─ SRR5168216_trimmed_2.fastq # 保留的高质量 paired reads
├─ SRR5168216_collapsed.fastq # 合并成一条的 overlapping reads
results/adapterremoval/
├─ SRR5168216_stats.txt # 修剪统计报告
```

## 5. 运行第五个脚本：参考基因组比对（使用BWA）

```
bash 5_alignment.sh
```

执行后生成：

```
results/bam/
├─ SRR5168216.bam           # 原始 BAM（可选保留）
├─ SRR5168216.sorted.bam   # 排序后的 BAM（主文件）
├─ SRR5168216.sorted.bam.bai # BAM 索引文件
```

## 6. 运行第六个脚本：变异检测（使用bcftools）

```
bash 6_variant_calling.sh
```

输出结构：

```
results/vcf/
├─ SRR5168216_raw.bcf # 原始变异检测结果
├─ SRR5168216_raw.vcf.gz # 原始变异检测结果
├─ SRR5168216_raw.vcf.gz.csi # 对应的索引文件
```

## 7. 运行第七个脚本：过滤低质量变异（使用 bcftools filter）


```
bash 7_filter_variants.sh
```

输出结果目录结构：











```
results/vcf/
├─ SRR5168216_filtered.vcf.gz # 筛选后的变异结果
└─ SRR5168216_filtered.vcf.gz.csi # 筛选后 VCF 文件的索引
```

## 结果

- 接头去除与质量修剪前，FASTQC报告：（以SRR5168216\_1.fastq.gz为例）

 **FastQC Report**

### Summary

-  [Basic Statistics](#)
-  [Per base sequence quality](#)
-  [Per sequence quality scores](#)
-  [Per base sequence content](#)
-  [Per sequence GC content](#)
-  [Per base N content](#)
-  [Sequence Length Distribution](#)
-  [Sequence Duplication Levels](#)
-  [Overrepresented sequences](#)
-  [Adapter Content](#)

### Basic Statistics

Measure	Value
Filename	SRR5168216_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	1572822
Total Bases	471.8 Mbp
Sequences flagged as poor quality	0
Sequence length	300
%GC	51

### Per base sequence quality

Per base sequence quality显示红色x，表示末端质量下降；Adapter Content显示红色x，表示接头污染。

- 接头去除与质量修剪后，FASTQC报告：（以SRR5168216\_1.fastq.gz为例）

# FastQC Report

## Summary

- ✓ [Basic Statistics](#)
- ✓ [Per base sequence quality](#)
- ✓ [Per sequence quality scores](#)
- ✗ [Per base sequence content](#)
- ! [Per sequence GC content](#)
- ✓ [Per base N content](#)
- ! [Sequence Length Distribution](#)
- ✓ [Sequence Duplication Levels](#)
- ✓ [Overrepresented sequences](#)
- ✓ [Adapter Content](#)

## ✓ Basic Statistics

Measure	Value
Filename	SRR5168216_trimmed_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	1572822
Total Bases	433.6 Mbp
Sequences flagged as poor quality	0
Sequence length	66-300
%GC	51

## ✓ Per base sequence quality

Per base sequence quality显示绿色√，=Adapter Content显示绿色√，成功移除接头序列和低质量碱基！

- BWA成功比对至参考基因组，平均比对率超过83%。
- 成功删除了低质量变异。
- 最终获得变异识别结果（SRR5168216\_filtered.vcf.gz），包含高质量突变位点。

最终的文件框架：

```
bio2503_final_project/
├── scripts/
│   ├── 1_create_env.sh          # 所有分析流程脚本（核心部分，上传 GitHub）
│   ├── 2_download_data.sh      # 创建 Conda 环境脚本
│   ├── 3_qc_fastqc.sh          # 下载参考基因组和测序数据
│   ├── 4_trim_adapter.sh       # 运行 FastQC 进行质量控制
│   ├── 5_alignment.sh          # 使用 AdapterRemoval 去除接头并质控
│   ├── 6_variant_calling.sh    # 使用 BWA 和 SAMtools 进行比对和排序
│   └── 7_filter_variants.sh    # 使用 BCFtools 进行变异检测
│                               # 筛选高质量变异
└──
```

```

└─ data/raw/                                # 存放原始数据（仅本地存在，不上传）
|   └─ SRR5168216_1.fastq.gz
|   └─ SRR5168216_2.fastq.gz
|
└─ data/clean/                              # 清洗后的 FASTQ 文件
|   └─ SRR5168216_trimmed_1.fastq
|   └─ SRR5168216_trimmed_2.fastq
|
└─ data/genome/                             # 参考基因组文件及索引
|   └─ Ecoli_K12_MG1655.fa
|   └─ Ecoli_K12_MG1655.fa.bwt  等等索引文件...
|
└─ results/
|   └─ fastqc/                             # FastQC 分析结果
|       └─ *.html / *.zip
|   └─ bam/                               # 比对结果（BAM 文件）
|       └─ SRR5168216.sorted.bam
|   └─ vcf/                               # 变异结果
|       └─ SRR5168216_raw.bcf
|       └─ SRR5168216_raw.vcf.gz
|       └─ SRR5168216_raw.vcf.gz.csi
|       └─ SRR5168216_filtered.vcf.gz
|       └─ SRR5168216_filtered.vcf.gz.csi
|
└─ env/                                    # 环境定义文件
    └─ environment.yml
|
└─ bio2503_final_report.md                # Markdown 格式的项目报告
└─ bio2503_final_report.pdf              # 转换后的 PDF 报告

```

## 讨论

本项目关键点在于：



- 使用Conda管理环境，Conda 部分包（如AdapterRemoval）需添加bioconda源
- 采用Shell脚本实现自动化分析流程，每一个脚本都不能出现报错
- 使用FastQC、AdapterRemoval、BWA、Samtools、BCFtools，进行数据的处理与分析，工具有一定使用难度

本项目也有稍许不足：

- 对于目标基因样本的处理方式较少，数据分析的处理质量可能会出现不佳的情况（本项目中，数据处理质量处于中上的水平，但并未达到最佳）
- 由于时间原因，流程可以进一步丰富扩展，实现功能丰富的全基因组流程变异测序。

---

## 贡献：

王誉凯负责了全部的内容，包括脚本的编写、项目报告的撰写、github网站的上传与维护等。

---

## 参考文献

1. Andrews, S. (2010). *FastQC: A quality control tool for high throughput sequence data*. <https://www.bioinformatics.babraham.ac.uk/projects/fastqc>
2. Schubert, M., Lindgreen, S., & Orlando, L. (2016). AdapterRemoval v2: rapid adapter trimming, identification, and read merging. *BMC Research Notes*, 9(1), 88. <https://doi.org/10.1186/s13104-016-1900-2>
3. Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25(14), 1754–1760. <https://doi.org/10.1093/bioinformatics/btp324>
4. Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., ... & Li, H. (2021). Twelve years of SAMtools and BCFtools. *GigaScience*, 10(2), giab008. <https://doi.org/10.1093/gigascience/giab008>
5. SRA Database. National Center for Biotechnology Information. *Sequence Read Archive*. Retrieved from <https://www.ncbi.nlm.nih.gov/sra>

6. Blattner, F. R., Plunkett, G., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., ... & Shao, Y. (1997). The complete genome sequence of *Escherichia coli* K-12. *Science*, 277(5331), 1453-1462. <https://doi.org/10.1126/science.277.5331.1453>
- 

## 附录

所有脚本均已上传至 GitHub 项目仓库 [wangyukai585/bio2503\\_final\\_project](https://github.com/wangyukai585/bio2503_final_project)。

下面列举本项目的关键脚本及其功能简述：

### 1. 1\_create\_env.sh

用于配置环境。

```
#!/bin/bash

set -e # 遇到错误立即退出脚本
ENV_NAME="bioinfo_course_project"

echo "正在准备 Conda 环境：$ENV_NAME"

# 添加 Bioconda 和 Conda-forge 频道（只需添加一次）
if ! grep -q "bioconda" ~/.condarc 2>/dev/null; then
    echo "首次添加 Conda 频道"
    conda config --add channels defaults
    conda config --add channels bioconda
    conda config --add channels conda-forge
    conda config --set channel_priority strict
else
    echo "Conda 频道已配置，无需重复添加"
fi

# 检查环境是否已存在
if conda info --envs | grep -q "^$ENV_NAME"; then
    echo "环境 $ENV_NAME 已存在，先删除..."
```

```
conda remove -y --name $ENV_NAME --all
fi

# 创建环境
echo "开始创建环境 $ENV_NAME..."
conda create -y -n $ENV_NAME fastqc bwa samtools seqtk adapterremoval
picard sra-tools bcftools

echo "Conda 环境创建成功，请运行以下命令激活："
echo ""
echo "    conda activate $ENV_NAME"
```

## 2. 2\_download\_data.sh

用于下载参考基因组及SRR5168216样本的原始FastQ数据。

```
#!/bin/bash

set -e

echo "正在创建数据目录..."
mkdir -p data/genome data/raw

echo "下载 E. coli K12 MG1655 参考基因组..."
cd data/genome
wget -O Ecoli_K12_MG1655.fa.gz
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845
.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz
gunzip -f Ecoli_K12_MG1655.fa.gz

echo "直接下载 SRR5168216 paired-end FastQ 文件..."
cd ../raw
```

```
wget -c
https://ftp.sra.ebi.ac.uk/vol1/fastq/SRR516/006/SRR5168216/SRR5168216_
1.fastq.gz
wget -c
https://ftp.sra.ebi.ac.uk/vol1/fastq/SRR516/006/SRR5168216/SRR5168216_
2.fastq.gz

echo "下载完成，跳过 SRA 解压，进入 FastQC 步骤。"
```

### 3. 3\_fastqc.sh

对原始测序数据进行质量控制分析。

```
#!/bin/bash
set -e

mkdir -p results/fastqc

fastqc data/raw/SRR5168216_1.fastq.gz data/raw/SRR5168216_2.fastq.gz -
o results/fastqc -t 4
```

### 4. 4\_adapter\_removal.sh

使用 AdapterRemoval 清除接头序列和低质量 reads。

```
#!/bin/bash

set -e

echo "创建清洗后的数据目录 data/clean"
mkdir -p data/clean

echo "正在执行 AdapterRemoval 去除接头序列和低质量 reads..."

AdapterRemoval \
```

```
--file1 data/raw/SRR5168216_1.fastq.gz \  
--file2 data/raw/SRR5168216_2.fastq.gz \  
--output1 data/clean/SRR5168216_trimmed_1.fastq \  
--output2 data/clean/SRR5168216_trimmed_2.fastq \  
--trimns --trimqualities --minquality 20 --minlength 50 \  
--adapter1 AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \  
--adapter2 AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT  
  
echo "AdapterRemoval 处理完成, 输出文件保存在 data/clean/"
```

## 5. 5\_alignment.sh

使用 BWA 进行比对, 并使用 Samtools 排序和索引 BAM 文件。

```
#!/bin/bash  
  
set -e  
  
# 创建输出目录  
echo "创建比对结果目录 results/bam"  
mkdir -p results/bam  
  
# 构建参考基因组索引 (只需一次)  
if [ ! -f data/genome/Ecoli_K12_MG1655.fa.bwt ]; then  
    echo "构建 BWA 索引..."  
    bwa index data/genome/Ecoli_K12_MG1655.fa  
else  
    echo "索引已存在, 跳过"  
fi  
  
# 执行 BWA 比对  
echo "正在进行 BWA 比对..."  
bwa mem -t 4 data/genome/Ecoli_K12_MG1655.fa \  
    data/clean/SRR5168216_trimmed_1.fastq \  
    data/clean/SRR5168216_trimmed_2.fastq |  
    samtools view -Sb - > results/bam/SRR5168216.bam
```

```
# BAM 文件排序
echo "排序 BAM 文件..."
samtools sort -@ 4 -o results/bam/SRR5168216.sorted.bam
results/bam/SRR5168216.bam

# 创建 BAM 索引
echo "索引 BAM 文件..."
samtools index results/bam/SRR5168216.sorted.bam

echo "比对完成，已生成排序并索引的 BAM 文件"
```

## 6. 6\_variant\_calling.sh

调用变异并生成压缩的 VCF 文件。

```
#!/bin/bash

set -e

# 创建输出目录
echo "创建 VCF 目录 results/vcf"
mkdir -p results/vcf

# 构建参考基因组 fa 的索引（如不存在）
if [ ! -f data/genome/Ecoli_K12_MG1655.fa.fai ]; then
    echo "构建参考基因组索引..."
    samtools faidx data/genome/Ecoli_K12_MG1655.fa
fi

# 生成 BCF 文件（调用变异）
echo "生成 BCF 文件..."
bcftools mpileup -Ou -f data/genome/Ecoli_K12_MG1655.fa
results/bam/SRR5168216.sorted.bam |
    bcftools call -mv -Ob -o results/vcf/SRR5168216_raw.bcf
```

```
# 转换为 VCF 格式
echo "转换 BCF 为 VCF 格式..."
bcftools view results/vcf/SRR5168216_raw.bcf -Oz -o
results/vcf/SRR5168216_raw.vcf.gz

# 创建 VCF 索引
echo "索引 VCF 文件..."
bcftools index results/vcf/SRR5168216_raw.vcf.gz

echo "变异检测完成，结果保存为 VCF 文件: results/vcf/SRR5168216_raw.vcf.gz"
```

## 6. 7\_filter\_variants.sh

对原始 VCF 文件进行质量和深度过滤。

```
#!/bin/bash

set -e

echo "创建 VCF 过滤后输出目录 results/vcf"
mkdir -p results/vcf

echo "过滤低质量和低深度变异..."

bcftools filter -e 'INFO/DP<10 || QUAL<20' \
  results/vcf/SRR5168216_raw.vcf.gz -Oz -o
results/vcf/SRR5168216_filtered.vcf.gz

echo "索引过滤后的 VCF..."
bcftools index results/vcf/SRR5168216_filtered.vcf.gz

echo "过滤完成，输出文件: results/vcf/SRR5168216_filtered.vcf.gz"
```