**clash in windows科学上网**

[**https://zhuanlan.zhihu.com/p/460493388**](https://zhuanlan.zhihu.com/p/460493388)

**ip route show查看自己的默认路由端口为10.0.2.2**

**ubuntu系统设置网络设置里面找到网络代理：**

**在里面选手动，在HTTP代理中填入刚才插到的IP和端口**

**10.0.2.2 7890**

**（科学上网的时候记得把国内镜像源给清除掉，改回默认）**

conda config --remove-key channels

#添加官方镜源，科学上网才挂官方镜源

conda config --add channels bioconda

conda config --add channels conda-forge

#conda太慢了用mamba代替conda

conda install -c conda-forge -n base mamba

配置环境报错：https://blog.csdn.net/qq\_49641239/article/details/122127213

后面有conda的都可以改成mamba，速度飞起,创建环境的时候vpn要关了，不然一直报错

**宏基因组测序实验分析方法-功能分析基于reads**

1 使用ctab法或相应试剂盒提取样本中的总 DNA；

2 DNA样品检测合格后，使用Covaris超声波破碎仪随机打断，再经末端修复、加A尾、加测序接头、纯化、PCR扩增等步骤完成整个文库制备工作；

3 利用 Illumina NovaSeq 高通量测序平台进行宏基因组测序，获得样品中细菌、真菌和病毒等的宏基因组序列；

4 数据质控和去宿主序列：采用KneadData软件对原始数据进行质控(基于Trimmomatic)和去宿主(基于Bowtie2)，KneadData前和KneadData后，用FastQC来检测质控合理性和效果。

5 物种注释：使用Kraken2和自建的微生物核酸数据库(筛选NCBI NT核酸数据库和RefSeq全基因组数据库中属于细菌，真菌，古菌，病毒的序列)比对来计算样本中所含有物种的序列数，再用Bracken来对样本中物种的实际丰度进行估计。

6 功能数据库注释：从质控以及去除宿主基因的reads出发，使用HUMAnN2软件（基于DIAMOND），将各个样本的reads比对到数据库（UniRef90），根据UniRef90 ID 和各个数据库的对应关系，得到各个功能数据库的注释信息和相对丰度表。

7 基于物种丰度表和功能丰度表，可以进行丰度聚类分析，PCoA和NMDS降维分析（仅物种），样品聚类分析；当有分组信息时，可以进行LEfSe biomarker挖掘分析以及代谢通路比较分析，挖掘样品之间的物种组成和功能组成差异。

8 抗性基因注释：从去除宿主基因的clean reads出发，使用DIAMOND软件将各个样本的质控以及去除宿主基因的reads与抗生素抗性基因数据库CARD进行比对注释，可以获得抗性基因丰度分布情况。

**宏基因组测序实验分析方法（功能分析基于组装）**

1 使用ctab法或相应试剂盒提取样本中的总 DNA；

2 DNA样品检测合格后，使用Covaris超声波破碎仪随机打断，再经末端修复、加A尾、加测序接头、纯化、PCR扩增等步骤完成整个文库制备工作。

3 利用 Illumina NovaSeq 高通量测序平台进行宏基因组测序，获得样品中细菌、真菌和病毒等的宏基因组序列；

4 数据质控和去宿主序列：采用KneadData软件对原始数据进行质控(基于Trimmomatic)和去宿主(基于Bowtie2)，KneadData前和KneadData后，会用FastQC来检测质控合理性和效果。

5 功能注释：运用MEGAHIT软件，将所有样本去宿主基因后的clean reads进行组装，得到contigs; 运用prodigal软件，预测contigs中的基因序列; 再用Cd-hit软件，对得到的基因进行去冗余，得到去冗余基因; 使用Salmon软件，对去冗余基因进行定量; 使用eggnog-mapper, DIAMOND软件，对去冗余基因进行各个数据库的注释。统计各个数据库的基因相对丰度表。

6 物种注释：用diamond 将非冗余蛋白比对到NCBI NR蛋白数据库，比对结果再用BASTA软件分析得到非冗余蛋白的物种注释信息；同时使用Kraken2和自建的微生物核酸数据库(筛选NCBI NT核酸数据库和RefSeq全基因组数据库中属于细菌，真菌，古菌，病毒的序列)比对来计算样本中所含有物种的序列数，再用Bracken来对样本中物种的实际丰度进行估计。

7 抗性基因注释：运用DIAMOND软件，将去冗余基因比对到CARD数据库，得到CARD数据库的抗性基因注释信息，根据去冗余基因的丰度信息，统计抗性基因的相对丰度表。

8 基于物种丰度表和功能丰度表，可以进行丰度聚类分析，PCoA和NMDS降维分析（仅物种），样品聚类分析；当有分组信息时，可以进行LEfSe biomarker挖掘分析以及代谢通路比较分析，挖掘样品之间的物种组成和功能组成差异。

pip源

清华大学 ：https://pypi.tuna.tsinghua.edu.cn/simple/

阿里云：http://mirrors.aliyun.com/pypi/simple/

中国科学技术大学 ：http://pypi.mirrors.ustc.edu.cn/simple/

华中科技大学：http://pypi.hustunique.com/

豆瓣源：http://pypi.douban.com/simple/

腾讯源：http://mirrors.cloud.tencent.com/pypi/simple

华为镜像源：https://repo.huaweicloud.com/repository/pypi/simple/

linux

安装conda

wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2022.05-Linux-x86\_64.sh

bash Anaconda3-2022.05-Linux-x86\_64.sh

source ~/.bashrc

#查看当前conda配置

conda config --show channels

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 --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud//pytorch/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/

conda config --set show\_channel\_urls yes

##添加北外镜像  
conda config --add channels https://mirrors.bfsu.edu.cn/anaconda/cloud/bioconda/  
conda config --add channels https://mirrors.bfsu.edu.cn/anaconda/cloud/conda-forge/  
conda config --add channels https://mirrors.bfsu.edu.cn/anaconda/pkgs/free/  
conda config --add channels https://mirrors.bfsu.edu.cn/anaconda/pkgs/main/  
conda config --set show\_channel\_urls yes

#设置搜索是显示通道地址

conda config --set show\_channel\_urls yes

#创建新的独立环境

conda/mamba create -n rnaseq

conda activate rnaseq

conda deactivate 退出

conda env list查看已有环境/conda info –env

设置共享文件夹

先创建linuxshare文件夹，virtualbox设备-共享文件夹-勾选自动分配

mkdir share

sudo mount –t vboxsf linuxshare share

#sf\_share 没有权限

sudo usermod -aG vboxsf linux

重启

安装aspera

conda/mamba install -y -c hcc aspera-cli

ascp-h #检查是否安装成功

$ which ascp # 输出下面内容，不同环境不一样，根据自己的来

/home/linux/.aspera/connect/bin/ascp

$ ls ~/software/miniconda3/envs/test/etc/asperaweb\_id\_dsa.openssh

/home/linux/software/miniconda3/envs/test/etc/asperaweb\_id\_dsa.openssh

手动下载地址：<https://ak-delivery04-mul.dhe.ibm.com/sar/CMA/OSA/09cne/0/ibm-aspera-connect-3.11.0.5-linux-g2.12-64.tar.gz>

$ tar -zxvf ibm-aspera-connect-3.11.0.5-linux-g2.12-64.tar.gz #解压

$ ./ibm-aspera-connect-3.11.0.5-linux-g2.12-64.sh#安装

echo 'export PATH=~/.aspera/connect/bin:$PATH' >> ~/.bashrc

source ~/.bashrc

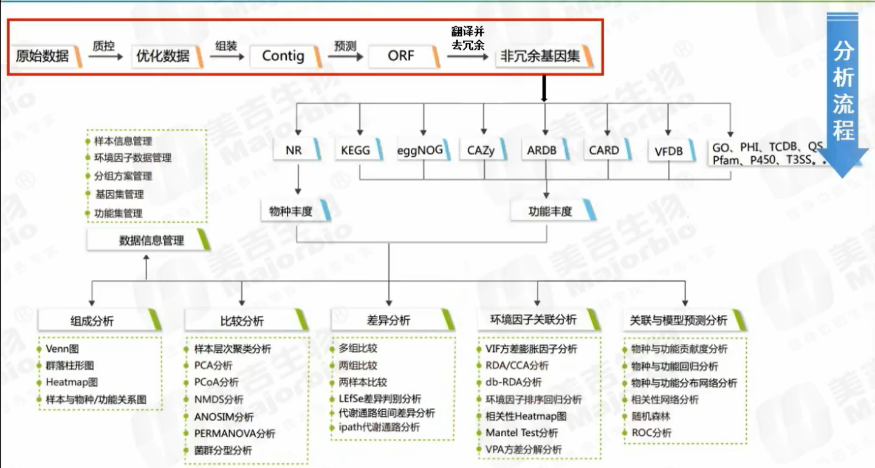
# 有输出帮助文档则安装成功

ascp -h

ascp -QT -l 300m -P33001 -k 1 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR110/087/SRR11080587/SRR11080587\_1.fastq.gz .

/home/linux/.aspera/connect/bin/ascp -QT -l 300m -P33001 -i /home/linux/.aspera/connect/etc/asperaweb\_id\_dsa.openssh [era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR109/041/SRR10972241/SRR10972241\_1.fastq.gz](mailto:era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR109/041/SRR10972241/SRR10972241_1.fastq.gz) .

<https://github.com/OpenGene/fastp>



**fastp质控**

简单粗暴

conda install fastp

mamba install fastp

**质量过滤**

SE单端：fastp -i in.fq.gz -o out.fq.gz

PE双端：fastp -i in.R1.fq.gz -I in.R2.fq.gz -o out.R1.fq.gz -O out.R2.fq.gz \

注意-i、-o、-I、-O分别forward输入和输出跟reverse输入和输出，前面小写后面大写

fastp --in1 SRR19023077\_1.fastq.gz --in2 SRR19023077\_2.fastq.gz –out1 SRR19023077\_1.clean.fq.gz. –out2 SRR19023077\_2.clean.fq.gz --html SRR19023077\_fastq.html --json SRR19023077\_fastq.json -e 25 -q 25 -u -r -M 30 -W 5

fastp -i SRR19023077\_1.fastq.gz -I SRR19023077\_2.fastq.gz -o SRR19023077\_1.clean.fq.gz. -O SRR19023077\_2.clean.fq.gz -h SRR19023077\_fastq.html -j SRR19023077\_fastq.json -e 25 -q 25 -u -r -M 30 -W 5

或者不进入conda 的base环境

wget http://opengene.org/fastp/fastp

chmod a+x fastp

/home/linux/fastp -i SRR19023077\_1.fastq.gz -I SRR19023077\_2.fastq.gz -o SRR19023077\_1.clean.fq.gz. -O SRR19023077\_2.clean.fq.gz -h SRR19023077\_fastq.html -j SRR19023077\_fastq.json -e 25 -q 25 -r -M 25 -W 5 -u 10

|  |  |
| --- | --- |
| -A, --disable\_adapter\_trimming | 默认是要进行接头去除的。如果这个参数被具体赋值，则接头去除就不默认进行，按参数执行 |
| -a, --adapter\_sequence | **设置read1的接头序列。如果是单端测序（SE），不指定软件就自动检测，如果是双端测序（PE），如果R1/R2没有重叠区域（overlap）就使用这个参数** |
| -W, --cut\_window\_size | **设置滑窗大小，默认是4** |
| -M, --cut\_mean\_quality | **滑窗的平均质量阈值，低于这个阈值就被切除，默认是Q20** |
| -Q, --disable\_quality\_filtering | **质量过滤默认是开启的，选择了这个参数就关闭质量过滤** |
| -q, --qualified\_quality\_phred | **设置碱基质量阈值，默认是15，就是phred质量评分≥Q15。** |
| -u, --unqualified\_percent\_limit | **允许百分之多少的碱基不合格（0-100），默认是40（就是说40%），超过这个比例，整条read就被删除了** |
| -n, --n\_base\_limit | **如果一条read中N碱基的数量超过了多少个，那么这条read就被删除，默认是5（即这条read里有5个N）。这需要根据你实际项目需求改，比如你的read是150bp, 如果你要求使N含量不超过5%，那么150\*0.05=7.5，但这里要填int型，不知道是填7.5会不会被软件接受** |
| -L, --disable\_length\_filtering | **长度过滤是默认开启的，如果填了这个参数那就关闭长度过滤。** |
| -l, --length\_required | **read小于这个参数设定长度会被丢弃或删除，默认是15** |
| -R, --report\_title | **设置报告标题，默认是“fastp report”** |
| -w, --thread | **设置运行的线程数，默认是3** |
| -e, --average\_qual | **设置为25，表示如果一条read的平均质量分数<25，那么这个read/pair会被丢弃，默认值为0表示没有要求。** |
| 滑窗质量修剪参数-r -M 25 -W 5 类似于Trimmomatic  SLIDINGWINDOW方法。 | **-r, --cut\_right 将滑动窗口从头移动到尾（front to tail），若遇到一个窗口内碱基的平均质量<25，则删除窗口中的碱基和余下的（right）部分，然后停止。** |

默认情况下会进行接头修剪，可以通过选项-A或--disable\_adapter\_trimming禁用，软件会自动检测单端或双端数据的接头序列。

如果由于低质量碱基找不到重叠区，推荐通过--adapter\_sequence为read1指定接头序列，通过--adapter\_sequence\_r2为read2指定接头序列。

对于双端数据，默认情况下不会进行接头序列自动检测，因为可以通过重叠分析修剪接头，但也可以使用选项--detect\_adapter\_for\_pe启用。

PS：--detect\_adapter\_for\_pe跟默认去除接头方法不一样，出来的结果估计有差异

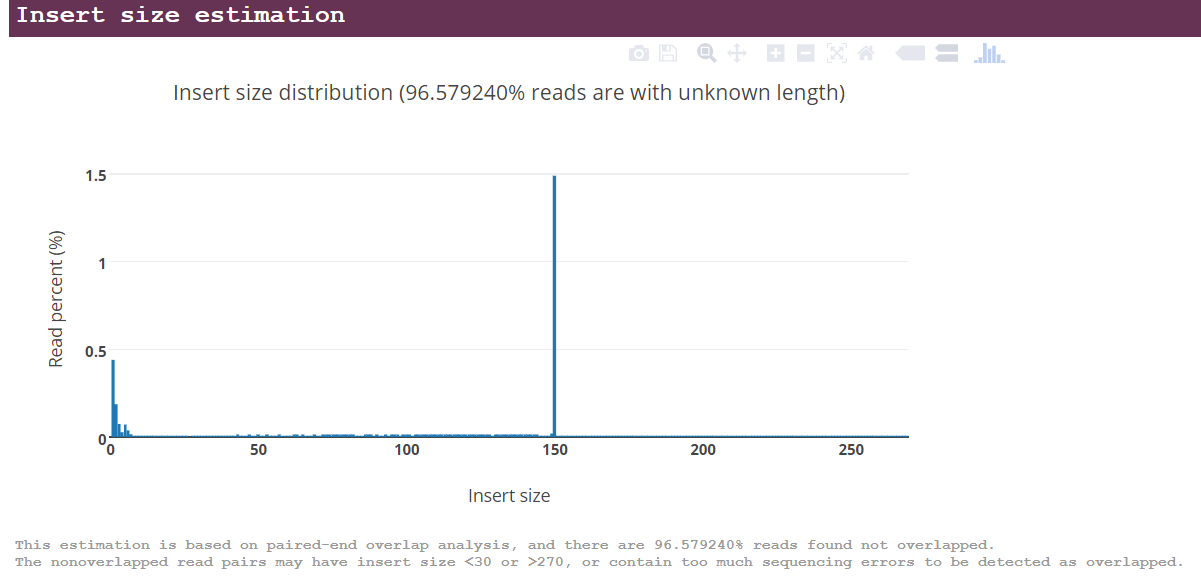
<https://blog.csdn.net/twocanis/article/details/109681242>

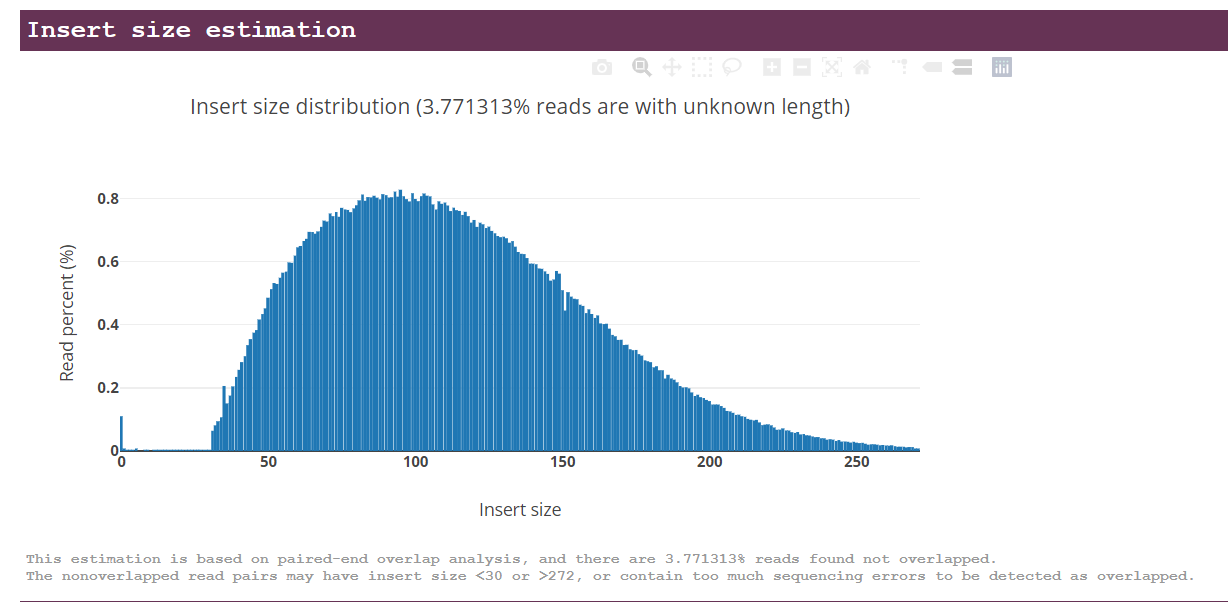
<https://blog.csdn.net/weixin_38279239/article/details/106781100>

<https://metagenome.blog.csdn.net/article/details/106552077?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-106552077-blog-109681242.pc_relevant_layerdownloadsortv1&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-106552077-blog-109681242.pc_relevant_layerdownloadsortv1&utm_relevant_index=2>

fastp质控报告

配对末端重叠分析，不同长度的Insert在reads中占的比例，相当于是DNA被打断后的长度分布。当插入片段大小<30或> 270，或包含太多错误，则不能被读取





其实就是说明，你测的reads含reads1，reads2，如果这两端的reads没有重叠，或者重叠太少，就不能预估整条插入片段的长度，因为中间的inner distance没法估算。这说明你的建的文库，被打断的片段都比较长或者太短；长的话，比如500bp，2端各测了150，中间还有200，因为这200是没法估算出来的，因此真实的插入片段长度500是不能被反推出来滴。如果是搞基因组组装，reads1，reads2有重叠可能会更好，因为可以通过重叠连成一个超级reads，进行基因组补洞。如果是转录组分析，则影响不会太大。个人理解。  
可以综合看看你的质控报告前面的summry，接头比例，q20，q30，N含量，有效数据比例（就是过滤了多少）。

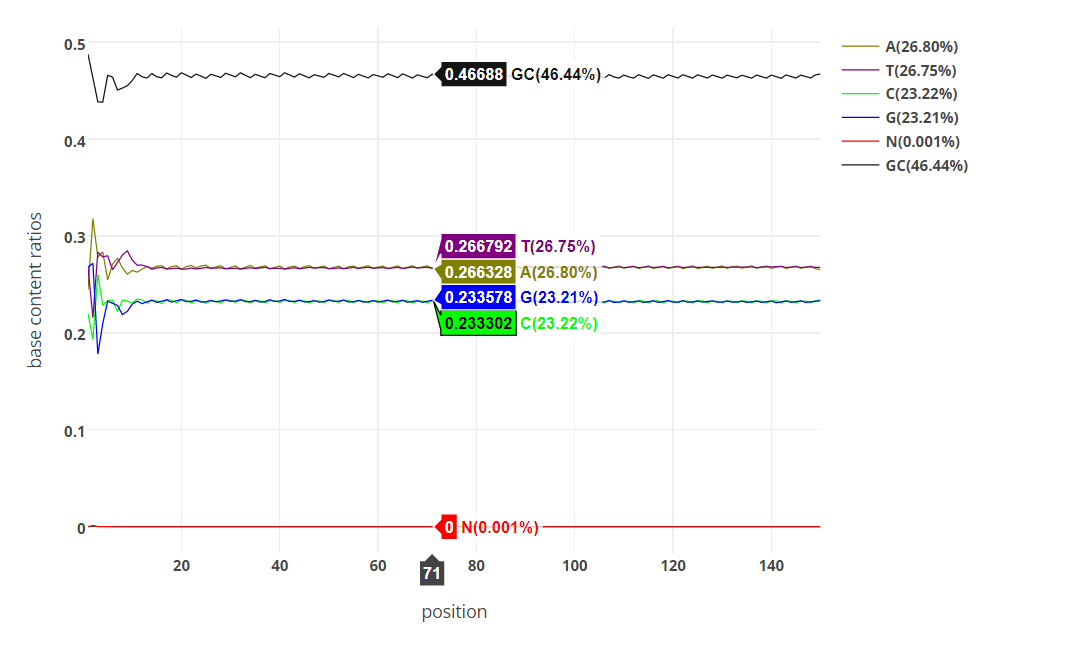
Q20、Q30一般来说，对于二代测序，最好是达到Q20的碱基要在95%以上（最差不低于90%），Q30要求大于85%（最差也不要低于80%）

碱基质量

read各个位置上碱基比例分布，这个是为了分析碱基的分离程度。何为碱基分离？已知AT配对，CG配对，假如测序过程是比较随机的话（随机意味着好），GC含量指的是G和C这两种碱基占总碱基的比例。二代测序平台或多或少都存在一定的测序偏向性，可以通过查看这个值来协助判断测序过程是否足够随机。

对于人类来说，基因组的GC含量一般在40%左右。因此，如果发现GC含量的图谱明显偏离这个值那么说明测序过程存在较高的序列偏向性，结果就是基因组中某些特定区域被反复测序的几率高于平均水平，除了覆盖度会有偏离之后，将会影响下游的变异检测和CNV分析

那么在每个位置上A和T比例应该差不多，C和G的比例也应该差不多，如上图所示，两者之间即使有偏差也不应该太大，最好平均在1%以内，如果过高，除非有合理的原因，比如某些特定的捕获测序所致，否则都需要注意是不是测序过程有什么偏差。



KMER计数  
fastp对5个碱基长度的所有组合的出现次数进行了统计，然后把它放在了一张表格中，表格的每一个元素为深背景白字，背景越深，则表示重复次数越多。这样，一眼望去，就可以发现有哪些异常的信息。鼠标可停留在某一具体组合上看出现次数和平均占比



fastp输出结果整合：<https://www.jianshu.com/p/9f3b694a4c30>

利用python脚本批量整合

fastp将raw data过滤成clean data，导入Bowtie2进行去宿主污染

conda/mamba install bowtie2

wget <http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.chromFa.tar.gz>

or

wget <http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz>

直接hg38.fa

#下载hg38参考基因组

gunzip hg38.chromFa.tar.gz

cat \*.fa > hg38.fa

conda activate

bowtie2-build /home/linux/hg38.fa hg38.fa #bowtie2 建立索并创建hg38为索引名字

#对单端序列比对，以reads\_1.fq为例

bowtie2 -x index/aaa -U reads\_1.fq -p 32 -S example.sam

##对双端序列比对

bowtie2 -x index/aaa -1 reads\_1.fq.gz -2 reads\_2.fq.gz -p 32 -S example.sam

bowtie2 -x /home/linux/index/hg38/hg38.fa -1 /media/sf\_linuxshare/SRR19023078\_1.clean.fastq.gz -2 /media/sf\_linuxshare/SRR19023078\_2.clean.fastq.gz -S SRR19023078.sam

**\*常用参数**：

-x 由bowtie2-build所生成的索引文件的前缀，需要指定路径及其共用文件名

       -1 质控后与read2配对（paired）的read1。可以为多个文件，并用逗号分开；多个文件必须和 -2中制定的文件一一对应

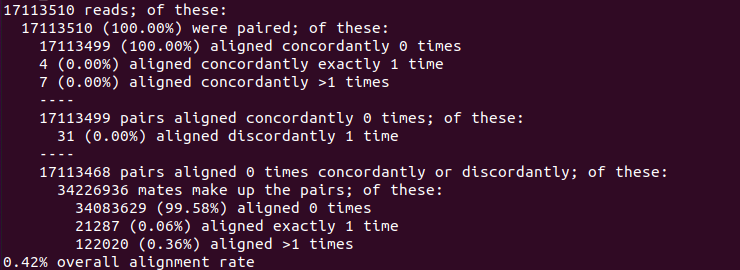
       -2 质控后与read1配对的read2

       -U 质控后未配对（unpaired）的reads

       -S 所生成的SAM格式的文件前缀。默认是输入到标准输出

       -p 线程数

结果解读：

****

检测到17113510个pairs，即17113510\*2=34227020reads

17113510个pairs都是双端

17113510个pairs都没有合理比对上

4个pairs只合理比对上一次

7个pairs合理比对大于一次

····

在17113510个没有合理比对上的pairs中：

31个pairs不合理比对上一次

在剩下17113468个pairs，即34226936个reads：

34083629个reads没有比对上

21287个reads比对上一次

122020比对大于一次

总比对率=（4\*2+7\*2+31\*2+21287+122020）/34227020=0.42%

（数据随便下的，正常样本比对率低不正常/要么别人上传就是clean data）

samtools转化bam+sort并提取unmapped

bowtie2 -x genome\_index -1 input\_1.fq -2 input\_2.fq | samtools view -bS | samtools sort > output.bam

这条命令把bowtie2 生成的sam文件通过管道|传递到samtools，将sam转换为bam文件，省去中间sam文件的空间占用

提取mapped序列信息

samtools view -bf 4 output.bam > output.mapped.bam

提取unmapped序列信息

samtools view -bF 4 output.bam >o output.unmapped.bam

#将unmapped的bam转成fastq

SE：samtools fastq output.unmapped.fq.gz > input.unmapped.bam

PE：samtools fastq -n -1 output.unmapped\_r1.fq.gz -2 output.unmapped\_r2.fq.gz > input.unmapped.bam

报错：samtools: error while loading shared libraries: libcrypto.so.1.0.0: cannot open shared object file: No such file or directory

# 找到samtools的位置

which samtools

# 进入lib

cd /root/anaconda3/lib

ls

# 建立软连接

ln -s libcrypto.so.1.1 libcrypto.so.1.0.0

samtools view -bS x.sam > x.bam

samtools view

**SAM （The Sequence Alignment / Map format）格式文件的解读**

SAM是短序列比对默认的标准格式，是以TAB为分割符的文本格式。BAM就是SAM的二进制文件，具有更小的存储空间，并且许多下游分析工具使用的是BAM格式。

SAM文件  
头部区：以’@'开始，体现了比对的一些总体信息。比如比对的SAM格式版本，比对的参考序列，比对使用的软件等。

主体区：比对结果，每一个比对结果是一行，有11个主列和一个可选列。  
第一列：QNAME，比对的序列名称，就是fq文件中的read ID，是一条测序read的名称。  
第二列：FLAG，比对上的情况  
第三列：染色体名称  
第四列：POS，比对上的最左面的定位  
第五列：MAPQ，比对的质量值。越高说明比对的越唯一，最高60  
第六列：CIGAR Extended CIGAR string，M表示匹配、I表示插入、D表示删除、N表示内含子和D类似、S表示替换、H表示剪切。87M表示87个碱基在比对时完全匹配。  
第七列：MRNM，这条reads第二次比对的位置，是比对上的参考序列名 。=表示参考序列与reads一模一样，\*表示没有完全一模一样的参考序列。  
第八列：MPOS，与该reads对应的mate pair reads的比对位置（即mate），若无mate,则为0。  
第九列：ISIZE 插入片段长度 例如：200。如果R1端的read和R2端的read能够mapping到同一条Reference序列上（即第三列RNAME相同），则该列的值表示第8列减去第4列加上第6列的值，  
第十列：SEQ，和参考序列在同一个链上比对的序列，即read的序列。  
第十一列：比对序列的质量（ASCII-33=Phred base quality）reads碱基质量值 例如：-8CCCGFCCCF7@E-

kneaddata可以调用Trimmomatic序列质控，bowtie2比对至对应数据库基因组去除宿主等序列。

当样本取自人类粪便样本时一般会包括人类基因组的序列以及微生物的序列，所以需要去除这些人类基因组的污染序列，否则会对后续的分析造成影响，采用Bowtie2将质控后的数据与人类参考基因组进行比对，可以区分出人类和微生物的序列，从而进一步剔除这些污染序列。

## 构建虚拟环境，安装kneaddata

conda/mamba create -n kneaddata

conda activate kneaddata

conda/mamba install -c biobakery kneaddata

#下载数据库

mkdir kneaddata\_database

cd kneaddata\_database/

kneaddata\_database --download human\_genome bowtie2 ./

kneaddata\_database --download mouse\_C57BL bowtie2 ./

kneaddata\_database --download human\_transcriptome bowtie2 ./

kneaddata\_database --download ribosomal\_RNA bowtie2 ./

个人建议手动网页下载

human\_genome : bowtie2 = <http://huttenhower.sph.harvard.edu/kneadData_databases/Homo_sapiens_hg37_and_human_contamination_Bowtie2_v0.1.tar.gz>

创建、解压  
mkdir human\_genome  
tar -zxvf Homo\_sapiens\_hg37\_and\_human\_contamination\_Bowtie2\_v0.1.tar.gz -c human\_genome

*#设置数据库地址(请修改为绝对路径）*

human\_genome\_db = ./kneaddata\_db/human\_genome

#运行bowtie2去除人体DNA污染，运行trimmomatic默认参数进行序列质控

kneaddata -i XXX\_R1.fq.gz \

-i XXX\_R2.fq.gz \

-o 01\_clean\_data/output\_dir \

-t 50 -p 50 -db $human\_genome\_db

## 分别解压数据库文件到自定义目录中

mkdir kneaddata\_db\_DATABASE\_NAME

tar -zxvf DATABASE\_NAME.tar.gz -C ./kneaddata\_db\_DATABASE\_NAME/

## 指定数据库，可以全列出来，也可以用到某个就写某个

KNEADDATA\_DB\_MOUSE\_C57BL\_6NJ=/home/dengysh/anaconda3/envs/kneaddata/kneaddata\_database/kneaddata\_db\_mouse\_C57BL\_6NJ/mouse\_C57BL\_6NJ

KNEADDATA\_DB\_HUMAN\_GENOME=/home/dengysh/anaconda3/envs/kneaddata/kneaddata\_database/kneaddata\_db\_human\_genome/Homo\_sapiens

KNEADDATA\_DB\_HUMAN\_TRANSCRIPTOME=/home/dengysh/anaconda3/envs/kneaddata/kneaddata\_database/kneaddata\_db\_human\_transcriptome/human\_hg38\_refMrna

KNEADDATA\_DB\_RIBOSOMAL\_RNA=/home/dengysh/anaconda3/envs/kneaddata/kneaddata\_database/kneaddata\_db\_ribosomal\_rna/SILVA\_128\_LSUParc\_SSUParc\_ribosomal\_RNA

KNEADDATA\_DB\_RNOR\_6=/home/dengysh/anaconda3/envs/kneaddata/kneaddata\_database/kneaddata\_db\_Rnor\_6/Rnor\_6

## 单端数据

kneaddata -i read-1.fastq.gz -o ./result-1 -t 20 -p 20 -db $KNEADDATA\_DB\_HUMAN\_GENOME

## 双端数据

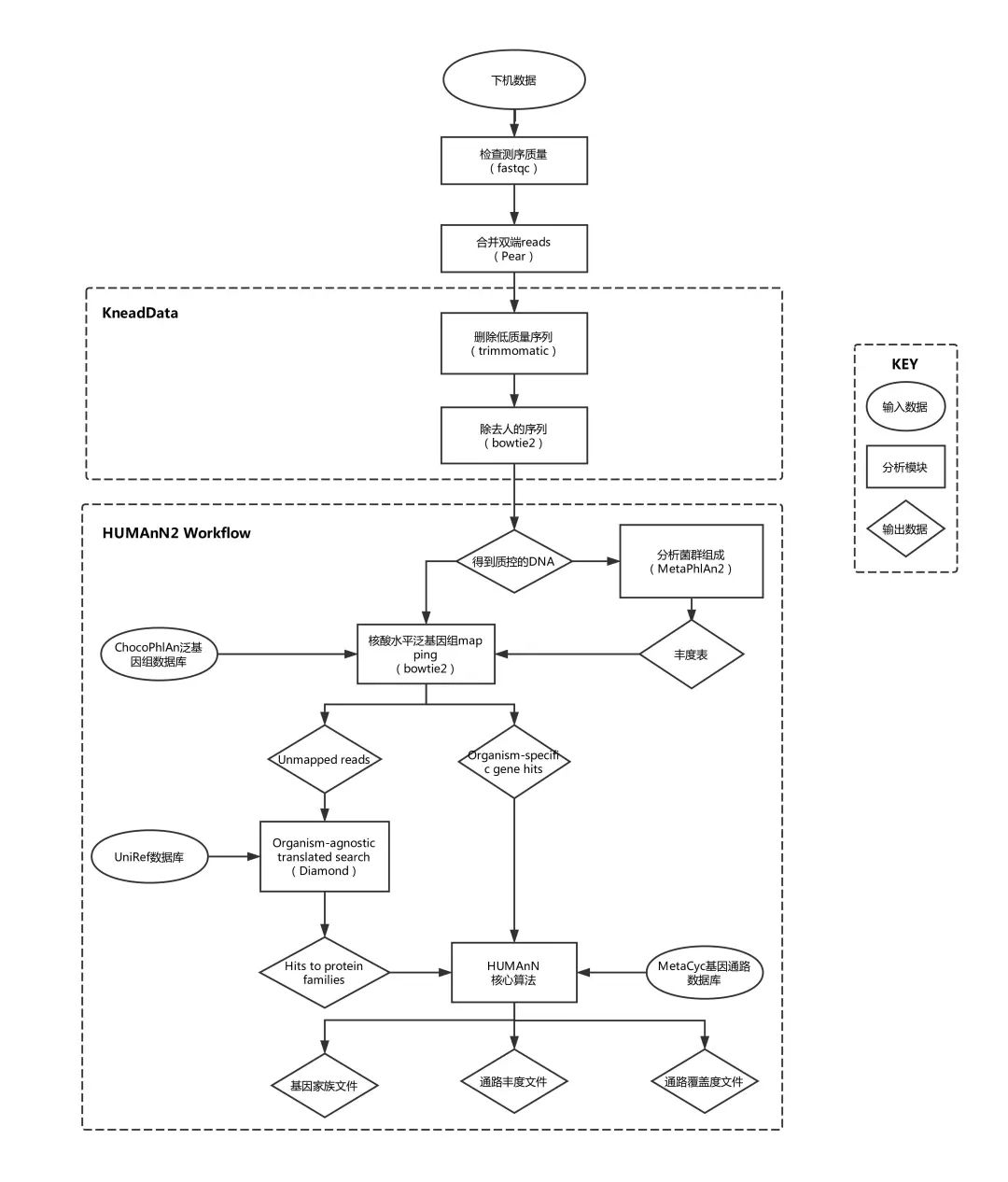
## 如果是使用多个数据库进行质控，建议添加‘--serial’参数，会将多个数据库的输入输出串联起来，同时建议添加‘--cat-final-output’参数，合并最终输出文件，便于后续分析

kneaddata -i 36\_1.out.fq.gz -i 36\_2.out.fq.gz -o ./kneaddata\_36 --output-prefix 36 -t 20 -p 20 -db /work\_directory/work\_result/kneaddata\_database/kneaddata\_db\_DATABASE\_NAME --trimmomatic /soft\_directory/Software/Trimmomatic-0.39

质控后结果统计

跑完质控后，还可以对跑完后的数据记性质控过程统计。使用kneaddata\_read\_count\_table功能，输入文件是输出目录中的log文件，里面记录了质控过程。如果是多个独立样本，可以将多个样本的log文件汇总在同一个目录下，对目录下所有log文件进行汇总统计。

kneaddata\_read\_count\_table --input log\_file/ --output kneaddata\_read\_count\_table.tsv



对于质量控制后的高质量序列，使用MetaPhlAn和HUMAnN软件进行物种组成和代谢通路注释，从而获得肠道菌群物种和代谢通路的组成及相对丰度信息表

metaplan3跟humann3 python（3.7）环境要一致，不然会运行报错

mataphlan3跟humann3数据库下载：

<https://blog.csdn.net/weixin_44223966/article/details/106875039>

<https://blog.csdn.net/zlabx/article/details/107226790>

(metaphlan直接导出的是相对丰度表)

metaphlan4

conda/mamba create -n mpa python=3.7

conda activate mpa

metaphlan4数据库：<http://cmprod1.cibio.unitn.it/biobakery4/metaphlan_databases/>（建议翻墙下载，或者直接找我百度云下吧）

bzip2 mpa\_vJan21\_CHOCOPhlAnSGB\_202103\_SGB.fna.bz2

#构建index

bowtie2-build --threads 30 -f mpa\_vJan21\_CHOCOPhlAnSGB\_202103\_SGB.fna mpa\_vJan21\_CHOCOPhlAnSGB\_202103

metaphlan input.fastq --input\_type fastq -o xx.txt --bowtie2db mpa\_vJan21\_CHOCOPhlAnSGB\_202103

#更换index路径

metaphlan --index ./mpa\_vJan21\_CHOCOPhlAnSGB\_202103

#没有index metaphlan会自动构建index，把数据库放到对应目录，构建index挺费时间的，加线程就完了

kraken2据说是比metaphlan更加敏感、比对速度更快

kraken结合bracken一起比对精准度更高（kraken2官网说的）

conda activate xx

conda/mamba install -c bioconda kraken2 -y

conda/mamba install bracken -y

数据库: https://benlangmead.github.io/aws-indexes/k2

国内就找到国家微生物科学数据中心的，目前是下不到的，觉得数据库大也去k2可以下minikraken，我是翻墙下标准库3（包含archaea, bacteria, viral, plasmid, human1, UniVec\_Core），也可以自己定义自己所需的库，解压即用，默认建好50, 75, 100, 150, 200, 250 and 300-mers，自己构建库需要去下ncbi的taxonomy分类数据库

#构建index

kraken2-build  --threads 20 --db ./kraken2\_db

bracken-build -d ./kraken2\_db -t20 -k 35 -l 150

-k，k-mer长度，Kraken1默认为31，Kraken2默认为32；  
-l，reads读长。

#比对

#使用Kraken2分析，不使用--use-mpa-style模式

mkdir result

kraken2 --db ~/kraken2\_db --threads 20 --report ./result/xx.report --output ./result/xx.output --paired xx.f1.clean.fq.gz xx.r2.clean.fq.gz

#数据：质控好的PE150（如果为其它长度，Bracken数据库索引的长度也要改变）数据

#使用Braken校正

bracken -d ~/kraken2\_db -i ./result/xx.report -o ./result/xx.output.bracken -w ./result/xx.output.bracken.report -r 150 -l S

for a in P C O F G S;do

bracken -d ~/kraken2\_db -i ./result/xx.report -o ./result/xx.output.bracken -w ./result/xx.output.bracken.report -r 150 -l $a(可以选你想要的物种分类水平)

done

-t , 阈值（没怎么研究）

-o，Bracken输出文件（校正详情）文件名；

-w，Bracken计算后的新报告（每个物种的reads数目）文件名；

-r， reads长度；

-l，分类水平（D,P,C,O,F,G,S）。

# 第三步，将braken的report格式转换成--use-mpa-style格式

# <https://github.com/jenniferlu717/KrakenTools/#kreport2mpapy>

github上面有py脚本

python kreport2mpa.py -r ./result/xx.output.bracken.report -o ./result/xx.txt --percentages --no-intermediate-ranks#默认default

结果为绝对丰度，若想要相对丰度，可在kreport2mpa.py中添加:--percentages

最近更新的krakentools可能会报错

name ‘l\_vals’is not defined

去下旧版本的py就好了

python combine\_mpa.py

-i/--input MYFILE1.MPA MYFILE2.MPA.......多个 MPA 样式文本文件（用空格分隔）

-o/--output MYFILE.COMBINED.MPA..........输出 MPA 样式文本文件

（整合所有样本的丰度表）

可以直接用那些py脚本算alpha、beta多样性，但没抽平，应该用的是相对丰度（不推荐）

python alpha\_diversity.py -i xxx -f xx.bracken -a Sh

-f 输出的文件

Sh......Shannon's alpha diversity

BP.....Berger-Parker's alpha

Si.....Simpson's diversity

ISi.....Inverse Simpson's diversity

F.......Fisher's index

python beta\_diversity.py 默认计算Bray-Curtis

humann3安装

conda activate humann3

conda/mamba install humann -y

三大数据库:微生物泛基因chocophlan、蛋白功能数据库uniref、注释数据库utility\_mapping

chocophlan：http://huttenhower.sph.harvard.edu/humann\_data/chocophlan/full\_chocophlan.v201901\_v31.tar.gz

uniref：http://huttenhower.sph.harvard.edu/humann\_data/uniprot/uniref\_annotated/uniref90\_annotated\_v220190b\_full.tar.gz

utility\_mapping：<http://huttenhower.sph.harvard.edu/humann_data/full_mapping_v201901b.tar.gz>

#上传数据库

mkdir chocophlan.v201901\_v31

mkdir uniref90\_annotated\_v201901b\_full

mkdir full\_mapping\_v201901b

tar -zxvf full\_chocophlan.v201901\_v31.tar.gz -C ./ chocophlan.v201901\_v31/

tar -zxvf uniref90\_annotated\_v220190b\_full.tar.gz -C ./uniref90\_annotated\_v201901b\_full/

tar -zxvf full\_mapping\_v201901b.tar.gz -C ./ full\_mapping\_v201901b /

#更新database路径

humann\_config --update database\_folders nucleotide ./ chocophlan.v201901\_v31/

humann\_config --update database\_folders protein ./ uniref90\_annotated\_v201901b\_full /

humann\_config --update database\_folders utility\_mapping ./ full\_mapping\_v201901b /

## 更新后查看设置

humann\_config

正常情况下用UniRef90，如果在UniRef90 注释结果不佳，也就UniRef90\_unknown 丰度很高的时候，再换为UniRef50.

## 运行必须参数，输入文件及输出路径。

## 输入文件可以为多种数据形式，包括测序文件及压缩格（fasta, fastq, fasta.gz, fastq,gz) , 比对文件（sam, bam等），还有gene table文件也可以。（不支持双端）

## 输出指定路径，运行后一般输出genefamilies，pathcoverage，pathabundance三个文件和比对过程中的中间文件（包括metaphlan的结果）

humann -i xx -o xx --threads 20#加线程得看内存够不够，跑不出来就默认1线程

--memory-use maximum#加内存跑，默认最小

--remove-temp-output  remove temp output files #是否删除中间文件, [默认: 不删除]

我的humann改不了metaphlan\_database路径

./metaphlan/metaphlan\_databases是默认路径，没有index会自己下，那就copy过去

第一次跑目录下要有md5、mpa\_latest、pkl、vsg.fna.bz2、marker\_info.txt.bz2、SGB\_fna 还有index

#标准化之前用 humann\_regroup\_table 将基因家族重新分组为不同的功能类别。

humann\_regroup\_table --input $TABLE --groups $GROUPS --output $TABLE2

参数说明

--input $TABLE #输入基因丰度文件，$SAMPLE\_genefamilies.tsv

--groups $GROUPS #对应数据库ID 可选数据库

{uniref90\_rxn,uniref50\_rxn,uniref50\_go,uniref90\_go,uniref50\_ko,uniref90\_ko,uniref50\_level4ec,uniref90\_level4ec,uniref50\_pfam,uniref90\_pfam,uniref50\_eggnog,uniref90\_eggnog}

--output $TABLE2 #输出对应数据库基因丰度文件

--custom CUSTOM #Custom groups file (.tsv or .tsv.gz format)

**标准化丰度输出文件（humann\_renorm\_table ）**

humann\_renorm\_table --input xx\_genefamilies.tsv --output xx\_genefamilies\_relab.tsv \--units relab

--input $TABLE=基因/通路表（tsv格式）

--units $CHOICE="relab"（relative abundance）或"cpm"（copies per million）

--output $TABLE2= 归一化后的基因/途径表

HUMAnN 3.0 以RPK为单位（每千碱基读数）定量基因和途径。下游分析之前大家还会重新归一化（官网说的）。“ relab”（相对丰度）加和为1， “ cpm”（百万份数）加和为100万，对于具有许多功能的表，如genefamilies.tsv，"cpm"通常更便捷。

默认是所有分层级别标准化为所有功能总和，即包括UNMAPPED, UNINTEGRATED, and UNGROUPED，不过可以通过--special 和 --mode 指定删除后再标准化。

**合并多样品结果（humann\_join\_tables）**

humann\_join\_tables --input hmp\_subset --output humann\_genefamilies.tsv --file\_name genefamilies\_relab

humann\_join\_tables --input hmp\_subset --output humann\_pathcoverage.tsv --file\_name pathcoverage

humann\_join\_tables --input hmp\_subset --output humann\_pathabundance.tsv --file\_name pathabundance\_relab

--input 合并文件所在目录

--output 输出文件名

--file\_name 仅合并含有这个字符的文件

（1）**SOAPdenovo：**这款软件由华大开发，SOAPdenovo2是用于short-read组装的软件，主要用于组装比较大的基因组， 组装速度快但是错误率较高。

（2）**SPAdes：**metaSPAdes是目前宏基因组领域组装指标较好的软件，尤其在株水平组装优势明显，组装效果优，但是拼接时间长，资源消耗高

（3）**IDBA：**适合预测深度不均一的数据，且资源消耗过高

（4）**Megahit：**MEGAHIT是NGS de novo汇编程序，在土壤等复杂环境样本组装、大量样本混合组装方面优势明显，速度很快，消耗的资源少。

megahit序列组装

分成多重混合拼接（样本量太大的话，可能组装失败）和单拼

单拼的结果（每一个样本单独拼接）

N50、N90值越大越好

**多重混合拼接（Multiple\_Megahit或Multiple\_IDBA\_UD）**

利用IDBA-UD[1]/Megahit[2]（https://github.com/voutcn/megahit）与Newbler (https://ngs.csr.uky.edu/Newbler)最大化利用测序数据，进行多重混合拼接组装。并获得各样品的组装结果统计表与组装结果contigs长度分布图。

推荐指数：Multiple\_Megahit ★★★★★ 推荐理由：reads利用率高，拼接速度快

                 Multiple\_IDBA\_UD★★ 推荐理由：reads利用率高，但拼接耗时非常长

**单拼Megahit**

利用Megahit（https://github.com/voutcn/megahit）软件针对不同测序深度序列进行组装。采用succinct de Bruijn graph方法，拼接参数从小k-mer开始迭代拼接至大k-mer，可以更快速的完成拼接，并且拼接结果contigs数量多、reads mapping率高，适用于测序数据量大、物种类型复杂的宏基因组样本拼接。可获得各样品的组装结果统计表与组装结果contigs长度分布图。

推荐指数：Megahit★★★★ 推荐理由：拼接速度非常快

conda install -c bioconda megahit

SE: megahit -r input\_f.fq.gz -o output\_megahit

PE: megahit -1 input\_f1.fq.gz -2 input\_r2.fq.gz -o output\_megahit

# 交错的双端序列：

megahit --12 interleaved.fq -o out

final.contigs.fa: 组装结果，fasta格式  
log: megahit程序运行时的log日志，便于查看具体执行进度和排错  
options.json: 执行组装程序参数的json格式  
intermediate\_contigs: 中间组装结果

常用参数

两种设置kmer参数的方法：

1）–k-list：组装的kmer size列表，支持多kmer组装，不同kmer size之间逗号分隔，可设置的范围15-255，相邻kmer size间隔必须小于或等于28，默认为21,29,39,59,79,99,119,141

2）组合参数：–k-min：设置最小的kmer size，应小于255，必须为奇数，默认为21

–k-max：设置最大的kmer size，应小于255，必须为奇数，默认为141

–k-step：多kmer组装的kmer size间隔，应小于等于28必须为偶数，默认为12

其他常用参数

1）-m/–memory：构建SdBG可以使用的最大内存，可设置0-1，也即占总内存的分数，默认为0.9

2）-t/–num-cpu-threads：程序运行使用的核数

3）–out-prefix：输出结果文件的前缀，例如contig文件会是OUT\_DIR/OUT\_PREFIX.contigs.fa

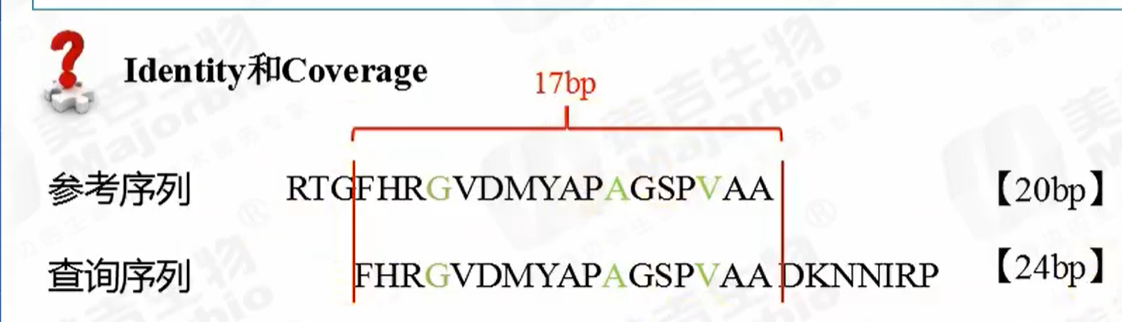
4）–min-contig-len输出的最短contigs，默认为200

https://blog.csdn.net/qq\_42491125/article/details/109843781

非冗余基因构建

CD-HIT（去冗余）进行聚类（阈值90%identity、90%coverage）类似构建97%的OTUs

每个类取最长的氨基酸序列作为序列，构建非冗余氨基酸集



根据参考序列跟查询序列比对

identity序列一致性（比对碱基的相似性）

14/17=82.35

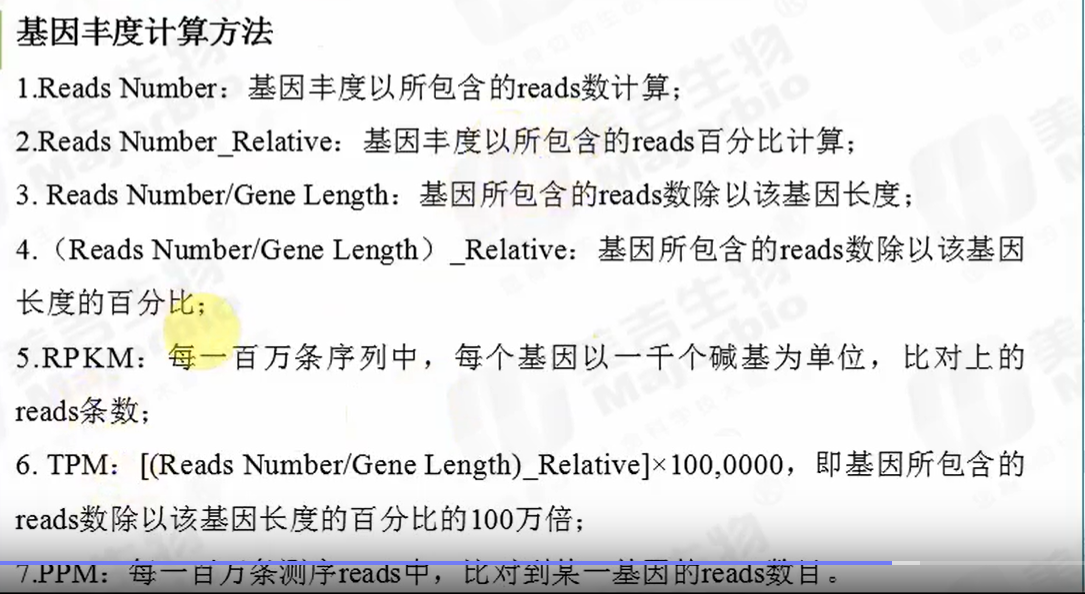
coverage序列覆盖度（比对参考序列与查询序列的比值）

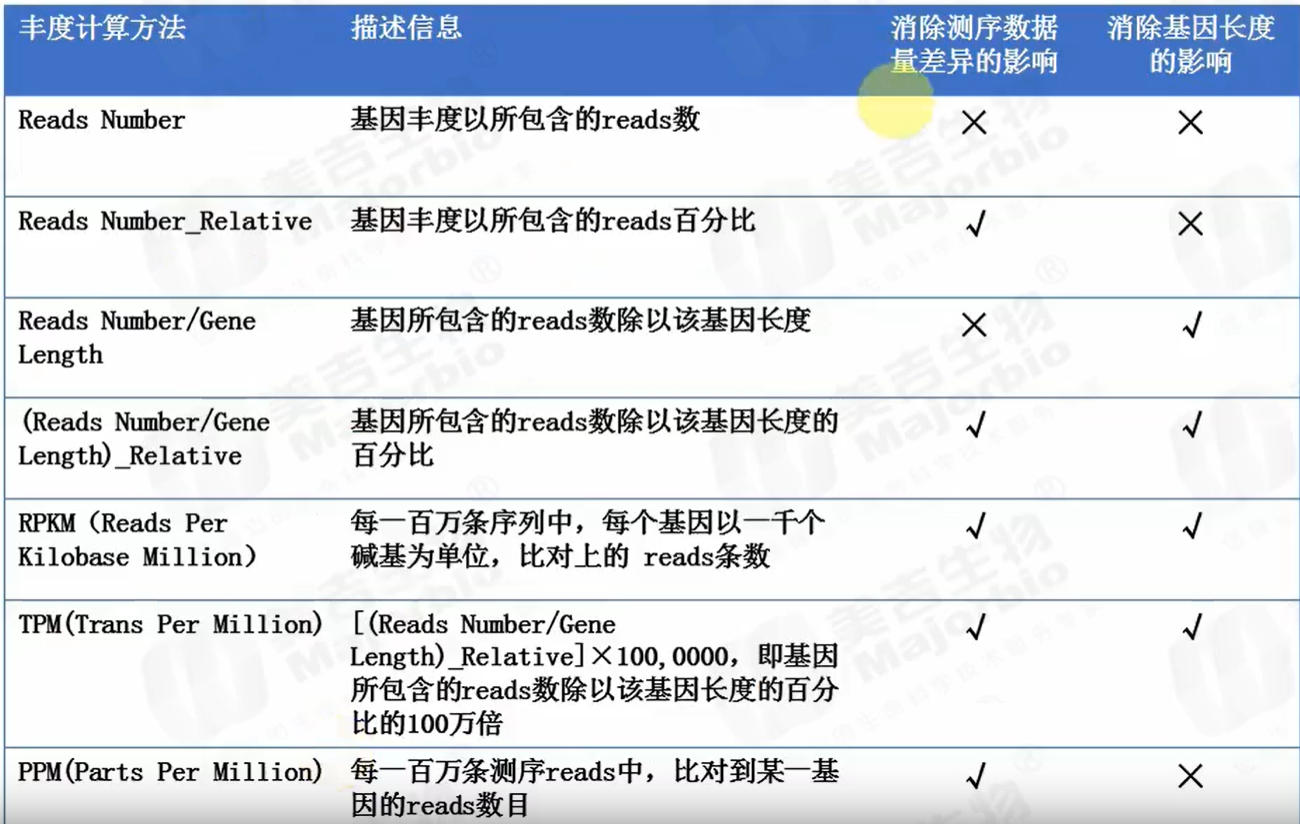
17/24=70.83

基因丰度计算

SOAPligner

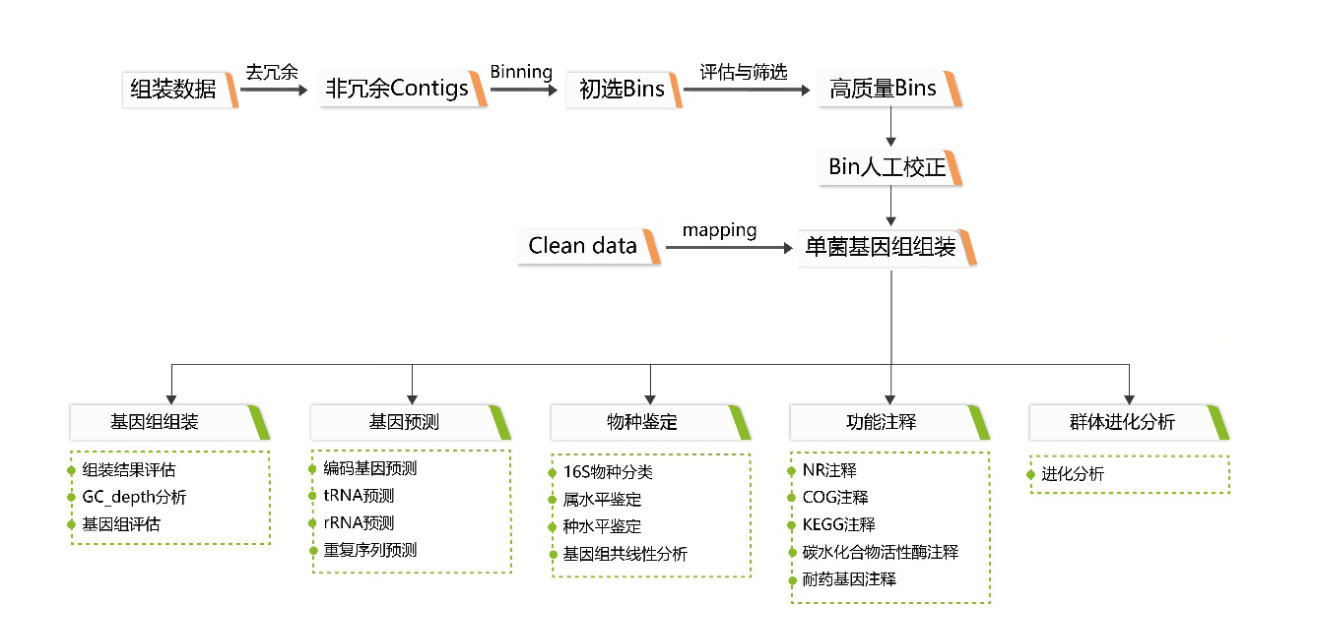
使用SOAPligner分别对每个样品的高质量reads与非冗余基因集进行比对（95%identity），统计基因在对应样品的丰度信息





物种、功能注释

KEGG基因功能注释



<https://blog.csdn.net/woodcorpse/article/details/118124686>

binning

从微生物群体序列中将不同个体的序列分离，将来自同一菌株的序列聚类，得到单菌的基因组信息

基于congtigs binning

组装data-去冗余-得到contigs-binning-初选bins（Metabat2，默认1000bp contigs长度）-通过评估与筛选（CheckM评估完整度（>50%）跟污染度）-高质量bins（可以再人工校正得到bins）-进行基因组组装（SOAPdenovo或SPAdes）

初选bins可以通过AMPHORA2进行物种注释

fastp质控

samtools 数据统计处理

picard 去除PCR重复序列

cd-hit 去冗余

Metabat2 binning

AMPHORA2 binning结果注释

DAS\_Tool binning结果合并

CheckM bins质量评估（古菌）

Busco bins质量评估（细菌）

SOAPdenovo 基因组装（测序深度大于等于60）

SPAdes（测序深度小于60）

**Binning组装**

首先对宏基因组拼接获得的contigs进行长度筛选，然后基于筛选后的contigs，使用MetaBAT[1]（https://bitbucket.org/berkeleylab/metabat, version 2.12.1）、CONCOCT[2]（https://github.com/BinPro/CONCOCT，version 0.5.0）或Maxbin2[3]（https://sourceforge.net/projects/maxbin/，version 2.2.5）进行binning组装；MetaBAT、CONCOCT和Maxbin2都是利用核酸组成和丰度变化信息（Nucleotide composition and abundance，NCA）综合计算距离矩阵，对contigs进行binning组装，产生用于重构环境样本中的微生物草图基因组的bins。

DAS\_tools[4]（https://github.com/cmks/DAS\_Tool，version 1.1.0 ）：可整合多款软件组装的bins结果，获得一个最佳的、非冗余bins集合。

**Bins评估**

利用CheckM[5]（https://github.com/Ecogenomics/CheckM/wiki）软件对获得的bins进行质量评估。CheckM工作原理：首先通过HMM识别bins序列中的marker genes，并利用HMMER（version 3.1.2，http://hmmer.janelia.org）与通用单拷贝基因进行比对，通过pplacer将bins置入参考进化树；确定特异的谱系marker set，对bins进行质量评估：

1）完整度（completeness）：计算该bin检测到的特异谱系marker gene占总marker gene的比例，得到完整度；

2）污染度（contamination）：计算检测到>1次的marker genes占该bin检测到的marker genes的比例，得到污染度；

3）异质性 （strain heterogeneity）：统计污染的marker genes相互间的AAI(Amino Acid Identity，氨基酸相似性)≥0.9的比例，得到异质性指标；数值越大，表明异质性越低。

**Bins物种注释**

利用AMPHORA2[6]（https://github.com/martinwu/AMPHORA2）软件对bin进行物种注释。

AMPHORA2物种注释的原理：通过HMMER3鉴定的marker gene序列，进行多序列比对筛选后，用RAxML对得到的marke genes进行进化分析，将bins内contigs在进化树中的位置的最小祖先作为bins的物种注释结果。

**Bin人工校正**

通过每个Bin中Contig的GC含量、覆盖度和四核苷酸频率（TNF）对Bin进行污染度评估。根据评估结果筛选保留较优的Contig并人工删除污染序列，对Bin进行人工校正。

**序列查找：**

输入关注的核苷酸或氨基酸序列，与宏基因组binning组装数据进行比对，获得该条参考序列比对上的Contig相关信息，包括参考序列与Contig的比对一致性，以及比对上的序列在Contig上的起止位置等信息，为后续分析提供理论基础。

**Bin人工校正：**

通过每个Bin中Contig的GC含量、测序深度和四核苷酸频率（TNF）对Bin进行污染度评估。根据评估结果从Bin筛选保留较优的Bin中的Contigs，并人工删除污染序列，完成对Bin进行人工校正，并重构新的Bin。

**重构Bin评估：**

利用CheckM软件对人工校正后的重构Bins进行质量评估，以确保其完整度并可进行后续的单菌基因组组装。

**基因组分析**

在此模块可选择完整度＞50%的bin进行分析，可选择是否进行重组装，若选择是，则会对选择的bin进行基因组组装再进行后续分析，选择否，则直接利用bin进行后续分析。

**基因组组装**

得到binning组装结果后，可以对指定的Bin进行基因组组装。

基因组组装步骤是将双端clean reads分别比对到该bin所包含的contigs上，合并所有样品中能够比对到这些contigs的clean reads并进行去冗余处理，然后使用常用的基因组组装软件SOAPdenovo[1]或SPAdes[2]将这些reads进行组装。

在组装前需要计算该bin的测序深度（所有比对上的reads碱基长度总和/bin包含的碱基数）。如果测序深度≥60X，则使用SOAPdenovo软件进行组装，如果测序深度＜60X，则使用SPAdes软件进行组装。

基因组组装完成后，使用Busco（细菌基因组）或者CheckM（古菌基因组）对组装结果进行评估。

**基因预测**

利用软件对基因组（bin或者重组装后的G\_bin）进行de novo基因预测，分别得到编码基因、tRNA及rRNA基因信息。

**编码基因预测：**

利用Glimmer [3]软件（http://ccb.jhu.edu/software/glimmer/index.shtml）对基因组中的编码序列（CDS）进行预测。通过预测获得功能基因的核酸序列和氨基酸序列，用于后续功能和系统进化分析。

**tRNA预测：**

利用 tRNAscan-SE v2.0[4]软件（http://trna.ucsc.edu/software/）对基因组中包含tRNA 进行预测，可以获得每个样本基因组中tRNA的核苷酸序列信息，反密码子信息及二级结构信息。

**rRNA预测：**

利用 Barrnap软件（https://github.com/tseemann/barrnap） 对基因组中包含的 rRNA 进行预测，获得每个样本基因组中所有rRNA的种类、位置、序列信息。

**重复序列预测**

        本分析主要针对基因组中的串联重复序列进行预测。串联重复序列是指在基因组邻近位置上出现两段或多段重复序列。使用Tandem Repeats Finder软件[5]进行串联重复序列预测，可以得到重复序列的位置、重复次数、核酸组成等信息。

**基因组鉴定**

**16S物种分类**

将基因组（bin或者重组装后的G\_bin）中预测得到的16S rRNA序列与16S数据库（如Silva，GreenGene，NT等，在云平台中可自行选择比对数据库）进行比对，获得物种注释信息。该分析会按照Identity排序，给出比对一致度TOP10的物种注释结果。

SILVA是一个rRNA基因序列的综合数据库，收录原核和真核微生物的小亚基rRNA基因序列（简称SSU，即16S和18SrRNA）和大亚基rRNA基因序列（简称LSU，即23S和28SrRNA）。该数据库由于更新比较及时，因此也是目前最常选用的参考数据库之一。由于是最大最全的数据库，其缺点是假阳性会更高。

Greengene数据库（http://greengenes.secondgenome.com/）是针对细菌和古菌16S rRNA基因的数据库，该数据库更新较慢，最新版本为2013年5月更新。由于是人工整理，比较准确。很多科研工作者依然选择使用该数据库。分类层级采用常用的七级界门纲目科属种，方便理解和阅读。同时，QIIME软件默认物种注释数据库也是它。PICRUST分析工具，即根据16S rRNA高通量测序结果预测微生物群落功能的分析，也是基于Greengne数据库开发的。

NT(Nucleotide Sequence Database)，核酸序列数据库，是NR库的子集。

**POCP分析**

如果基因组注释到科水平，可以与参考物种进行属级别的POCP[1] ( percentage of conserved proteins ) 分析。综合基因组调查表明，POCP可以作为原核生物划分属边界的重要指标。理论上，POCP值可以在0％至100％之间变化，这取决于两个基因组的蛋白质含量的相似性。但基本上，属于同一属的两个物种至少一半的蛋白是相同的。所以当POCP值低于50%时，一般认为是新属。POCP值具体计算公式如下：

POCP(%)=（C1+C2）/(T1+T2)

其中，C1和C2分别代表被比较的两个基因组中保守蛋白的数量，T1和T2分别代表被比较的两个基因组中蛋白的总数量。

**ANI分析**

如果基因组注释到属级别，可以尝试进行细化到种级别的ANI[2]（average nucleotide identity）分析。ANI分析即平均核苷酸一致性分析，是基于物种全基因组序列，通过分析比较同源基因序列来判定物种间的遗传关联性的重要参数，可以直观表示物种间的亲缘关系远近。

ANI 的计算利用了大量的基因，与16S rRNA 相比，其亲缘关系计算中并不受单个基因或者少数基因变化的遗传速率和基因水平转移影响，即使物种基因组中很大一部分基因发生不同进化历史，比如某些基因的进化快于基因组的速率，而某些基因的进化慢于基因组速率，其对物种 ANI 的影响也是很小几乎可以忽略。当两菌间的 ANI≤95% （等同于 DDH<70%）时，认为它们属于不同种。

**基因组共线性分析**

共线性一般是指不同物种中染色体之间的某些区域有着相似的基因排列顺序。在进化过程中，影响基因共线性的因素有染色体重组，基因转座等。一般而言，进化距离越远的物种之间基因共线性越差，因此两个物种之间的共线性程度可以作为衡量它们之间进化距离的标尺。同时，我们还可以通过共线性分析来获得物种间基因组在进化过程中所发生结构变异的情况。利用Sibelia进行共线性分析，再通过Circos软件绘制共线性圈图。