Zhang-Lab 生信小课堂 第十七期

Applied Bioinformatics Club (ABC)

和趣求真 秉实生信

（张建伟生物信息学课题组 https://zhang.hzau.edu.cn）

基因结构注释

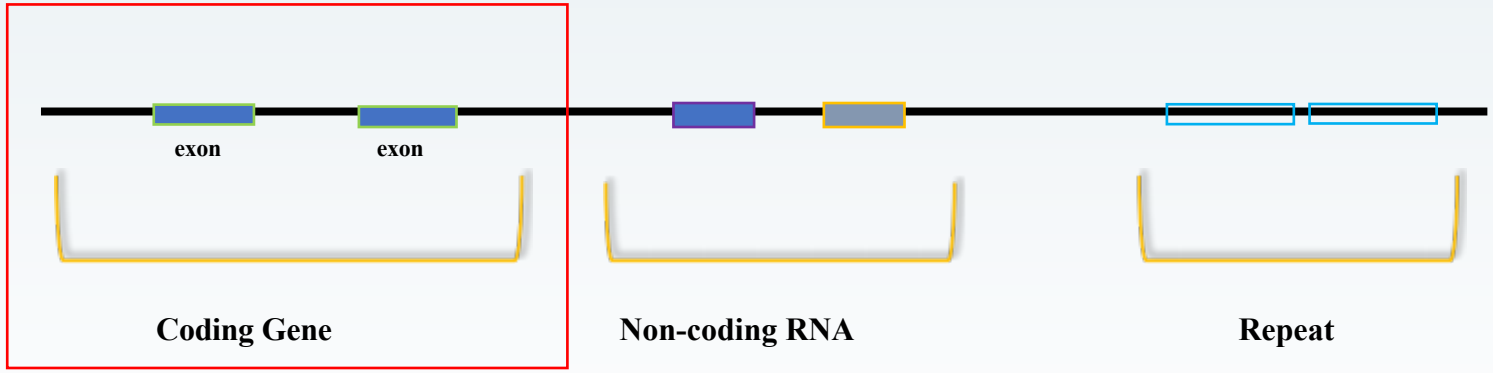
2024.05.10 二综C102 15:00 欢迎大家交流学习！

主讲人：夏丹丹

时间：2024/05/10

基因组注释定义

- **基因组注释：即在一条DNA序列上， 通过从头、同源、结构定义等多种方法，搜寻并定义基因组原件，得到其位置、序列、结构、功能等信息。**



这里以**甘蓝型油菜**为例，介绍基因组编码基因注释基本流程

数据：

- 甘蓝型油菜初步组装的文件 Genome_sequence.fasta
- 甘蓝型油菜的转录组数据 WLC_1.R1.raw.fastq.gz
  WLC_1.R2.raw.fastq.gz
- 甘蓝型油菜参考基因组 zs11.genome.fa

分析流程

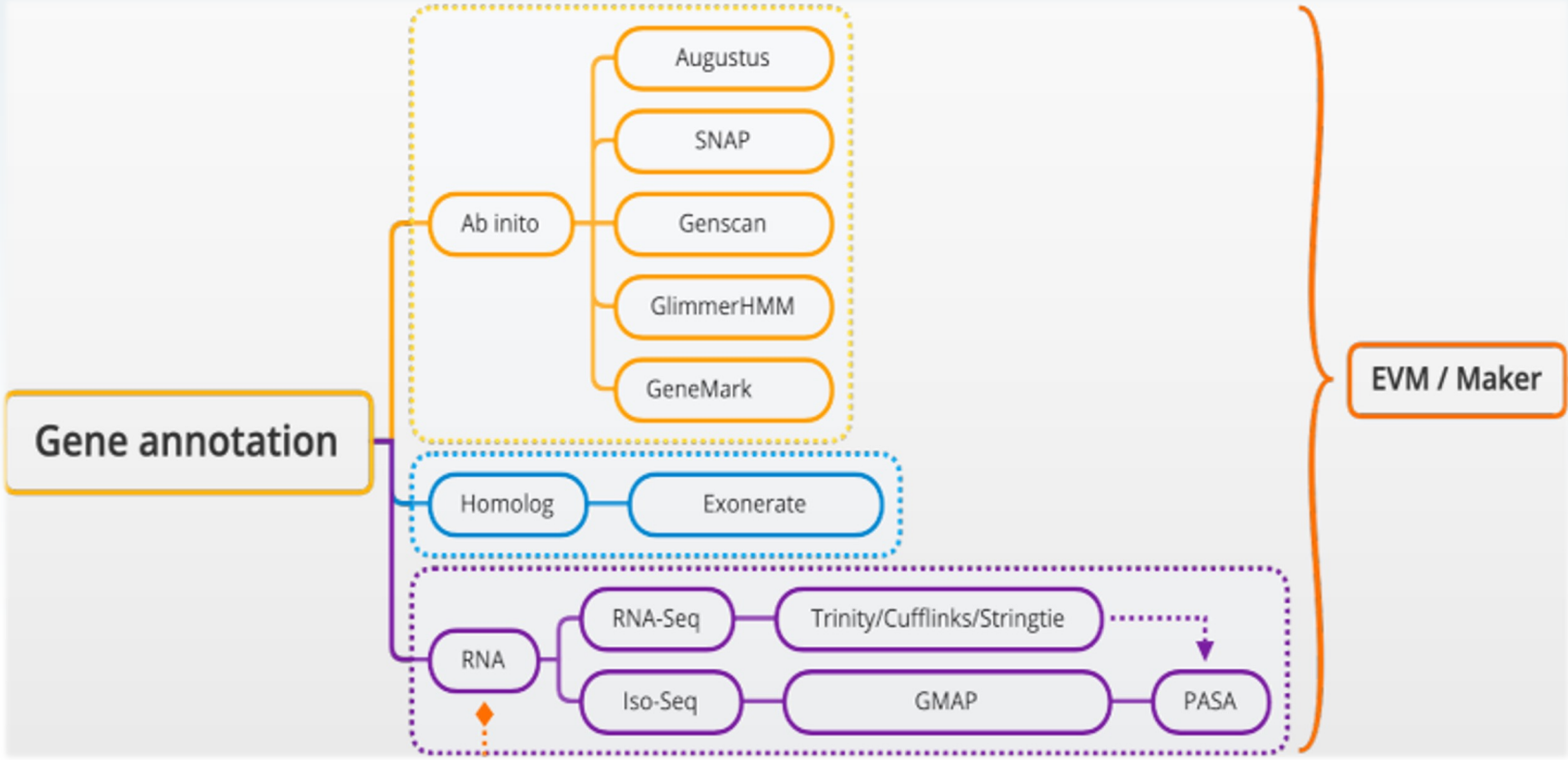
00 重复序列屏蔽

01 从头注释

02 基于转录组预测

03 同源预测

04 EVM整合



重复序列屏蔽： 真核生物的基因组存在大量的重复序列，植物基因组的重复序列甚至可以高达80%。尽管重复序列对维持染色体的空间结构、基因的表达调控、遗传重组等都具有重要作用，但是却会导致BLAST的结果出现大量假阳性，增加基因结构的预测的计算压力甚至影响注释正确性。

1.RepeatMasker：基于与已知的重复序列数据库比对来寻找重复序列

2.repeatmodeler：通过重续序列的结构特征基于自身序列比对，寻找一些物种特有的重复序列

```
Brassica [ 408 ]
    Brassica carinata [ 1 ]
    Brassica oleracea [ 23 ]
    Brassica rapa [ 381 ]
    Brassica nigra [ 1 ]
    Brassica napus [ 2 ]
```

```
$ RepeatMasker/util/queryRepeatDatabse.pl -tree > species.txt
```

```
$ RepeatMasker -e ncbi -species "Brassica napus" -gff Genome_sequence.fasta
```

```
# -e 选择搜索引擎
# -species 选择物种
# -gff 输出gff注释
```

```
$ cat Genome_sequence.fasta.tbl
==================================================
file name: Genome_sequence.fasta
sequences:          3393
total length: 1049012309 bp  (1049012309 bp excl N/X-runs)
GC level:         37.16 %
bases masked:   35051283 bp ( 3.34 %)
==================================================
```

重复序列屏蔽：

Step1: 构建数据库

$ BuildDatabase -name Brassica -engine ncbi Genome_sequence.fasta

生成Brassica.nhr,.nin,.nnd,.nni,.nog,.nsq,.translation 7个文件

Step2: 自我训练

$ RepeatModeler -database Brassica

生成 Brassica-families.fa （用于repeatmasker指定lib）
Brassica-families.stk
RM_371135.SatDec161609382023目录

Step3: RepeatMasker屏蔽重复序列

$ RepeatMasker -nolow -e ncbi -lib Brassica-families.fa Genome_sequence.fasta

生成文件Genome_sequence.fasta.masked用于后续从头预测

01

PART ONE

从头预测

目前的从头预测软件大多是基于HMM(隐马尔科夫链)和贝叶斯理论，通过已有物种的注释信息对软件进行训练，从训练结果中去推断一段基因序列中可能的结构，在这方面做的最好的工具是 AUGUSTUS 它可以仅使用序列信息进行预测。

- Augustus(真核)
- GlimmerHMM (真核, 一般用于植物)
- Genscan (真核，其预测的内含子较大，一般用于动物)
- Genemark-ES/ET (真核)

…

denovo的软件很多,两个软件就可以了,太多软件会增加较多的假阳性,

01

PART ONE

从头预测（屏蔽重复序）

如果有AUGUSTUS的已训练物种，可以直接使用对应的species名称

$ augustus --species=help

查看被训练的物种信息

Step1: 训练模型

```
$ autoAugTrain.pl \
--genome=00.reference_genome/ZS11/zs11.genome.fa \
--trainingset=00.reference_genome/ZS11/zs11.v0.gff3 \
--species=Brassica_napus
```

Step2: 目标基因组预测

$ augustus --species=Brassica_napus --gff3=on Genome_sequence.fasta.masked >aug.gff

Step3: 转换成EVM的输入形式

perl augustus_GFF3_to_EVM_GFF3.pl aug.gff > denovo_prediction.gff

02

PART TWO

RNA-seq辅助注释

fastp进行质控 → hisat2比对到参考基因组 → stringtie构建转录本 → 预测ORF

Step1: 对RNA-seq数据进行质控

$ fastp -i WLC_1.R1.raw.fastq.gz -o output.R1.fq -I WLC_1R2.fq.gz -O output.R2. fq

Step2: 将RNAseq数据比对到参考基因组

$ hisat2-build Genome_sequence.fasta index

$ hisat2 --dta -p 20 -x index -1 output.R1.fq -2 output.R2. fq -S WLC_1.sam

$ samtools view -bS WLC_1.sam | samtools sort -o WLC_1.bam

Step3: 构建转录本

$ stringtie -p 10 -o WLC_1.gtf WLC_1.bam

02

PART TWO

RNA-seq辅助注释

Step4: 使用TransDecoder在构建的转录本上预测Open Reading Frame(ORF)

```
# 从GTF文件中提取FASTA序列
$ TransDecoder-TransDecoder-v5.7.1/util/gtf_genome_to_cdna_fasta.pl WLC_1.gtf
Genome_sequence.fasta.masked > transcripts.fasta
# 将GTF文件转成GFF3格式
TransDecoder-TransDecoder-v5.7.1/util/gtf_to_alignment_gff3.pl merged.gtf > transcripts.gff3
# 预测转录本中长的开放阅读框
$ TransDecoder-TransDecoder-v5.7.1/TransDecoder.LongOrfs -t transcripts.fasta
#预测可能的编码区
$ ransDecoder-TransDecoder-v5.7.1/TransDecoder.Predict -t transcripts.fasta
# 生成基于参考基因组的编码区注释文件
$ TransDecoder-TransDecoder-v5.7.1/util/cdna_alignment_orf_to_genome_orf.pl \
    transcripts.fasta.transdecoder.gff3 \
    transcripts.gff3 \
    transcripts.fasta > transcripts.fasta.transdecoder.genome.gff3
```

03

PART THREE

同源注释

利用近缘物种已知基因进行序列比对，找到同源序列。然后在同源序列的基础上，根据基因信号如剪切信号、基因起始和终止密码子对基因结构进行预测。

相对于从头预测的“大海捞针”，同源预测相当于先用一块磁铁在基因组大海中缩小了可能区域，然后从可能区域中鉴定基因结构。

1.利用TBlastn将同源物种的蛋白比对回基因组，得到候选区域。

2.利用Exonerate/Genewise进行精确的蛋白-核酸比对，以得到剪接位点。

- Exonerate解决了GeneWise存在的很多问题，并且速度快了1000倍，默认选择Exonerate分析

03

PART THREE

同源注释

Exonerate(基于同源蛋白数据)

Step1: 蛋白比对和预测

```
$ exonerate -q $protein -t Genome_sequence.fasta \
      --model protein2genome \
      --showtargetgff yes \
      --showcigar no >exonerate.gff

# --model protein2genome 指定比对模式
# -q 指定参考基因组
# -t 指定需要注释的基因组
# --showtargetgff 输出格式为GFF
# --showalignment no 关闭显示比对细节。
```

Step2: 使用evm脚本转为evm格式的gff

```
$ perl exonerate_gff_to_alignment_gff3.pl exonerate.gff exo_pro.gff
```

04

PART FOUR

EVM整合

Step1: 创建权重文件

```
# copy ~/EVidenceModeler-v2.1.0/testing/weights.txt 下的weights.txt进行修改
$ cp ~/EVidenceModeler-1.1.1/simple_example/weights.txt ./
$ vi weights.txt
TRANSCRIPT      transdecoder      10
ABINITIO_PREDICTION    Augustus          1
## 第一列为来源类型；分为：TRANSCRIPT, ABINITIO_PREDICTION
## 第二列对应着gff3文件第二列
## 第三列为权重
```

Step2: 分割原始数据用于后续并行

```
EVidenceModeler-v2.1.0/EvmUtils/partition_EVM_inputs.pl \
        --partition_dir split \
        --genome Genome_sequence.fasta \
        --gene_predictions aug_evm.gff \
        --transcript_alignments transcripts.fasta.transdecoder.genome.gff3 \
        --segmentSize 100000 --overlapSize 10000 \
        --partition_listing partitions_list.out
```

04

PART FOUR

EVM整合

Step3: 创建并行运算命令并且执行

```
$ EVidenceModeler-v2.1.0/EvmUtils/write_EVM_commands.pl \
      --genome Genome_sequence.fasta \
      --weights weights.txt \
      --gene_predictions aug_evm.gff \
      --transcript_alignments transcripts.fasta.transdecoder.genome.gff3 \
      --output_file_name evm.out  --partitions partitions_list.out >  commands.list
$ EVidenceModeler-v2.1.0/EvmUtils/execute_EVM_commands.pl commands.list
```

Step4: 合并运行结果，并转换成GFF3

```
$ EVidenceModeler-v2.1.0/EvmUtils/recombine_EVM_partial_outputs.pl \
      --partitions partitions_list.out \
      --output_file_name evm.out
$ EVidenceModeler-v2.1.0/EvmUtils/convert_EVM_outputs_to_GFF3.pl \
      --partitions partitions_list.out \
      --output evm.out \
      --genome Genome_sequence.fasta
$ find . -regex ".*evm.out.gff3" -exec cat {} \; | bedtools sort -i - > EVM.all.gff
```

THANK YOU

谢谢观看