**RSNAP开发文档**

**目录**

[一、 开发目的 2](#_Toc136617678)

[二、 总体设计 2](#_Toc136617679)

[（一） 总体框架流程 2](#_Toc136617680)

[（二） 详细设计 3](#_Toc136617681)

[**1.** **配置参数模块** 3](#_Toc136617682)

[**2.** **质量控制模块** 3](#_Toc136617683)

[**3.** **从头组装模块** 3](#_Toc136617684)

[**4.** **表达水平评估模块** 3](#_Toc136617685)

[**5.** **差异表达基因鉴别模块** 3](#_Toc136617686)

[**6.** **转录本鉴别模块** 3](#_Toc136617687)

[**7.** **UniProtKB数据库的自动检索和下载模块** 3](#_Toc136617688)

[**8.** **功能注释分析模块** 3](#_Toc136617689)

[**9.** **主模块** 3](#_Toc136617690)

[三、 平台和主要模块 3](#_Toc136617691)

[（一） 开发平台 3](#_Toc136617692)

[（二） 主要模块 3](#_Toc136617693)

[四、 所需依赖 3](#_Toc136617694)

[五、 应用测试 5](#_Toc136617695)

[**1.** **数据介绍** 5](#_Toc136617696)

[**2.** **参数设置和命令示例** 5](#_Toc136617697)

[**3.** **结果** 8](#_Toc136617698)

# 开发目的

本工具的开发目的是为了实现RNA-seq无参分析的自动化流程。RNA-seq无参分析是指在没有参考基因组序列或注释信息的情况下，对RNA-seq数据进行转录本的组装、定量、鉴定和注释等分析。这种分析对于那些基因组信息缺乏或不完善的物种，如许多植物、动物和微生物，具有重要的意义，可以发现新的基因或转录本，揭示其生物学功能和进化关系。然而，RNA-seq无参分析涉及多个步骤和软件，需要考虑数据质量、测序深度、组装算法、表达量估计、差异表达分析、功能注释分析等多方面的因素。每个步骤都可能影响最终的分析结果和解释，因此需要选择合适的方法和参数，以及进行有效的评估和验证。目前已有一些工具或流程可以实现部分或全部的RNA-seq无参分析，如Trinity、ExpressAnalyst等，但它们仍然存在一些不足或局限性，如运行速度慢、内存占用大、功能不完善或不灵活等。为了解决这些问题，我们开发了这个工具，它基于Python语言编写，使用多个开源软件和数据库，可以根据用户输入的参数配置文件，完成从原始测序数据到功能注释结果的一系列分析，并生成一个html格式的报告。该工具具有易用性、灵活性和可扩展性等特点，可以为RNA-seq无参分析提供一个方便快捷的解决方案，有助于那些缺乏完整基因组注释信息的物种的转录组学研究。

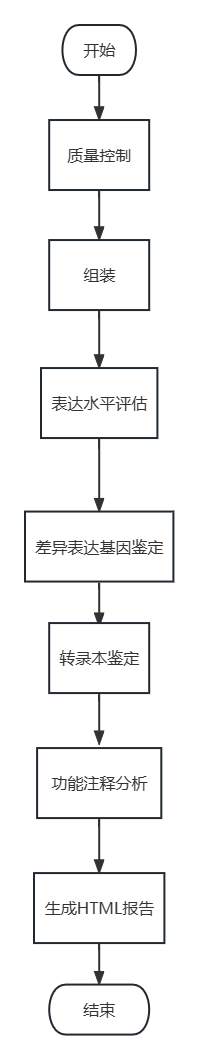


图 1 RSNAP总体框架流程图

# 总体设计

## 总体框架流程

为了完成RNA-seq无参分析，编写了若干脚本，执行RNA测序分析的一系列步骤，如质量控制、组装、表达量评估、差异表达基因鉴定、转录本鉴定和功能注释分析。该流程亦是RSNAP.py主模块的流程。

本流程的总体框架设计如下：

1. 开始：导入所需模块和包，检查和安装依赖，设置日志文件，读取配置文件和用户输入。

2. 质量控制：使用fastqc和trimmomatic或其他剪切程序对原始数据进行质量评估和过滤。

3. 组装：使用Trinity对过滤后的数据进行转录组组装，并生成组装结果和摘要文件。

4. 表达量评估：使用RSEM对组装结果进行表达量评估，并生成表达矩阵文件。

5. 差异表达基因鉴定：使用DESeq2或其他差异表达R包对表达矩阵进行差异分析，并生成差异表达基因列表和摘要文件。

6. 转录本鉴定：使用blastx对组装结果进行转录本鉴定，并生成转录本列表和摘要文件。

7. 功能注释分析：使用gseapy对差异表达基因进行功能注释分析，并生成注释结果和富集图文件。

8. 生成HTML报告：使用jinja2模板渲染生成HTML报告文件。

9. 结束：输出日志信息，结束脚本。

## 详细设计

1. **配置参数模块**

该模块使用Python编写的软件项目配置文件，用于设置用户输入参数和各个功能模块的默认参数。用户输入参数包括样本路径、样本分组信息、测序类型、测序文件类型、结果目录和检索关键词等。各个功能模块包括质量控制、转录本组装、表达量评估、差异表达基因鉴定、转录本鉴定和特征注释分析等。每个功能模块有一些可调节的参数，如线程数、阈值、软件选择、输出文件名等。这些参数可以根据用户的需要进行修改或保持默认值。

在设计上，该模块首先定义了用户输入参数，包括样本的路径、分组信息、测序类型、文件类型、输出目录和检索关键词等。然后定义了六个子模块的默认参数，分别是质量控制（Quality\_Control）、组装（Assemble\_trinity）、表达评估（Expression\_evaluation\_rsem）和差异表达基因鉴定（Differential\_expression\_gene\_identification）、转录本鉴定（Transcript\_identification）和特征注释分析（Feature\_annotation\_analysis）。

其中的用户输入参数需用户指定，遵循python代码的规范，使用字典和字典嵌套的格式记录变量和值。而六个子模块的参数有默认参数，如有需要也可更改。

1. **质量控制模块**

该模块的主要功能是对RNA-seq的原始数据（raw data）进行质量评估、过滤、裁剪和比对，并生成相应的日志文件和结果文件。其使用Python语言编写，使用click库来处理命令行参数，使用glob、os、zipfile、pickle等库来处理文件和目录，使用logging库来记录日志信息，使用config.py文件来配置用户参数。

本模块的基本功能流程为：1. main函数首先从config.py文件中导入用户参数，然后根据input\_files参数判断是否为双端测序数据，并将输入文件字符串转换为列表。然后调用run\_fastqc函数对输入文件进行质控分析，并生成zip格式的结果文件。接着调用fastqc\_parse函数对zip文件进行解析，并生成summary.txt文件和trim\_dict字典。如果not\_cut参数为否，则调用trim函数对输入文件进行裁剪，并生成trim\_files字典。最后将trim\_files字典保存为pkl格式的文件。2. run\_fastqc函数接收input\_files、output\_dir、format和threads四个参数，它的主要功能是执行fastqc命令对输入文件进行质控分析，并将结果保存在output\_dir目录中。该函数使用logging库来记录日志信息，并检查输入文件和输出目录是否存在。如果输出目录中已经存在zip格式的结果文件，则跳过fastqc命令的执行；否则，根据单端或双端参数设置fastqc命令，并调用os.system函数执行该命令，并判断是否执行成功。3. fastqc\_parse函数接收zip\_file\_name、out\_dir和threshold三个参数，它的主要功能是解析zip格式的结果文件，并打印相关信息。该函数使用logging库来记录日志信息，并获取zip文件中的data\_file\_name和summary\_file\_name两个子文件名。然后调用get\_data函数获取全局变量bases（读长）、gc\_percent（GC含量）和all\_base（所有碱基数）。接着调用calculate\_Q20\_Q30函数传入data\_file\_name，得到Q20和Q30的百分比，并打印出来。然后调用check\_quality\_status函数传入summary\_file\_name，得到是否通过质量检测的布尔值，并打印出来。最后调用find\_stable\_base\_number函数传入data\_file\_name和threshold，得到跳过的碱基数，并打印出来。该函数还将相关信息写入summary.txt文件，并返回一个元组(quality\_status, skip\_numbers)。4. get\_data函数接收data\_file参数，它的主要功能是从zip文件中的data文件中获取全局变量bases（读长）、gc\_percent（GC含量）和all\_base（所有碱基数）。该函数使用zipfile库来打开zip文件和data文件，并按行读取内容。然后遍历每一行，根据不同的关键字，提取相应的数值，并赋值给全局变量。5. calculate\_Q20\_Q30函数接收data\_file参数，它的主要功能是计算Q20和Q30的百分比。该函数使用zipfile库来打开zip文件和data文件，并按行读取内容。然后遍历每一行，判断是否在每条序列质量值的部分，并将其分割为质量值和计数两个字段。然后根据质量值的大小，累加质量大于等于20或30的碱基数，并计算其占所有碱基数的比例。最后返回一个元组(Q20\_percent, Q30\_percent)。6. check\_quality\_status函数接收summary\_file参数，它的主要功能是判断是否通过质量检测。该函数使用zipfile库来打开zip文件和summary文件，并按行读取内容。然后遍历每一行，判断是否是关于quality的指标，并将其状态添加到一个列表中。最后判断列表中是否都是PASS或者WARN，如果是，则返回True，否则返回False。7. find\_stable\_base\_number函数接收data\_file和threshold两个参数，它的主要功能是判断当碱基ATCG含量趋于稳定时候的跳过的碱基数。该函数使用zipfile库来打开zip文件和data文件，并按行读取内容。然后遍历每一行，判断是否是关于Per base sequence content的数据，并将其添加到一个列表中。然后去掉第一行和最后一行，只保留数据部分。然后定义一个内部函数inter\_fun，用于计算跳过的碱基数。该函数遍历每个位置的数据，将每个位置的A,T,C,G的百分比转换为浮点数，并计算其标准差（std）。然后判断三个标准差的值是否小于某个阈值（例如0.01），如果是，则认为该位置开始碱基含量趋于稳定，跳出循环；否则，累加跳过的碱基数。最后返回跳过的碱基数。find\_stable\_base\_number函数调用inter\_fun函数两次，分别对正向和反向的数据进行计算，并返回一个元组(skip\_number, skip\_number\_minus)。8. trim函数接收input\_files、trimdict、out\_dir和software四个参数，它的主要功能是对输入文件进行裁剪，并生成裁剪后的数据分组。该函数根据单端或双端参数，对trimdict字典进行处理，并构造裁剪后的数据分组trim\_files字典。然后根据software参数选择不同的裁剪程序，并构造相应的命令行cmd。然后检查输出文件是否存在，如果不存在，则调用os.system函数执行cmd命令，并判断是否执行成功；如果存在，则跳过执行。（图 2）

该模块接收的命令行参数如下：

1. input\_files: 输入测序数据文件，多个文件用百分号(%)隔开。（双端测序文件左右端用逗号隔开），默认为空字符串。
2. output\_dir: 输出质控分析结果目录，默认为'./qc\_fastqc\_output'。
3. format: 测序数据文件格式，可选'fastq'、'sam'或'bam'，默认为'fastq'。
4. not\_cut: 是否执行裁剪测序数据，默认为否。
5. threads: 线程数，默认为10。
6. threshold: 进行裁剪长度判断时的阈值，默认为0.01。
7. software: 进行裁剪的程序，可选'trimmomatic'或'cutadapt'，默认为'trimmomatic'。
8. config: 配置文件目录，默认为'./config.py'。

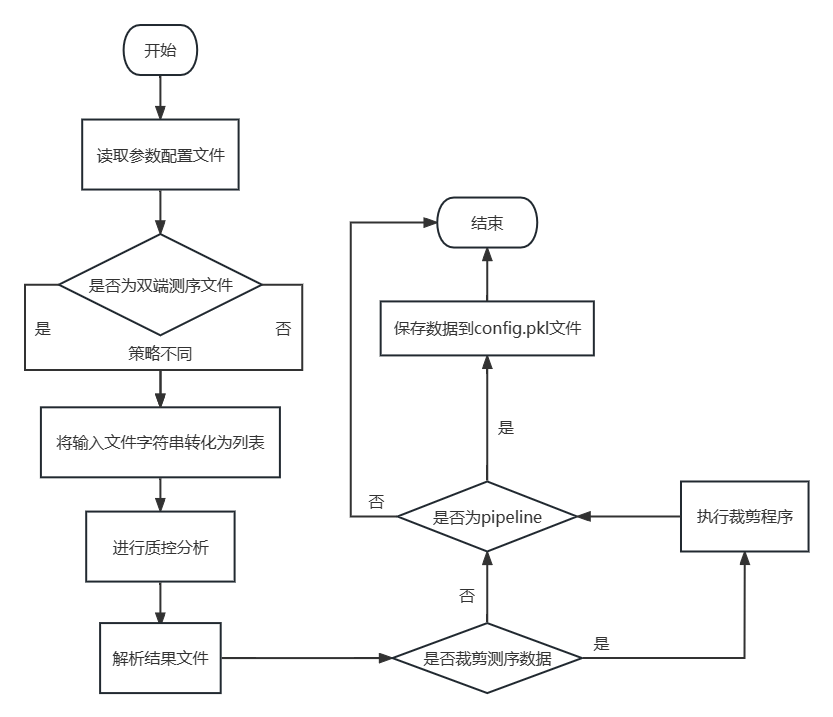


图 2 质量控制模块流程图

1. **从头组装模块**

该模块是一个使用click包构建的命令行接口，用于执行trinity rsem组装，提取组装摘要信息，并保存配置文件。它可以接收用户输入的参数，如测序数据文件格式、单端或双端数据、输出目录、结果文件名、最大内存占用数、CPU核数等。它还可以从配置文件中读取参数，以便与其他模块进行整合。

本模块的基本功能流程为：1. 定义函数run\_trinity，用于执行Trinity命令，根据单端或双端数据，设置不同的参数，如seqType、single、left、right、max\_memory、CPU、output等。如果输出目录中不存在结果文件，则执行命令并判断是否成功；如果存在，则直接返回结果文件的路径。2. 定义函数extract\_assembly\_summary，用于提取组装摘要信息，并输出到指定文件中。使用Trinity自带的TrinityStats.pl脚本，生成组装摘要信息，并判断是否成功。3. 定义click命令行接口，用于接收用户输入的参数，并设置默认值和帮助信息。根据是否提供配置文件，决定是否进行整合分析。4. 设置日志文件，记录程序的执行过程和结果，包括时间、文件名、行号、消息等。使用logging模块，创建记录器和处理器，并设置日志级别和格式。5. 检查输入参数是否合法，如果没有提供必要的参数，则抛出异常。6. 调用run\_trinity函数，进行组装，并获取组装结果文件路径。7. 调用extract\_assembly\_summary函数，提取组装摘要信息，并输出到指定文件中。8. 如果是pipeline分析，则读取原始pkl文件中的列表数据，并在原始列表中添加新的字符串元素，表示组装结果文件的路径。然后将原始列表和新的字符串一起写入新的pkl文件中。（图 3）

该模块接收的命令行参数如下：

1. format: 测序数据文件格式，可以是fq或fa，默认为fa
2. input: 单端测序数据存放目录或文件（多个文件使用逗号隔开）
3. left: 双端测序中(左段)数据存放目录或文件（多个文件使用逗号隔开）
4. right: 双端测序中(右段)数据存放目录或文件（多个文件使用逗号隔开）
5. output: 组装结果文件
6. paired: 是否为双端测序数据，默认为否
7. result\_file: 组装结果文件名，默认为assemblies.fa
8. summary\_file: 组装摘要结果文件，默认为summary.txt
9. max\_memory: 最大内存占用数（G）（默认为6G）
10. cpu: cup核数，默认为4
11. config: 配置文件目录

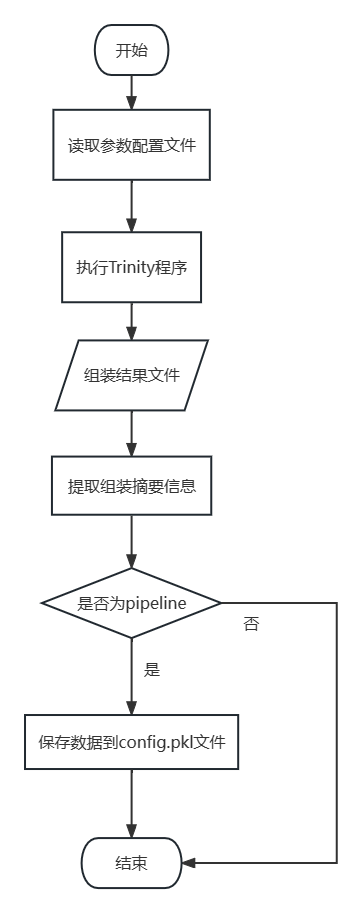


图 3 从头组装模块流程图

1. **表达水平评估模块**

该模块的主要功能是使用 rsem 软件对 RNA-seq 数据进行表达量定量分析，并生成表达矩阵。其使用 click 包来定义命令行参数，包括单端或双端测序数据文件列表、参考转录本文件、输出目录、配置文件和线程数，使用os 包来调用系统命令执行 rsem 相关的命令，包括 rsem-prepare-reference、rsem-calculate-expression 和 rsem-generate-data-matrix，使用 logging 包来记录日志信息，包括错误、警告和成功信息，使用 pickle 包来读取和写入配置文件中的数据，包括分组和样本信息。

本模块的基本功能流程为：1. 定义了一个函数 longest\_common\_prefix 来获取最大前缀公共子字符串，用于处理双端测序数据文件的前缀名。2. 检查输入参数是否合法，不能同时指定单端和双端测序数据文件，必须指定参考转录本文件，参考转录本文件必须存在。3. 检查输出目录是否存在，如果不存在则创建。4. 设置日志文件的路径、级别和格式。5. 如果是从配置文件中读取数据，则根据测序类型将 trim\_files 中的值转换为字符串，并赋值给 singles 或 paireds 参数。6. 构造 rsem-prepare-reference 命令，使用参考转录本文件作为输入，输出目录下的 Trinity 文件夹作为输出。7. 调用系统命令执行 rsem-prepare-reference 命令，并判断是否执行成功。8. 根据单端或双端构造 rsem-calculate-expression 命令，使用 single 或 paired 作为输入，输出目录下的样本名文件夹作为输出，并记录结果文件的路径。9. 调用系统命令执行 rsem-calculate-expression 命令，并判断是否执行成功。10. 构造 rsem-generate-data-matrix 命令，使用结果文件作为输入，输出目录下的 genes.expected\_count.matrix 文件作为输出。11. 调用系统命令执行 rsem-generate-data-matrix 命令，并判断是否执行成功。12. 读取配置文件中的数据，并添加新的字符串元素 expression\_matrix ，表示表达矩阵的路径。13. 将原始数据和新的字符串一起写入配置文件中。（图 4）

该模块接收的命令行参数如下：

1. singles: 单端测序数据文件列表（用逗号分割）
2. paireds: 双端测序数据文件列表（用逗号分割）（两端文件依次排列）
3. reference: 参考转录本文件
4. output: 输出目录
5. config: 配置文件目录，默认为./config.py
6. threads: 线程数，默认为8

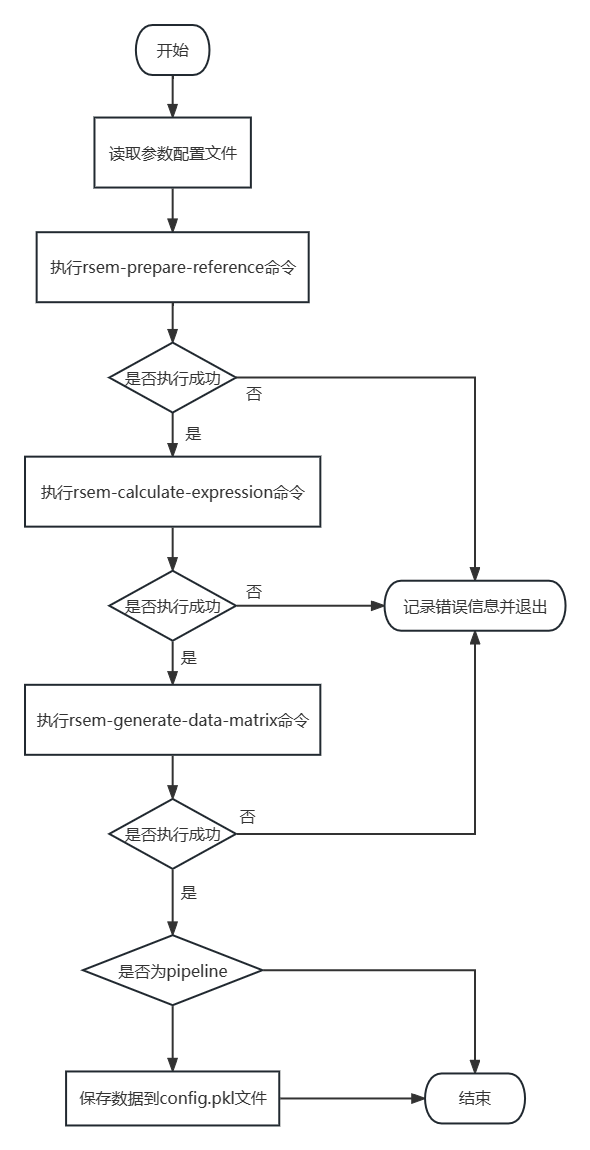


图 4 表达水平评估模块流程图

1. **差异表达基因鉴别模块**

该模块使用click包构建的命令行，使用rpy2包使python与R交互，并且执行R命令进行差异分析。

本模块的基本功能流程为：1. 设置日志文件，记录程序的运行情况和错误信息。2. 检查输入文件和输出目录是否存在，如果不存在则报错或创建。3. 从配置文件中获取分组和样本信息，构建一个分组向量 gs\_vector。4. 根据选择的 R 包，生成相应的 R 代码 cmd。5. 在 R 环境中执行 cmd ，进行以下步骤：读取输入文件，将其转换为 DGEList 或 DESeqDataSet 对象。对数据进行归一化和离散度估计。输出归一化后的表达矩阵文件。绘制 MDS 图和聚类树图，评估样本间的差异和相似性。进行精确检验或 Wald 检验，得到所有基因的差异分析结果。根据 log2FD 和 padj 阈值，筛选出差异表达显著的基因，包括上调、下调和总体表达水平变化，并输出差异表达基因结果文件。绘制火山图、散点图和聚类分析热图，可视化差异表达基因的分布和表达模式。6. 读取原始 pkl 文件中的列表数据，并将输出文件添加到列表中，再写入新的 pkl 文件中。（图 5）

该模块接收的命令行参数如下：

1. input\_file: 输入计数矩阵文件，必须提供。
2. output\_dir: 输出质控分析结果目录，默认为'DEG\_identification\_output'。
3. software: 差异分析 R 包，可以选择 edgeR 或 DESeq2，默认为 edgeR。
4. foldchange: 差异分析 log2foldchange 筛选阈值，默认为 1。
5. padj: 差异分析矫正 p 值 padj 筛选阈值，默认为 0.05。
6. config: 配置文件目录，默认为 ./config.py。

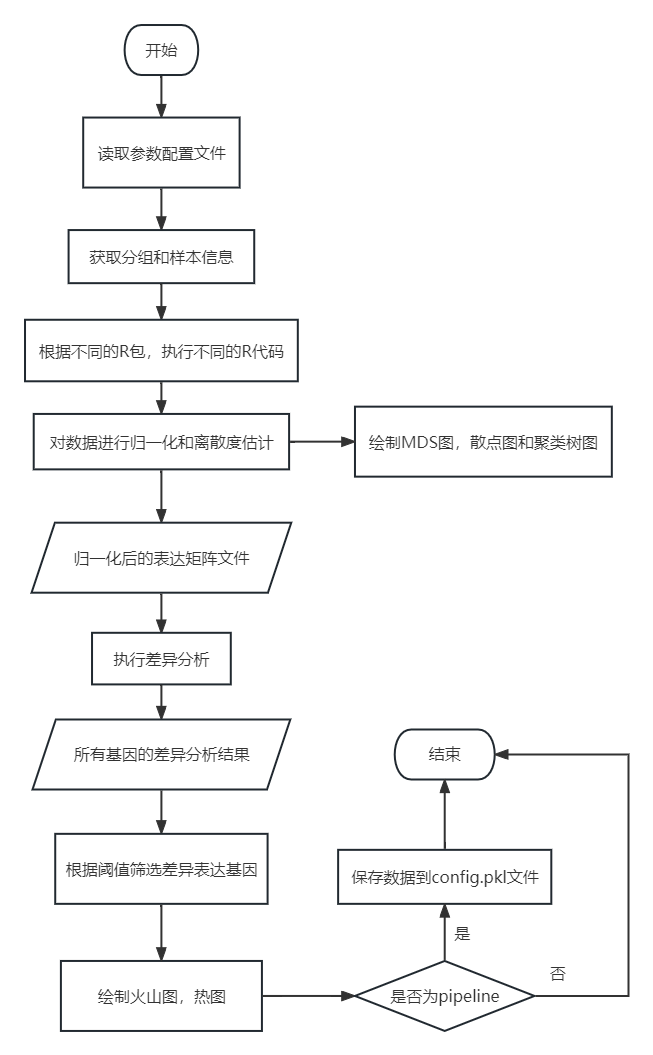


图 5 差异表达基因鉴别模块

1. **转录本鉴别模块**

这个模块的主要功能是使用blast工具将组装转录本序列文件与已知蛋白序列文件或检索关键词对应的UniProt数据库中的蛋白序列进行比对，然后根据比对结果识别转录本并进行统计和摘要分析。

本模块的基本功能流程为：1. 检查输入参数是否合法，如果不合法则报错并退出。2. 设置日志文件，记录程序执行过程中的信息和错误。3. 如果没有提供已知蛋白序列文件，则调用Uniprot\_query.py子程序根据检索关键词从UniProt数据库下载相应的蛋白序列文件，并解压缩得到fasta格式的文件。4. 使用makeblastdb工具将蛋白序列文件构建成blast数据库，并保存在blastdb目录下。5. 使用blastx工具将组装转录本序列文件与blast数据库进行比对，并将结果保存在blast\_result文件中6. 调用get\_res\_file函数，根据blast\_result文件生成stat\_file, summary\_file和transcript\_out三个输出文件，并保存在output目录下。7. 如果提供了配置文件，则将所有输出文件的路径保存在配置文件对应的pickle文件中。（图 6）

该模块接收的命令行参数如下：

1. input\_file: 组装转录本序列文件，必须提供。
2. protein\_file: 已知蛋白序列文件，可以为空。
3. query: 检索关键词，例如"(taxonomy\_id:4751) AND (reviewed:true)"，可以为空。
4. output: 结果输出文件的绝对路径，默认为"./transcript\_indentification\_output/"。
5. outfmt: 结果输出格式，默认为"7"。
6. evalue: evalue值，默认为10e-5。
7. cpu: CPU核数，默认为"1"。
8. config: 配置文件目录，可以为空。

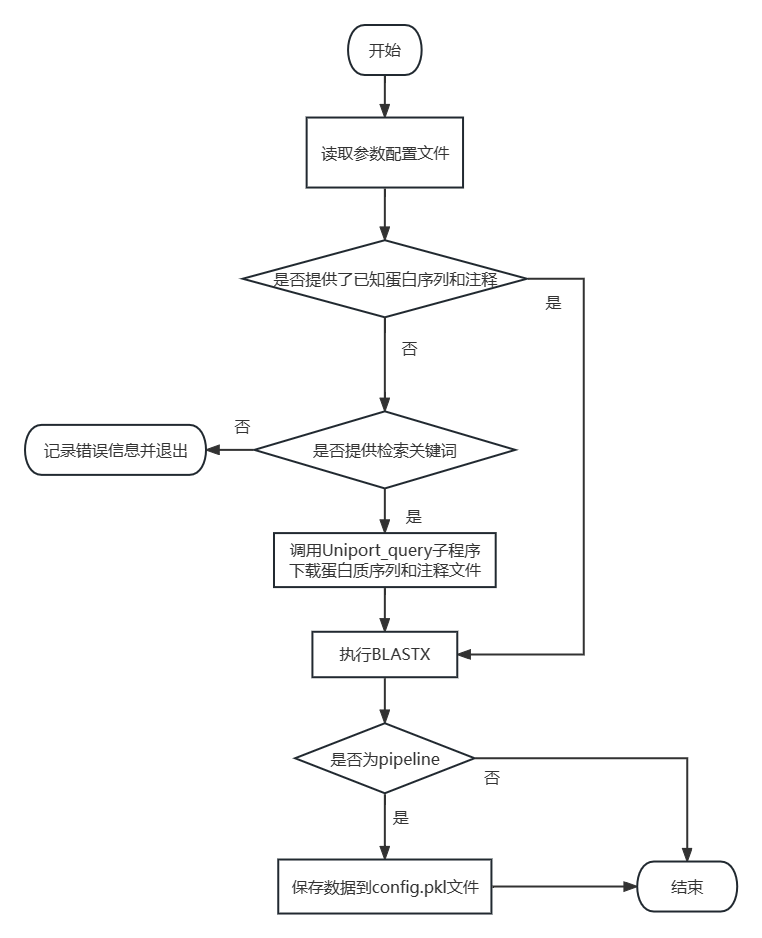


图 6 转录本鉴别模块流程图

1. **UniProtKB数据库的自动检索和下载模块**

该模块使用Python3语言编写，需要导入os, pickle, re, time, urllib.parse, logging, click, requests等python包。其定义了一个get\_next\_link函数，用于从响应头中提取下一页的链接，定义了一个get\_batch函数，用于发送请求并获取响应，并返回一个生成器，每次生成一个响应对象和总结果数，定义了一个get\_uniprot\_data函数，用于根据查询条件、格式、字段和文件名，从UniprotKB\_API中获取并保存summary文件，并设置一个全局变量data\_success表示是否成功，定义了一个get\_uniprot\_fasta函数，用于根据查询条件和文件名，从UniprotKB\_API中获取并保存fasta文件，并设置一个全局变量fasta\_success表示是否成功，定义了一个record\_download\_info函数，用于记录下载信息到本地文件，包括检索关键词、文件名、额外字段和时间等，定义了一个format\_filename函数，用于将不规范的字符替换为下划线，并转换为小写。

本模块的基本功能流程为：1. 获得config文件的内容，如果存在，则设置pipeline为True，并导入config模块。2. 检查输入参数是否有效，如果不有效，则退出该模块。3. 根据query生成file\_prefix，并根据是否指定fasta\_output和summary\_output生成相应的文件名。4. 获取data的准备工作，将默认字段和额外字段合并为fields列表。5. 检查下载文件是否存在，判断是否进行后续下载步骤，并设置data\_flag和fasta\_flag。6. 如果data\_flag为True，则调用get\_uniprot\_data函数保存summary文件。7. 如果fasta\_flag为True，则调用get\_uniprot\_fasta函数保存fasta文件。8. 根据data\_flag和fasta\_flag判断是否执行成功，并设置flag变量。9. 如果flag为True，则调用record\_download\_info函数记录下载信息到本地文件，并打印"执行成功！"。10. 如果flag为False，则根据data\_success和fasta\_success打印相应的错误信息或异常信息。11. 如果pipeline为True，则读取原始pkl文件中的列表数据，并在原始列表中添加新的字符串元素，包括fasta\_file和summary\_file，并将原始列表和新的字符串一起写入新的pkl文件中。（图 7）

该模块接收的命令行参数如下：

1. query: 检索关键词，例如"(taxonomy\_id:4751) AND (reviewed:true)"
2. addition\_fields: 检索该物种信息的额外条目(列)，用逗号分割，默认为:"accession", "xref\_geneid","gene\_names", "organism\_id", "organism\_name", "cc\_pathway", "go\_id", "go"
3. format: 指定下载总结文件的格式
4. output\_dir: 结果输出目录，默认为当前目录
5. fasta\_output: fasta文件名称，默认为{query}.fasta
6. summary\_output: summary表格文件名称，默认为{query}.tsv
7. uniprotKB\_records: 下载记录文件，默认为UniprotKB\_records.txt
8. config: 配置文件目录

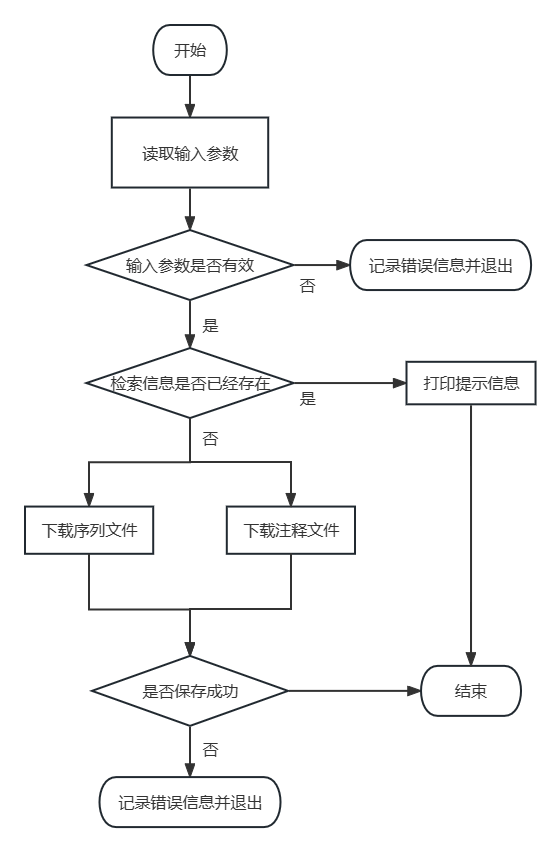


图 7 UniProtKB数据库的自动检索和下载模块

1. **功能注释分析模块**

该模块使用用 Python3编写用于对 RNA-seq 无参数分析的结果进行进一步的解释和富集分析。该模块定义了一些函数，用于从 BLAST 比对结果中提取可鉴定的转录本，从 UniprotKB 数据库中获取已知蛋白的注释信息，从注释信息中提取不同类型的注释（如 GO 和 Pathway），以及绘制富集分析的条形图和点图。

本模块的基本功能流程为：1. 使用 click 库来接收命令行参数，包括注释文件、BLAST 结果文件、差异表达基因列表文件、输出目录和文件名等。2. 使用 gseapy 库来进行基因集富集分析，使用 matplotlib 库来进行绘图，使用 pandas 库来处理数据框，使用 re 库来进行正则表达式匹配，使用 pickle 库来保存配置文件。3. 根据 BLAST 结果文件或者解析后的基因列表文件，获取所有可鉴定转录本和差异表达基因，并与注释文件进行合并，输出所有可鉴定转录本和差异表达基因的各类注释信息。4. 根据注释信息，构建不同类型的基因集（如 GO 和 Pathway），并对上调和下调的差异表达基因进行富集分析，绘制条形图和点图，并保存为 PDF 格式。5. 该模块还可以根据配置文件的内容，将结果文件添加到原始列表中，并保存为 pkl 格式。（图 8）

该模块接收的命令行参数如下：

1. annotation\_file：来自UniprotKB数据库的已知蛋白注释信息，必须参数
2. blast\_outfile：BLAST比对鉴别结果文件，可选参数
3. blast\_list：BLAST比对鉴别结果文件解析后的基因列表文件（两列，列名为query\_id和entry），可选参数
4. deg\_list：差异表达基因列表文件（两列，列名为query\_id和logFC），默认为空
5. output：结果输出文件(绝对路径)，默认为当前目录
6. all\_out：所有可鉴别转录本的各类注释信息结果文件名，默认为all\_anno.tsv
7. deg\_out：差异表达基因的各类注释信息结果文件名，默认为deg\_anno.tsv
8. enrichment\_out：差异表达基因的各类注释信息的富集分析结果文件目录名,默认为enrichment\_out（即在output目录下的enrichment\_out子目录）
9. config：配置文件目录，可以为空

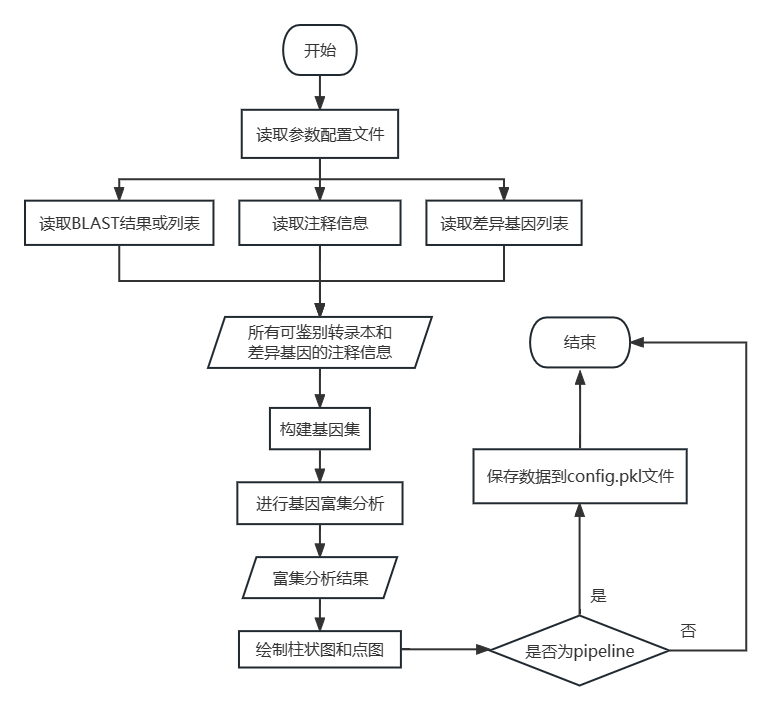


图 8功能注释分析模块流程图

1. **主模块**

该模块将上述模块组合起来，它的主要功能是从原始的测序数据开始，进行质量控制、转录本组装、表达量评估、差异表达分析、转录本鉴定和功能注释分析等步骤，最后生成一个HTML格式的报告。

本模块的基本功能流程为：1. 导入必要的包和模块，如os, sys, subprocess, click, rpy2, jinja2等，以及用户自定义的配置文件config.py。2. 检查系统环境和依赖软件，如python版本，R版本，R包，linux命令等，如果缺少则尝试安装或提示用户手动安装。3. 定义主函数main，并使用click库解析用户输入的参数，如样本信息，分组信息，测序类型，输出目录等。4. 设置日志文件，记录程序的执行过程和结果。5. 调用Quality\_Control.py模块对原始数据进行质量控制，使用fastqc, trimmomatic等软件对reads进行质量评估和过滤，并将结果保存在qc\_fastqc\_output目录下。6. 调用Assemble\_trinity.py模块对质控后的数据进行转录本组装，使用Trinity软件对reads进行拼接，并将结果保存在assemble\_trinity\_output目录下。7. 调用Expression\_evaluation\_rsem.py模块对组装后的转录本进行表达量评估，使用bowtie2和rsem软件对reads进行比对和定量，并将结果保存在expression\_rsem\_output目录下。8. 调用Differential\_expression\_gene\_identification.py模块对表达量数据进行差异表达分析，使用limma或edgeR软件对不同组之间的转录本进行统计检验，并将结果保存在DEG\_identification\_output目录下。9. 调用Transcript\_identification.py模块对差异表达的转录本进行鉴定，使用blastx软件对转录本与UniProt数据库中的蛋白序列进行比对，并将结果保存在transcript\_indentification\_output目录下。10. 调用Feature\_annotation\_analysis.py模块对鉴定后的转录本进行功能注释分析，使用gseapy软件对转录本进行富集分析，并将结果保存在feature\_annotation\_analysis\_output目录下。11. 使用jinja2库和report\_template模板生成HTML格式的报告，并将结果保存在output\_dir目录下。（图 9）

本模块只有一个参数config，默认为./config.py。

# 平台和主要模块

## 开发平台

硬件系统：不限制；

操作系统：Unix/Linux及其衍生版本；

编程语言：python3。

## 主要模块



# 所需依赖

1. **Python（版本大于3.7）**

os：使用操作系统相关功能的便携方式，比如文件和目录操作、进程管理和环境变量。

sys：提供了一些与Python解释器和它的环境相关的变量和函数，比如标准输入输出、命令行参数和退出状态。

subprocess：创建和管理子进程的方式，可以灵活地控制它们的输入输出和错误流。

zipfile：读写ZIP文件的工具，支持压缩和解压缩。

logging：记录日志信息的功能，可以灵活地配置日志级别、格式、输出目标等。

glob：根据通配符匹配文件名的方法，比如\*.txt或a?.py。

#gseapy：进行基因集富集分析（GSEA）和富集图（Enrichment Map）的方法，可以从多个数据库中获取基因集。

#jinja2：创建动态HTML模板的方法，支持变量、控制结构、过滤器等。

#click：创建命令行界面（CLI）的方法，支持参数、选项、子命令等。

#rpy2：在Python中调用R语言的方法，支持R对象、函数、包等。

#numpy：处理多维数组和矩阵的方法，支持高效的数学运算和随机数生成等。（版本大于等于1.9.0）

#scipy：进行科学计算的方法，支持线性代数、优化、统计、信号处理等。

#matplotlib：进行数据可视化的方法，支持绘制折线图、柱状图、散点图等。（版本大于等于1.4.3）

#pandas：处理结构化数据的方法，支持数据读取、清洗、分析、聚合等。（版本大于等于0.16）

#requests：发送HTTP请求的方法，支持GET、POST、PUT等方法和各种参数。

注：上述python包中，不带#号的为python内置模块，带#号的需要安装。

1. **R（版本大于3.7）**

还需安装如下R包：

DESeq2：一种进行差异表达分析的方法，可以处理RNA测序数据，使用负二项分布模型和Wald检验。

limma：一种进行线性模型分析的方法，可以处理基因芯片和RNA测序数据，使用经验贝叶斯方法和t检验。

edgeR：一种进行差异表达分析的方法，可以处理RNA测序数据，使用负二项分布模型和精确检验。（版本大于等于3.34.0）

pheatmap：一种绘制热图的方法，可以显示数据的聚类和相关性。

data.table：一种处理大型数据表的方法，支持快速的子集、排序、分组和连接操作。

1. **依赖程序**

还需安装如下程序：

Pigz：一个并行实现的gzip，可以利用多个处理器和多核心来压缩数据12。

blast：一个用于比较生物序列的工具，可以找到相似或同源的序列。

fastqc：一个用于检查原始测序数据质量的工具，可以生成各种质量指标的图表和报告。

trimmomatic：一个用于去除测序数据中的低质量或接头序列的工具，可以提高后续分析的准确性。

gzip：一个用于压缩和解压缩文件的工具，可以节省存储空间和传输时间。

Trinity：一个用于从RNA测序数据中组装转录本的工具，可以重建基因结构和表达量。

Bowtie：一个用于将短读比对到参考基因组的工具，可以快速且准确地找到比对位置。

Bowtie2：一个改进版的Bowtie，可以比对更长或更不规则的短读，也可以处理配对端测序数据。

RSEM：一个用于估计基因和转录本表达量的工具，可以从RNA测序数据中计算相对丰度。

jellyfish：一个用于统计k-mer频率的工具，可以从DNA或RNA测序数据中生成k-mer分布。

注：jellyfish应该为Trinity程序自带，但在测试中发现某些版本中执行Trinity时会报错不存在jellyfish。

# 应用测试

1. **数据介绍**

本次的数据来源于NCBI BioProject数据库，ID为PRJNA350822。为3对3的样本，其中实验组为使用NaCl处理（3g），对照组为做对照处理（S0g）。（https://www.ncbi.nlm.nih.gov/bioproject/PRJNA350822）

1. **参数设置和命令示例**
2. **参数设置**

本案例中的配置参数模块的内容如下，其含义为：

在用户输入参数中，样本绝对路径为/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/目录下的六个样本文件；样本分组信息为group1为对照组，group2为实验组；测序类型为双末端；测序文件类型为fastq；结果目录为当前目录下的RSNAP\_out目录；Uniport检索关键词为"(taxonomy\_id:4751) AND (reviewed:true)"检索物种id为4751（真菌），并且经过人工审查。

在Quality\_Control参数中，进行裁剪数据；线程数目为10；进行裁剪长度判断时的阈值为0.02；进行裁剪的程序为trimmomatic。

在Assemble\_trinity参数中，组装结果文件名为assemblies.fa；组装摘要结果文件名为summary.txt；最大内存占用数（G）为60；cup核数为4。

在Expression\_evaluation\_rsem参数中，线程数为10。

在Differential\_expression\_gene\_identification参数中，差异分析R包为edgeR；差异分析log2foldchange筛选阈值为1；异分析矫正p值padj筛选阈值为0.05。

在Transcript\_identification参数中，CPU核数为10；evalue筛选阈值为10e-5。

在Feature\_annotation\_analysis参数中，所有可鉴别转录本的各类注释信息结果文件名为all\_anno.tsv；差异表达基因的各类注释信息结果文件名为deg\_anno.tsv；差异表达基因的各类注释信息的富集分析结果文件目录名为enrichment\_out。

config.py文件具体设置如下：

user\_args = {

# 样本绝对路径（双末端）

"samples": { # 左 #右

"S4": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-1\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-1\_combined\_R2.fastq.gz"],

"S5": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-2\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-2\_combined\_R2.fastq.gz"],

"S6": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-3\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/3g-L-3\_combined\_R2.fastq.gz"],

"S1": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-1\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-1\_combined\_R2.fastq.gz"],

"S2": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-2\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-2\_combined\_R2.fastq.gz"],

"S3": ["/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-3\_combined\_R1.fastq.gz", "/home/zhanggaochuan/Dr.Wei/2019-AP-RNA-Seq-NaCl/combined/S0g-L-3\_combined\_R2.fastq.gz"],

},

# 样本分组信息,对照组放前面，实验组放在后面

"groups": {

"group1": ["S1", "S2", "S3"],

"group2": ["S4", "S5", "S6"]

},

},

# 测序类型（单双末端） paired或者single

"layout": "paired",

# 测序文件类型 fastq或者fasta

"format": "fastq",

# 结果目录（默认为./RSNAP\_out/）

"output\_dir": "./RSNAP\_out/",

# 检索关键词，例如"(taxonomy\_id:4751) AND (reviewed:true)"\n详细请查看手册：https://www.uniprot.org/help/query-fields

"query": "(taxonomy\_id:4751) AND (reviewed:true)"

}

# Quality\_Control默认参数

qc\_args = {

"not\_cut": False, # 是否执行裁剪测序数据，默认为是(False表示不不切割，如过为True则为切割)

"threads": "10", # 线程数，默认为10

"threshold": 0.02, # 进行裁剪长度判断时的阈值，默认为0.01

"software": "trimmomatic", # 进行裁剪的程序，默认为trimmomatic

}

# Assemble\_trinity默认参数

at\_args = {

"result\_file": "assemblies.fa", # 组装结果文件名

"summary\_file": "summary.txt", # 组装摘要结果文件名

"max\_memory": "60", # 最大内存占用数（G）（默认为6G）

"cpu": "4", # cup核数，默认为4

}

# Expression\_evaluation\_rsem默认参数

eer\_args = {

"threads": "10", # 线程数，默认为8

}

# Differential\_expression\_gene\_identification默认参数

degi\_args = {

"software": "edgeR", # 差异分析R包,默认为'edgeR'。支持("edgeR", "DESeq2")

"foldchange": "1", # 差异分析log2foldchange筛选阈值，默认为1

"padj": "0.05", # 异分析矫正p值padj筛选阈值，默认为0.05

}

# Transcript\_identification默认参数

ti\_args = {

"cpu": "10", # CPU核数(默认为1)

"evalue": 10 \*\* -5, # evalue值(默认为10e-5)

}

# Feature\_annotation\_analysis默认参数

faa\_args = {

"all\_out": "all\_anno.tsv", # 所有可鉴别转录本的各类注释信息结果文件名，默认为all\_anno.tsv

"deg\_out": "deg\_anno.tsv", # 差异表达基因的各类注释信息结果文件名，默认为deg\_anno.tsv

"enrichment\_out": "enrichment\_out", # 差异表达基因的各类注释信息的富集分析结果文件目录名,默认为enrichment\_out（即在output目录下的enrichment\_out子目录）

}

1. **命令行实现**

* 若在本地目录

在运行程序时，首先确保所有文件具有可执行权限：

chmod 755 ./\*

执行RSNAP主程序，由于config文件默认为./ config.py，故不用指定。因为程序运行需要很长时间。故让其后台运行，并且将日志输出到mylog.log文件：

nohup ./RSNAP.py > mylog.log &

* 放入bin目录

也可将所有程序及其模板放入/urs/bin目录

sudo cp ./\* /usr/bin/

然后执行RSNAP主程序，由于config文件与主程序不在同一目录，故需要指定绝对路径。因为程序运行需要很长时间。故让其后台运行，并且将日志输出到mylog.log文件：

nohup ./RSNAP.py –config /home/ubuntu/RSNAP/config.py > mylog.log &

1. **结果**
2. **结果文档和目录结构**

在./RSNAP\_out/目录下，结果目录如下表格 1 结果文档和目录结构表：

表格 1 结果文档和目录结构表

|  |  |
| --- | --- |
| **结果文档和目录结构** | **描述** |
| --APIGMT.log | RSNAP日志文件 |
| --all\_data.csv | 结果总表 |
| --report.html | 运行结果报告文件 |
| --qc\_fastqc\_output  |--quality\_control.log  |--\*\_fastqc.zip  |--\*\_summary.txt  |--\*\_fastqc.html  |--\*\_ trim\_unpaired.fq.gz  |--\*\_trim.fq.gz | 质量控制结果目录  质量控制日志  Fastqc结果文件  质控摘要文件  Fastqc报告文件  裁剪后的非配对文件  裁剪后的结果文件 |
| --assemble\_trinity\_output  |--assemble\_trinity.log  |--summary.txt  |--assemblies.fa  |--其他文件及其目录 | 组装结果目录  组装日志  组装摘要文件  组装结果文件  Trinity运行结果 |
| --expression\_rsem\_output  |--expression\_evaluation\_rsem.log  |--genes.expected\_count.matrix  |--其他文件及其目录 | 表达水平评估结果目录  表达水平评估日志  基因计数矩阵  RESM运行结果 |
| --DEG\_identification\_output  |--differential\_expression\_gene\_identification.log  |--scatter.pdf  |--plotMDS2.pdf  |--volcano.pdf  |--clustering\_tree\_plot.pdf  |--sigDEGs\_pheatmap.pdf  |--sigDEG.tsv  |--edgeR\_allGenes.out  |--edgeR\_upDEGs\_FDR0.05\_logFC1.out  |--edgeR\_downDEGs\_FDR0.05\_logFC1.out  |--edgeR\_sigDEGs\_FDR0.05\_logFC1.out  |--edgeR\_sigDEGs\_FDR0.05.out | 差异表达基因鉴别结果目录  差异表达基因鉴别日志  scatter map 散点图  多维缩放（MDS）聚类图  火山图  层次聚类树状图  差异表达基因聚类热图  差异表达基因摘要结果  所有转录本差异分析结果  上调基因差异分析结果  下调基因差异分析结果  按FDR和logFC筛选的差异基因差异分析结果  按FDR筛选的差异基因差异分析结果 |
| --transcript\_indentification\_output  |--differential\_expression\_gene\_identification.log  |--blastdb  |--all\_identified\_transcripts.tsv  |--blastx.stat  |--blastx.summary  |--taxonomy\_id\_4751\_and\_reviewed\_true.fasta  |--taxonomy\_id\_4751\_and\_reviewed\_true.fasta.gz  |--taxonomy\_id\_4751\_and\_reviewed\_true.outfmt7  |--taxonomy\_id\_4751\_and\_reviewed\_true.summary  |--UniprotKB\_records.txt | 转录本鉴别结果目录  转录本鉴别日志  Blast数据库文件夹  所有鉴别到的转录本  Blastx结果统计信息  Blastx结果总结  Uniport下载的测序文件  Uniport下载的测序文件(压缩)  Blastx结果文件  Uniport下载的注释信息文件  Uniport检索关键字日志 |
| --feature\_annotation\_analysis\_output  |--feature\_annotation\_analysis.log  |--all\_anno.tsv  |--deg\_anno.tsv  |--gene2anno\_Gene Ontology (GO).tsv  |--gene2anno\_Pathway.tsv  |--enrichment\_out  |--up\_pathway\_dotplot.pdf  |--up\_pathway\_barplot.pdf  |--up\_gene\_ontology\_go\_dotplot.pdf  |--up\_gene\_ontology\_go\_barplot.pdf  |--down\_pathway\_dotplot.pdf  |-- down\_pathway\_barplot.pdf  |-- down\_gene\_ontology\_go\_dotplot.pdf  |-- down\_gene\_ontology\_go\_barplot.pdf | 功能注释分析结果目录  功能注释分析日志  所有鉴定到的转录本的注释信息  差异基因的注释信息  GO富集分析结果表  KEGG通路富集分析结果表  富集分析可视化结果目录  上调基因KEGG通路富集分析气泡图  上调基因KEGG通路富集分析柱状图  上调基因GO富集分析气泡图  上调基因GO富集分析柱状图  下调基因KEGG通路富集分析气泡图  下调基因KEGG通路富集分析柱状图  下调基因GO富集分析气泡图  下调基因GO富集分析柱状图 |

1. **HTML报告结果**

report.html报告页面如下图所示。上方为标题栏，下方展示各个栏目的内容。每个栏目均有一个表格，收录了各个栏目的结果文件，可以点击超链接进入查看。在表格下方还有一些文件的示例数据。如果涉及到可视化的栏目，则会显示相应的图片以供查看。(图 10)



图 10 HTML报告结果页面示意图

1. **详细结果分析**

为了描述的方便，在这里我们规定。对照组为group1:S1，S2，S3。实验组为group2:S4，S5，S6。其中S0g-L-1\_combined\_R1.fastq.gz为S1\_1, S0g-L-1\_combined\_R2.fastq.gz为S2\_2。以此类推，我们的12个测序文件为：S1\_1, S1\_2, S2\_1, S2\_2, S3\_1, S3\_2, S4\_1, S4\_2, S5\_1, S5\_2, S6\_1, S6\_2。

1. **质量控制结果**

查看质控报告中各个样本的fastqc质量控制的总体情况。可以看到每个样本都是Per tile sequence quality, Per base sequence content, Per sequence GC content和Sequence Duplication Levels这四个指标出现警告和错误。

其中Per tile sequence quality是一种用来评估测序数据在不同区域和循环的质量情况的图表。它显示了每个tile（flow cell的一小块区域）在每个碱基位置的平均质量得分与全局平均质量得分的偏差。如果一个tile的质量得分低于其他tile，它会显示为暖色，反之则为冷色。一个理想的图表应该是全蓝色的，没有明显的偏差。出现警告或错误的原因可能是：flow cell过载，导致不同位置和循环的质量随机下降；测序序列有偏差，导致整个flow cell或某些区域的质量普遍低于预期；flow cell表面或内部有污渍、气泡、碎屑等阻碍物，导致某些区域或循环的质量持续或暂时下降；拍照系统出现故障，导致某些区域或循环的质量无法正确拍摄等。

Per base sequence content是一种用来评估测序数据在每个碱基位置的碱基组成的图表。它显示了每个碱基位置的A、T、G、C的比例。在一个随机的文库中，我们期望每个碱基位置的碱基组成没有太大的差异，所以图表中的四条线应该平行于彼此。每个碱基的相对比例应该反映你的基因组中这些碱基的总体比例，但无论如何它们不应该相互之间有很大的不平衡。Per base sequence content可能会出现警告或报错的原因有：测序序列有偏差，导致某些碱基在某些位置出现异常波动或偏离预期值；测序数据中存在过度表示的序列，如接头二聚体或rRNA，这些序列可能会影响整体的碱基组成；测序数据经过过度的接头剪切，导致序列末端出现碱基组成的偏差等。

Per sequence GC content是一种用来评估测序数据中每条序列的GC含量分布的图表。它显示了每条序列的平均GC含量（%）与序列数量的关系。在一个随机的文库中，你期望看到一个大致符合正态分布的GC含量分布，其中中心峰对应于基因组的整体GC含量。由于我们不知道基因组的GC含量，所以从观察到的数据中计算出模态GC含量，并用它构建一个参考分布。一个异常形状的分布可能表明一个受污染的文库或其他一些有偏差的子集。一个正态分布但是发生了平移的情况表明存在一些与碱基位置无关的系统性偏差。Per sequence GC content可能会出现警告或报错的原因有以下几种：测序数据中存在过度表示的序列，如polyA，polyT，单碱基重复序列，导致GC含量异常；测序数据中存在不同物种的污染，导致GC含量分布与目标物种不匹配；测序数据经过甲基化处理，导致大部分胞嘧啶转变为胸腺嘧啶，导致GC含量极低。

Sequence duplication levels是一种用来评估测序数据中完全一致的序列的频率的图表。它显示了序列重复的次数（1表示unique的序列，2表示有2条完全相同的reads，以此类推）与重复序列（duplicated reads）所占的百分比，以unique reads的总数作为100%。在一个多样性比较好的文库中，大部分的reads都应该是unique的，

或者只有少量的重复。如果有很高比例的重复序列，可能表明存在一些偏差或污染。Sequence duplication levels可能会出现警告或报错的原因有以下几种：测序数据中存在过度表示的序列，如接头二聚体或rRNA，这些序列可能会导致重复率增加；测序数据中存在不同物种或不同样本的污染，导致某些序列出现异常高频；测序数据经过PCR扩增，导致某些序列被过度扩增，产生偏差或假阳性。

我们规定Per base sequence quality，Per tile sequence quality，Per sequence quality scores这三个指标全部为PASS或者WARN，则认为通过质量检测，经过程序处理计算后，各个样本的情况和裁剪碱基数目如下表格 2 各样本质量和裁剪总结表：其中开头裁剪的碱基数目都为15，与fastqc报告中显示的Per base sequence content指标结果较为吻合。每个样本的Q20和Q30都较高。其中每对双端测序文件中有都只有一个通过了质量检测。

表格 2 各样本质量和裁剪总结表



1. **组装结果**

经过Trinity组装后，得到了assemblies.fa文件。组装的结果为：组装了12个样本的数据，每个样本有两个paired-end的fastq文件。Trinity组装得到了14,262个基因和24,459个转录本，GC含量为50.53%。基于所有转录本的统计，转录本的平均长度为2,757.79 bp，中位数长度为1,834 bp，N50为5,099 bp，总长度为67,452,772 bp。基于每个基因的最长转录本的统计，转录本的平均长度为1,947.76 bp，中位数长度为845 bp，N50为4,106 bp，总长度为27,778,970 bp。

组装摘要文件如下：

################################

## Counts of transcripts, etc.

################################

Total trinity 'genes': 14262

Total trinity transcripts: 24459

Percent GC: 50.53

########################################

Stats based on ALL transcript contigs:

########################################

Contig N10: 11786

Contig N20: 8866

Contig N30: 7218

Contig N40: 6074

Contig N50: 5099

Median contig length: 1834

Average contig: 2757.79

Total assembled bases: 67452772

#####################################################

## Stats based on ONLY LONGEST ISOFORM per 'GENE':

#####################################################

Contig N10: 10087

Contig N20: 7628

Contig N30: 6105

Contig N40: 5023

Contig N50: 4106

Median contig length: 845

Average contig: 1947.76

Total assembled bases: 27778970

1. **表达水平评估结果**

经过RSEM表达水平评估后，得到了表达矩阵文件：genes.expected\_count.matrix。

其格式如下：

"RSNAP\_out/expression\_rsem\_output/S4.genes.results"

"RSNAP\_out/expression\_rsem\_output/S5.genes.results"

"RSNAP\_out/expression\_rsem\_output/S6.genes.results"

"RSNAP\_out/expression\_rsem\_output/S1.genes.results"

"RSNAP\_out/expression\_rsem\_output/S2.genes.results"

"RSNAP\_out/expression\_rsem\_output/S3.genes.results"

"TRINITY\_DN0\_c0\_g1\_i1" 113.42 93.82 54.46 203.43 106.87 282.71

"TRINITY\_DN0\_c0\_g1\_i4" 669.78 397.55 574.66 687.57 475.90 765.13

"TRINITY\_DN0\_c0\_g1\_i5" 2984.46 3349.01 2694.34 3674.33 2781.32 4611.93

"TRINITY\_DN0\_c0\_g1\_i6" 12504.63 11772.03 12619.61 13966.80 12868.63 15468.20

"TRINITY\_DN0\_c0\_g1\_i8" 2086.72 2677.59 1160.93 4035.87 2831.28 5628.04

"TRINITY\_DN0\_c0\_g2\_i1" 2108.16 2357.20 2415.59 1669.19 3071.59 3271.24

"TRINITY\_DN0\_c0\_g2\_i2" 1578.02 1740.82 1747.50 1175.43 1775.55 2278.62

"TRINITY\_DN0\_c0\_g2\_i3" 95.62 0.00 87.58 39.60 140.00 105.71

"TRINITY\_DN0\_c0\_g2\_i5" 1456.31 1157.18 874.32 1346.67 2541.50 2104.22

"TRINITY\_DN0\_c0\_g2\_i7" 825.90 976.80 1385.01 489.12 25.34 790.21

"TRINITY\_DN0\_c0\_g3\_i2" 1721.52 1401.12 1095.37 1858.15 2344.14 3793.08

"TRINITY\_DN0\_c0\_g3\_i3" 7311.42 8254.46 7510.71 10509.47 10462.61 13106.51

1. **差异表达基因鉴别结果**

使用edgeR包进行差异表达基因鉴别后，得到的结果如下：以FDR=0.05阈值得到的差异基因有96个。以FDR=0.05和logFC=1为阈值得到的差异基因有83个，其中上调基因有35个，下调基因有48个。

下图 11 scatter map 散点图，展示了两组样本之间基因表达水平的相关性。它显示了每个基因在两组样本中的平均表达量（log10 transformed）的散点分布。显示了两组样本之间基因表达水平的高度相关性。散点分布在y=x直线附近，表示大部分基因在两组样本中表达量相近，没有显著上调或下调的基因。

下图 12 多维缩放（MDS）聚类图是一种用于展示样本之间基因表达差异的图表。它使用一种降维的方法，将样本的基因表达矩阵转换为二维的坐标，使得样本之间的距离反映了样本之间的差异程度。在该图中，每个点代表一个样本，不同的颜色表示不同的分组，横轴和纵轴是两个主成分（PC），表示样本差异的主要方向。logFC方法中S2和S3聚在了一类，但与S1较远。而S4, S5, S6则较为离散。Bcv方法中S2和S6聚在了一类，且其他样本均离群，总体效果不如logFC方法。

下图 13 层次聚类树状图展示了样本之间基因表达相似性。它使用一种自上而下的方法，将样本按照基因表达的相似度进行分组和合并，形成一个树状结构。该图中，每个叶子节点代表一个样本，不同的颜色表示不同的分组，每个内部节点代表一个聚类，节点之间的距离反映了聚类之间的差异程度。在没有经过标准化前S1单独一个分枝，其他5个样本一个分枝，总体效果不理想。在经过标准化后，将S2和S3，S4和S6归为一类，效果总体得到提示。

下图 14 火山图显示了每个基因在两个分组（3g-L和S0g-L）之间的表达变化（log2 fold change）和显著性水平（-log10 FDR）的散点分布，横轴为log2 fold change，纵轴为-log10 adjusted p-value。在图中，红色点表示显著上调的基因，绿色点表示显著下调的基因，而黑色点表示没有显著差异表达的基因。

下图 15 差异基因热图展示了样本和基因之间表达量。它使用一种颜色编码的方法，将样本和基因的表达量（log2 transformed）转换为不同的颜色，从而形成一个矩阵。图中的每一行代表一个基因，每一列代表一个样本，不同颜色代表不同的表达量，颜色越深表示表达量越高，颜色越浅表示表达量越低[28]。图中还显示了样本和基因的聚类情况，用树状图表示。热图显示了6个样本差异表达基因的表达量情况，可以看出两个分组（3g-L和S0g-L）在热图上有明显的分离，表示两个分组之间存在显著的基因表达差异。同时，同一分组内的样本聚类较紧密，表示同一分组内的样本差异较小，具有较好的重复性。在基因方面，可以看出有些基因在3g-L组中表达量较高，而在S0g-L组中表达量较低，反之亦然，这些基因可能是与实验条件相关的关键基因。

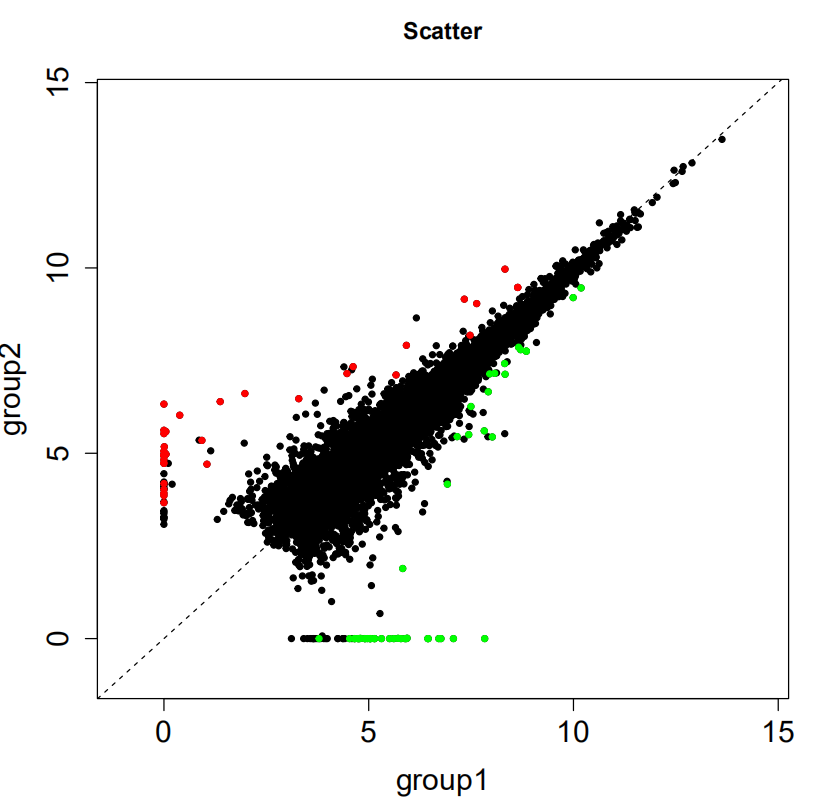


图 11 scatter map 散点图

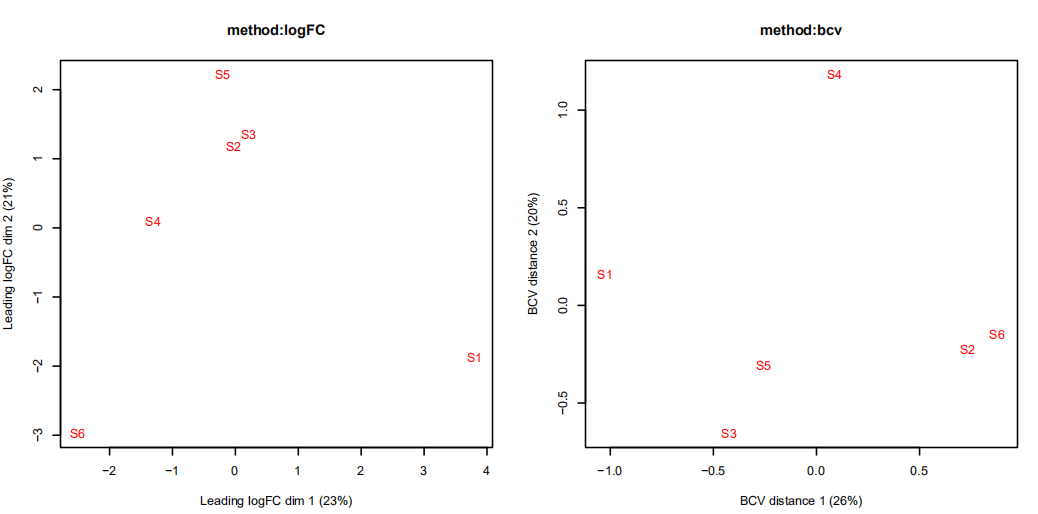


图 12 多维缩放（MDS）聚类图

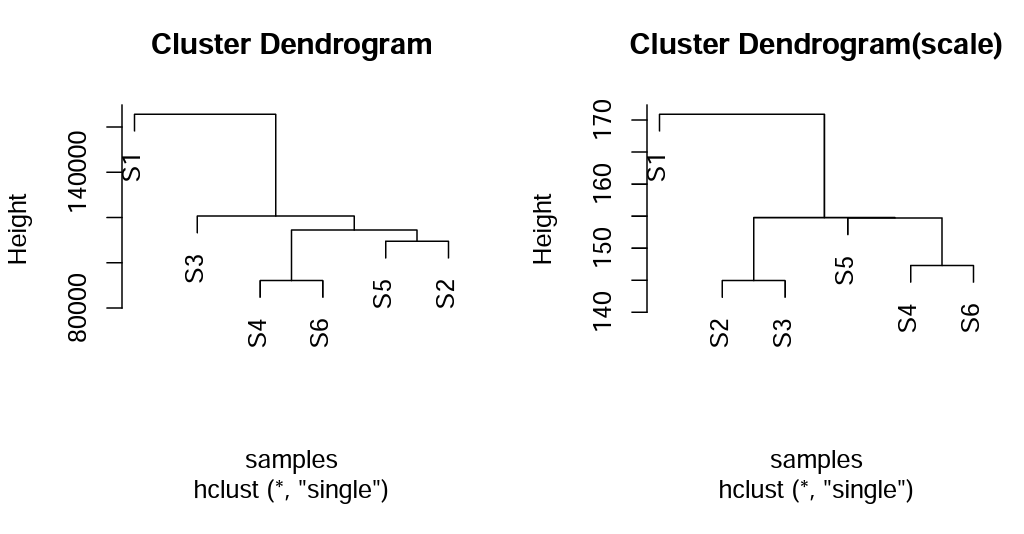


图 13 层次聚类树状图

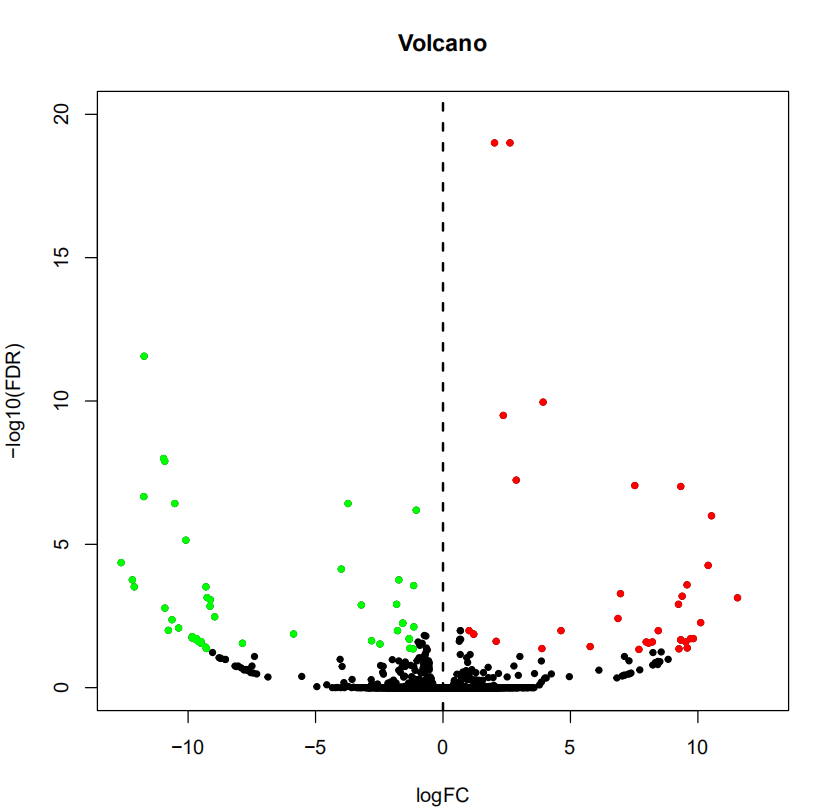


图 14 火山图

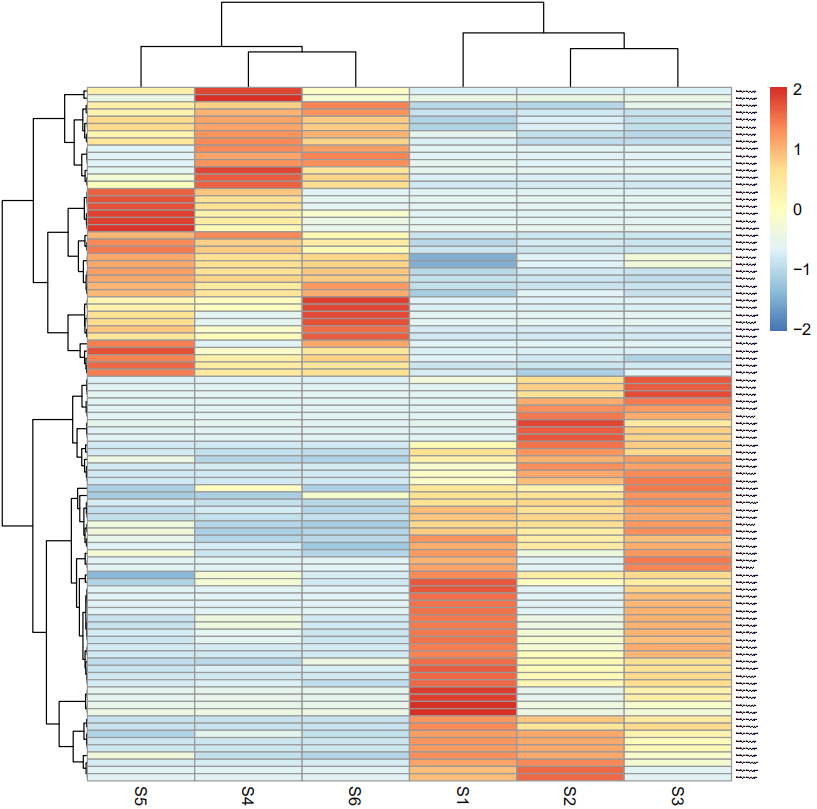


图 15 差异基因热图

1. **转录本鉴别结果**

在这一部分，首先使用Uniport API 检索关键字下载fasta序列文件和注释信息。其中序列文件：taxonomy\_id\_4751\_and\_reviewed\_true.fasta大小为22.8 MB ，共有36325条序列。Uniport下载的注释信息文件：taxonomy\_id\_4751\_and\_reviewed\_true.summary大小为43.9 MB。

然后使用下载的序列文件与组装的转录本进行序列比对。Blastx的总结结果blastx.summary如下：0 hits found的共有11228条，1 hits found的共有10712条，2 hits found的共有2128条，3 hits found的共有276条，4 hits found的共有84条，5 hits found的共有12条，6 hits found的共有13条，7 hits found的共有1条，8 hits found的共有4条，10 hits found的共有1条。共鉴别到13231条转录本。

1. **功能注释结果**

在功能注释中，将所有鉴定到的转录本的注释信息：all\_anno.tsv作为背景基因集，对所有差异基因的注释信息：deg\_anno.tsv进行基因富集分析。得到了KEGG通路富集分析结果表：gene2anno\_Pathway.tsv和GO富集分析结果表：gene2anno\_Gene Ontology (GO).tsv。

GO富集分析的结果如下图 16 上调基因GO富集分析点图显示了前10个富集的GO相关条目。下图 17 下调基因GO富集分析点图显示了前10个富集的GO相关条目。值得注意的是下调基因大多富集在氧化还原酶活性相关通路，由于氧化还原酶在代谢通路中发挥着重要作用，这意味着与代谢相关的基因可能会被下调。根据综合的GO富集结果，可以得知在NaCl条件下培养的出芽酵母中，与代谢相关的基因表达量明显下降。这可能是因为高压环境的NaCl迫使细胞减缓代谢过程，以减少对维持生命所需能量的消耗。

KEGG通路富集分析的结果如下图 18 上调基因KEGG富集分析柱状图所示。下图 19 下调基因KEGG富集分析柱状图。综合KEGG富集结果，在NaCl浓度为3g/L条件下，出芽短梗霉样本中的基因表达发生了一些变化，从而导致了通路富集分析结果的差异。上调基因富集在次生代谢物的生物合成可能表明在NaCl浓度为3g/L条件下，出芽短梗霉样本中的次生代谢物生产增加了。这可能是细胞为了适应NaCl胁迫而启动的一种生物合成途径，该途径可能与细胞的应激响应和适应性有关。蛋白质糖基化和鞘脂代谢是两个重要的细胞生物学过程，它们的上调可能反映了细胞在NaCl浓度为3g/L条件下对环境变化做出的适应性调整。L-缬氨酸的生物合成可能与蛋白质合成和氨基酸代谢有关，它的上调可能是为了满足细胞在NaCl浓度为3g/L条件下的需求。下调基因富集在从D-赤藓糖4-磷酸和磷酸烯醇丙酮中合成分枝氨基酸、代谢中间体生物合成、分枝氨基酸生物合成、代谢中间体生物合成、肌肉果糖生物合成、果胶降解等通路，可能表明在NaCl浓度为3g/L条件下，出芽短梗霉样本中这些代谢途径的活性下降了。这可能是为了节省细胞的能量和物质代谢，以应对NaCl胁迫所带来的负面影响。

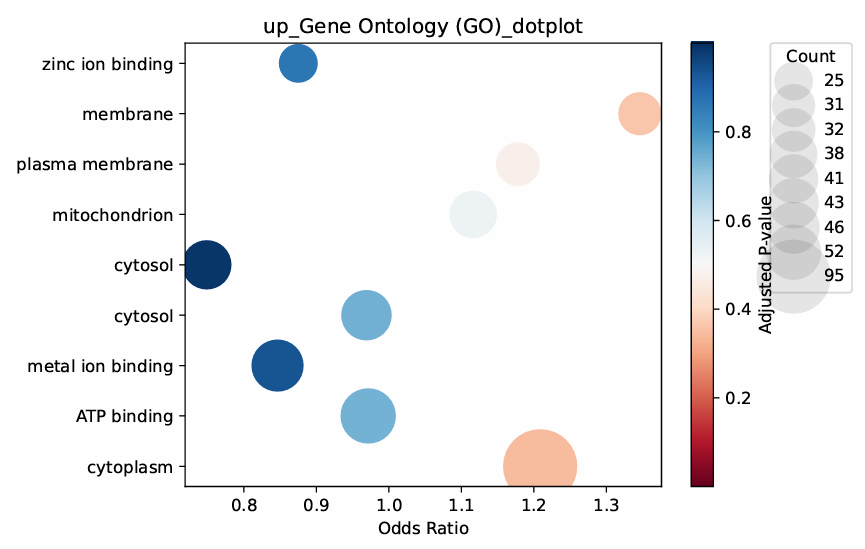


图 16 上调基因GO富集分析点图

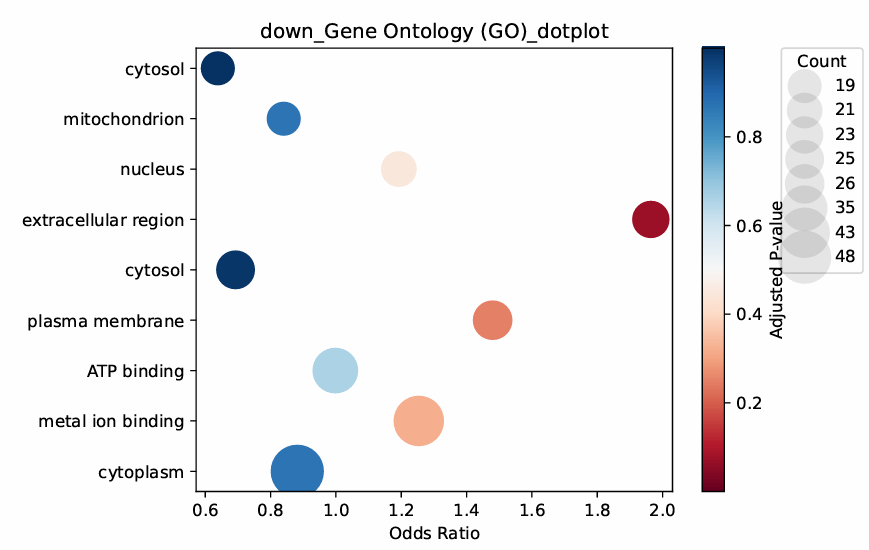


图 17 下调基因GO富集分析点图

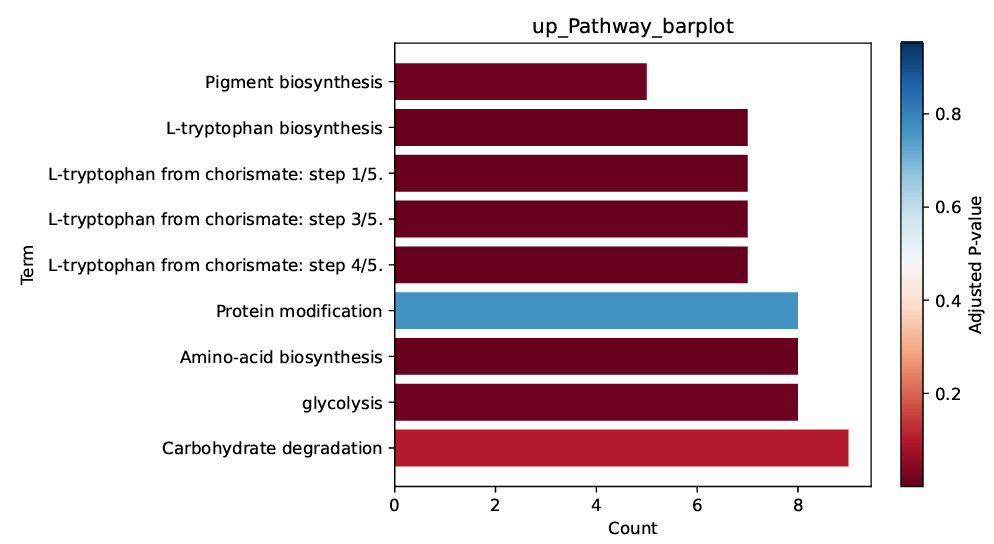


图 18 上调基因KEGG富集分析柱状图

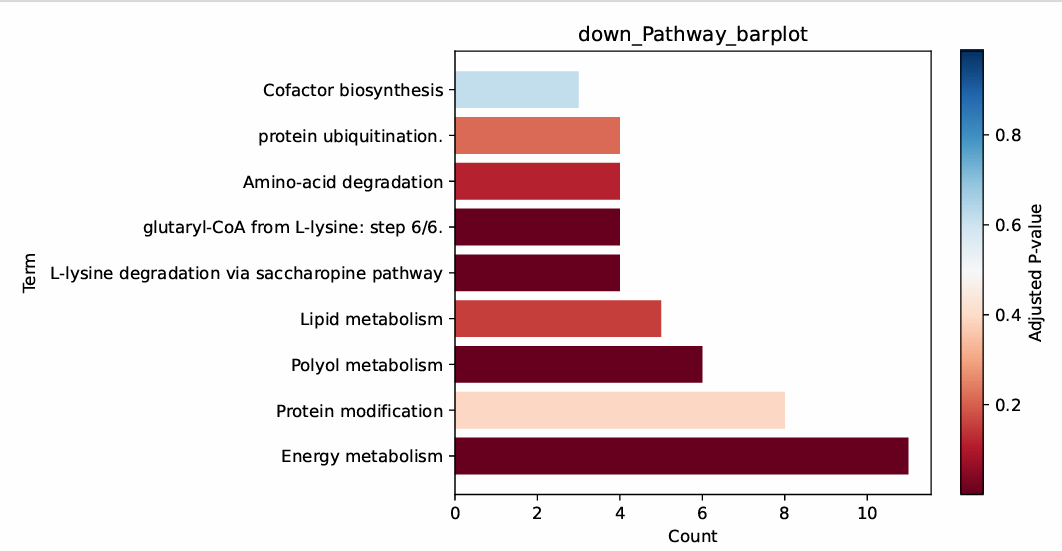


图 19 下调基因KEGG富集分析柱状图