The detailed tour of running PEPG

1 Adding MAKER2's Quality metrics to Preexisting Annotation

1.1 Description

With all new RNA and Protein homology evidence, MAKER2 offers a way to accomplish measuring the quality of gene prediction from any preexisting annotation by adding the same quality metrics, like Annotation Edit Distance (AED) and mRNA Quality index (QI). AED is a number between 0 and 1, with an AED score of 0 indicating a perfect prediction with available evidence and a value of 1 without any evidence supporting the annotated gene model. A preexisting annotation file in GFF3 format and the same kinds of evidence we used above for consistency were used here.

1.2 Running MAKER2-Legacy

With the same strategy like normal MAKER2 one-pass, we need to edit the three control files for MAKER2 to get access to the genome, RNA, Proteins evidence. However, there is no need to utilize any gene predictor because preexisting annotation provided gene models for MAKER2 to exam with the evidence it has.

1.2.1 Editing three control files

With three control files existing in didiren/ folder, the maker_opts.ctl file was edited to specify all inputs, like Genome sequence, RNA evidence, Protein evidence and the preexisting GFF3 file.

```
genome=/work/didiren/Cparasitica.genome.fasta
organism_type=eukaryotic
est=/work/didiren/assembled.transcriptome.fasta
protein=/work/didiren/uniprot_sprot.fasta.gz
model_gff=/work/didi/Cparasiticav2.GeneCatalog20091217.gff3
est2genome=0
protein2genome=0
```

 # 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*
#model_gff	*to give external preexisting annotated gene models*
#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*

1.2.2 To run MAKER2-Legacy

Once maker_opts.ctl have been edited, run Maker2 by creating this bash script file maker_legacy.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_legacy
# -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_legacy.sh file is.

```
qsub maker_legacy.sh
```

1.2.3 MAKER2-Legacy outputs

Now in the current working directory, MAKER2 have created a folder named maker2_legacy.maker.output. Inside this directory, maker2_legacy_master_datastore_index.log need to be 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_combine1.sh bash scripts file.

```
#!/bin/bash
#PBS -N Run_MAKER_COMBINE1
#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_legacy
# 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

#RUN INTERPROSCAN USING (COMMANDS IN ONE LINE)
/PATH/TO/interproscan-5.22-61.0/interproscan.sh -f TSV, HTML -goterms -
iprlookup –i maker2_legacy.maker.output.all.proteins.fasta -b
/work/didiren/interproscan/maker2_legacy_interproout
#ADD INTERPROSCAN ID AND GO TERMS TO GFF3 (COMMANDS IN ONE LINE)
/usr/local/igbb/maker/bin/ipr_update_gff maker2_legacy_interproout.tsv
maker2_legacy.maker.all.gff > maker2_legacy.all.ipr.gff
```

Now submit the job to the sever by typing this command at the directory of where your maker_combine1.sh file is.

```
   qsub maker_combine1.sh
```

2 AED Comparison between preexisting and new annotation

2.1 Extraction of AED score from the GFF3 file.

In a GFF3 format file, there are 9 columns, containing seqid, source, type, start, end, score, strand, phase and attributes, separately. In the ninth column, gene ID, gene name, AED score, etc were presented by a semicolon-separated list. A R script were written to extract AED score using regulator expression function as followed.

```
#IN R STUDIO, CREATE A AEDSCORE FUNCTION TO EXTRACT AED VALUES FROM EACH GENE.

AEDscore <- function(gfffile){
#obtain maker.gff file to see how many genes it has and their AED scores
maker.gff <- read.table(gfffile,sep='\t')
#ONLY EXTRACT THE COLUMN 9 OF GFF FILE
maker.gff.col9 <- maker.gff[,9]
pattern <- "_AED=([0-9].[0-9][0-9])"
maker.gff.AED=c()
index=1
for (i in 1:length(maker.gff.col9)){
  maker.gff.match <- regexec(pattern, as.character(maker.gff[i,9]))
  if (maker.gff.match[[1]][1] != -1){
    maker.gff.AED[index]<-
regmatches(as.character(maker.gff[i,9]),maker.gff.match)[[1]][2]
    index = index + 1
  }
}
return (maker.gff.AED)
}
```

2.2 Distribution of AED score presenting Gene Prediction quality

R installed function ecdf{} was used to perform cumulative distribution of the AED scores from the whole genome. Subsequently, plot{} function were used to demonstrate the differences of gene prediction qualities from two annotated genome.

```
#RUN AEDSCORE FUNCTION WITH NEW GFF3 AND PREEXISTING GFF3 FILES
maker2.new.AED <- AEDscore('maker2_second_run.all.functional.ipr.gff')
```

```
    maker2.legacy.AED <- AEDscore('maker2_legacy.all.gff')

    #ECDF{} FUNCTION WAS USED TO DO CUMULATIGVE DISTRIBUTION
    maker2.new.AED.plot<-ecdf(maker2.new.AED)
    maker2.legacy.AED.plot <- ecdf(maker2.legacy.AED)

    #RUN PLOT{} TO VISULIZE THE DIFFERENCE OF TWO ANNOTATIONS' QUALITY
    plot(maker2.new.AED.plot, pch=. ,col = "red", main="Cumulative distribution of AED
Value", xlab ="AED value", ylab="Cumulative fraction of annotation", xlim=(0:1),
ylim=(0:1))
    lines(maker2.legacy.AED.plot,col='green',pch=.)
    legend('bottomright',legend=c('maker2.new.AED.plot','maker2.legacy.AED.plot'),
lty=1, col = c('red','green'), bty='n',cex=.75)
```

3 Run the PEPG.py with two annotation GFF files

3.1 Prepare two GFF files

There are only two files should be provided for the PEPG.py step. The maker2_legacy.all.ipr.gff and the maker2_newer.all.ipr.gff two files for example in my case.

```
    python PEPG.py –g1 maker2_legacy.all.ipr.gff –g2 maker2_newer.all.ipr.gff
```

#-g1 provide the first annotation file name, which the prior annotation
#-g2 provide the second annotation file name, which the newer annotation
#The order is critical.

3.2 Outputs of PEPG.py

There were four categories, Match, Similar, Different, Noexist that the predicted gene models from the prior version were sorted into depending on the discrepancies of its coding region (start/end coordinates) and coding domains (InterPro ID) with the newer version. The script ultimately generated a text file for each category with the gene model ID, the coordinates, the AED scores and the protein homolog names from both annotation versions attached.