28 January 2021

Kevin Childs

Overview of the MAKER annotation process

The key to obtaining an excellent annotation is a good genome assembly and good transcript and protein evidence. With MAKER, transcript and protein evidence strongly influences gene prediction, and so, spending some extra time to ensure that transcripts and proteins are aligned correctly can greatly improve the quality of the final annotation.

It is no longer recommended to use Illumina RNA-seq for the purpose of obtaining transcript assemblies. The noisy assemblies that result from Illumina RNA-seq data will only degrade gene prediction quality. Instead, cDNA sequences derived from Nanopore or PacBio sequencing are preferred. With Nanopore cDNA sequences, reads should be aligned to the genome with minimap and subsequently assembled with StringTie. The alignments must then be reviewed in a genome browser such as IGV to compare them against aligned protein sequences. StringTie will sometimes fuse transcripts from adjacent genes, but adjusting the StringTie parameters can often reduce these fusion events. The resulting StringTie alignments can be output in gtf format. The transcript gtf file can be converted to gff format, and the transcript alignment gff file can be directly input to MAKER as an entry for the est\_gff= parameter. PacBio IsoSeq reads should be assembled using the PacBio pipeline, and these reads should also be reviewed in a genome browser to compare them against aligned protein sequences. PacBio assembled IsoSeq reads can be aligned to the genome with exonerate.

Transcripts and proteins can be aligned by MAKER using exonerate. However, it is not possible to adjust exonerate's alignment parameters within MAKER. Therefore, we are now performing transcript and protein alignments outside of MAKER. Nanopore cDNA sequences will be aligned with StringTie when they are assembled. PacBio IsoSeq transcript assemblies and protein sequences can be aligned with exonerate. Exonerate outputs alignments in gtf format, which can be converted to gff format.

Before running maker, a custom repeat library should be made. That repeat library can be used during the first MAKER run to mask repetitive regions of the genome. We have been using RepeatModeler to create a custom repeat library (a fasta file), which is then entered in the rmlib parameter in the maker\_opts.ctl file.

MAKER must be run twice. The first run will perform repeat masking and will incorporate transcript and protein evidence into the gff output file. This first run will also create initial gene models from the best quality transcript alignments. This is done by setting est2genome=1. This MAKER run should be performed in a new directory (usually "maker\_1").

The key parameters for the first MAKER run:

genome=/path/to/the/genome/assembly/fasta

organism\_type=eukaryotic # The eukaryotic option is the only option that works.

est\_gff=/path/to/the/premade/transcript/alignment/gff/file

protein\_gff=/path/to/the/premade/protein/alignment/gff/file

model\_org= # Must be blank if an entry is made in rmlib.

rmlib=/path/to/the/premade/repeat/library/fasta/file

est2genome=1 # This will cause MAKER to create gene models from high quality transcript alignments

TMP=/path/to/a/writeable/tmp/directory # On Childs Lab Servers, use /data/run/tmp/

After the initial MAKER run, extract the gff results file and transcript and protein fast files using the instructions in Bowman et al. (2017). Use this initial MAKER result to train SNAP and Augustus. The instructions for training SNAP and AUGUSTUS are also found in Bowman et al. (2017). With the trained SNAP and AUGUSTUS HMMS, the second MAKER run can be started.

Bowman MJ, Pulman JA, Liu TL, Childs KL (2017) A modified GC-specific MAKER gene annotation method reveals improved and novel gene predictions of high and low GC content in *Oryza sativa*. BMC Bioinformatics 18:522

For the second MAKER run the gff results file from the first MAKER run is used. That gff file contains the repeat masking, transcript alignments and protein alignments. There is no need to rerun those steps. In the maker\_opts.ctl file for the second MAKER run, the SNAP and AUGUSTUS HMMs must be specified. The est2genome option must be turned off.

The key parameters for the second MAKER run:

genome=/path/to/the/genome/assembly/fasta

organism\_type=eukaryotic # The eukaryotic option is the only option that works.

maker\_gff=/path/to/gff/results/file/from/first/MAKER/run

est\_gff= # Leave blank

protein\_gff= # Leave blank

model\_org= # Leave blank

rmlib= # Leave blank

snaphmm=/path/to/SNAP/HMM

augustus=name\_of\_hmm\_that\_was\_specified\_when\_training\_augustus

est2genome=0 # Must be set to zero

always\_complete=1 # This forces MAKER to take extra steps so models contain start and stop codons.

single\_exon=1 # If your transcript data is noisy, you might set this to 0.

TMP=/path/to/a/writeable/t maker mp/directory # On Childs Lab Servers, use /data/run/tmp/

After this second MAKER run, follow the steps in Bowman et al. (2017) to collect the gff file, the transcript fasta file and the protein fasta file. The Bowman et al. paper also has information about how to filter the initial gene predictions down to a high-quality set of MAKER Standard gene predictions. Instructions are also in that paper about how to rename the genes in the gff and fasta files and how to generate functional descriptions for the genes.

Some instructions for creating the repeat database and for handling transcript and protein alignments.

**Creating a repeat database with RepeatModeler. #alternatively, you can use EDTA.**

module purge && module load RepeatModeler/2.0.1  
BuildDatabase -name effusus\_DB -engine ncbi /data/run/1\_genome\_assembly/effusus\_v1.0.fasta  
RepeatModeler -database effusus\_DB -pa 32 >& effusus\_v1.0.out

The fasta database from RepeatModeler can be entered into the rmlib parameter in the maker\_opts.ctl file. The model\_org parameter in the Repeat Masking section of the maker\_opts.ctl file should be removed. If model\_org=all, then the entry in the rmlib= will be ignored.

**Exonerate alignments.**

genome=/data/run/plantajo/finalized\_asm/inflexus/1\_genome\_assembly/inflexus\_v1.0.fasta

#genome=/data/run/plantajo/finalized\_asm/inflexus/scaf1\_21/scaf1\_21\_inflexus.fasta

protein\_evid=/data/run/plantajo/databases\_models/ricePep\_noTE\_uniprotPlants.fasta

exonerate --model protein2genome --bestn 5 --minintron 10 --maxintron 3000 --query $protein\_evid --target $genome --showtargetgff yes --showalignment no --ryo ">%qi length=%ql alnlen=%qal\n>%ti length=%tl alnlen=%tal\n" --targetchunkid 1 --targetchunktotal 20 > inflexus\_exonerate\_chunk1.output &

**Convert gtf to gff3.**

module purge && module load cufflinks/2.2.1

gffread -E inflexus\_forMaker\_transcripts.gtf -o- > inflexus\_forMaker\_transcripts.gff3

**Correct the feature source and feature types in the gff file so that they are recognized by MAKER as alignments.**

cp inflexus\_forMaker\_transcripts.gff3 inflexus\_forMaker\_transcripts\_features\_fixed.gff3

For transcripts, the feature source must be est2genome, and the feature types must be expressed\_sequence\_match and match\_part.

sed -i 's/StringTie/est2genome/g' inflexus\_forMaker\_transcripts\_features\_fixed.gff3

sed -i 's/transcript/expressed\_sequence\_match/g' inflexus\_forMaker\_transcripts\_features\_fixed.gff3

sed -i 's/exon/match\_part/g' inflexus\_forMaker\_transcrsshipts\_features\_fixed.gff3

For proteins, the feature source must be protein2genome, and the feature types must be protein\_match and match\_part.

sed -i 's/exonerate:protein2genome:local/protein2genome/g' inflexus\_protein\_features\_fixed.gff3

sed -i 's/mRNA/protein\_match/g' inflexus\_protein\_features\_fixed.gff3

sed -i 's/CDS/match\_part/g' inflexus\_protein\_features\_fixed.gff3

**To order a fasta by length…**

Module load BBMap

sortbyname.sh in=file.fa out=sorted.fa length descending