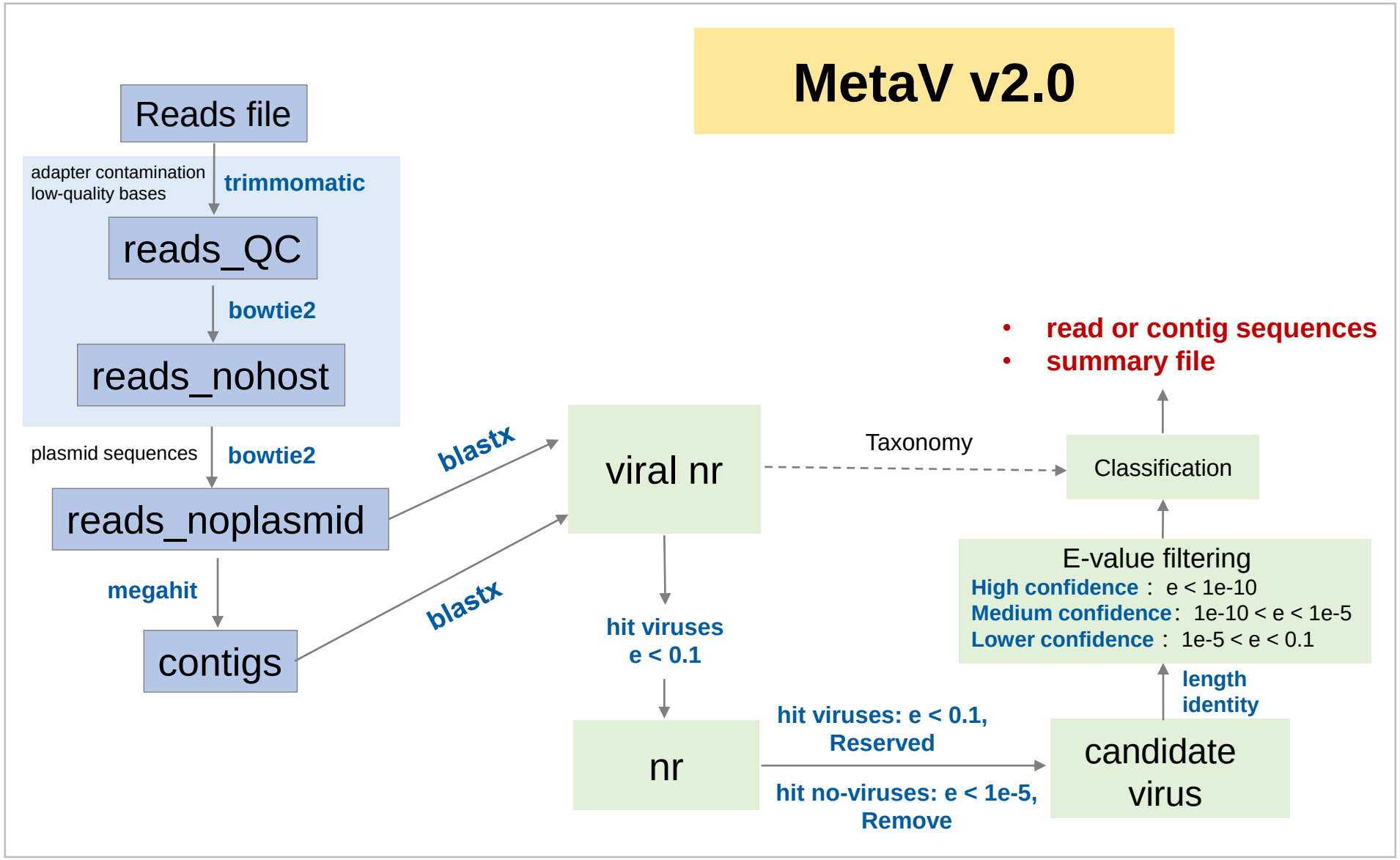
宏基因组病毒检测管道
MetaV v2.0 使用介绍

Zhi-Jian Zhou

2025.12.14



Note: all E-value thresholds in the figure can be adjusted in the profiles.xml file and input parameters of metav.

MetaV v2.0.0流程解读

MetaV管道，基于串联现有的工具，从宏基因组二代测序数据中鉴定病毒序列。

主要步骤：

1. **去除接头引物污染、低质量碱基片段，软件trimmomatic**
2. **去除宿主污染，软件bowtie2**
3. **去除质粒污染，软件bowtie2**
4. **采用两条独立的子管道（diamond软件）**
   - **管道1（reads blastx）：**先将reads比对到病毒nr库，初步筛选出命中病毒nr的reads（会有来自非病毒序列的非特异性命中，即假阳性），再将这些reads与更大的nr库进行比对，进一步剔除非病毒的成分。
   - **管道2（contigs blastx）：**先将reads拼接为contigs，将contigs比对到病毒nr库，初步筛选出命中病毒nr的contigs（会有来自非病毒序列的非特异性命中，即假阳性），再将这些contigs与更大的nr库进行比对，进一步剔除非病毒的成分。

5. 假阴性和假阳性

假阴性是指病毒序列被误删，例如因筛选条件太严苛导致一些和现有病毒同源性低的新病毒被剔除掉。假阳性是指非病毒序列被误识别为病毒序列，例如筛选条件过于宽松时，非病毒序列可能因非特异性匹配而被误识别为病毒序列。

为了兼顾假阴性和假阳性，MetaV v2.0.0采用一套组合策略鉴定病毒：

- （i）在比对病毒nr库时，使用宽松的E值阈值（默认值0.1，**可调整**），尽可能保留潜在的病毒reads/contigs。
- （ii）再将上一步得到的reads/contigs比对到nr库，如果最佳命中依旧为病毒蛋白且E值低于0.1（可调整）则保留；如果最佳命中为非病毒蛋白且E值低于1×10^{-5}（**可调整**）则剔除，反之则保留。
- （iii）将上一步得到的候选病毒reads/contigs，在限定长度和同一性值下（**可调整**），按照三个E值区间（分隔点为1×10^{-10}、1×10^{-5}、0.1；**可调整**）进行分级提取，包括**E值<1×10^{-10}、1×10^{-10} < E值 < 1×10^{-5}、1×10^{-5} < E值 < 0.1**。

其中E值<1×10^{-10}的结果置信度较高。可以在其他两个阈值的输出中搜寻新病毒，但同时要注意排除假阳性。

MetaV v2.0.0安装和配置指南

1. 安装前准备

metav已分发到conda和pypi平台，推荐通过conda工具安装metav，会自动下载依赖的软件。

项目github地址：https://github.com/ZhijianZhou01/metav

安装conda环境（已有则忽略）

```
# 安装anaconda3示例代码
    cd /home/zzj/software/  # 假设存在这个文件夹

    wget https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh

    chmod -R 775 Anaconda3-2024.10-1-Linux-x86_64.sh

    ./Anaconda3-2024.10-1-Linux-x86_64.sh

# 安装过程中根据提示输入信息，安装位置可以选择默认，例如为/home/zzj/anaconda3
```

2. 安装metav

```
# (1) add bioconda origin
    conda config --add channels defaults
    conda config --add channels bioconda
    conda config --add channels conda-forge

# (2) install metav
## (i) create a separate environment for metav (recommend)
    conda create -n metav_env python=3.7  # python >=3.5
    conda activate metav_env
    conda install metav    # or 'conda install bioconda::metav'

## (ii) or installation without creating separate environment (slow)
    conda install metav  # or 'conda install bioconda::metav'

# (3) view the help documentation
    metav -h
```

3. 修改trimmomatic软件的最大java内存

默认值仅为1Gb, 非常不推荐，较低的值会导致后续reads无法全部输出。

假设anaconda3安装在/home/zzj目录下：

打开文件 /home/zzj/anaconda3/envs/metav_env/share/trimomatic-0.39-2/trimomatic,
找到default_jvm_mem_opts = ['-Xms512m', '-Xmx1g']
修改为：default_jvm_mem_opts = ['-Xms512m', '-Xmx20g']

即将-Xmx1g调整为-Xmx20g，将java最大内存1Gb调整为20Gb，只要不超过电脑本身的内存，尽量给多点。

4. 构建宿主库

下载宿主的基因组序列并建库，以人类参考基因组GCF_000001405.40为例。

```
# （1）安装ncbi-genome-download工具
  conda activate metav_env
  pip install ncbi-genome-download

# （2）下载GCF_000001405.40
  cd /home/zzj/database     # 假设存在这个文件夹
  mkdir -p human_genome     # 创建新文件夹，储存该数据
  cd human_genome
  ncbi-genome-download  --assembly-accessions  GCF_000001405.40  --section  refseq  --formats
fasta vertebrate_mammalian --flat-output       # 下载人类基因组参考序列，详情了解该工具参数。
  gunzip -c GCF_000001405.40_GRCh38.p14_genomic.fna.gz > human.fna  # 解压缩

# （3）建库
  bowtie2-build /home/zzj/database/human_genome/human.fna
/home/zzj/database/human_genome/human --threads 20    # 记住这里宿主库的路径，后续需要用到
```

5. 构建质粒库

质粒基因组数据来自：https://ftp.ncbi.nlm.nih.gov/refseq/release/plasmid/

截至2025年11月18日，最大压缩分卷号为plasmid.8.genomic.gbff.gz

```
cd /home/zzj/database     # 假设存在这个文件夹
mkdir -p plasmids
cd plasmids

# 下载
curl -O https://ftp.ncbi.nlm.nih.gov/refseq/release/plasmid/plasmid.[1-8].[1-
2].genomic.fna.gz

gunzip -k *.fna.gz
cat *.fna > combined_plasmids_nt.fna   # 合并多个文件为一个，还可以根据需要自己再增加序列

# 建库
bowtie2-build --threads 30 /home/zzj/database/plasmids/combined_plasmids_nt.fna
/home/zzj/database/plasmids/combined_plasmids   # 记住这里质粒库的路径，后续需要用到
```

6. 构建病毒nr库（viral nr）

病毒非冗余蛋白（viral nr）数据来自： https://ftp.ncbi.nlm.nih.gov/refseq/release/viral/

6. 1 构建病毒nr库

```
cd /home/zzj/database     # 假设存在这个文件夹
mkdir -p viral_nr
cd viral_nr

# 下载病毒nr序列和注释信息
wget https://ftp.ncbi.nlm.nih.gov/refseq/release/viral/viral.1.protein.faa.gz
wget https://ftp.ncbi.nlm.nih.gov/refseq/release/viral/viral.1.protein.gpff.gz

gunzip -k *.gz

# 建库
diamond makedb -p 20 --in /home/zzj/database/viral_nr/viral.1.protein.faa --db
/home/zzj/database/viral_nr/ViralProtein.dmnd     # 记住这里病毒nr库的路径，后续需要用到
```

MetaV v2.0.0安装和配置指南

6. 2 提取病毒accession对应的分类信息（目、科、物种/毒株名）

将右侧命令保存为：
viral_taxonomy_information.sh文件，放入/home/zzj/database/viral_nr目录下

运行：

```
chmod -R 775 viral_taxonomy_information.sh
./viral_taxonomy_information.sh
```

viral_extracted_info.txt（记住它的路径）

Accession	order	family	Organism (strain or specise)
YP_009237256.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237257.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237258.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237259.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237260.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237277.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237261.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237262.1	Elliovirales	Fimoviridae	Emaravirus tritici
YP_009237265.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237266.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237267.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237268.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237278.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237279.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237280.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237274.1	Elliovirales	Fimoviridae	Emaravirus idaeobati
YP_009237269.1	Elliovirales	Fimoviridae	Emaravirus fici
YP_009237275.1	Elliovirales	Fimoviridae	Emaravirus fici
YP_009237270.1	Elliovirales	Fimoviridae	Emaravirus fici

```
awk '
BEGIN {
    OFS="\t";
    print "Accession", "order", "family", "Organism (strain or specise)"
}
/^VERSION/ {
    version = $2
}
/\/organism="/ {
    split($0, parts, "\"");
    organism = parts[2];
}
/^  ORGANISM/ {
    taxonomy = "";
    getline;
    while (getline > 0 && $0 !~ /^REFERENCE/ && $0 !~ /^[[:space:]]*$/) {
        taxonomy = taxonomy " " $0;
    }

    order = "Unknown";
    family = "Unknown";

    if (taxonomy ~ /[A-Z][a-z]+ales/) {
        match(taxonomy, /[A-Z][a-z]+ales/);
        order = substr(taxonomy, RSTART, RLENGTH);
    }
    if (taxonomy ~ /[A-Z][a-z]+dae/) {
        match(taxonomy, /[A-Z][a-z]+dae/);
        family = substr(taxonomy, RSTART, RLENGTH);
    }

    if (order == "Unknown" || family == "Unknown") {
        split(taxonomy, taxa_arr, ";");
        for(i in taxa_arr) {
            gsub(/^[ \t]+|[ \t]+$/, "", taxa_arr[i]);
            if (order == "Unknown" && taxa_arr[i] ~ /ales$/) {
                order = taxa_arr[i];
            }
            if (family == "Unknown" && taxa_arr[i] ~ /dae$/) {
                family = taxa_arr[i];
            }
        }
    }
}
/^\/\// {
    if (version != "" && organism != "") {
        print version, order, family, organism;
    }
    version = "";
    organism = "";
    order = "Unknown";
    family = "Unknown";
}
' viral.1.protein.gpff > viral_extracted_info.txt
```

代码请访问metav项目主页

6.3. 或使用已经整理好的病毒nr序列和对应的分类信息文件

下载地址： https://github.com/ZhijianZhou01/metav/releases/tag/data

备注：
（1）病毒nr序列和accession分类信息要配套使用，如果使用上述地址里的accession分类信息文件，也需要使用对应的病毒nr序列来建库，两者要保持一致。

（2）也可以更改这个文件。比如从官网下载viral.1.protein.faa，在它的基础上增加序列，同时在accession分类信息文件里增加对应的条目。

7. 构建nr库

```
cd /home/zzj/database     # 假设存在这个文件夹
mkdir -p nr
cd nr

# nr数据库（FASTA格式，大约100多Gb）
  wget https://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nr.gz
  gunzip -c nr.gz > nr.faa

# 蛋白质accession到taxid映射文件
  wget https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/accession2taxid/prot.accession2taxid.gz
  gunzip -c prot.accession2taxid.gz > prot.accession2taxid

# 完整分类学数据
  wget https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/new_taxdump/new_taxdump.tar.gz
  tar -xzf new_taxdump.tar.gz
```

7. 构建nr库

```
# 构建包含序列Taxonomy信息的NR数据库
diamond makedb --in nr.faa -d /home/zzj/database/nr/nr_taxid.dmnd \
        --taxonmap prot.accession2taxid \
        --taxonnodes nodes.dmp \
        --taxonnames names.dmp \
        --threads 20 \
        --verbose

# 记住nr_taxid.dmnd这个库文件的路径

# 从taxidlineage.dmp中提取所有病毒（Viruses）对应的taxid
awk -F'\t\\|\t' '$2 ~ /(^| )10239( |$)/ {print $1}' taxidlineage.dmp >
/home/zzj/database/nr/virus_taxids.txt    # 记住virus_taxids.txt文件的路径
```

8. 配置xml文件

在实际中，可能会根据不同的需求，调整被metav调用的软件参数（trimmomatic、 bowtie2、megahit、diamond），或者调整数据库的路径。metav旨在提供一个灵活可变的框架，为了统一管理参数，metav使用.xml文件进行配置。

xml文件下载地址： https://github.com/ZhijianZhou01/metav/blob/main/profiles.xml

8. 1 配置宿主库路径

```
        <!-- the path of host database which was created in advance by bowtie2-build program, and usually needs to be
adjusted. -->
        <!-- for example: bowtie2-build -p 20 /path/Human_Gallus_gallus.fas /path/Human_Gallus_gallus -->
        <database name="hostdb">
                <path>
                /home/zzj/database/human_genome/human
                </path>
        </database>
```

将先前构建的宿主库的路径（不要带引号）填入标红色的位置，如果用到多个宿主库，则使用英文逗号隔开，例如：/home/zzj/database/human_genome/human, /home/zzj/database/pig_genome/pig

MetaV v2.0.0安装和配置指南

8. 2 配置质粒库路径

```
<!-- the path of plasmid database, which was created in advance by diamond program. -->
<!-- for example: bowtie2-build -p 20 /path/plasmid.fna /path/plasmid -->
<database name="plasmid">
        <path>
        /home/zzj/database/plasmids/combined_plasmids
        </path>
</database>
```

8. 3 配置病毒nr库和分类信息

```
<!-- the path of viral nr database (*.dmnd), which was created in advance by diamond program. -->
<database name="viral_nr">
        <path>
        /home/zzj/database/viral_nr/ViralProtein.dmnd
        </path>
</database>

<!-- the path of information of viral taxonomy.  -->
<taxonomy name="viral_taxonomy">
        <path>
        /home/zzj/database/viral_nr/viral_extracted_info.txt
        </path>
</taxonomy>
```

8. 4 配置nr库和病毒taxid文件路径

```
        <!-- the path of nr database (*.dmnd) with taxid, which was created in advance by
diamond program. -->
        <database name="nr_taxid">
                <path>
                /home/zzj/database/nr/nr_taxid.dmnd
                </path>
        </database>


        <!-- the path of information of viral taxonomy.  -->
        <taxonomy name="nr_taxid_viruses">
                <path>
                /home/zzj/database/nr/virus_taxids.txt
                </path>
        </taxonomy>
```

将先前构建的nr库路径、病毒taxid文件对应的路径填入标红色的位置。

8. 5 配置trimmomatic软件参数

```
        <!-- the parameters of trimmomatic, please adjust the path of merge_adapter.fas -->
        <app name="trimmomatic">
                <parameter>
                ILLUMINACLIP:/home/zzj/anaconda3/envs/metav_env/share/trimmomatic-0.40-
0/adapters/adapter_combined.fa:2:40:15:8:true LEADING:3 TRAILING:3 SLIDINGWINDOW:4:20 MINLEN:50
                </parameter>
        </app>
```

备注：
（1）这里的adapter_combined.fa是将trimmomatic软件安装目录（/home/zzj/anaconda3/envs/metav_env/share/trimmomatic-0.40-0）下adapters文件夹中所有接头序列进行了合并，还可以根据需要自己增加序列。

（2）标紫色的参数是一个中等严格的reads质控参数，具体请查阅trimmomatic软件说明书。

（3）注意不要在这里设置输入/输出文件的路径，后续是在metav的命令行中输入（下同）。

8. 6 配置bowtie2软件参数

```
<!-- the parameters of bowtie2 for host database-->
<app name="bowtie2_host">
        <parameter>
        --very-sensitive --dovetail
        </parameter>
</app>

<!-- the parameters of bowtie2 for plasmid database-->
<app name="bowtie2_plasmid">
        <parameter>
        --very-sensitive --dovetail
        </parameter>
</app>
```

可以分别为**宿主库**和**质粒库**做bowtie2的步骤设置不同的参数， bowtie2的其他参数说明请查阅说明书。

注意这里不需要设置线程参数，后续是在metav的命令行中统一设置。

8. 7 配置megahit软件参数

```
<!-- the parameters of megahit-->
<app name="megahit">
        <parameter>
        --memory 0.8 --min-contig-len 200
        </parameter>
</app>
```

megahit软件的参数使用请查询说明书。

注意这里不需要设置线程参数，后续是在metav的命令行中统一设置。

8. 8 配置diamond软件参数

```
        <!--  the key parameter of diamond program in viral nr database, note, the max-target-seqs is fixed at 1.  -->
        <app name="diamond_viral_nr">
                <paremeter>
                -e 0.1 -b5 -c1
                </paremeter>
        </app>


        <!--  the key parameter of diamond program in nr database, note, the max-target-seqs is fixed at 1.-->
        <app name="diamond_nr">
                <paremeter>
                -e 0.1 -b5 -c1
                </paremeter>
        </app>
```

可以分别为**病毒nr库**、**nr库**做diamond blastx的步骤设置不同的参数，diamond的其他参数请查阅它的说明书。

注意：（1） --max-target-seqs不需要设置，已经使用固定值1。
（1）e值可以根据需要调整，预设使用宽松的阈值0.1，因为后续还会对命中结果做e值分类输出。
（2）这里不需要设置线程参数，后续是在metav的命令行中统一设置。

MetaV v2.0.0 运行

查看帮助文档:

```
metav -h
```

运行示例:

(i) paired-end sequencing:

```
 metav -pe -i1 reads_R1.fq -i2 reads_R2.fq -xml
profiles.xml -r1 -r2 -t 10 -o outdir
```

(ii) single-end sequencing:

```
 metav -se -u reads.fq -xml profiles.xml -r1 -
r2 -t 10 -o outdir
```

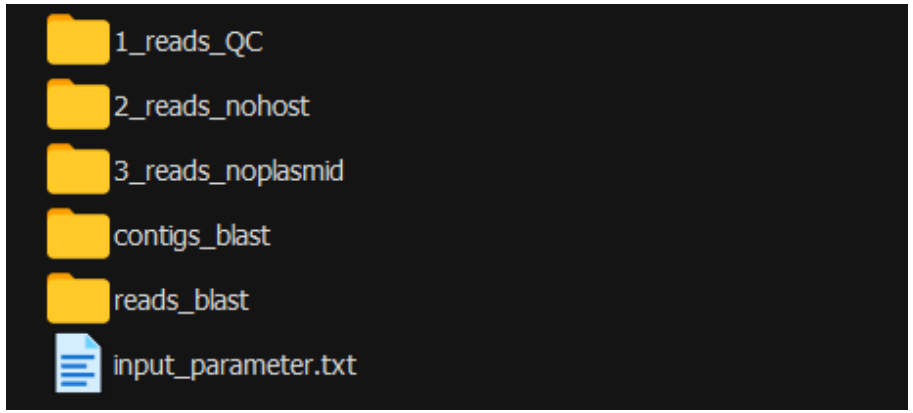
备注:

（1）前面配置好xml文件后，metav本身运行需要的参数比较少。xml文件可以重复利用。

（2）最终结果在reads_blast或contigs_blast目录下的finally_result目录里，包含统计表和提取的reads/contigs序列。

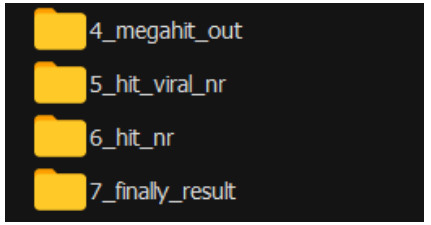
MetaV v2.0.0 输出

输出包含每一步的关键结果和最终结果：



reads_blast

contigs_blast



目录及文件说明：

1_reads_QC：去除接头污染和低质量碱基后的reads。

2_reads_nohost：进一步去除宿主污染后的reads。

3_reads_noplasmid：进一步去除质粒污染后的reads。

reads_blast：使用reads_noplasmid做blastx的结果。

contigs_blast：使用reads_noplasmid先拼接为contigs，再做blastx的结果。

input_parameter.txt：记录metav的参数（不含profiles.xml中的）。

注意

对于contigs，在megahit输出的基础上简化了序列名称，以k141_3783_F1_M3.0000_L317为例：

k141_3783 ：contig的名称（未更改）。

F1：由flag=1简化而来。

M3.0000：由multi=3.0000简化而来。

L317：由len=317简化而来。

MetaV v2.0.0 输出

1. 在测试中发现，相比**reads blast，先将reads拼接为contigs，**再和viral-nr和nr库做blast比对得到的结果会更加干净，假阳性更少。因此**建议使用contigs的blast结果来验证reads的blast**。考虑到当病毒载量较低、或一些新病毒与已知病毒同源性较低时，reads的blast结果也是不可忽视的重要参考。

2. 管道只能在计算上鉴定病毒序列，如果是样本源头受到其他病毒的污染，则无法区分。

3. 任何生物信息学工具都无法完全避免假阳性，建议对metav报告的病毒结果进行验证。

特别注意（Important Notes）

1. As a pipeline that calls existing software, metav aims to reduce the complexity of switching between tools. During the blast step, it is difficult to avoid false positives, especially with reads. Therefore, the final output of metav requires further evaluation. For instance, applying additional filters based on the number of hit reads or performing alignment against reference genomes can help reduce false positives. Additionally, a more stringent E-value threshold can be set for blast, but it is also important to balance the risk of false negatives.

2. Not applicable to metatranscriptomic sequencing data.

1. metav作为一个调用现有软件的流程工具，其初衷是简化多软件切换的繁琐操作。在blast步骤，假阳性难以完全避免，特别是对于短序列reads，因此还需要对metav最后的输出结果进一步评估。例如，根据命中reads数量设置阈值再次进行过滤，或者比对到参考基因组，均可帮助降低假阳性。此外，也可以为blast设置更严格的E值阈值，但也要兼顾假阴性。
2. 不适用于宏转录组测序数据。