Gene annotation pipeline

Note: Code is highlighted in green, changes to parameter files are highlighted in gray.

References: http://gmod.org/wiki/MAKER_Tutorial_2012
https://groups.google.com/forum/#!forum/maker-devel

Training SNAP (*ab initio* gene prediction)

Computation time: ~8 hours total (you may be able to do each step on standby)
You will need:

1) A fasta file with chicken, turkey, zebra finch and pigeon proteins (my file was called avianUniprot.fasta)
2) Approximately 10 Mb of your longest contigs (I used all scaffolds > 1200000 bp, my file was called kmer70_min1200000_scaffolds)

```
module load MAKER/2.28b

maker –CTL

nano maker_opts.ctl
```

Edit the file so the following applies and save/exit:

```
genome=kmer70_min1200000_scaffolds.fasta
protein=avianUniprot.fasta
protein2genome=1

mpiexec -n 48 maker maker_opts.ctl maker_bopts.ctl maker_exe.ctl &> log_file.txt

cd kmer70_min1200000_scaffolds.maker.output

maker2zff –n –d kmer70_min1200000_scaffolds_master_datastore_index.log
```

This will create two files: genome.ann and genome.dna

The following commands give you an idea of what your data looks like:

```
fathom genome.ann genome.dna -gene-stats
```

```
fathom genome.ann genome.dna -validate
```

To train SNAP:

```
cd..
```

```
mkdir snap
```

copy genome.ann and genome.dna files to snap folder

```
cd snap
```

```
fathom -categorize 1000 genome.ann genome.dna
```

```
fathom -export 1000 -plus uni.ann uni.dna
```

```
forge export.ann export.dna
```

```
hmm-assembler.pl Pult . > Pult.hmm
```

```
cd ..
```

```
nano maker_opts.ctl
```

Edit the file so that the following applies and save/exit:

```
snaphmm=snap/Pult.hmm
protein2genome=0
```

```
mpiexec -n 48 maker maker_opts.ctl maker_bopts.ctl maker_exe.ctl &> log_file.txt
```

Wait for MAKER to finish running and manually end job. This will be much faster than the first run.

cd kmer70_min1200000_scaffolds.maker.output

delete the previous genome.ann and genome.dna files that are in the maker.output directory

maker2zff –n –d kmer70_min1200000_scaffolds_master_datastore_index.log

mkdir snap2

copy genome.ann and genome.dna files to snap2 folder

fathom -categorize 1000 genome.ann genome.dna

fathom -export 1000 -plus uni.ann uni.dna

forge export.ann export.dna

hmm-assembler.pl Pult . > Pult2.hmm

cd ..

SNAP is now trained.

1st MAKER run

Computation time: 120-200 hours (run on fnrgenetics, I usually set the run time for the job as 300 hrs)

You will need:

1) A fasta file with chicken, turkey, zebra finch and pigeon proteins (my file was called avianUniprot.fasta)
2) The SNAP file you just generated (my file was called Pult2.hmm)
3) EST sequences from a closely-related species (I used falcon ESTs, my file was called falconTrinity.fasta)
4) All of your scaffolds with a minimum length of 10000 bp (my file was called kmer70_min10000_scaffolds.fasta)
5) MIKE_compare_annotations_3.1.pl script

```
module load MAKER/2.28b

maker –CTL

nano maker_opts.ctl
```

Edit the file so the following applies and save/exit:

```
genome=kmer70_min10000_scaffolds.fasta
alttest=falconTrinity.fasta
protein=avianUniprot.fasta
snaphmm=snap2/Pult2.hmm
augustus_species=chicken
keep_preds=1
tries=1
```

Note that alttest is used when you're providing ests from a closely-related species. If you have est sequences for your own species, use est= instead of alttest=. This will allow MAKER to run MUCH faster, as it won't be translating in 6 different reading frames.

```
mpiexec -n 48 maker maker_opts.ctl maker_bopts.ctl maker_exe.ctl &> log_file.txt
```

You have finished your first MAKER run. Use fasta_merge to create genome-level fasta files with annotations based on different types of evidence (snap, augustus, ab initio + blast evidence, etc.). Use gff3_merge to create a genome-level gff file. I also make a zff file just so I can generate summary statistics.

```
fasta_merge -d kmer70_min10000_scaffolds_revisedassembly_master_datastore_index.log

gff3_merge -d kmer70_min10000_scaffolds_revisedassembly_master_datastore_index.log

maker2zff –n –d kmer70_min1200000_scaffolds_master_datastore_index.log
```

I use several scripts to generate summary statistics for the genome-level fasta and gff files. It is possible for incomplete gff files to be generated if there isn't enough tmp room, so it is important to make sure everything matches up before moving onto the next step.

```
perl MIKE_compare_annotations_3.1.pl kmer70_min10000_scaffolds_revisedassembly.all.gff
```

```
fathom genome.ann genome.dna -gene-stats
```

```
perl fastaStats.pl –I kmer70_min10000_scaffolds_revisedassembly.fasta
```

InterProScan and downstream processing

You will need:

1) The all.maker.proteins.fasta file from you MAKER run (mine was called kmer70_min10000_scaffolds.all.maker.proteins.fasta)
2) The .gff file from your MAKER run (mine was called kmer70_min10000_scaffolds_revisedassembly.all.gff)
3) InterProScan installed locally in your scratch drive (you might check the bioinformatics modules as well, InterProScan is supposed to be added at some point)
4) quality_filter.pl and iprscan_parser.pl scripts

```
./interproscan.sh -appl PfamA -iprlookup -goterms -f tsv –I kmer70_min10000_scaffolds.all.maker.proteins.fasta
```

If needed, modify the merged tsv file so that the first column is only the maker id

```
perl -lane 'my @array = split(/\t/, $_); $array[0] = $F[0]; print join("\t", @array);' merged_iprscan_output.tsv > merged_iprscan_output_fixed_ids.tsv
```

Update the gff3 file

```
module load MAKER/2.28b
```

```
ipr_update_gff kmer70_min10000_scaffolds_revisedassembly.all.gff  merged_iprscan_output_fixed_ids.tsv
```

NOTE: This modifies the gff3 file in place so if you kill it before it is done you will truncate your gff3 file, so it is a good idea to make a backup copy. Now you can use the quality_filter_2.pl script to make gff files with genes annotated with different types of evidence. The updated gff file is the “max” build with all possible gene annotations (including ab initio predictions for which there is no other evidence). The maker “default” build includes gene annotations generated from ab initio predictions + protein or EST evidence. The maker “standard” build includes gene annotations generated from ab initio predictions + protein or EST evidence or an InterProScan protein domain hit.

Make the default, standard, and max builds

quality_filter.pl -d <ipr_updated_gff3_file> > maker_default.gff

quality_filter.pl -s <ipr_updated_gff3_file> > maker_standard.gff

The maker max is the original file

2nd MAKER run:

Computation time: 10-20 hours (run on fnrgenetics, I usually set the run time for the job as 48 hrs)

This second MAKER retains your MAKER standard annotations while discarding the ab initio predictions that don't have any other evidence. Start by getting a gff3 file with only the gene models and none of the evidence or appended fasta sequence. You can use gff3_merge for this.

gff3_merge -g -n MAKER_standard.gff

The output will be a file named genome.all.gff. This file only has the MAKER gene models and the appended fasta sequence has been removed. You need to copy this file to the same directory that holds the maker_opts.ctl, genome, protein, etc. files from your first MAKER run.

Next edit the maker_opts.ctl file and turn off all of the gene predictors (i.e. get rid of your snap2 file and augustus chicken setting) and put the genome.all.gff file in as model_gff in the gene prediction section and set map_forward=1.

```
snaphmm=
augustus_species=
model_gff=genome.all.gff #annotated gene models from an external GFF3 file (annotation pass-through)
map_forward=1 #map names and attributes forward from old GFF3 genes, 1 = yes, 0 = no
keep_preds=0 #Add unsupported gene prediction to final annotation set, 1 = yes, 0 = no
```

If you run this in the same directory you did your original run in MAKER will be able to find its kmer70_min10000_scaffolds_revisedassembly.maker.output directory and won't have to realign the evidence and will make a directory for each scaffold with a gff3 file that contains the gene models you passed in and the evidence. This will run much faster than a de novo annotation because it doesn't have to realign the evidence or redo the repeat masking.

```
mpiexec -n 48 maker maker_opts.ctl maker_bopts.ctl maker_exe.ctl &> log_file_2ndrun.txt
```

You have finished your second MAKER run. Use fasta_merge to create genome-level fasta files.

```
fasta_merge -d kmer70_min10000_scaffolds_revisedassembly_master_datastore_index.log
```