The guide to run MAKER2-two-pass pipeline

1 Computer/program requirements

- In a UNIX-based system, first make sure GeneMark-ES, MAKER2, and their necessary external dependencies or additional programs including SNAP, AUGUSTUS, BLAST+, Exonerate, RepeatMaker. MAKER Tutorial with all above tools' installation source and usage information can be found in (http://weatherby.genetics.utah.edu/MAKER/wiki/index.php/MAKER_Tutorial_for_GMOD_Online_Training_2014).
- InterProScan were properly installed.
- Genome Browser, like Apollo or igv can be run on Mac or PC.
- GeneMark-ES can be find how to use in website (https://wiki.gacrc.uga.edu/wiki/GeneMark).

2 Run GeneMark-ES

2.1 Description

The GeneMark-ES program determine the protein-coding potential of a DNA sequence (within a sliding window) by using species specific parameters of the Markov models of coding and non-coding regions. In the bash file for running command in shell of UNIX system, starting with # indicating this line is comments.

2.2 Running Program

First, please copy the gm_key to your home dir and move genome fasta, assembled transcriptome fasta, protein dataset fasta files to working dir before run any program.

```
cd /YOUR/PATH/TO/STORE/THESE/FILES
scp gmes_petap/gm_key_64.gm_key didiren@severname:/home/didiren/
scp Cryphonectria.genome.fasta didiren@severname:/work/didiren/
scp assembled.transcriptome.fasta didiren@severname:/work/didiren/
scp uniprot_sprot.fasta.gz didiren@severname:/work/didiren/
```

Then, at the queue type sever (IGBB) in this study, all jobs always will be submitted in a bash script format to the queue. Here, in order to run geneMark-ES, a bash script file called genemark_run.sh was created as followed frame:

```
#!/bin/bash
#PBS –N Run_Genemark_Default
#PBS –l nodes=1:ppn=20
#PBS –l walltime=48:00:00

#======================== SETUP GENEMARK =====================
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#---------------------------command to run genemark------------
# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd /work/didiren
# RUN GeneMark WITH MAX NUMBER OF THREADS
#COMMANDS:
/PATH/TO/genemark-es-et_4.30/gmes_petap.pl --ES --fungus --min_contig 10000 --
sequence Cryphonectria.genome.fasta --cores $PBS_NUM_PPN
```

 # OPTIONS I USED FOR GENEMARK-ES:

#--cores	*to run with multiple threads*
#--fungus	*to run the algorithm with branch point model (most useful for fungal genomes)*
#–ES	*to run self-training*
#--min_contigs	*to only take contigs when its size is over 10000 in training.*
#--sequence	*to take genome sequence*

Now submit the job to the sever by typing this command at the directory of where your genemark_run.sh file is.

```
qsub genemark_run.sh
```

2.3 GeneMark-ES output

This should work fine in most cases, but for different sever, it may need setup a few library paths for running any software that need specific help from technicians. The most important output file you need to feed next step from GeneMark-ES is gmhmm.mod.

3 Run MAKER2

3.1 Description

MAKER is a genome annotation and curation pipeline that can be used to perform de novo genome annotation and legacy genome annotation by updating it with new quality metrics called Annotation Edit Distance (AED), so a comparison between a new with a preexisting annotation could be proceed next to exhibit the improvement of the new annotation. This tutorial assumes you are starting with three input files, a well-assembled assembled genome (Here I use C. parasitica genome with N50 equaling to 5118729 in FASTA format provided by JGI), a newly produced transcriptome file from Additional file 1 , assembled.transcriptome.fasta that provides intrinsic RNA evidence, and a well-characterized proteins dataset named uniprot_sprot.fasta.gz obtained from Swiss-Prot database (http://www.uniprot.org/downloads#uniprotkblink) to help building gene models with extrinsic protein evidence. Also, running them with the -h option will display the help message including a description of the program, the usage, and all options.

3.2 Running MAKER2-First pass

MAKER uses a set of control (ctl) files to manage its process automatically. Through these control files created first, all input is given to MAKER and all parameters are modified. First, MAKER need to run by creating three control files: maker_opts.ctl, maker_bopts.ctl, and maker_exe.ctl in working directory by writing this bash script file maker_ctl.sh and submit it to the sever.

```
#!/bin/bash
#PBS -N Run_MAKER_Default
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00
#======================== SETUP MAKER ========================
swsetup () { eval `/usr/erc-share/etc/swsetup/swsetup.pl $*`; }
swsetup maker
swsetup augustus
PERL5LIB=/usr/local/igbb/maker/lib/perl5:$PERL5LIB
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#-----------------commands to create maker control files---------
```

```
Cd /work/didiren
maker -CTL
```

Now submit the job to the sever by typing this command at the directory of where your maker_ctl.sh file is.

```
qsub maker_ctl.sh
```

3.2.1 Editing three control files

With three control files existing in didiren/ folder, the maker_exe.ctl file was edited by adding the path to two parameters for MAKER to get access to the model built with GeneMark-ES. All the other parameters in the maker_exe.ctl file should not need to be edited because it simply tells MAKER2 where to find the its own executables.

```
gmhmme3=/usr/local/igbb/genemark-es-et_4.30/gmhmme3
probuild=/usr/local/igbb/genemark-es-et_4.30/probuild
```

Edit the maker_opts.ctl file to specify all inputs, like Genome sequence, RNA evidence, Protein evidence and the gene finder to use.

```
genome=/work/didiren/Cparasitica.genome.fasta
organism_type=eukaryotic
est=/work/didiren/assembled.transcriptome.fasta
protein=/work/didiren/uniprot_sprot.fasta.gz
gmhmm=/work/didiren/genemark_output/output/gmhmm.mod
est2genome=1
protein2genome=1
keep_preds=1
single_exon=1
```

 # OPTIONS I USED FOR MAKER2 HERE:

#genome	*to give genome sequences*
#organism_type	*to indicate the type of organism*

#est	*to give RNA evidence sequences*
#protein	*to give protein evidence sequences*
#gmhmm	*to give GeneMark trained HMM file*
#est2genome	*to infer gene predictions directly from RNA sequence, 1=yes, 0=no*
#protein2genome	*to infer gene predictions directly from protein homology, 1=yes, 0=no*
#keep_preds	*to concordance threshold to add unsupported gene prediction, for genomes with high gene density like fungi, oomycetes, etc, setting to 1 can be beneficial.*
#single_exon	*to consider single exon EST evidence with allowing one exon gene exist.*

3.2.2 To run MAKER2-First pass

Once maker_opts.ctl and maker_bopts.ctl have been edited, run Maker2 by creating this bash script file maker_firstpass.sh in the /work/didiren folder and submit it to the sever.

```
#!/bin/bash
#PBS -N Run_Maker_FirstPass
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00

#========================= SETUP MAKER ========================
swsetup () { eval `/usr/erc-share/etc/swsetup/swsetup.pl $*`; }
swsetup maker
swsetup augustus
PERL5LIB=/usr/local/igbb/maker/lib/perl5:$PERL5LIB
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#--------------------------------------------------------------

# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd /work/didiren

# RUN MAKER2 WITH MAX NUMBER OF THREADS
maker -c $PBS_NUM_PPN -base maker2_first_run
# -C        TO USE MULTIPLE PROCESSORS
# -BASE     TO USE SET THE BASE NAME MAKER USES TO SAVE OUTPUT FILES
```

Now submit the job to the sever by typing this command at the directory of where your maker_firstpass.sh file is.

```
qsub maker_firstpass.sh
```

3.2.3 MAKER2-First pass outputs

Now in the current working directory, MAKER2 have created a folder named maker2_first_run.maker.output. The name came from the –b option and always followed with maker.output. Inside this directory, MAKER2 created a few log files, one of them, maker2_first_run_master_datastore_index.log need to looked into first because it shows if there is any failure of each scaffold MAKER2 has taken. Since MAKER2 has stored separately the output of each scaffold MAKEKR has generated for that, it is required to combine all of them to generate an integrated gff, proteins.fasta, transcripts.fasta first. And then, the gene model from the MAKER2-first pass need to be converted to HMM model to train gene predictors, SNAP and Augustus in the MAKER2-second pass.

3.3 Running MAKER2-interim

3.3.1 To run MAKER2-interim

Preparation work of training the SNAP and Augustus with the output from MAKER2-first pass were completed by running this bash scripts named maker_interim.sh.

```
#!/bin/bash
#PBS -N Run_SNAP_Augustus
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00

#========================= SETUP MAKER ==========================
swsetup () { eval `/usr/erc-share/etc/swsetup/swsetup.pl $*`; }
swsetup maker
swsetup augustus
PERL5LIB=/usr/local/igbb/maker/lib/perl5:$PERL5LIB
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#----------------------------------------------------------------

# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd /work/didiren/
```

```
base=maker2_first_run

# TO MERGE ALL GFF3 FRILE FROM EACH SCAFFOLD TO ONE GFF3 FILE
gff3_merge -d $base.maker.output/$base\_master_datastore_index.log
#TO MERGE ALL FASTA FRILE FROM EACH SCAFFOLD TO ONE FASTA FILE
fasta_merge -d $base.maker.output/$base\_master_datastore_index.log
#CREATE A FOLDER HERE CALLED MAEKR2_FIRST_RUN.HMM
mkdir $base.hmm
#CHANGE WORKING DIRECTORY TO MAEKR2_FIRST_RUN.HMM
cd $base.hmm
#TO CONVERT GFF3 GENE MODELS TO ZFF FORMAT
maker2zff ../$base.all.gff
#HERE TWO FILES WERE GENERATED, GENOME.ANN, GENOME.DNA. THEY ARE USED TO TRAIN SNAP
#First TO FILTER THE INPUT GENE MODELS, CAPTURE GENOMIC SEQUENCES IN EACH MODEL
#LOCUS TO PRUDUCE HMM
fathom genome.ann genome.dna -categorize 1000
fathom -export 1000 -plus uni.ann uni.dna
forge export.ann export.dna
hmm-assembler.pl $base . > ../$base.snap.hmm
#THE FINAL OUTPUT IS .SNAP.HMM FILE

#CONVERT MAKER2 GFF3 PREDICTIONS INTO AUGUSTUS HMM
cd /work/didiren/$base.maker.output/$base.hmm
zff2gff3.pl genome.ann | perl -plne 's/\t(\S+)$/\t\.\t$1/' >genome.gff3

#CREATE SYMLINKS FOR CONFIG/SCRIPTS FILES OF AUGUSTUS IN YOUR WORKING FOLDER
ln -s /usr/local/igbb/augustus/bin src
ln -s /usr/local/igbb/augustus/bin bin
ln -s /usr/local/igbb/augustus/scripts scripts
cp -r /usr/local/igbb/augustus/config ./
#NOW you create @src @bin @scripts @config in /$base.hmm folder, SO YOU CAN ADD NEW
#SPECIES MODEL TO CONFIGS FILE

# PREPARE cdna and species names for augustus:
species=cryphonectria_parasitica
AUGUSTUS_CONFIG_PATH=/work/didiren/$base.hmm/config
#run augustus using autoAug:
./scripts/autoAug.pl      --genome=/work/didiren/Cryphonectria.genome.fasta      --
species=$species --cdna=/work/didiren/assembled.transcriptome.fasta
```

```
--trainingset=genome.gff3 -v --useexist
```

 # OPTIONS I USED HERE:

#--genome	*to provide the access to genome seuqence*
#--species	*to provide species name of your organism (for most non-model organisms only, Augustus config folder already have model-organisms' training set)*
#--cdna	*to give RNA evidence sequences*
#--traningset	*to offer a training set to start with*
#-v	*to print more status info. To make this script verbose*
#--useexist	*To use and change the present config and parameters file, here new species files were added.*

Now submit the job to the sever by typing this command at the directory of where your maker_interim.sh file is.

```
qsub maker_interim.sh
```

3.3.2 MAKER2-interim outputs

After MAKER2-interim run, there are two training set ready for SNAP and Augustus separately. One of them is maker2_first_run.snap.hmm in maker2_first_run.hmm/ folder, the other is the cryphonectria_parasitica file in maker2_first_run.hmm/config/ folder.

3.4 Running MAKER2-Second pass

The purpose of the second pass of MAKER2 is to improve the quality of genome annotation by combining the outputs of several different gene predictors with newer training set because MAKER2 is a iterative fashion pipeline.In order to accomplish this, all newest training set obtained from last step were fed in MAKER2 by editing maker_opts.ctl file.

3.4.1 Editing three control files

With three control files existing in work/didiren/ folder, maker_exes.ctl was exited for the MAKER-Frist pass and it need to be kept same. However, for the maker_opts.ctl file, several parameters need to specified to add more inputs, like training set for SNAP and Augustus to use. One thing needs

to be careful is that the config path must present in the special line augustus_species besides adding training set file.

```
genome=/work/didiren/Cparasitica.genome.fasta
organism_type=eukaryotic
est=/work/didiren/assembled.transcriptome.fasta
protein=/work/didiren/uniprot_sprot.fasta.gz
snaphmm=/work/didiren/maker2_first_run.hmm/maker2_first_run.snap.hmm
gmhmm=/work/didiren/genemark_output/output/gmhmm.mod
augustus_species=cryphonectria_parasitica --AUGUSTUS_CONFIG_PATH=/work/didiren/maker2_first_run.hmm/config
est2genome=0
protein2genome=0
keep_preds=1
single_exon=1
pred_stats=1
```

 # OPTIONS I USED FOR MAKER2 HERE:

#genome	*to give genome sequences*
#organism_type	*to indicate the type of organism*
#est	*to give RNA evidence sequences*
#protein	*to give protein evidence sequences*
#gmhmm	*to give GeneMark trained HMM file*
#snaphmm	*to give the training HMM file to SNAP*
#augustus_species	*to give the training model to Augustus (all letters must in one line)*
#est2genome	*to infer gene predictions directly from RNA sequence, 1=yes, 0=no*
#protein2genome	*to infer gene predictions directly from protein homology, 1=yes, 0=no*
#keep_preds	*to concordance threshold to add unsupported gene prediction, for genomes with high gene density like fungi, oomycetes, etc, setting to 1 can be beneficial.*
#single_exon	*to consider single exon EST evidence with allowing one exon gene exist.*
#pred_stats	*to report AED and QI quality metrics in gff file, 1=yes, 0=no*

3.4.2 To run MAKER2-Second pass

Once maker_opts.ctl have been edited, run Maker2 by creating this bash script file maker_secondpass.sh in the /work/didiren folder and submit it to the sever.

```
#!/bin/bash
#PBS -N Run_Maker_SecondPass
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00

#======================== SETUP MAKER ========================
swsetup () { eval `/usr/erc-share/etc/swsetup/swsetup.pl $*`; }
swsetup maker
swsetup augustus
PERL5LIB=/usr/local/igbb/maker/lib/perl5:$PERL5LIB
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#-------------------------------------------------------------

# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd /work/didiren

# RUN MAKER2 WITH MAX NUMBER OF THREADS
maker -c $PBS_NUM_PPN -base maker2_second_run
# -C        TO USE MULTIPLE PROCESSORS
# -BASE     TO USE SET THE BASE NAME MAKER USES TO SAVE OUTPUT FILES
```

Now submit the job to the sever by typing this command at the directory of where your maker_secondpass.sh file is.

```
qsub maker_secondpass.sh
```

3.4.3 MAKER2-Second pass outputs

Now in the current working directory, MAKER2 have created a folder named maker2_second_run.maker.output. Again, inside this directory, MAKER2 created a few log files, one of them, maker2_second_run_master_datastore_index.log need to looked into first to make sure that all scaffolds were taken successfully. As we have done before, they were combined to generate an integrated gff, proteins.fasta, transcripts.fasta first by creating this maker_combine.sh bash scripts file.

```
#!/bin/bash
```

```
#PBS -N Run_MAKER_COMBINE
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00

#========================= SETUP MAKER ========================
swsetup () { eval `/usr/erc-share/etc/swsetup/swsetup.pl $*`; }
swsetup maker
swsetup augustus
PERL5LIB=/usr/local/igbb/maker/lib/perl5:$PERL5LIB
PERL5LIB=/usr/local/igbb/genemark-es-et_4.30/lib/perl5:$PERL5LIB
#--------------------------------------------------------------

# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd /work/didiren/

base=maker2_second_run
# TO MERGE ALL GFF3 FRILE FROM EACH SCAFFOLD TO ONE GFF3 FILE
gff3_merge -d $base.maker.output/$base\_master_datastore_index.log
#TO MERGE ALL FASTA FRILE FROM EACH SCAFFOLD TO ONE FASTA FILE
fasta_merge -d $base.maker.output/$base\_master_datastore_index.log
```

Now submit the job to the sever by typing this command at the directory of where your maker_combine.sh file is.

```
qsub maker_combine.sh
```

4 Gene name and function assignment

4.1 Description

The most important output files from the MAKER2-Second pass are maker2_second_run.all.gff, maker2_second_run.all.transcripts.fasta, and maker2_second_run.all.proteins.fasta, which can be forward to functional prediction and name assignment step. However, we strongly recommended you to generate pure CDS, exons and genes FASTA files also using a perl scripts named gff2fasta.pl from GitHub (https://github.com/minillinim/gff2fasta/blob/master/gff2fasta.pl).

4.2 To assign gene name and functions

By creating a bash scripts file named aftermaker.sh, unique IDs were assigned to each gene in gff file as well as fasta files. Functional information for each gene, like Uniprot homology protein names and InterproScan ID were also appended in this job.

```
#!/bin/bash
#PBS -N Run_
#PBS -l nodes=1:ppn=20
#PBS -l walltime=48:00:00
# CHANGE DIRECTORY TO WHERE JOB WAS SUBMITTED
cd $PBS_O_WORKDIR
#EXTRACT GENE,CDS,EXON,PROTEIN FASTA FROM NEW GENERATED GFF FILE
#THIS STEP WILL CREATE FIVE FILES
perl gff2fasta.pl /work/didiren/Cryphonectria.genome.fasta maker2_second_run.all.gff 2017versionfasta

#CREATE UNIQUE ID FOR GFF3 FILE (COMMANDS IN ONE LINE)
/usr/local/igbb/maker/bin/maker_map_ids --prefix cp_ep155_ --abrv_gene G --abrv_tran T --suffix _
--iterate 1 maker2_second_run.all.gff > genome.all.id.map

#MAP ID TO EACH GENE IN GFF3 FILE (COMMANDS IN ONE LINE)

/usr/local/igbb/maker/bin/map_gff_ids genome.all.id.map maker2_second_run.all.gff

#MAP ID TO EACH GENE IN ALL FASTA FILE (COMMANDS IN ONE LINE)
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map maker2_second_run.all.maker.proteins.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map maker2_second_run.all.maker.transcripts.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map 2017versionfast.gene.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map 2017versionfast.cdna.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map 2017versionfast.CDs.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map 2017versionfast.exon.fasta
/usr/local/igbb/maker/bin/map_fasta_ids genome.all.id.map 2017versionfast.pep.fasta

#MAKER BLAST DATABASE USE UNIPROT/SWISS PROTIENS DATASET (COMMANDS IN ONE LINE)
/usr/local/igbb/blast/bin/makeblastdb -in uniprot_sprot.fasta.gz -dbtype prot -out uniprotdb
#RUN BLASTP USING MAXIUM THREADS (COMMANDS IN ONE LINE)
/usr/local/igbb/blast/bin/blastp -num_threads 20 -max_target_seqs 1 -query maker2_second_run.all.proteins.fasta -outfmt
6 -db uniprotdb -out maker2_second_run.blastpout
#RUN INTERPROSCAN USING (COMMANDS IN ONE LINE)
/PATH/TO/interproscan-5.22-61.0/interproscan.sh -f TSV, HTML -goterms -iprlookup -i maker2_second_run.all.proteins.fasta
-b /work/didiren/interproscan/maker2_second_run_interproout

#ADD PROTEIN HOMOLOGY TO GFF3 AND FASTA FILE (COMMANDS IN ONE LINE)
```

```
/usr/local/igbb/maker/bin/maker_functional_gff      uniprot_sprot.fasta.gz      maker2_second_run.blastpout
maker2_second_run.all.gff > maker2_second_run.all.functional.gff
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
maker2_second_run.all.maker.proteins.fasta > maker2_second_run.all.functional.proteins.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
maker2_second_run.all.maker.transcripts.fasta > maker2_second_run.all.functional.transcripts.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
2017versionfast.gene.fasta > 2017versionfast.gene.functional.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
2017versionfast.pep.fasta > 2017versionfast.pep.functional.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
2017versionfast.cdna.fasta > 2017versionfast.cdna.functional.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
2017versionfast.CDs.fasta > 2017versionfast.CDs.functional.fasta
/usr/local/igbb/maker/bin/maker_functional_fasta     uniprot_sprot.fasta.gz     maker2_second_run.blastpout
2017versionfast.exon.fasta > 2017versionfast.exon.functional.fasta

#ADD INTERPROSCAN ID AND GO TERMS TO GFF3 (COMMANDS IN ONE LINE)
/usr/local/igbb/maker/bin/ipr_update_gff                                    maker2_second_run_interproout.tsv
maker2_second_run.all.functional.gff > maker2_second_run.all.functional.ipr.gff
```

Now submit the job to the sever by typing this command at the directory of where your aftermaker.sh file is.

```
qsub aftermaker.sh
```