使用MAKER进行注释: 如何避免多轮MAKER时的重复运算

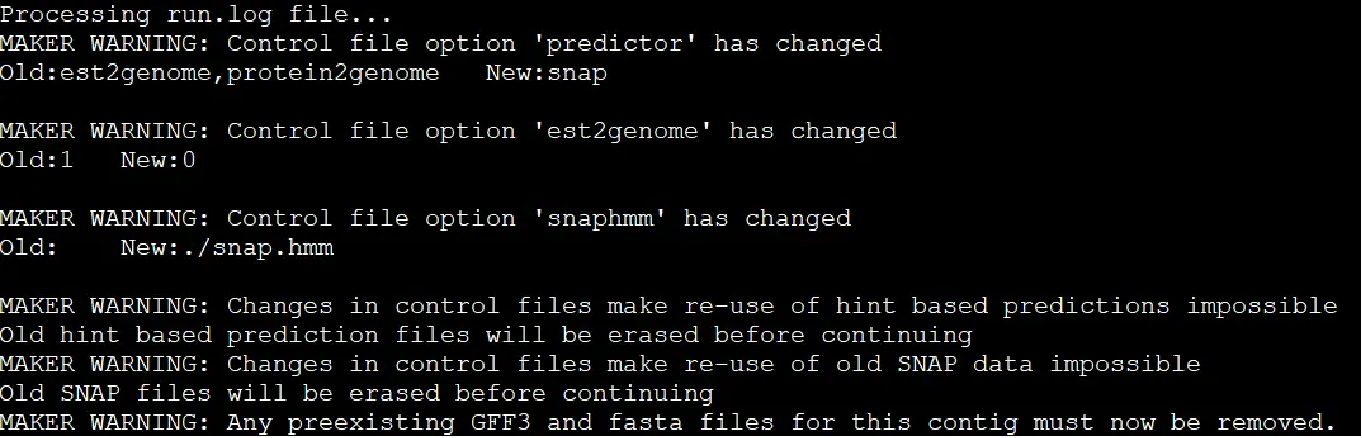
通常而言，我们会运行不只一轮的MAKER。如果参考组序列没有变化，那么有一些计算只需要做一次就行了，例如将EST, Repeat和Protein序列比对到参考基因组，得到它们对应的位置。

我们有三种方法可以避免不必要的运算，第一种方法是直接修改配置文件，让MAKER重复利用之前的运行结果；第二种方式是利用之前输出的GFF文件，通过配置"Re-annotation Using MAKER Derived GFF3"里的选项来跳过对应的计算；第三种方法于是利用之前输出的GFF文件，从中提取EST/Repeat/Protein的位置信息保存为GFF文件，通过配置"est\_gff", "protein\_gff", "rm\_gff"来避免重新计算位置信息。

后续分析建立MAKER高级篇-SNAP模型训练基础上，也就是通过protein和est序列直接输出基因模型，然后训练出初步的HMM模型

方法1

方法1最为简单，我们只需要修改之前的maker\_opts.ctl里的参数，然后重新运行即可。运行时会输出如下的警告信息。



警告信息

注意: MAKER是通过对比maker\_opt.ctl里的配置信息和自己运行时记录的maker\_opts.log来判断哪些参数发生了改变。因此，如果SNAP第二次训练生成的文件，要是和上一次命名相同，**那么它会认为你这次输入的模型文件和上次相同，就会跳过SNAP预测这一步**。

**实际运行时，MAKER会跳过BLAST步骤，但是依旧会调用"exonerate"来处理BLAST结果**。

方法2

如果你不小心把maker的输出文件删掉了，但是你保留着之前gff3\_merge默认参数输出的文件，那么你可以使用该文件来跳过BLAST和Exonerate运算。

Step1: 配置"Re-annotation Using MAKER Derived GFF3"里的参数

#-----Re-annotation Using MAKER Derived GFF3

maker\_gff=round1.gff #MAKER derived GFF3 file

est\_pass=1 #use ESTs in maker\_gff: 1 = yes, 0 = no

altest\_pass=1 #use alternate organism ESTs in maker\_gff: 1 = yes, 0 = no

protein\_pass=1 #use protein alignments in maker\_gff: 1 = yes, 0 = no

rm\_pass=1 #use repeats in maker\_gff: 1 = yes, 0 = no

model\_pass=1 #use gene models in maker\_gff: 1 = yes, 0 = no

pred\_pass=1 #use ab-initio predictions in maker\_gff: 1 = yes, 0 = no

other\_pass=1 #passthrough anyything else in maker\_gff: 1 = yes, 0 = no

此处的round1.gff通过gff3\_merge从上一论的maker输出中提取，代码如下

gff3\_merge -d genome.maker.output/genome\_master\_datastore\_index.log -o round1.gff

Step2: 将"EST Evidence"和"Protein Homology Evidence"里的配置清空，如下

#-----EST Evidence (for best results provide a file for at least one)

est= #set of ESTs or assembled mRNA-seq in fasta format

altest= #EST/cDNA sequence file in fasta format from an alternate organismest\_gff= #aligned ESTs or mRNA-seq from an external GFF3 file

altest\_gff= #aligned ESTs from a closly relate species in GFF3 format

#-----Protein Homology Evidence (for best results provide a file for at least one)

protein= #protein sequence file in fasta format (i.e. from mutiple organisms)

protein\_gff= #aligned protein homology evidence from an external GFF3 file

#-----Repeat Masking (leave values blank to skip repeat masking)

model\_org= #select a model organism for RepBase masking in RepeatMasker

rmlib= #provide an organism specific repeat library in fasta format for RepeatMasker

repeat\_protein= #provide a fasta file of transposable element proteins for RepeatRunner

rm\_gff= #pre-identified repeat elements from an external GFF3 file

prok\_rm=0 #forces MAKER to repeatmask prokaryotes (no reason to change this), 1 = yes, 0 = no

softmask=1 #use soft-masking rather than hard-masking in BLAST (i.e. seg and dust filtering)

Step3: 配置"Gene Prediction"，例如SNAP, 同时将"est2genome"和"protein2genome"设置为0

#-----Gene Prediction

snaphmm=snap.hmm #SNAP HMM file

gmhmm= #GeneMark HMM file

augustus\_species= #Augustus gene prediction species model

# 略过其他参数

est2genome=0 #infer gene predictions directly from ESTs, 1 = yes, 0 = no

protein2genome=0 #infer predictions from protein homology, 1 = yes, 0 = no

# 略过其他参数

会跳过exonerate步骤，直接从snap预测开始。

方法3

我们还可以通过设置est2gff, protein\_gff和rm\_gff，来避免重复序列屏蔽和BLAST+Exonerate运算

Step1: 从之前的MAKER输出的GFF文件种提取EST/Protein/Repeat的位置信息

# transcript alignment

awk '{ if ($2 ~ "est") print $0 }' round1.gff > est.gff

# protein alignments

awk '{ if ($2 == "protein2genome") print $0 }' round1.gff > protein2genome.gff

# repeat alignments

awk '{ if ($2 ~ "repeat") print $0 }' round1.gff > repeats.gff

Step2: 修改EST Evidence / rotein Homology Evidence /Repeat Masking里的配置参数

#-----EST Evidence (for best results provide a file for at least one)

est= #set of ESTs or assembled mRNA-seq in fasta format

altest= #EST/cDNA sequence file in fasta format from an alternate organismest\_gff=est.gff #aligned ESTs or mRNA-seq from an external GFF3 file

est\_gff=./est.gff

altest\_gff= #aligned ESTs from a closly relate species in GFF3 format

#-----Protein Homology Evidence (for best results provide a file for at least one)

protein= #protein sequence file in fasta format (i.e. from mutiple organisms)

protein\_gff=protein2genome.gff #aligned protein homology evidence from an external GFF3 file

#-----Repeat Masking (leave values blank to skip repeat masking)

model\_org= #select a model organism for RepBase masking in RepeatMasker

rmlib= #provide an organism specific repeat library in fasta format for RepeatMasker

repeat\_protein= #provide a fasta file of transposable element proteins for RepeatRunner

rm\_gff=repeats.gff #pre-identified repeat elements from an external GFF3 file

prok\_rm=0 #forces MAKER to repeatmask prokaryotes (no reason to change this), 1 = yes, 0 = no

softmask=1 #use soft-masking rather than hard-masking in BLAST (i.e. seg and dust filtering)

Step3: 配置"Gene Prediction"，例如SNAP, 同时将"est2genome"和"protein2genome"设置为0

#-----Gene Prediction

snaphmm=snap.hmm #SNAP HMM file

gmhmm= #GeneMark HMM file

augustus\_species= #Augustus gene prediction species model

# 略过其他参数

est2genome=0 #infer gene predictions directly from ESTs, 1 = yes, 0 = no

protein2genome=0 #infer predictions from protein homology, 1 = yes, 0 = no

# 略过其他参数

同样也会跳过exonerate步骤，直接从snap开始。

结果比较

对于这三种方法，从运行日志中看，三者都会跳过重复序列屏蔽，将EST和蛋白序列回帖到参考基因组的步骤，然而最终预测的基因数却不一致。

分析方法2和方法1的输出GFF文件时，发现方法2输出包括exonerate\_protein2genome-gene和exonerate\_est2genome-gene。推测其原因在第二种方法的model\_pass, pred\_passs参数在设置为1时会使用之前est2genome和protein2genome输出的基因模型，而由于模型本身就来自于EST和Protein，就变成自我验证，于是输出结果就变多了。当设置model\_pass, pred\_passs参数为0时，最终保证方法2和方法1输出结果一致。

之后设置model\_pass, pred\_passs参数为0，然后比较方法1，方法2和方法3输出的GFF。我发现方法1和方法2的第二列信息完全相同，是blastn, blastx, est2genome, maker, protein2genome, repeatmasker, snap\_masked, 而方法3的第二列为est\_gff:est2genome, maker, protein\_gff:protein2genome, repeat\_gff:repeatmasker, snap\_masked. 目前只能推测是MAKER对这些证据使用方式不同引起了最终输出结果的差异，但具体的原理我没有分析清楚，不过不妨碍使用。

最后，三种方法使用优先级分别是方法1 > 方法2 > 方法3，其中方法2要注意设置model\_pass, pred\_passs的设置。

作者：xuzhougeng

链接：https://www.jianshu.com/p/f9bf0a00b06a

来源：简书

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。