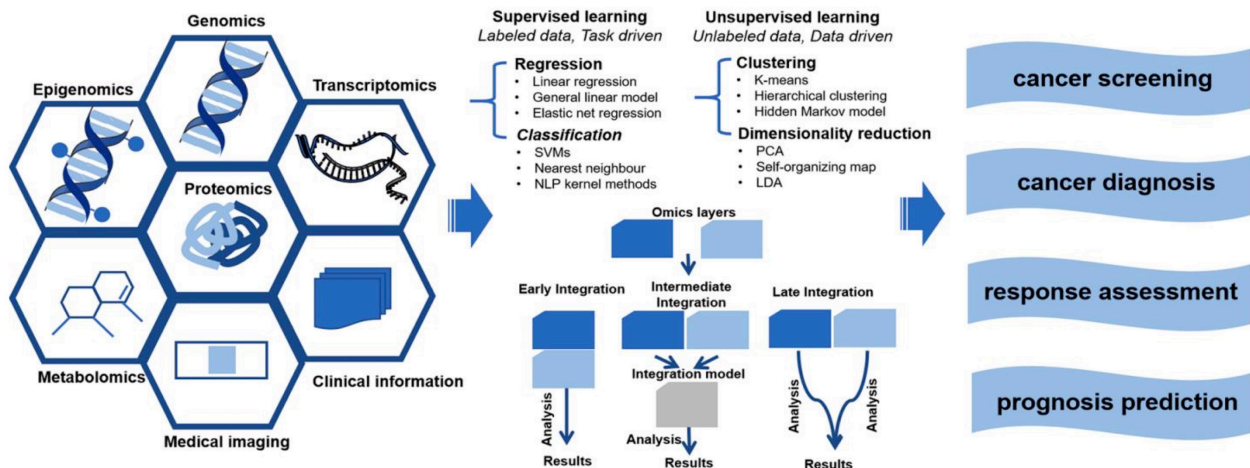


Data sources

Machine learning and
Integration analysis

Clinical tasks



多模态数据分析平台——使用手册

Multimodal Data Analysis Platform —User Guide

作者：课题四：基于专病队列的儿童重要肿瘤与血液性疾病致病性遗传改变及驱动机制研究

组织：“十四五”国家科技部重点研发计划项目（课题编号：2022YFC2705004）

时间：December 25, 2025

版本：2.0

Together we can save more lives

目录

第 1 章 平台概述	1
1.1 平台定位	1
1.2 核心价值	1
1.3 适用人群	1
1.4 快速入门	1
1.4.1 访问方式	1
1.4.2 界面导航	1
1.4.3 核心功能模块详解	2
1.4.3.1 多模态数据来源	2
1.4.3.2 分析工具集	2
1.4.3.3 临床应用场景	3
1.4.4 特色资源与研究成果	3
1.4.4.1 重点特色论文	3
1.4.4.2 优秀论文列表	3
1.4.5 常见问题	4
1.4.5.1 数据相关问题	4
1.4.5.2 权限与使用限制	4
1.4.5.3 成果引用规范	5
1.4.6 联系与支持	5
1.4.7 免责声明	5
第 2 章 新用户初始化配置	6
2.1 准备工作	6
2.2 首次登录操作	6
2.2.1 登录步骤	6
2.2.2 首次登录密码修改	6
2.3 熟悉 RStudio Server 界面	7
2.3.1 区域 1: 源代码编辑器	7
2.3.2 区域 2: 控制台	7
2.3.3 区域 3: 环境/历史/连接/指导	7
2.3.4 区域 4: 文件/plots/包/帮助/查看器/演示	7
2.4 初始化配置	7
2.4.1 工作目录设置	7
2.4.2 界面个性化配置	8
2.4.3 R 包安装与配置	8
2.4.3.1 安装单个包	8
2.4.3.2 批量安装常用包	8
2.4.3.3 加载已安装的包	8
2.4.4 新建 R 脚本文件	9
2.5 常用操作指引	9
2.5.1 文件保存与打开	9

2.5.2	数据导入	9
2.5.3	退出登录	9
2.6	常见问题解决	10
2.6.1	登录失败	10
2.6.2	包安装失败	10
2.6.3	界面中文乱码	10
2.7	附录	10
2.7.1	常用快捷键	10
2.7.2	国内 CRAN 镜像推荐	10
2.7.3	求助方式	11
第 3 章	RStudio Server 进阶使用	12
3.1	系统配置优化	12
3.1.1	核心配置文件	12
3.1.2	配置生效步骤	12
3.1.3	服务管理常用命令	12
3.2	多人协作与权限管理	13
3.2.1	新增协作用户	13
3.2.2	共享目录设置	13
3.3	Git 版本控制集成	13
3.3.1	前期准备	13
3.3.2	RStudio 配置 Git	13
3.3.3	项目绑定远程仓库	14
3.3.4	常用 Git 操作	14
3.4	高效 workflow 技巧	14
3.4.1	批量操作与自动化	14
3.4.2	R Markdown 报告生成	14
3.4.3	结果保存与导出	15
3.4.4	常见问题排查	15
3.4.5	登录失败	15
3.4.6	代码运行报错	15
3.4.7	会话崩溃/卡顿	15
3.4.8	常用快捷键汇总	15
3.4.9	官方资源链接	16
第 4 章	通过 RStudio Server 调用 Python	17
4.1	前期准备	17
4.1.1	服务器环境要求	17
4.1.2	工具准备	17
4.1.3	Conda	17
4.1.3.1	下载 Miniconda 安装包	17
4.1.3.2	安装 Miniconda	18
4.1.3.3	验证 Miniconda 安装	18
4.2	在 RStudio Server 中配置 conda 环境	18
4.2.1	创建 conda 虚拟环境	18

4.2.2	RStudio Server 关联 conda 虚拟环境	19
4.2.3	验证 conda 环境关联成功	19
4.3	在 RStudio Server 中使用 Python	19
4.3.1	直接在 Console 中执行 Python 代码	19
4.3.2	创建并运行 Python 脚本	20
4.4	生成随机数组	20
4.5	计算数组均值	20
4.5.1	在 conda 环境中安装 Python 包	20
4.6	常见问题与排查	21
4.6.1	conda 命令无法识别	21
4.6.2	RStudio 无法找到 conda 虚拟环境的 Python 解释器	21
4.6.3	安装 Python 包时速度慢或失败	21
4.6.4	RStudio 中调用 Python 提示 reticulate 包未找到	21
第 5 章	在 RStudio Server 中使用 Terminal	22
5.1	概述	22
5.2	Terminal 的开启与关闭	22
5.2.1	打开 Terminal	22
5.2.2	新建与切换多终端会话	22
5.2.3	关闭 Terminal 会话	22
5.3	Terminal 基础功能使用	23
5.3.1	基础命令操作	23
5.3.2	与 RStudio 编辑器的交互	23
5.3.3	外观与配置调整	23
5.4	进阶应用	23
5.4.1	运行长时间作业	23
5.4.2	RStudio Server 系统管理	24
5.4.3	远程登录与文件传输	24
5.5	常见问题与解决方案	24
5.5.1	终端显示异常	24
5.5.2	长时间作业中断	24
5.5.3	权限不足	24
5.5.4	无法新建 Terminal 会话	25
5.6	快捷键汇总	25
第 6 章	多模态医学数据预处理与下游分析核心工具汇总	26
6.1	多组学数据	26
6.1.1	基因组学数据	26
6.1.1.1	bigstatsr & bigsnpr	26
6.1.1.2	maftools	26
6.1.2	转录组学数据	27
6.1.2.1	GEO2RNAseq	27
6.1.2.2	DESeq2	27
6.1.3	表观基因组学数据	28
6.1.3.1	easyEWAS	28

6.1.3.2	minfi	28
6.1.4	蛋白组学数据	28
6.1.4.1	MSstats	28
6.1.5	代谢组学数据	29
6.1.5.1	XCMS	29
6.1.5.2	MetaboAnalystR	29
6.1.6	单细胞组学数据	30
6.1.6.1	核心 R 包	30
6.1.6.2	补充 Python 库	30
6.1.7	微生物组多模态数据	30
6.1.7.1	multimedia	31
6.1.7.2	MultiMed	31
6.1.8	多组学整合分析工具	31
6.1.8.1	mixOmics	31
6.1.8.2	omicade4	32
6.1.8.3	MultiModalGraphics	32
6.2	医学影像数据	32
6.2.1	ITKR	32
6.2.2	ANTsR	33
6.2.3	SimpleITKR	33
6.2.4	4. RNifti & oro.nifti	33
6.2.5	5. EBImage	34
6.2.6	6. RIA	34
6.3	数字病理图像数据	34
6.3.1	核心 Python 库	35
6.3.1.1	OpenSlide-python	35
6.3.1.2	HistomicsTK	35
6.3.2	主流软件与深度学习模型	35
6.3.2.1	QuPath	35
6.3.2.2	CellPose	35
6.4	临床文本数据	36
6.4.1	PubMedMining	36
6.4.2	tidytext	36
6.5	总结	36

第 1 章 平台概述

1.1 平台定位

本平台是一款专注于儿童重要肿瘤与血液性疾病的专业化数据分析工具，核心目标是通过整合多维度生物医学数据，为临床诊疗与科研工作提供精准化、系统化的分析支持，助力疾病研究向精准医疗方向发展。

1.2 核心价值

- 数据整合：打破单一数据维度局限，融合基因组学、转录组学、影像组学等多模态数据，实现全方位数据视角。
- 精准分析：依托生物信息学分析与机器学习算法，提供疾病亚型分类、治疗反应预测等核心分析能力。
- 临床转化：打通科研与临床的壁垒，将多组学研究成果快速转化为临床可用的筛查、诊断、治疗决策工具。

1.3 适用人群

- 临床医师：儿童肿瘤与血液科医护人员，用于疾病诊断、治疗方案选择及预后评估。
- 科研人员：从事儿童肿瘤、血液疾病、精准医疗相关研究的高校教师、研究员、研究生。
- 医疗研发人员：肿瘤靶向药物、免疫治疗药物研发团队，用于靶点验证与疗效预测研究。

1.4 快速入门

1.4.1 访问方式

- 直接访问官网：<https://fudan-sph.github.io/ceh/>
- 浏览器兼容：建议使用 Chrome、Firefox、Edge 等现代浏览器（版本 90.0 及以上），确保功能正常运行。

1.4.2 界面导航

平台界面设计简洁直观，主要导航栏包括：首页、使用手册、分析工具、帮助中心、关于我们、联系我们以及公告栏。

- 首页：平台简介、项目总体介绍、核心功能展示及最新科研成果。
- 使用手册：详细的功能介绍与操作指南。
- 分析工具：多模态数据分析流程与算法，单模态数据分析工具，以及数据清洗/可视化工具。
- 帮助中心：常见问题解答与技术支持信息。
- 关于我们：课题 PI、课题组背景、研究方向及团队成员。
- 联系我们：技术支持与科研合作联系方式。
- 公告栏：平台更新与重要通知发布。

1.4.3 核心功能模块详解

1.4.3.1 多模态数据来源

平台支持以下全维度数据类型的导入与整合分析，覆盖疾病研究的核心数据维度：

数据类别	具体类型	应用场景
组学数据	基因组学、表观遗传组学	基因突变、基因表达调控分析
	Bulk/单细胞转录组学	细胞异质性、基因表达谱分析
	蛋白质组学、代谢组学	蛋白功能、代谢通路机制研究
	微生物组学	肠道菌群与疾病关联性分析
影像/病理数据	医学影像组学 (Radiomics)	肿瘤病灶特征提取与量化分析
	病理组学 (Pathomics)	病理切片特征与疾病关联分析
临床数据	患者基本信息、诊疗记录、随访数据	临床特征与多组学数据整合分析

1.4.3.2 分析工具集

1.4.3.2.1 基础数据统计分析

- 数据清洗：缺失值处理、异常值检测与修正、数据标准化与归一化。
- 描述性统计：均值、中位数、标准差、频数分布等基本统计量计算。
- 可视化展示：柱状图、饼图、箱线图、散点图等多种图表类型。
- 相关性分析：Pearson/Spearman 相关系数计算与热图展示。
- 假设检验：t 检验、方差分析 (ANOVA)、卡方检验等常用统计检验方法。
- 生存分析：Kaplan-Meier 生存曲线绘制、Cox 比例风险模型构建。
- 临床特征关联分析：临床变量与组学数据的关联性分析。
- 亚组分析：基于临床特征或分子特征进行患者亚组划分与比较。
- 风险因素分析：多变量回归模型评估疾病风险因素。

1.4.3.2.2 生物信息学分析

- 基因组学分析：基因突变检测 (生殖系突变/体细胞突变)、基因拷贝数变异分析 (如 inferCNV 分析)。
- 表观遗传组学分析：DNA 甲基化水平评估、差异甲基化区域识别 (如肝癌相关甲基化位点分析)。
- 转录组学分析：基因表达量定量、差异表达基因筛选、通路富集分析 (如 STAT3 信号通路分析)。
- 蛋白质组学与代谢组学分析：蛋白质鉴定与定量、差异蛋白筛选、代谢通路分析 (如氨基酸代谢通路)。
- 代谢组学分析：代谢物鉴定与定量、差异代谢物筛选、代谢通路分析 (如脂肪酸代谢通路)。
- 微生物组学分析： α 多样性分析 (Simpson、Chao1、ACE 指数)、物种丰度分析 (如肠球菌属)、功能通路分析 (如氧化磷酸化通路)。
- 单细胞组学分析：细胞聚类与注释、细胞通讯分析 (如肿瘤微环境细胞互作)、伪时序分析 (如细胞分化轨迹)。
- 影像组学分析：肿瘤病灶分割、影像特征提取 (如纹理特征、形状特征)、影像特征与临床结局关联分析。
- 病理组学分析：病理切片预处理、细胞核检测与分割、病理特征提取与分类 (如肿瘤浸润淋巴细胞评估)。
- 多组学整合分析：组学数据间关联性分析、核心靶点筛选 (如 TMCO6、DDAH1 等潜在靶点验证)。

1.4.3.2.3 机器学习算法 平台可集成多种经典与前沿机器学习算法，支持不同分析场景需求：

算法类型	具体算法	适用场景
回归分析	线性回归、逻辑回归	疗效评分预测、风险因素量化分析
聚类分析	K 均值聚类、层次聚类	疾病亚型划分、细胞群体分类
分类算法	随机森林、朴素贝叶斯、决策树	疾病诊断分型、治疗反应分类
深度学习	神经网络	复杂影像特征提取、高维数据建模

1.4.3.3 临床应用场景

平台的核心分析功能已通过大量临床研究验证，可直接应用于以下场景：

- 1. 肿瘤早期筛查与诊断：结合影像组学与基因组学数据，提高早期肿瘤检出率。
- 2. 癌症精准分型与亚型预测：通过多组学数据聚类分析，明确疾病分子亚型。
- 3. 治疗方案选择与疗效预测：基于算法模型预测不同治疗方案的响应率，如抗 PD-1 免疫治疗、顺铂化疗等。
- 4. 预后评估与复发风险预测：整合临床信息与多组学数据，评估患者长期生存与复发风险。
- 5. 耐药风险预测与机制解析：分析耐药相关基因与通路，揭示耐药机制。

1.4.4 特色资源与研究成果

1.4.4.1 重点特色论文

项目组代表性论文包括：

- 1. «DNA of Neutrophil Extracellular Traps Binds TMCO6 to Impair CD8+ T-cell Immunity in Hepatocellular Carcinoma» - Cancer Res. 2024 (JCR Q1, IF=16.6)：揭示 NET-TMCO6 轴是克服肝癌免疫抑制的潜在靶点。
- 2. «DDAH1 Promotes Cisplatin Chemoresistance in Patients with Locally Advanced Nasopharyngeal Carcinoma via the EGFR-JAK2-STAT3 Pathway» - Adv Sci. 2025 (JCR Q1, IF=14.1)：证实 DDAH1 是 EGFR+ 癌症的预后标志物和潜在治疗靶点。
- 3. «Multimodality deep learning radiomics predicts pathological response after neoadjuvant chemoradiotherapy for esophageal squamous cell carcinoma» - Insights into imaging, 2024 (JCR Q1, IF=4.5)：多模态深度学习预测放化疗反应。
- 4. «Prenatal PFAS exposure, gut microbiota dysbiosis, and neurobehavioral development in childhood» - J Hazard Mater. 2024 (JCR Q1, IF=13.6)：阐明肠道菌群介导环境污染物神经毒性的机制。

1.4.4.2 优秀论文列表

优秀论文列表版块集中展示了项目组近年来的代表性研究成果，收录论文均发表于 JCR Q1 区 SCI 高水平期刊。研究内容涵盖两大核心方向：一是多模态数据在癌症诊断、治疗、预后评估及分子驱动机制领域的突破性探索；二是依托队列研究平台开展的微生物组与儿童神经发育关联性的前瞻性研究。

论文按研究方向分类：

1. 儿科肿瘤与血液疾病治疗

- Lu S, et al. EClinicalMedicine. 2024 (IF=10): PEG 化脂质体阿霉素联合化疗治疗儿童复发/难治性实体瘤。
- Que Y, et al. J Transl Med. 2024 (IF=7.5): 儿童淋巴瘤样癌的基因组与临床特征。

2. 儿童健康与发育研究

- Zhou Y, et al. J Hazard Mater. 2024 (IF=13.6): 产前 PFAS 暴露与儿童神经行为发育。
- Gui Y, et al. Int J Epidemiol. 2024 (IF=7.7): 上海亲子队列 (MCPC) 研究。

3. 肝癌与免疫治疗研究

- Song M, et al. Cancer Res. 2024 (IF=16.6): NET-TMCO6 轴与肝癌免疫抑制研究。

4. 鼻咽癌研究

- Yang JH, et al. Adv Sci. 2025 (IF=14.1): DDAH1 与鼻咽癌顺铂耐药机制。
- He JR, et al. NPJ Precis Oncol. 2025 (IF=8): 新发转移性鼻咽癌局部区域放疗省略策略。
- Liu LT, et al. Eur J Nucl Med Mol Imaging. 2025 (IF=7.6): 12 周 PET/CT 与液体活检在鼻咽癌监测中的应用。

5. 卵巢癌研究

- Gao Y, et al. J Ovarian Res. 2025 (IF=4.2): 卵巢高钙血症型小细胞癌的单细胞分析。
- Zheng K, et al. J Gynecol Oncol. 2024 (IF=3.9): 中国人群卵巢高钙血症型小细胞癌的临床特征 Meta 分析。
- Zheng K, et al. Biomedicines. 2024 (IF=3.9): mTOR 抑制剂依维莫司增强卵巢癌顺铂敏感性。

6. 其他肿瘤/疾病相关研究

- Yao Y, et al. Genes Dis. 2024 (IF=6.8): TMEM217 作为急性髓系白血病预后标志物。
- Xu J, et al. Int J Mol Sci. 2023 (IF=4.9): RAD51D 二次突变介导 HGSOC 对 PARP 抑制剂耐药。
- Liu Y, et al. Insights Imaging. 2024 (IF=4.5): 多模态深度学习影像组学预测食管鳞癌新辅助放化疗病理反应。
- Wang P, et al. J Clin Immunol. 2024 (IF=5.7): 脐带血移植治疗 IL-10 受体缺陷所致极早发炎症性肠病。

1.4.5 常见问题

1.4.5.1 数据相关问题

- Q: 平台支持哪些数据格式?
A: 支持 FASTQ、BAM、VCF、CSV、Excel 等标准格式。
- Q: 数据存储周期是多久?
A: 平台仅在分析期间临时存储数据, 分析完成后可下载本地保存, 服务器端数据将定期删除, 保障数据安全。

1.4.5.2 权限与使用限制

- Q: 平台是否需要注册登录? 是否收费?
A: 目前平台网站展示功能为公开访问, 无需注册登录; 所有分析功能仅对项目组内部研究人员开放。后续如有更新扩展, 将在平台网站【公告栏】另行通知。

1.4.5.3 成果引用规范

- Q: 使用平台数据或分析结果发表论文时，如何引用？

A: 平台引用格式：儿童重要肿瘤与血液性疾病多模态数据分析平台. <https://fudan-sph.github.io/ceh/>, 访问日期.

1.4.6 联系与支持

若在使用过程中遇到技术问题、功能建议或数据解析疑问，可通过以下方式获取支持：

- 科研合作咨询：建议联系平台相关研究团队（参考论文作者邮箱）。
- 技术支持：可通过平台官网预留的反馈入口提交问题，工作人员将定期回复。
- 资源获取：如需论文原文或补充分析资料，可在对应论文期刊官网下载，或联系论文通讯作者获取。

1.4.7 免责声明

本平台分析结果仅作为临床研究与科研参考，不可直接替代临床诊断结论，临床诊疗需由专业医师结合患者实际情况综合判断。

第 2 章 新用户初始化配置

本章节旨在指导 RStudio Server 新用户完成首次登录后的初始化配置操作，核心涵盖界面熟悉、环境设置、基础工具配置等关键步骤，助力用户快速上手并搭建契合个人使用习惯的工作环境。适用于所有首次使用 RStudio Server 的用户，无论是否具备 R 语言基础，均可依照手册步骤顺利完成初始化配置。

2.1 准备工作

- 已获取 RStudio Server 用户名及初始密码；
- 本地设备已连接互联网，且能正常访问 RStudio Server 服务器地址。

2.2 首次登录操作

2.2.1 登录步骤

1. 打开本地浏览器（推荐 Chrome、Firefox、Edge 等主流浏览器）；
2. 通过网站首页**立即使用**按钮跳转至 RStudio Server 登录页面；进入登录页面后，输入管理员分配的用户名和初始密码；
3. 点击【Sign In】按钮，完成首次登录。

2.2.2 首次登录密码修改

为保障账号安全，首次登录后需立即修改初始密码，步骤如下：

1. 登录成功后，点击页面顶部菜单栏的【Tools】（工具）；
2. 在下拉菜单中选择【Terminal】（终端），打开终端窗口；
3. 在终端中输入命令：`passwd`，按下回车键；
4. 系统会提示 `Current password`（当前密码），输入初始密码后回车；
5. 提示 `New password`（新密码），输入自定义的新密码（建议包含大小写字母、数字及特殊符号，长度 ≥ 8 位），回车；
6. 提示 `Retype new password`（再次输入新密码），重复输入新密码后回车；
7. 若提示 `passwd: password updated successfully`，说明密码修改成功；若修改失败，检查密码格式是否符合要求，重新操作即可。

2.3 熟悉 RStudio Server 界面

登录成功后，RStudio Server 界面默认分为 4 个核心区域，各区域功能如下，帮助用户快速定位操作位置：

2.3.1 区域 1：源代码编辑器

源代码编辑器位于界面左上方，用于编写 R 脚本（.R 文件）、Markdown 文档（.Rmd 文件）等，支持语法高亮、代码补全、行号显示等功能，是核心的代码编写区域。

2.3.2 区域 2：控制台

控制台位于界面左下方，用于执行 R 代码（直接输入代码回车即可运行），显示代码运行结果。同时支持终端（Terminal）切换，可执行 Linux 系统命令（如前文的密码修改操作）。

2.3.3 区域 3：环境/历史/连接/指导

环境/历史/连接区域位于界面右上方，包含以下功能：

- **Environment**（环境）：显示当前工作空间中的变量、数据框、函数等对象，支持查看、删除对象；
- **History**（历史）：记录之前在控制台执行过的所有代码，支持复制、重新运行历史代码；
- **Connections**（连接）：用于连接数据库、云存储等外部数据源。
- **Tutorials**（指导）：提供 R 语言基础教程、数据分析案例等学习资源，新手可通过此区域学习 R 语言知识。

2.3.4 区域 4：文件/plots/包/帮助/查看器/演示

文件/plots/包/帮助区域位于界面右下方，包含以下功能：

- **Files**（文件）：显示当前工作目录下的所有文件和文件夹，支持新建、上传、下载、删除、重命名文件/文件夹；
- **Plots**（绘图）：显示 R 代码生成的图形（如直方图、折线图），支持保存、导出图形；
- **Packages**（包）：显示已安装的 R 包，支持安装新包、更新包、加载/卸载包；
- **Help**（帮助）：提供 R 语言基础语法、函数、包的详细帮助文档，新手可通过此区域查询知识点。
- **Viewer**（查看器）：用于显示 HTML 文档、Shiny 应用等交互式内容。
- **Presentation**（演示）：用于预览、展示与控制幻灯片、报告等演示文稿内容。

2.4 初始化配置

2.4.1 工作目录设置

工作目录是 RStudio Server 默认读取和保存文件的路径，建议设置为自定义的专用文件夹（避免使用默认目录导致文件混乱），步骤如下：

1. 在右侧 **【Files】** 区域，点击 **【New Folder】**（新建文件夹），输入文件夹名称（如 **R_Workspace**），点击 **【OK】**；
2. 选中新建的文件夹，点击 **【More】**（更多），在下拉菜单中选择 **【Set As Working Directory】**（设为工作目录）；

3. 验证: 在左侧【Console】(控制台)输入代码 `getwd()`, 回车后若显示的路径为新建文件夹的路径(如 `/home/用户名/R_Workspace`), 说明设置成功。

2.4.2 界面个性化配置

根据个人使用习惯调整界面主题、字体、字号等, 步骤如下:

1. 点击顶部菜单栏【Tools】(工具) → 【Global Options】(全局选项);
2. 在弹出的窗口中, 选择左侧【Appearance】(外观): Theme (主题): 推荐新手选择浅色主题 (如 Modern、Classic) 或深色主题 (如 Dracula、Solarized Dark), 根据视觉习惯选择;
3. Font (字体): 选择支持中文的字体 (如 Microsoft YaHei、SimHei、Consolas);
4. Font Size (字号): 调整为清晰可见的大小 (如 12-14 号);
5. Zoom (缩放): 根据屏幕尺寸调整界面缩放比例 (默认 100%)。
6. 设置完成后, 点击【Apply】(应用) → 【OK】(确认), 界面会立即更新为设置后的样式。

2.4.3 R 包安装与配置

R 语言的核心功能通过“包”(Packages)扩展, 首次使用需安装常用包 (如数据处理包 `dplyr`、绘图包 `ggplot2` 等), 步骤如下:

2.4.3.1 安装单个包

1. 点击右侧【Packages】区域的【Install】(安装)按钮;
2. 在弹出的窗口中, Packages 输入框内输入要安装的包名 (如 `dplyr`);
3. 勾选 `Install dependencies` (安装依赖, 必选, 避免包安装后无法使用);
4. 点击【Install】, 等待安装完成 (底部会显示安装进度, 提示 `DONE` 即安装成功)。

2.4.3.2 批量安装常用包

若需安装多个常用包, 可通过控制台代码批量安装, 步骤如下:

1. 在左侧【Console】输入以下代码 (包含数据处理、绘图、读写文件等常用包): `install.packages(c("dplyr", "ggplot2", "readr", "tidyr", "stringr", "knitr"))`
2. 回车后, 系统会提示选择 CRAN 镜像 (包下载源), 建议选择国内镜像 (如 China (Beijing 4) [https] - Tsinghua University, 清华镜像, 下载速度更快);
3. 等待安装完成, 控制台显示 `DONE` 即全部安装成功。

2.4.3.3 加载已安装的包

包安装完成后, 需【加载】才能使用, 步骤如下:

- 图形化方式：在 **【Packages】** 区域找到已安装的包，勾选包名前的复选框即可加载；
- 代码方式：在控制台输入 `library(包名)`（如 `library(dplyr)`），回车后无报错即加载成功。

2.4.4 新建 R 脚本文件

完成初始化配置后，可新建 R 脚本文件开始编写代码，步骤如下：

1. 点击顶部菜单栏 **【File】**（文件）→ **【New File】**（新建文件）→ **【R Script】**（R 脚本）；
2. 左上源代码编辑器区域会新建一个名为 `Untitled1.R` 的空白脚本；
3. 可在脚本中输入测试代码（如 `print("Hello RStudio Server!")`）；
4. 点击编辑器右上角的 **【Run】**（运行）按钮，或选中代码后按快捷键 **【Ctrl+Enter】**，控制台会显示运行结果（`[1] "Hello RStudio Server!"`），说明环境配置正常。

2.5 常用操作指引

2.5.1 文件保存与打开

- 保存：新建脚本/文档后，点击编辑器顶部的 **【Save】**（保存）按钮，或按快捷键 **【Ctrl+S】**，选择保存路径（建议保存在之前设置的工作目录 `R_Workspace` 下），输入文件名后点击 **【Save】**；
- 打开：点击顶部菜单栏 **【File】** → **【Open File...】**，在弹出的窗口中选择要打开的文件（.R、.Rmd 等），点击 **【Open】** 即可。

2.5.2 数据导入

新手常用数据格式为 CSV、Excel，导入步骤如下（以 CSV 为例）：

1. 将本地 CSV 文件上传至工作目录：右侧 **【Files】** 区域点击 **【Upload】**（上传），选择本地文件，点击 **【OK】**；
2. 使用 `readr` 包导入：在脚本中输入代码 `library(readr)`，回车加载包后，输入代码 `data <- read_csv("文件名.csv")`（替换为实际文件名），运行代码；
3. 使用 `import` 包导入：在脚本中输入代码 `library(import)`，回车加载包后，输入代码 `data <- import("文件名.csv")`，运行代码；
4. 验证：在 **【Environment】** 区域会显示 `data` 对象，点击对象名称可查看数据内容，说明导入成功。

2.5.3 退出登录

完成操作后，点击页面右上角的用户名，在下拉菜单中选择 **【Sign Out】**（退出登录），即可安全退出 RStudio Server。

2.6 常见问题解决

2.6.1 登录失败

- 问题现象：输入用户名/密码后提示 `Invalid credentials`（无效凭据）；
- 解决方法：检查用户名、密码是否输入正确（注意大小写、空格）；
- 若忘记密码，联系管理员重置密码；
- 检查登录地址是否正确（确认服务器 IP 和端口 8787 是否正确）。

2.6.2 包安装失败

- 问题现象：安装包时提示 “installation of package ‘xxx’ had non-zero exit status”；
- 解决方法：确保勾选 `texttt`（安装依赖）；
- 更换 CRAN 镜像（选择国内镜像，如清华、中科大镜像）；
- 若为系统依赖缺失，联系管理员安装对应系统库（如 Ubuntu 系统可能需要安装 `libcurl4-openssl-dev` 等）。

2.6.3 界面中文乱码

- 问题现象：导入中文数据或界面显示中文时出现乱码；
- 解决方法：设置系统编码：在控制台输入 `Sys.setlocale(category = "LC_ALL", locale = "zh_CN.UTF-8")`，回车后生效；
- 导入数据时指定编码：使用 `read_csv(" 文件名.csv", locale = locale(encoding = "GBK"))`（若文件编码为 GBK）。

2.7 附录

2.7.1 常用快捷键

- 运行代码：Ctrl+Enter（选中代码）、Ctrl+Shift+Enter（运行整个脚本）；
- 保存文件：Ctrl+S；
- 新建脚本：Ctrl+Shift+N；
- 查找替换：Ctrl+F；
- 清空控制台：Ctrl+L。

2.7.2 国内 CRAN 镜像推荐

- 清华镜像：<https://mirrors.tuna.tsinghua.edu.cn/CRAN/>；
- 中科大镜像：<https://mirrors.ustc.edu.cn/CRAN/>；
- 阿里云镜像：<https://mirrors.aliyun.com/CRAN/>。

2.7.3 求助方式

若遇到手册未覆盖的问题，可通过以下方式求助：

- 联系服务器管理员（解决登录、系统依赖、服务器配置相关问题）；
- 参考 RStudio 官方文档：<https://docs.rstudio.com/>；
- R 语言社区：Stack Overflow (<https://stackoverflow.com/>)、RStudio 社区 (<https://community.rstudio.com/>)。

第 3 章 RStudio Server 进阶使用

3.1 系统配置优化

通过修改配置文件可自定义 RStudio Server 服务参数，需使用 root 权限操作（普通用户无修改权限）：

3.1.1 核心配置文件

1. 服务配置文件：/etc/rstudio/rserver.conf（默认不存在，需手动创建）；
2. 常用配置项：
 - `www-port=8080`：修改默认端口（默认 8787）；
 - `rsession-which-r=/usr/local/bin/R`：指定 R 程序路径（解决 R 版本冲突问题）。
3. 会话配置文件：/etc/rstudio/rsession.conf（默认不存在，需手动创建）；
4. 常用配置项：
 - `session-timeout-minutes=60`：设置会话超时时间（默认 30 分钟，无操作自动断开）；
 - `r-cran-repos=http://ftp.ctex.org/mirrors/CRAN/`：设置 CRAN 镜像源（加速包安装）。

3.1.2 配置生效步骤

1. 使用 root 权限编辑配置文件（例：编辑 rserver.conf）：`sudo vim /etc/rstudio/rserver.conf`；
2. 添加配置项后，保存并退出；
3. 重启 RStudio Server 服务使配置生效：`sudo rstudio-server restart`。

3.1.3 服务管理常用命令

1. 启动服务：`sudo rstudio-server start`；
2. 停止服务：`sudo rstudio-server stop`；
3. 重启服务：`sudo rstudio-server restart`；
4. 查看运行状态：`sudo rstudio-server status`；
5. 查看活跃会话：`sudo rstudio-server active-sessions`；
6. 强制终止指定会话（替换 [PID] 为会话进程 ID）：`sudo rstudio-server force-suspend-session [PID]`；
7. 临时下线服务（禁止新登录，保留现有会话）：`sudo rstudio-server offline`；
8. 上线服务：`sudo rstudio-server online`。

3.2 多人协作与权限管理

RStudio Server 基于 Linux 系统用户实现多用户隔离与协作，管理员可通过 Linux 命令管理用户，用户间可通过共享目录或版本控制实现文件共享。

3.2.1 新增协作用户

管理员可通过以下步骤创建新用户账号：

1. 创建用户组（例：创建 `hadoop` 组）：`sudo groupadd hadoop`;
2. 创建用户并加入用户组（例：创建 `hadoop` 用户）：`sudo useradd hadoop -g hadoop`;
3. 设置用户密码（密码至少 8 位）：`sudo passwd hadoop`;
4. 赋予 `sudo` 权限（可选，允许用户执行管理员命令）：`sudo adduser hadoop sudo`;
5. 创建用户主目录并设置权限：`sudo mkdir /home/hadoop`
`sudo chown -R hadoop:hadoop /home/hadoop`;
6. 测试用户：新用户可使用 `hadoop` 账号登录 RStudio Server，验证是否正常访问。

3.2.2 共享目录设置

为实现多用户文件共享，可创建共享目录并设置权限：

1. 创建共享目录：`sudo mkdir /home/shared_data`;
2. 设置目录权限（允许用户组读写）：`sudo chmod -R 775 /home/shared_data`;
3. 将用户添加到共享组（例：添加 `hadoop` 用户到 `hadoop` 组）：`sudo usermod -a -G hadoop 用户名`;
4. 用户可在 RStudio Server 右下角「Files」区域访问 `/home/shared_data`，实现文件共享。

3.3 Git 版本控制集成

通过 Git 可实现代码版本管理、多人协作开发，RStudio Server 内置 Git 集成功能，需先完成基础配置。

3.3.1 前期准备

1. 服务器安装 Git：`sudo apt-get install git` (Ubuntu) 或 `sudo yum install git` (CentOS);
2. 注册 GitHub/GitLab 账号，创建远程仓库（例：github.com 上创建 `rstudio-demo` 仓库）。

3.3.2 RStudio 配置 Git

1. 安装 `usethis` 包（辅助配置 Git）：`install.packages("usethis")` `library(usethis)`;
2. 设置 Git 用户名和邮箱（需与 GitHub/GitLab 账号一致）：`use_git_config(user.name = " 你的用户名", user.email = " 你的邮箱")`;

3. 生成 SSH 密钥（实现无密码访问远程仓库）：`ssh-keygen -t rsa`（按回车默认设置即可）；
4. 查看并复制公钥：`cat /home/用户名/.ssh/id_rsa.pub`；
5. 将公钥添加到 GitHub/GitLab：登录账号 → 进入「Settings」→「SSH and GPG keys」→「New SSH key」→ 粘贴公钥并保存。

3.3.3 项目绑定远程仓库

1. 在 RStudio 中打开项目（或新建项目）；
2. 绑定远程仓库：
 - 点击顶部菜单栏「Tools」→「Version Control」→「Project Setup」；
 - 选择「Version control system」为「Git」，点击「OK」；
 - 控制台输入命令绑定远程仓库（替换为你的仓库地址）：`git remote add origin git@github.com: 用户名/rstudio-demo.git`。

3.3.4 常用 Git 操作

1. 提交代码：
 - 点击右下角「Git」标签 → 勾选需要提交的文件；
 - 点击「Commit」→ 输入提交说明（Commit message）→ 点击「Commit」。
2. 推送代码到远程仓库：
 - 提交完成后，点击「Push」按钮；
 - 首次推送可能需要输入 GitHub/GitLab 账号密码（或验证 SSH 密钥）。
1. 拉取远程代码（同步他人修改）：
 - 点击「Git」标签 → 点击「Pull」按钮。

3.4 高效 workflow 技巧

3.4.1 批量操作与自动化

1. 批量安装包：使用 `install.packages(c("包 1", "包 2", "包 3"))` 一次性安装多个包；
2. 自动化脚本：编写 .R 脚本包含完整分析流程，通过「Source」一键运行；
3. 任务调度：结合 Linux cron 定时运行 R 脚本（例：每日凌晨自动运行数据处理脚本）。

3.4.2 R Markdown 报告生成

1. 新建 R Markdown 文档：「File」→「New File」→「R Markdown」；
2. 编辑文档：支持文本、代码块混合编写，代码块可直接运行；

3. 生成报告：点击「Knit」按钮，选择输出格式（HTML、PDF、Word 等），自动生成包含代码、结果、图形的完整报告。

3.4.3 结果保存与导出

1. 保存分析结果：使用 `save(df, file = " 结果.RData")` 保存数据框，`load(" 结果.RData")` 加载；
2. 导出表格：使用 `writexl::write_xlsx(df, " 数据.xlsx")` 导出 Excel 文件；
3. 导出图形：在「Plots」区域点击「Export」→ 选择保存格式（PNG、PDF 等），设置分辨率和尺寸。

3.4.4 常见问题排查

3.4.5 登录失败

1. 问题现象：输入账号密码后无法登录，提示「Invalid credentials」；解决方案：检查用户名/密码是否正确（区分大小写）；联系管理员确认用户是否存在（`cat /etc/passwd | grep 用户名`）；重置密码（`sudo passwd 用户名`）。
2. 问题现象：无法访问登录页面；解决方案：检查服务器是否启动 RStudio Server 服务（`sudo rstudio-server status`）；检查端口是否开放（`sudo netstat -tuln | grep 8787`）；检查防火墙是否放行 8787 端口（`sudo ufw allow 8787`）。

3.4.6 代码运行报错

1. 问题现象：提示「libR.so: cannot open shared object file: No such file or directory」；解决方案：在 `/etc/rstudio/rserver.conf` 中添加 `rsession-which-r=/usr/local/bin/R`（指定正确的 R 路径），重启服务。
2. 问题现象：包安装失败；解决方案：检查网络连接，更换 CRAN 镜像源；安装依赖库（Ubuntu：`sudo apt-get install libssl-dev libcurl4-openssl-dev`；CentOS：`sudo yum install openssl-devel curl-devel`）。

3.4.7 会话崩溃/卡顿

1. 问题现象：界面无响应、代码运行卡住；解决方案：关闭当前浏览器标签，重新登录 RStudio Server；联系管理员终止异常会话（`sudo rstudio-server force-suspend-session [PID]`）；检查代码是否存在死循环、大数据量处理导致内存不足（可通过 `rm(对象名)` 释放内存）。

3.4.8 常用快捷键汇总

1. 运行代码：Ctrl+Enter（单行）、Ctrl+Shift+Enter（全部）；
2. 保存文件：Ctrl+S；
3. 注释代码：Ctrl+/（单行注释/取消注释）；
4. 查找替换：Ctrl+F；

5. 新建项目: `Ctrl+Shift+N`;
6. 清除控制台: `Ctrl+L`。

3.4.9 官方资源链接

1. RStudio Server 官方文档: <http://www.rstudio.com/ide/download/server>;
2. CRAN 镜像源列表: <https://cran.r-project.org/mirrors.html>;
3. RStudio 快捷键官方指南: <https://rstudio.github.io/rstudio-shortcuts-guide/>。

第 4 章 通过 RStudio Server 调用 Python

本章节旨在帮助用户完成 conda 环境的安装配置，并实现在 RStudio Server 中调用 Python 进行开发。全程包含环境准备、conda 安装、环境配置、Python 使用验证及常见问题排查，步骤清晰且适配主流服务器系统。

4.1 前期准备

服务器管理员已为用户完成如下前置配置工作：核心包括在服务器的系统级路径下安装 Miniconda，并将 Miniconda 的执行路径添加到系统级环境变量配置文件中，以此保障所有用户均可全局调用 conda 命令。

4.1.1 服务器环境要求

- 操作系统：Linux 发行版（推荐 Ubuntu 20.04+/CentOS 7+，64 位）；
- 已安装 RStudio Server（确认可通过浏览器访问，地址格式：`http://服务器 IP:8787`）；
- 服务器具备网络连接（用于下载 Miniconda 及 Python 包）；
- 拥有服务器管理员权限（sudo 权限，用于安装软件及配置环境变量）。

4.1.2 工具准备

- SSH 客户端（如 Xshell、Putty、Terminal）：用于远程连接服务器执行命令；
- 浏览器（如 Chrome、Firefox）：用于访问 RStudio Server 界面。

4.1.3 Conda

Conda 是开源的包管理器和环境管理器，Miniconda 是精简版的 conda 发行版（仅包含 conda、Python 及基础依赖），适合服务器环境部署。以下步骤通过 SSH 客户端远程连接服务器执行。

4.1.3.1 下载 Miniconda 安装包

1. 连接服务器：打开 SSH 客户端，输入服务器 IP、端口（默认 22），登录用户名及密码；
2. 进入临时下载目录（如/tmp）：`cd /tmp`；
3. 下载 Miniconda 安装包（选择 Python 3.x 版本，适配 Linux 64 位）：

推荐使用国内镜像源（加快下载速度），命令如下：

```
# Ubuntu/CentOS 通用(Python 3.9 版本,可替换版本号) wget https://mirrors.tuna.tsinghua.edu.cn/anaconda
```

若未安装 wget，先执行安装：

```
Ubuntu: `sudo apt update && sudo apt install -y wget`
```

```
CentOS: `sudo yum install -y wget`
```

4.1.3.2 安装 Miniconda

1. 赋予安装包执行权限: `chmod +x Miniconda3-py39_23.10.0-1-Linux-x86_64.sh`;
2. 执行安装脚本: `./Miniconda3-py39_23.10.0-1-Linux-x86_64.sh`;
3. 安装过程交互配置:

按 **【Enter】** 键查看许可协议，持续按Enter直到协议末尾；

输入 ``yes`` 接受许可协议；

确认安装路径 `\textbf{(系统级路径: /opt/miniconda3)}`，按 **【Enter】** 确认；

询问是否初始化Miniconda3，输入 `\texttt{yes}`（关键步骤，自动配置环境变量）；

`\begin{verbatim}`

安装完成后，提示 “Thank you for installing Miniconda3!”，表示安装成功。

4.1.3.3 验证 Miniconda 安装

1. 刷新环境变量（使 conda 命令立即生效）: `source ~/.bashrc`（若使用 zsh 终端，执行 `source ~/.zshrc`）;
2. 执行 conda 命令验证: `conda --version`；若输出类似 `conda 23.10.0` 的版本信息，说明安装成功；若提示 `conda: command not found`，需重新检查 **【安装 Miniconda】** 步骤的 是否选择 `yes`，或手动配置环境变量（见下方补充说明）。

补充说明：手动配置环境变量

若 conda 命令无法识别，执行以下命令（替换用户名为实际登录用户名）：`echo 'export PATH="/opt/miniconda3/bin:$PATH'`
`>> ~/.bashrc` 再执行 `source ~/.bashrc` 刷新环境变量即可。

4.2 在 RStudio Server 中配置 conda 环境

本章节分为创建 conda 虚拟环境和 RStudio Server 关联 conda 环境两步，前者通过 SSH 完成，后者在 RStudio Server 浏览器界面操作。

4.2.1 创建 conda 虚拟环境

推荐创建独立的虚拟环境（如命名为 `rstudio-python`），避免与系统 Python 或其他环境冲突。

1. 通过浏览器访问平台网页并登录服务器，在 RStudio Server 界面的终端（Terminal）中执行以下命令创建环境（如指定 Python 3.9 版本，可根据需求修改）: `conda create -n rstudio-python python=3.9`;
2. 交互确认：输入 `y`，等待环境创建完成（自动下载 Python 及基础依赖）;
3. 激活虚拟环境: `conda activate rstudio-python`;

激活成功后，终端提示符前会显示 “(rstudio-python)”，表示当前处于该环境中。

4.2.2 RStudio Server 关联 conda 虚拟环境

1. 过浏览器访问平台网页并登录 RStudio Server;
2. 查看并配置 Python 解释器:

点击RStudio界面顶部菜单栏 **【Tools】** → **【Global Options】** ;

在弹出的窗口左侧选择 **【Python】** ;

勾选 **【Use a specific Python interpreter】** , 点击右侧 **【Browse】** ;

在文件浏览器中, 导航到conda虚拟环境的Python解释器路径, 默认路径为:

`/opt/miniconda3/envs/rstudio-python/bin/python`

选中“python”文件, 点击 **【Open】** ;

点击 **【Apply】** → **【OK】** , 完成Python解释器配置。

4.2.3 验证 conda 环境关联成功

1. 在 RStudio 界面中, 打开 **【Console】** 控制台 (右下角面板);
2. 输入以下 R 代码 (通过 `reticulate` 包调用 Python, RStudio 默认集成该包) : `library(reticulate)`
`py_config()`;
3. 查看输出结果:若 `python:` 对应的路径为上述 conda 虚拟环境的 Python 路径 (如/home/用户名/miniconda3/envs/rstudio-python/bin/python), 且 `version:` 为 3.9.x, 说明关联成功。

4.3 在 RStudio Server 中使用 Python

配置完成后, 可在 RStudio 中通过两种方式使用 Python:

1. 直接在 Console 中执行 Python 代码;
2. 创建 Python 脚本文件 (.py) 编写并运行。

4.3.1 直接在 Console 中执行 Python 代码

1. 在 RStudio 的 Console 中, 输入 `repl_python()`, 按 Enter 键, 进入 Python 交互环境 (提示符变为 `>>>`);
2. 输入 Python 代码测试:`print("Hello, RStudio + Conda + Python!")` # 输出字符串 `import sys print(sys.version)`
查看 Python 版本;
3. 退出 Python 交互环境: 输入 `exit`, 按 Enter 键, 返回 R 控制台。

4.3.2 创建并运行 Python 脚本

1. 创建 Python 脚本：点击 RStudio 界面左上角 **【File】** → **【New File】** → **【Python Script】**，新建一个空白脚本文件；
2. 编写 Python 代码（示例：读取数据并简单处理）：‘# 导入 numpy 包（需先安装）import numpy as np‘

4.4 生成随机数组

```
data = np.random.randn(5, 3) print(" 随机数组： ") print(data)
```

4.5 计算数组均值

```
1 # 计算数组均值 (Python代码)
2 import numpy as np
3 print("\n数组均值： ")
4 print(np.mean(data))
```

3. 保存脚本：点击脚本编辑器顶部的 **【Save】**，命名为 `test_python.py`，选择保存路径（如/home/用户名/）；
4. 运行脚本：

方式1：点击脚本编辑器顶部的 **【Run】** 按钮，逐行运行代码；

方式2：选中需要运行的代码，按 **Ctrl+Enter** 键运行；

运行结果会在 **Console** 面板中显示。

4.5.1 在 conda 环境中安装 Python 包

若需要安装第三方 Python 包（如 numpy、pandas、matplotlib 等），可通过两种方式安装：

1. 方式 1：通过 SSH 终端安装（推荐，速度稳定）：

登录SSH，激活conda虚拟环境：`conda activate rstudio-python`；

使用conda安装包：`conda install numpy pandas matplotlib`（多个包用空格分隔）；

或使用pip安装（若conda源中无该包）：`pip install requests`；

2. 方式 2：在 RStudio 中直接安装：在 **Console** 中进入 Python 交互环境（`repl_python()`），执行 pip 安装命令：`pip install scikit-learn`

4.6 常见问题与排查

4.6.1 conda 命令无法识别

- 原因：环境变量未配置或未刷新；
- 解决方案：执行 `source ~/.bashrc` 刷新环境变量，若仍无效，参考【验证 Miniconda 安装】“补充说明”中的手动配置环境变量。

4.6.2 RStudio 无法找到 conda 虚拟环境的 Python 解释器

- 原因：路径错误或虚拟环境未激活；
- 解决方案： 确认虚拟环境路径正确（默认路径：`/home/用户名/miniconda3/envs/环境名/bin/python`）；

通过SSH激活虚拟环境后，执行 `which python`，复制输出的路径，在RStudio中手动粘贴该路径配置。

4.6.3 安装 Python 包时速度慢或失败

- 原因：默认 conda/pip 源为国外源，网络不稳定；
- 解决方案：配置国内镜像源：

配置conda国内源（清华镜像）：

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/ conda
config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/ conda config
--set show_channel_urls yes
```

配置pip国内源（临时使用）：

```
pip install 包名 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4.6.4 RStudio 中调用 Python 提示 reticulate 包未找到

- 原因：RStudio 未安装 reticulate 包；
- 解决方案：在R控制台执行 `install.packages("reticulate")`，安装完成后再执行 `library(reticulate)` 加载包。

第 5 章 在 RStudio Server 中使用 Terminal

5.1 概述

RStudio Server 的 Terminal 功能为用户提供了在 IDE 内部直接访问服务器系统 Shell 的能力，支持 xterm 仿真、全屏终端应用（如 vim、emacs、tmux）以及常规命令行操作，具备行编辑和命令历史记录功能。其核心应用场景包括：高级版本控制操作、长时间运行作业的执行、远程登录管理、RStudio Server 系统运维以及全屏终端程序运行等。

与 R Console 不同，Terminal 直接对接服务器系统环境，可执行系统级命令（如软件安装、文件权限配置），且操作独立于 R 会话，适合处理需脱离 R 环境的任务。

5.2 Terminal 的开启与关闭

5.2.1 打开 Terminal

RStudio Server 的 Terminal 标签默认位于 Console 标签旁，常见打开方式有 3 种：

- 界面点击：直接点击 Console 面板上方的「Terminal」标签，若此前无活跃会话，将自动创建新会话；
- 菜单操作：通过顶部菜单栏「Tools」→「Terminal」→「Move Focus to Terminal」打开；
- 快捷键：使用「Shift+Alt+M」快速切换焦点至 Terminal（Mac 系统为「Shift+Option+M」）。

若 Terminal 标签未显示，通过「Tools」→「Terminal」→「Move Focus to Terminal」可强制显示面板。

5.2.2 新建与切换多终端会话

RStudio Server 支持多个独立 Terminal 会话并行，会话间互不干扰，即各自拥有独立的 Shell 进程和缓冲区。

- 新建会话：点击 Terminal 面板工具栏的「New Terminal」按钮，或使用快捷键「Shift+Alt+R」，也可通过「Tools」→「Terminal」→「New Terminal」创建；
- 切换会话：通过 Terminal 面板顶部的下拉菜单操作-点击菜单后可看到所有活跃会话列表，点击会话名称或使用菜单旁的左右箭头即可切换；也可使用快捷键「Ctrl+Alt+F11」（上一个会话）和「Ctrl+Alt+F12」（下一个会话）快速切换；
- 重命名会话：通过「Terminal」下拉菜单的「Rename Terminal」选项为会话命名（如数据处理、软件安装），便于区分不同用途。

5.2.3 关闭 Terminal 会话

关闭会话的 3 种常用方式：

- 命令关闭：在 Terminal 中输入「exit」并回车，当前会话将终止；
- 界面操作：点击 Terminal 面板工具栏右侧的「×」按钮，或通过「Tools」→「Terminal」→「Close Terminal」关闭当前会话；

- 强制中断：若 Terminal 处于繁忙状态（如执行长时间命令），「×」将变为「Stop」按钮，点击可中断当前命令，但不会关闭会话。

5.3 Terminal 基础功能使用

5.3.1 基础命令操作

Terminal 支持服务器系统的所有 Shell 命令（如 Linux 的 bash 命令），常用基础操作示例：

- 目录与文件操作：「ls」（查看当前目录文件）、「cd 目录路径」（切换目录）、「mkdir 目录名」（创建目录）、「rm -rf 文件名」（删除文件/目录）；
- 软件安装：「sudo apt-get install 软件名」（Debian/Ubuntu 系统）、「sudo yum install 软件名」（CentOS 系统）；
- R 相关操作：「R -version」（查看 R 版本）、「Rscript 脚本路径.R」（执行 R 脚本）；
- 清空缓冲区：使用快捷键「Ctrl+Shift+L」（或 Terminal 获焦时按「Ctrl+L」）清空终端显示内容。

5.3.2 与 RStudio 编辑器的交互

Terminal 可与 RStudio 编辑器无缝协作，提升代码执行效率：

- 代码发送：在编辑器中选中代码（无选中则默认当前行），使用快捷键「Ctrl+Alt+Enter」发送至活跃 Terminal 执行；若编辑的是 Shell 脚本（.sh 文件），直接使用「Ctrl+Enter」即可单步执行；
- 内容导出：通过「Tools」→「Terminal」→「Copy Terminal to Editor」，可将 Terminal 的输出缓冲区内容导出至新的编辑器文档，便于保存和后续编辑。

5.3.3 外观与配置调整

Terminal 的视觉样式可通过 RStudio 全局设置自定义：

1. 点击顶部菜单栏「Tools」→「Global Options」；
2. 在弹出的窗口中选择「Appearance」选项卡；
3. 可调整字体、字体大小、颜色主题等，设置完成后点击「Apply」生效。

5.4 进阶应用

5.4.1 运行长时间作业

由于 RStudio Server 会在用户长时间无操作时断开会话，导致作业中断，可通过「screen」工具在 Terminal 中后台运行长作业：

1. 创建 screen 会话：在 Terminal 中输入「screen -S 会话名」（如「screen -S long_job」），回车后进入新的会话环境；
2. 启动作业：在 screen 会话中输入 R 命令或执行脚本（如「Rscript long_task.R」）；
3. 脱离会话：按「Ctrl+A」后再按「D」，即可脱离 screen 会话，此时作业仍在后台运行，可关闭 RStudio 窗口；

4. 重新连接：再次登录 RStudio Server 并打开 Terminal，输入「screen -r 会话名」（如「screen -r long_job」），即可回到之前的会话查看作业进度。

5.4.2 RStudio Server 系统管理

若用户拥有服务器管理员权限，可通过 Terminal 执行 RStudio Server 的管理命令（需输入 sudo 密码）：

- 启动服务：「sudo rstudio-server start」；
- 停止服务：「sudo rstudio-server stop」；
- 重启服务：「sudo rstudio-server restart」（修改配置后需重启生效）；
- 查看状态：「sudo rstudio-server status」；
- 用户管理：「sudo rstudio-server list-users」（查看授权用户）、「sudo rstudio-server add-user 用户名」（添加授权用户）。

5.4.3 远程登录与文件传输

通过 Terminal 可直接进行远程登录和文件传输：

- SSH 远程登录：输入「ssh 用户名 @ 远程服务器 IP -p 端口号」（如「ssh user@192.168.1.100 -p 22」），按提示输入密码即可登录远程服务器；
- 文件传输：使用「scp」命令传输文件，如「scp 本地文件路径用户名 @ 远程服务器 IP: 远程目录路径」（上传文件）、「scp 用户名 @ 远程服务器 IP: 远程文件路径本地目录路径」（下载文件）。

5.5 常见问题与解决方案

5.5.1 终端显示异常

问题现象：打开 Terminal 后，光标闪烁正常，但输入字符、执行命令后无内容显示，或提示符为空白字符。

解决方案：该问题多为浏览器主题与 RStudio Server 终端组件不兼容导致，优先排查浏览器配置：

1. 若使用 Firefox 浏览器，禁用自定义主题（如 **Maker Theme- Stainless steel theme**），切换回默认主题；
2. 尝试更换浏览器（如 Chrome、Edge）重新登录；
3. 检查浏览器是否为最新版本，更新后重试。

5.5.2 长时间作业中断

问题现象：长时间运行的作业在 RStudio Server 会话断开后终止。

解决方案：使用「screen」或「tmux」工具创建后台会话（具体操作见 4.1 节），确保作业脱离当前 RStudio 会话独立运行。

5.5.3 权限不足

解决方案：

- 执行系统级命令（如软件安装、修改系统配置）时，在命令前添加「`sudo`」（需知道管理员密码）；
- 若为文件操作权限不足，使用「`chmod`」命令修改文件权限（如「`chmod 755 文件名`」）。

5.5.4 无法新建 Terminal 会话

解决方案：

1. 检查服务器资源是否充足（如内存、进程数），释放资源后重试；
2. 通过「Tools」→「Terminal」→「Terminal Diagnostics」生成诊断报告，联系服务器管理员排查问题。

5.6 快捷键汇总

功能	快捷键	备注
切换焦点至 Terminal	Shift+Alt+M（Mac：Shift+Option+M）	无
新建 Terminal 会话	Shift+Alt+R	无
切换至上一个会话	Ctrl+Alt+F11	无
切换至下一个会话	Ctrl+Alt+F12	无
清空终端缓冲区	Ctrl+Shift+L（获焦时可按 Ctrl+L）	无
编辑器代码发送至 Terminal	Ctrl+Alt+Enter（.sh 文件：Ctrl+Enter）	无

RStudio Server 的 Terminal 功能是连接 IDE 与服务器系统的重要桥梁，支持常规命令操作、多会话管理、长作业运行及服务器运维等核心场景。通过本章节的操作指引，可快速掌握其基础使用与进阶技巧；遇到问题时，优先排查浏览器配置、服务器资源及权限问题，或使用诊断工具辅助定位。合理运用 Terminal 的快捷键与交互功能，能显著提升数据处理与服务器管理的效率。

第 6 章 多模态医学数据预处理与下游分析核心工具汇总

随着精准医学的发展，多模态医学数据（如医学影像、多组学、临床文本、微生物组数据等）的整合分析成为挖掘临床价值与生物机制的关键。R 语言凭借丰富的开源包生态，为多模态医学数据的预处理、整合分析及可视化提供了高效工具。本文按数据模态分类，汇总各类数据处理的核心 R 包/软件，详细说明其核心功能（含预处理与下游分析），并附上官方帮助文档链接，方便科研人员快速检索与应用。

6.1 多组学数据

多组学数据涵盖基因组学、转录组学、表观基因组学、蛋白组学、代谢组学、微生物组学等多个分支，不同分支数据的预处理与下游分析需求存在差异：预处理核心需求包括各分支数据的专属质控（如基因组 SNP 过滤、转录组测序质量控制）、标准化、缺失值填充；下游分析涵盖各分支专属差异分析、功能注释及跨分支数据整合等。以下按具体组学分支分类阐述核心 R 包：

6.1.1 基因组学数据

基因组学数据预处理核心需求为 SNP 质量控制（过滤低质量位点、去除连锁不平衡位点）、数据格式转换、大样本数据内存优化；下游分析包括全基因组关联研究（GWAS）、主成分分析（PCA）、多基因风险评分（PRS）计算等。

核心 R 包：

6.1.1.1 bigstatsr & bigsnpr

核心功能：

- 预处理：采用内存映射技术高效处理大规模基因组数据（百万级变异位点、数万级样本），无需加载全量数据至内存；支持 SNP 质量控制（过滤低检出率位点）、连锁不平衡（LD）过滤等核心预处理步骤；兼容 VCF、PLINK 等标准基因组数据格式。
- 下游分析：快速实现主成分分析（PCA）以校正群体分层；提供多基因风险评分（PRS）计算算法；支持全基因组关联研究（GWAS）的关联分析，可在普通台式机上处理 50 万样本 +100 万标记的大规模数据集。

帮助文档链接：

- bigstatsr: [官方文档](#)
- bigsnpr: [官方文档](#)

6.1.1.2 maftools

核心功能：

- 预处理：专为癌症体细胞突变数据（MAF 格式）设计，支持 MAF 文件的读取、验证与数据清洗（过滤低质量突变位点、去除胚系突变污染）；可整合临床信息实现样本层面的数据匹配与筛选。

- 下游分析：提供多样化的突变图谱可视化功能（瀑布图、突变类型分布饼图、染色体突变位置热图等）；支持驱动基因识别、突变信号通路富集分析；可关联临床数据进行生存分析，挖掘突变特征与预后的关联。

帮助文档链接：[Bioconductor 官方文档](#)

补充 Python 库：PyMAFIA

- 核心功能：MAF 文件的快速解析与数据预处理；支持突变特征提取与可视化；可与 scikit-learn 联动进行机器学习建模。
- 帮助文档链接：[PyPI 官方文档](#)

6.1.2 转录组学数据

转录组学数据（RNA-seq）预处理核心需求为测序数据质量控制、比对结果过滤、基因表达量定量与标准化；下游分析包括差异表达基因（DEG）筛选、功能富集分析、样本聚类等。核心 R 包：

6.1.2.1 GEO2RNAseq

核心功能：

- 预处理：提供完整的 RNA-seq 数据预处理流程，支持从 GEO 数据库自动下载原始 FASTQ 数据，或导入本地 FASTQ 数据；涵盖测序质量控制、reads 比对、基因表达量定量等全流程标准化步骤；支持单物种、双物种（dual RNA-seq）、三物种（triple RNA-seq）转录组数据处理。
- 下游分析：内置差异表达基因（DEG）分析模块，输出差异基因列表及相关统计表格；提供中间结果与最终结果的可视化图表（如表达量分布箱线图、差异基因火山图）。

帮助文档链接：[官方文档（含安装指南与示例）](#)

补充 Python 库：Salmon + pandas

- Salmon：支持 RNA-seq 数据的快速定量（无需比对），输出基因/转录本表达量；帮助文档链接：[官方文档](#)
- pandas：用于转录组定量数据的清洗、标准化与格式转换，支持缺失值处理与数据筛选；帮助文档链接：[官方文档](#)

6.1.2.2 DESeq2

核心功能：

- 预处理：针对 RNA-seq 计数数据进行标准化（校正测序深度与文库组成差异）；支持缺失值过滤与低表达基因过滤，提升后续分析可靠性。
- 下游分析：基于负二项分布模型进行差异表达基因筛选，适用于两组或多组比较、配对样本等多种实验设计；支持差异基因的功能注释与可视化（火山图、热图）。

帮助文档链接：[Bioconductor 官方文档](#)

补充 Python 库：edgeRpy2

- 核心功能：R 包 edgeR 的 Python 接口，支持 RNA-seq 计数数据的标准化与差异表达分析；适配多组比较、配对样本等实验设计；输出差异基因列表及统计指标，可直接用于后续功能注释。
- 帮助文档链接：[PyPI 官方文档](#)

6.1.3 表观基因组学数据

表观基因组学数据（以 DNA 甲基化为主）预处理核心需求为探针质量控制、背景校正、批次效应去除；下游分析包括差异甲基化位点（DMP）/区域（DMR）识别、功能注释、可视化等。核心 R 包：

6.1.3.1 easyEWAS

核心功能：

- 预处理：针对 Illumina DNA 甲基化芯片数据设计，支持探针质量控制（过滤低检出率探针）、背景校正与批次效应去除；提供数据标准化功能，适配不同芯片平台数据。
- 下游分析：支持多种场景下的差异甲基化位点分析；集成 DMRcate 方法实现差异甲基化区域（DMR）识别；提供差异甲基化结果的功能注释与可视化工具；内置 bootstrap 内部验证模块，提升结果稳健性。

帮助文档链接：[官方研究论文](#)（含详细使用说明）

补充 Python 库：methilpy

- 核心功能：支持 DNA 甲基化测序数据（WGBS、RRBS）的预处理，包括序列比对、甲基化位点调用与定量；提供批次效应去除与数据标准化功能；适配大规模甲基化数据集分析。
- 帮助文档链接：[官方文档](#)

6.1.3.2 minfi

核心功能：

- 预处理：支持 Illumina 甲基化芯片（450K、EPIC）数据的读取与探针质量控制；提供多种背景校正与标准化方法（如 BMIQ、SWAN），校正不同探针类型的偏差。
- 下游分析：实现差异甲基化位点与区域的识别；支持甲基化数据与临床表型的关联分析；提供甲基化水平的可视化工具（如甲基化热图、散点图）。

帮助文档链接：[Bioconductor 官方文档](#)

补充 Python 库：PyMethylProcess

- 核心功能：针对 Illumina 甲基化芯片数据的预处理工具，支持探针质量控制、背景校正、标准化与批次效应去除；提供甲基化数据可视化功能（如甲基化水平分布箱线图）；可导出标准化数据用于差异分析。
- 帮助文档链接：[官方文档](#)

6.1.4 蛋白组学数据

蛋白组学数据预处理核心需求为肽段/蛋白鉴定结果过滤、定量数据标准化、缺失值填充；下游分析包括差异蛋白筛选、蛋白互作网络分析、功能注释等。核心 R 包：

6.1.4.1 MSstats

核心功能：

- 预处理：支持多种质谱平台（如 LC-MS/MS）的蛋白定量数据导入；提供数据过滤（去除低置信度鉴定结果）、缺失值填充（基于不同实验设计的专属算法）、标准化（校正系统误差）等预处理功能。
- 下游分析：基于线性混合模型进行差异蛋白筛选，适配标记定量（TMT/iTRAQ）、非标记定量等多种实验设计；支持差异蛋白的功能富集分析与可视化；可集成蛋白互作数据库进行网络分析。

帮助文档链接：[Bioconductor 官方文档](#)

补充 Python 库：Pyteomics

- 核心功能：支持质谱蛋白组数据的解析（mzML、mzXML 等格式）、肽段/蛋白鉴定结果过滤与定量数据提取；提供缺失值填充与标准化功能；可与 pandas 联动进行数据后续分析。
- 帮助文档链接：[官方文档](#)

6.1.5 代谢组学数据

代谢组学数据预处理核心需求为峰提取与匹配、基线校正、噪声过滤、数据标准化；下游分析包括差异代谢物筛选、代谢通路富集、多组学关联分析等。核心 R 包：

6.1.5.1 XCMS

核心功能：

- 预处理：支持基于 LC-MS/MS 的代谢组学数据峰提取、峰匹配与峰对齐；提供基线校正、噪声过滤、同位素峰识别等预处理功能；适配不同样品类型（生物体液、组织）的代谢组数据。
- 下游分析：实现差异代谢物的筛选与鉴定；支持代谢组数据的主成分分析（PCA）、偏最小二乘判别分析（PLS-DA）等多元统计分析；可与转录组、蛋白组数据联动进行多组学关联分析。

帮助文档链接：[Bioconductor 官方文档](#)

6.1.5.2 MetaboAnalystR

核心功能：

- 预处理：支持代谢组数据的标准化（如自动缩放、pareto 缩放）、缺失值填充与异常值检测；提供数据过滤功能，保留高质量代谢物特征。
- 下游分析：集成多种差异代谢物筛选算法；提供代谢通路富集分析（基于 KEGG、HMDB 等数据库）；支持代谢组数据的可视化（热图、火山图、通路图）。

帮助文档链接：[Bioconductor 官方文档](#)

补充 Python 库：pymzML、MetaboAnalystPy

- pymzML：支持质谱原始数据（mzML 格式）的解析与预处理，提供峰提取、基线校正等功能；帮助文档链接：[官方文档](#)
- MetaboAnalystPy：MetaboAnalyst 工具的 Python 接口，支持代谢组数据标准化、差异代谢物筛选与通路富集分析；帮助文档链接：[官方文档](#)

6.1.6 单细胞组学数据

单细胞组学数据（单细胞 RNA-seq、单细胞 ATAC-seq 等）预处理核心需求为细胞过滤、基因过滤、批次效应去除、归一化；下游分析包括细胞聚类、细胞类型注释、差异表达分析、轨迹分析等。核心 R 包与 Python 库：

6.1.6.1 核心 R 包

6.1.6.1.1 Seurat 核心功能：

- 预处理：支持单细胞 RNA-seq、ATAC-seq 等多类型数据的读取；提供细胞过滤（去除低质量细胞、线粒体基因高比例细胞）、基因过滤（保留高可变基因）、归一化（LogNormalize、SCTransform）、批次效应去除（Harmony、CCA 整合）等完整预处理流程。
- 下游分析：实现细胞聚类（Louvain 算法）、细胞类型注释（参考数据集匹配）、差异表达基因筛选、细胞轨迹分析；提供丰富的可视化工具（UMAP、t-SNE 降维图、小提琴图、热图）。

帮助文档链接：[官方文档](#)

6.1.6.1.2 SingleR 核心功能：

- 预处理：支持与 Seurat 等单细胞分析包无缝衔接，导入预处理后的单细胞表达矩阵。
- 下游分析：基于参考数据集（如 Blueprint、ENCODE）实现单细胞的精准细胞类型注释；支持批量注释与可视化验证，提升细胞类型识别的可靠性。

帮助文档链接：[Bioconductor 官方文档](#)

6.1.6.2 补充 Python 库

6.1.6.2.1 scanpy 核心功能：

- 预处理：支持单细胞数据的高效读取与预处理（细胞/基因过滤、归一化、对数转换、批次效应去除）；内存优化较好，适用于大规模单细胞数据集（10 万 + 细胞）。
- 下游分析：提供细胞聚类、UMAP/t-SNE 降维、细胞类型注释、差异表达分析、轨迹推断等功能；支持与 scikit-learn、pandas 等库联动，扩展性强。

帮助文档链接：[官方文档](#)

6.1.6.2.2 scikit-learn 核心功能：

- 预处理：提供标准化、缺失值填充、特征选择等通用预处理工具，可辅助单细胞数据的特征优化。
- 下游分析：支持聚类算法（KMeans、DBSCAN）、降维方法（PCA、t-SNE）、分类模型（SVM、随机森林），可用于细胞聚类与细胞类型预测建模。

帮助文档链接：[官方文档](#)

6.1.7 微生物组多模态数据

微生物组多模态数据（微生物丰度 + 临床指标 + 环境因素）作为多组学的重要分支，预处理核心需求为数据标准化、中介变量筛选；下游分析包括中介效应分析、因果关系挖掘等。核心 R 包及对应 Python 库：

6.1.7.1 multimedia

核心功能：

- 预处理：支持微生物组数据（如 ASV 丰度）与临床数据（如 PHQ 抑郁评分）的整合与标准化；提供数据校验功能，确保中介变量、处理因素、结局变量的逻辑对齐。
- 下游分析：实现多模态中介效应分析，量化微生物组在处理因素（如治疗干预）与结局（如疾病预后）间的直接效应与间接效应；支持线性模型、稀疏回归、贝叶斯分层模型等多种统计方法。

帮助文档链接： [CRAN 官方文档](#)

补充 Python 库： qiime2

- 核心功能：微生物组数据处理全流程工具，支持测序数据质量控制、ASV/OTU 聚类、物种注释等预处理；提供数据标准化与缺失值填充功能；可整合临床数据进行关联分析与可视化。
- 帮助文档链接： [QIIME 2 官方文档](#)

6.1.7.2 MultiMed

核心功能：

- 预处理：支持多生物中介变量的筛选与标准化，兼容多种数据格式（数据框、矩阵）。
- 下游分析：提供基于置换检验的多中介变量联合检验方法，校正多重比较误差；适用于微生物组-临床指标-疾病结局的因果链分析。

帮助文档链接： [Bioconductor 官方文档](#)

补充 Python 库： scikit-bio

- 核心功能：支持微生物组数据的读取、标准化与转换；提供 β 多样性分析、物种组成可视化等功能；可与 pandas、scikit-learn 联动实现微生物组与临床数据的多模态中介效应分析。
- 帮助文档链接： [scikit-bio 官方文档](#)

6.1.8 多组学整合分析工具

适用于跨多个组学分支的数据整合分析，实现多维度数据的协同挖掘。核心 R 包：

6.1.8.1 mixOmics

核心功能：

- 预处理：支持多组学数据（转录组、蛋白组、微生物组等）的标准化、缺失值填充及数据过滤；提供数据维度压缩的预处理方法。
- 下游分析：核心功能为多组学数据整合（样本水平 N-整合、变量水平 P-整合）；支持稀疏多元统计方法（如 PLS-DA、CCA）进行生物标志物筛选；提供丰富的可视化工具（热图、散点图）展示组间差异与关联。

帮助文档链接： [官方网站（含教程与文档）](#)

补充 Python 库： PyMultiOmics

- 核心功能：支持多组学数据（转录组、蛋白组、代谢组）的整合与标准化；提供 PLS-DA、CCA 等多元统

计方法实现多组学数据关联分析；支持结果可视化（热图、散点图）。

- 帮助文档链接： [官方文档](#)

6.1.8.2 omicade4

核心功能：

- 预处理：支持多平台多组学数据（如不同芯片的基因表达数据）的格式对齐与样本排序（确保多数据集样本顺序一致）。
- 下游分析：基于协方差分析（CIA）实现多组学数据集的整合，挖掘不同组学数据间的协同模式；提供图形化展示功能，直观呈现多数据集的相似性与差异。

帮助文档链接： [Bioconductor 官方文档](#)

补充 Python 库： MOFA+ (Python 版)

- 核心功能：基于因子分析模型实现多组学数据的整合，挖掘跨组学的共享变异模式；支持缺失值插补与多组学数据的可视化解释；适用于大规模多组学数据集。
- 帮助文档链接： [官方文档](#)

6.1.8.3 MultiModalGraphics

核心功能：

- 预处理：与 Bioconductor 生态（如 MultiAssayExperiment、limma）无缝集成，支持多组学数据的预处理结果导入（如差异分析后的 p 值、fold-change 数据）。
- 下游分析：专注于多组学数据的统计注释可视化，提供注释热图、复合特征热图、阈值散点图等工具，可直接在图形中嵌入 p 值、差异倍数等统计指标，适用于癌症异质性、神经疾病时空动态等分析场景。

帮助文档链接： [Bioconductor 官方文档（开发版）](#)

补充 Python 库： Plotly、Seaborn

- Plotly：支持交互式多组学数据可视化（热图、散点图、箱线图），可实现图表的缩放、hover 详情查看，便于多组学结果的交互式探索；帮助文档链接： [官方文档](#)
- Seaborn：基于 matplotlib 的高级可视化库，提供简洁的 API 实现多组学数据的统计可视化（聚类热图、小提琴图、联合分布图），支持多维度数据的关联展示；帮助文档链接： [官方文档](#)

6.2 医学影像数据

医学影像数据预处理核心需求为格式标准化、配准、去噪、重采样等；下游分析包括特征提取、影像组学分析、可视化等。以下为核心 R 包及对应 Python 库：

6.2.1 ITKR

核心功能：

- 预处理：封装 ITK 框架的经典配准方法，支持医学影像的刚性（平移、旋转、缩放）与仿射变换，实现多模态影像（如 MRI 与 PET）的空间对齐；支持 DICOM、NIfTI 等格式的标准化处理。

- 下游分析：为后续影像融合、病灶定位等分析提供配准后的数据基础。

帮助文档链接：[ITKR 官方文档 \(ITK R 接口\)](#)

补充 Python 库：SimpleITK

- 核心功能：ITK 的 Python 接口，支持医学影像的刚性/仿射/非线性配准；兼容 DICOM、NIfTI 等格式，提供格式转换、重采样、去噪等预处理功能；可与 NumPy 联动实现影像数据的数组运算。
- 帮助文档链接：[SimpleITK 官方文档](#)

6.2.2 ANTsR

核心功能：

- 预处理：基于 ITK 框架，集成先进的 SyN (Symmetric Normalization) 非线性配准算法，适用于脑部等精细结构的多模态影像对齐；支持高维形变场估计与拓扑保持优化。
- 下游分析：可结合影像特征提取工具实现病变区域的精准量化分析。

帮助文档链接：[ANTsR 官方文档](#)

补充 Python 库：ANTsPy

- 核心功能：ANTs 工具的 Python 实现，集成 SyN 非线性配准算法，支持多模态影像精准对齐；提供形变场估计与应用功能，适配脑部、腹部等多部位影像分析；支持与 PyTorch 联动用于深度学习数据预处理。
- 帮助文档链接：[ANTsPy 官方文档](#)

6.2.3 SimpleITKR

核心功能：

- 预处理：简化医学影像读取与预处理流程，支持 DICOM、NIfTI 等多种格式一键读取；提供影像标准化（强度范围统一）、重采样（体素间距调整）、去噪（各向异性扩散滤波）等核心预处理功能。
- 下游分析：支持影像数据的数组转换，便于与其他统计分析包联动进行特征挖掘。

帮助文档链接：[SimpleITKR 官方文档 \(R 语言版\)](#)

补充 Python 库：NiBabel

- 核心功能：专注于神经影像格式 (NIfTI、ANALYZE 等) 的读取与写入；支持影像元数据解析与内存优化；可直接将影像数据转换为 NumPy 数组，便于与其他 Python 科学计算库联动。
- 帮助文档链接：[NiBabel 官方文档](#)

6.2.4 4. RNifti & oro.nifti

核心功能：

- 预处理：专为 NIfTI 格式影像（神经影像主流格式）设计，支持高效读取与内存优化（RNifti 的内存映射功能可降低大型影像的资源消耗）；oro.nifti 支持多维 NIfTI-1/2 格式及 GZIP 压缩文件，提供元数据解析接口。
- 下游分析：与神经影像分析生态（如 fMRI 处理包）无缝集成，支持影像切片分析与可视化。

帮助文档链接：

- RNifti: [CRAN 官方文档](#)
- oro.nifti: [CRAN 官方文档](#)

补充 Python 库: PyRadiomics

- 核心功能: 医学影像组学特征提取专用库, 支持从 NiftI、DICOM 格式影像中提取一阶统计量、纹理特征等; 提供特征筛选与标准化功能; 可与 SimpleITK 联动实现影像预处理-特征提取全流程。
- 帮助文档链接: [PyRadiomics 官方文档](#)
- RNifti: [CRAN 官方文档](#)
- oro.nifti: [CRAN 官方文档](#)

6.2.5 5. EBImage

核心功能:

- 预处理: 支持多通道生物学医学影像(如荧光切片)的读取、通道分离、去噪、形态学滤波(膨胀/腐蚀)等预处理; 自动识别 TIFF 等常见格式。
- 下游分析: 提供细胞分割、形态学特征提取(面积、纹理)及可视化功能, 适用于微观医学影像(如病理切片)的定量分析。

帮助文档链接: [Bioconductor 官方文档](#)

补充 Python 库: OpenCV-Python

- 核心功能: 支持多通道生物学医学影像的读取、通道分离、形态学滤波(膨胀/腐蚀)、去噪等预处理; 提供高效的图像数组运算接口; 可用于微观医学影像(如荧光切片)的预处理与初步分割。
- 帮助文档链接: [OpenCV-Python 官方文档](#)

6.2.6 6. RIA

核心功能:

- 预处理: 支持 DICOM、NiftI 等多种格式影像的读取与标准化; 提供快速的特征计算优化(向量化公式), 降低大规模参数计算的耗时。
- 下游分析: 专注于影像组学分析, 可计算一阶统计量、灰度共生矩阵、灰度游程矩阵等核心影像组学特征, 适用于肿瘤等疾病的影像标志物挖掘。

帮助文档链接: [CRAN 官方文档](#)

补充 Python 库: Radiomics (PyRadiomics 进阶版)

- 核心功能: 在 PyRadiomics 基础上扩展了更多高级影像组学特征; 支持自定义特征提取参数; 提供特征可视化与特征相关性分析功能, 适配肿瘤影像标志物挖掘场景。
- 帮助文档链接: [Radiomics 官方文档](#)

6.3 数字病理图像数据

数字病理图像(如 HE 染色切片、免疫组化切片、荧光切片等)具有大尺寸、高分辨率、染色异质性强等特点, 其预处理核心需求为大尺寸图像解析、染色归一化、去噪、背景分割; 下游分析包括细胞分割、组织分型、肿瘤区域检测、病理分级、免疫组化评分等。以下为核心处理工具(含 Python 库、开源软件及深度学习模型):

6.3.1 核心 Python 库

6.3.1.1 OpenSlide-python

核心功能：

- 预处理：专为大尺寸数字病理切片设计，支持读取 SVS、TIFF、NDPI 等主流病理图像格式；提供多分辨率图像解析、区域裁剪功能，解决大尺寸图像内存占用过高问题；支持图像元数据（如扫描放大倍数、切片信息）提取。
- 下游分析：为后续细胞分割、特征提取提供标准化的图像数据接口，可与其他 Python 库联动处理。

帮助文档链接：[官方文档](#)

6.3.1.2 HistomicsTK

核心功能：

- 预处理：提供病理图像染色归一化（校正不同批次染色差异）、颜色空间转换、噪声过滤等功能；支持组织区域自动分割，去除空白背景与杂质区域。
- 下游分析：实现病理图像的纹理特征、形态学特征提取（适配影像组学分析）；支持细胞计数、核分割结果量化；提供病理图像可视化工具（如标注叠加、特征热力图）。

帮助文档链接：[官方文档](#)

6.3.2 主流软件与深度学习模型

6.3.2.1 QuPath

核心功能：

- 预处理：支持大尺寸病理切片读取与可视化；提供手动/自动标注工具，辅助区域筛选；支持染色标准化与图像增强。
- 下游分析：集成细胞分割、组织分型算法；支持肿瘤浸润程度评估、免疫组化评分（如 Ki-67 评分）；可通过 Python 脚本扩展自定义分析流程，适配批量数据处理。

帮助文档链接：[官方文档](#)

6.3.2.2 CellPose

核心功能：

- 预处理：支持病理图像的自动对比度调整、噪声抑制，适配不同染色类型（HE、免疫组化）的图像输入。
- 下游分析：基于预训练深度学习模型实现高精度细胞分割（细胞核、细胞质）；支持自定义数据集微调，提升特定病理场景的分割效果；输出分割掩码与细胞形态参数（面积、圆形度）。

帮助文档链接：[官方文档](#)

6.4 临床文本数据

临床文本数据预处理核心需求为文本清洗、分词、术语标准化；下游分析包括关键词提取、关联分析、情感分析、文献挖掘等。以下为核心 R 包及对应 Python 库：

6.4.1 PubMedMining

核心功能：

- 预处理：自动化从 PubMed 数据库提取相关文献标题与出版年份，支持关键词（固定术语、发布术语）的精准匹配与文本过滤。
- 下游分析：计算术语间的逐点相互信息（PMI）评分，挖掘研究主题相关术语的隐含关联，助力医学文献热点识别与领域知识图谱构建。

帮助文档链接： [GitHub 官方文档](#)（含安装与示例）

补充 Python 库： BioPython（PubMed 挖掘模块）

- 核心功能：支持从 PubMed 数据库批量检索文献摘要、标题等信息；提供文本过滤与关键词匹配功能；可提取文献中的作者、期刊、发表时间等元数据，助力文献计量分析。
- 帮助文档链接：[BioPython PubMed 模块官方文档](#)

6.4.2 tidytext

核心功能：

- 预处理：将临床文本（如电子病历、问卷回复）转换为整洁数据格式（每行 1 个词-文档对），支持停用词去除、词汇标准化（词干提取）等预处理步骤。
- 下游分析：集成 3 种情感词典（bing/nrc/afinn）实现临床文本的情感分析；支持关键词频率统计与可视化，适用于患者反馈分析、临床症状文本挖掘。

帮助文档链接： [CRAN 官方文档](#)

补充 Python 库： nltk、spaCy

- nltk：支持临床文本的分词、停用词去除、词干提取/词形还原等预处理；集成多种情感词典与文本分类模型，适用于患者反馈情感分析；帮助文档链接：[官方文档](#)
- spaCy：工业级自然语言处理库，支持临床术语识别（如疾病、药物名称）、依存句法分析；提供预训练的医学文本模型（如 en_ner_bc5cdr_md），适配电子病历文本挖掘；帮助文档链接：[官方文档](#)

6.5 总结

上述 R 包与 Python 库覆盖了多模态医学数据处理的全流程需求，不同模态数据可选择针对性工具：医学影像优先选择 ITKR、ANTsR、RIA 等 R 包，Python 可搭配 SimpleITK、NiBabel；数字病理图像推荐 OpenSlide-python、HistomicsTK（Python 库）及 QuPath 软件、CellPose 模型；临床文本挖掘可选用 PubMedMining、tidytext（R 包）及 nltk、spaCy（Python 库）；多组学（含基因组、转录组、表观基因组、蛋白组、代谢组、单细胞组学、微生物组）推荐对应分支专属工具，整合分析可选用 mixOmics、omicade4（R 包）及 PyMultiOmics、MOFA+（Python 库）。

使用时建议结合具体数据类型与分析目标，参考官方帮助文档构建标准化分析流程，同时可利用 **R** 与 **Python** 生态的协同性（如数据格式互转、工具联动）实现复杂多模态数据的深度挖掘。