GO\_annotation\_scripts

## Convert gff to gtf using gffread

#gffread conversion  
gffread -E Qrob\_PM1N\_genes\_20161004.gff -T -o Qrob\_PM1N\_genes\_20161004.gtf  
  
#edit into correct format for trinotate preparation  
<Qrob\_PM1N\_genes\_20161004.gtf awk -F '\\;' '{print $1}' > edit1\_Qrob\_PM1N\_genes\_20161004.gtf  
  
<edit1\_Qrob\_PM1N\_genes\_20161004.gtf awk '{print $0,$9,$10}' > trinotate\_Qrob\_PM1N\_genes\_20161004.gtf  
  
sed -i 's/transcript\_id/gene\_id/' trinotate\_Qrob\_PM1N\_genes\_20161004.gtf   
  
rm edit1\_Qrob\_PM1N\_genes\_20161004.gtf

## Prepare files required for trinotate

GO\_analysis/Trinotate/util/Trinotate\_GTF\_or\_GFF3\_annot\_prep.pl \  
 --annot Reference/trinotate\_Qrob\_PM1N\_genes\_20161004.gtf \ #gtf  
 --genome\_fa Reference/Qrob\_PM1N.fa \ #ref genome  
 --out\_prefix GO\_analysis/trinotate #out file prefix  
  
#output files = gene-transcript map, protein fasta, transcript fasta

## Download databases

Trinotate/Trinotate --create \  
 --db myTrinotate.sqlite \ #database for GO data  
 --trinotate\_data\_dir TRINOTATE\_DATA\_DIR \ #directory for databases  
 --use\_diamond #use diamond for faster BLAST

## Transdecoder

#extract long open-reading frames (ORFs)  
TransDecoder.LongOrfs -t trinotate.transcripts.cdna.fa  
  
#make a BLAST database for homology search  
makeblastdb -in uniprot\_sprot.fasta -dbtype prot -blastdb\_version 5  
  
#blastp homology using BLAST +  
blastp -query ${OUT\_DIR}/longest\_orfs.pep -db uniprot\_sprot.fasta -outfmt 6 -evalue 1e-5 -num\_threads 10 > ${OUT\_DIR}/blastp.outfmt6  
  
#pfam search using HMMER3   
hmmsearch --cpu 8 -E 1e-10 \  
 --domtblout ${OUT\_DIR}/pfam.domtblout \   
 ${HMM\_DATADIR}/Pfam-A.hmm \ #pfam database to search against  
 ${OUT\_DIR}/longest\_orfs.pep #use longest orf peptides for search  
  
#retain hits from homology search  
#keeps both longest ORFs and regions of homology  
TransDecoder.Predict -t trinotate.transcripts.cdna.fa \  
 --retain\_pfam\_hits ${OUT\_DIR}/pfam.domtblout --retain\_blastp\_hits ${OUT\_DIR}/blastp.outfmt6

## Initialise sqlite database for GO terms

Trinotate --db myTrinotate.sqlite --init \  
 --gene\_trans\_map trinotate.gene-to-trans-map \ #gene-transcript   
 --transcript\_fasta trinotate.transcripts.cdna.fa \ #transcript fasta  
 --transdecoder\_pep trinotate.transcripts.cdna.fa.transdecoder.pep #protein fasta

## Perform main trinotate analysis/annotation

Trinotate --db myTrinotate.sqlite --run ALL --CPU 10 --transcript\_fasta trinotate.transcripts.cdna.fa \ #run ALL does all possible analysis based on installed tools  
 --transdecoder\_pep trinotate.transcripts.cdna.fa.transdecoder.pep --trinotate\_data\_dir TRINOTATE\_DATA\_DIR --use\_diamond #for faster BLAST searches

## Signal protein prediction

Done separately or else doesnt work.

#predict signalling peptides  
signalp6 --fastafile trinotate.transcripts.cdna.fa.transdecoder\_dir/longest\_orfs.pep --output\_dir sigP6outdir --format none --organism euk --mode fast --model\_dir "$BB\_APPS\_DATA/SignalP/models"  
  
#add to sqlite database  
Trinotate --db myTrinotate.sqlite --LOAD\_signalp sigP6outdir/output.gff3

## Generate final report

#create full excel report based on sqlite database  
Trinotate --db myTrinotate.sqlite --report > myTrinotate.xls

## Extract GO annotations

Trinotate/util/extract\_GO\_assignments\_from\_Trinotate\_xls.pl --Trinotate\_xls myTrinotate.xls --trans --include\_ancestral\_terms > ${OUT\_DIR}/trinotate\_GO\_terms.tsv  
  
#this step extracts all the GO annotations required for GO enrichment