How to Run Maker, A Practical Genome Annotation Guide for Fish Nerds

By Andrew (Drew) W. Thompson, a fish nerd

Before you start:

**This guide assumes you are working on a server with all prerequisites and modules installed and that you have an assembled genome in fasta file format.*

1.) First sort your contigs largest to smallest in Geneious or with sorting options on Unix/a sorting script.

2.) Make a map file csv or tsv (like below) to rename contigs as you like. Make sure your new names contain no special characters like "=" or ";" as it will cause problems downstream with MAKER.

ScDou1A_293;HRSCAF=313 Fish_name_scaf_1
ScDou1A_312;HRSCAF=333 Fish_name _scaf_2
ScDou1A_1939;HRSCAF=2079 Fish_name _scaf_3
……

3.) Run a script called this4that.py (download and directions found here: https://github.com/ballesterus/Utensils?files=1#2) to rename your genome fasta file using your map file:

Submit your fasta file to NCBI, labeling mitogenome as so:
> Fish_name _scaf_1950 [location=mitochondrion]
AACAGTAC……..

NCBI genome submissions are started here:
https://submit.ncbi.nlm.nih.gov/subs/genome/

(you will need to register an account at NCBI and login)

*You should do this before you start annotation. This is because NCBI will do an initial scan for contamination and you will want to remove that before proceeding.

1.) Making a custom repeat library

It is best to make a custom library for your newly sequenced fishy if one doesn't already exist for that species. If you use a library from a related species or those available from other model organisms you will likely underestimate the percentage of repetitive elements in the genome. Repeat Modeler can do this and it is easy to use:

```
##############################################################################
#! /bin/bash
module load RepeatModeler/1.0.8
BuildDatabase -name NAME_OF_YOUR_DB -engine ncbi PATH_to_FASTA
nohup RepeatModeler -database name NAME_OF_YOUR_DB >& seqfile.out
##############################################################################
```

Repeat Modeler will output a file called "consensi.fa.classified" You can then use this as the input library to Repeat Masker when running MAKER, as well as combining it with other repeat libraries to use for MAKER.

You can also do this with RepeatScout like so:

```
##############################################################################
#! /bin/bash
module load RepeatScout/1.0.5
build_lmer_table -sequence PATH_to_FASTA -freq output_lmer.frequency
RepeatScout -sequence PATH_to_FASTA -output output_repeats.fas  -freq
output_lmer.frequency
##############################################################################
```

For more information see:
http://weatherby.genetics.utah.edu/MAKER/wiki/index.php/Repeat_Library_Construction--Basic

2.) Getting Control Files:

The first part of running maker is obtaining and establishing the options you want in your control files:

After loading your MAKER module, type: "maker –CTL" This creates control file templates: maker_bopts.ctl maker_exe.ctl maker_opts.ctl in the working directory.

Now you need to fill in the options you want for each control file. The main file is maker_opts.ctl. I usually start by leaving the options in the other two files set as default.

for more info go to:
http://weatherby.genetics.utah.edu/MAKER/wiki/index.php/The_MAKER_control_files_explained

3.) Setting up the Initial Run:

If you are running Maker for the first time on your favorite newly sequenced fishy, you need to use the following options as so in "maker_opts.ctl":

#-----Re-annotation Using MAKER Derived GFF3
maker_gff= This is blank because you haven't run it previously to get an initial gff file. This is useful when re-running maker as everything does not need to be realigned, masked, etc. Set remaining options in this section to "0."

#-----EST Evidence (for best results provide a file for at least one)
est= This is where you put the full path to the EST dataset(s) you are using as evidence. Note that this should be the same species you are annotating. If you are using multiple files, the paths can be separated by "," and no spaces. The rest of the options here I leave blank as they are only for alternate sets from different species other than the one you are annotating, or if you already have a gff file of the ESTs

#-----Protein Homology Evidence (for best results provide a file for at least one)
protein= This is where you put the full path to the protein datasets you are using from this species as well as related species.
protein_gff= Again, I leave this option blank as they are only for if you already have a gff file of the proteins

#-----Repeat Masking (leave values blank to skip repeat masking)
model_org=all This option takes advantage of the dataset of repetitive elements from multiple species that come with the MAKER package.
-OR-
rmlib= Path_to_FASTA This is where you can give the path for your species-specific repeat library that you created as described in the previous section.

repeat_protein= I leave this blank.
rm_gff= Again, I leave this option blank as they are only for if you already have a GFF file repetative elements
prok_rm=0 There is no reason to change this.
softmask=1 I use soft masking

#-----Gene Prediction
snaphmm= On your first run you have to leave this blank. You will make a SNAP model based on results from this first run you will use later.
gmhmm= leave this blank-I usually use SNAP and AUGUSTUS for gene models
augustus_species= On your first run you have to leave this blank. You will make a AUGUSTUS model based on results from this first run you will use later.

fgenesh_par_file= I leave this blank.
evm_weights= I leave this blank.
pred_gff= Again, I leave this option blank as they are only for if you already have a gff file.
model_gff= Again, I leave this option blank as they are only for if you already have a gff file.
est2genome=1 Turn this on with "1". This causes MAKER to infer "MAKER genes" directly from your EST evidence. This is necessary because you haven't created any models yet and the ESTs for this species are all you have.
protein2genome=0 Leave this off if your proteins are from different species from the one you are annotating. Leave the remainder of the options in this section on default.

#-----Other Annotation Feature Types (features MAKER doesn't recognize)
leave as defaults

#-----External Application Behavior Options
leave as defaults

#-----MAKER Behavior Options
leave as defaults

4.) Run MAKER

Make a shell script to run maker as so:

```
#################################################################################
#! /bin/bash
#$ -cwd
export TMP=/data/run/tmp/
module load MAKER/r1128
module load MPICH2
mpiexec -n 10  maker -q -fix_nucleotides
#################################################################################
```

Run MAKER via the above shell script like this:

```
#################################################################################
/PATH_to/the_shell_script > maker_text.txt  2>&1
Note that the shell script and the .ctl files should all be in the same directory.
#################################################################################
```

Now, you wait for it to run! Once MAKER has finished, run the following regular expressions on the "*datastore_index.log" to make sure there were no failures:

```
#################################################################################
grep -c 'START' *_datastore_index.log
grep -c 'FINISH' *_datastore_index.log
grep -c 'FAIL' *datastore_index.log
#################################################################################
```

If there are no fails, run gff_merge on *datastore_index.log as so:

```
##############################################################################
module load MAKER/r1128
gff3_merge -d /PATH_to/datastore_index_log.
##############################################################################
```

This will search through a directory tree of each contig or scaffold and merge the maker results from the first run into a single gff called "*.all.gff".

5.) Checking percent of repeat masking (optional):

```
##############################################################################
module purge
module load GNU/4.4.5
module load BEDTools/2.24.0
grep "repeat" *.all.gff >> repeats.gff
bedtools sort -i repeats.gff >> sorted.gff
bedtools merge -i sorted.gff >> merged.gff
##############################################################################
```

This will give you a GFF of all of the regions in the genome that were repeat masked, accounting for any overlap. You can total the number of masked nucleotides from this file and divide by total bases in the genome.

6.) Training SNAP:

Make and cd into SNAP directory. Make and run a shell script to execute SNAP model training based on results from your initial MAKER run. This will work by using the "best" "MAKER genes" promoted by EST2genome with an AED cutoff of "0.2" in your first run:

```
##############################################################################
#! /bin/bash
module load MAKER/r1128
maker2zff -x 0.2 -l 200 -n *.all.gff #this is the merged gff file
fathom -categorize 1000 genome.ann genome.dna
fathom -export 1000 -plus uni.ann uni.dna
forge export.ann export.dna
hmm-assembler.pl NAME_SNAP1_hmm /PATH/SNAP >
/PATH/SNAP/NAMESNAP1.hmm
##############################################################################
```

7.) Training AUGUSTUS:

Make and cd into an AUGUSTUS directory. Use the train_augustus.sh (available here: https://github.com/Childs-Lab/GC_specific_MAKER) written by the Childs Lab and detailed below to train AUGUSTUS based on results from your initial maker run. This will work by using the "best" "MAKER genes" promoted by EST2genome with an AED cutoff of "0.2" in your first run. Make sure you have access to fathom_to_genbank.pl

get_subset_of_fastas.pl randomSplit.pl scripts in the path for this to run (available here: https://github.com/Childs-Lab/GC_specific_MAKER) Execute AUGUSTUS training with a shell script as follows:

```
#########################################################################
#! /bin/bash
bash /PATH_to/train_augustus.sh /PATH/AUGUSTUS /PATH_to/*.all.gff
"Species_Name" /PATH_to*_EST.fasta
#########################################################################
```

8.) Re-Run MAKER with Gene Models from SNAP and AUGUSTUS:

Now you are ready to re-run MAKER a final time. This time MAKER will use alignments from ESTs and Proteins to guide the models that you trained in SNAP and AUGUSTUS with the best scoring (AED score) MAKER genes from your initial MAKER run with the EST2genome option

Run it as you did before only make the following changes to the maker_opt.ctl file:

Change “est2genome=0”
Change “snaphmm=/PATH/SNAP/NAMESNAP1.hmm #SNAP HMM file”
Change “augustus_species="Species_Name" #Augustus gene prediction species model”
Make sure ”keep_preds=1” This will retain predictions supported by models only -if evidence is lacking.
Also, if you have an idea of what the maximum intron length could be, you may want to adjust “split_hit=10000” #length for the splitting of hits (expected max intron size for evidence alignments).

Run MAKER and GFFMerge as you did before

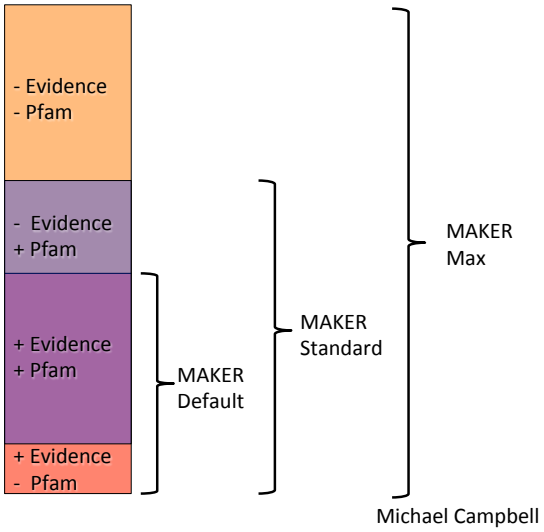
9.) Getting "MAKER Standard: genes:

First run Fasta_merge like so:

```
#################################################################
module load MAKER/r1128
fasta_merge -d /PATH/datastore index log.
#################################################################
```

This has given you the “MAKER Max” gene set. The next step is to run a Pfam search for genes with known protein domains. The goal is to get a “MAKER Standard” gene set that only keeps genes with EST or protein evidence or that have been detected by models but also contain Pfam domains. See figure below:

Three MAKER Gene Sets



Perform pfam search using hmmscan from the iprscan package:

```
##############################################################################
module load iprscan/4.8
hmmscan --domE 1e-3  -E 1e-5  --cpu 3  --acc --tblout out.txt /PATH_to/Pfam-A.hmm
/PATH_to/.all.maker.proteins.fasta  > PFam_output.txt  2>&1
##############################################################################
```

Then you identify the maker standard genes using a number of scripts provided by the Childs Lab (https://github.com/Childs-Lab/GC_specific_MAKER):

```
##############################################################################
perl /PATH_to/generate_maker_standard_gene_list.pl --input_gff /PATH_to/*.all.gff --
pfam_results out.txt --pfam_cutoff 1e-10 --output_file
NAME_complete_w_single_exons_standard_gene_list.txt
##############################################################################
```

Generate the transcript and protein fasta files for the maker standard gene set.
Generate the gff file for the maker standard gene set.

```
##############################################################################
module load BioPerl
perl /PATH_to/get_subset_of_fastas.pl  -l
/PATH_to/complete_w_single_exons_standard_gene_list.txt  -f
/PATH_to/complete_w_single_exons.all.maker.proteins.fasta  -o
/PATH_to/complete_w_single_exons_maker_standard_set.proteins.fasta

perl /PATH_to//get_subset_of_fastas.pl   -l
/PATH_to/complete_w_single_exons_standard_gene_list.txt  -f
```

/PATH_to/complete_w_single_exons.all.maker.transcripts.fasta -o /PATH_to/complete_w_single_exons_maker_standard_set.transcripts.fasta

#options: -l list_gene_ids -f multifasta_file -o output_fasta_file

perl /PATH_to/create_maker_standard_gff.pl --input_gff /PATH_to/*.all.gff --output_gff /PATH_to/complete_w_single_exons_standard_genes.gff --maker_standard_gene_list /PATH_to/complete_w_single_exons_standard_gene_list.txt

/PATH_to/create_maker_standard_gff.pl --input_gff /PATH_to/*.all.gff --output_gff /PATH_to/complete_w_single_exons_standard_genes.gff --maker_standard_gene_list /PATH_to/complete_w_single_exons_standard_gene_list.txt
##

Congrats you ran MAKER and should have an annotated genome of your newly sequenced fish!

**For another, detailed description see: http://gmod.org/wiki/MAKER_Tutorial_2013*