## #Align Reads with Tophat

#all\_tophat.job

tophat2 Nlec\_genome\_v1\_NCBI\_reference.fasta FemaleA1\_R1.paired.trimmed12.fastq,FemaleA2\_R1.paired.trimmed12.fastq,FemaleA3\_R1.paired.trimmed12.fastq,FemaleA4\_R1.paired.trimmed12.fastq,FemaleO1\_R1.paired.trimmed12.fastq,FemaleO2\_R1.paired.trimmed12.fastq,FemaleO3\_R1.paired.trimmed12.fastq,FemaleO4\_R1.paired.trimmed12.fastq,FemaleH1\_R1.paired.trimmed12.fastq,FemaleH2\_R1.paired.trimmed12.fastq,FemaleH3\_R1.paired.trimmed12.fastq,FemaleH4\_R1.paired.trimmed12.fastq,FemaleL1\_R1.paired.trimmed12.fastq,FemaleL2\_R1.paired.trimmed12.fastq,FemaleL3\_R1.paired.trimmed12.fastq,FemaleL4\_R1.paired.trimmed12.fastq,FemaleM1\_R1.paired.trimmed12.fastq,FemaleM2\_R1.paired.trimmed12.fastq,FemaleM3\_R1.paired.trimmed12.fastq,FemaleM4\_R1.paired.trimmed12.fastq,FemaleT1\_R1.paired.trimmed12.fastq,FemaleT2\_R1.paired.trimmed12.fastq,FemaleT3\_R1.paired.trimmed12.fastq,FemaleT4\_R1.paired.trimmed12.fastq,MaleA1\_R1.paired.trimmed12.fastq,MaleA2\_R1.paired.trimmed12.fastq,MaleA3\_R1.paired.trimmed12.fastq,MaleA4\_R1.paired.trimmed12.fastq,MaleC1\_R1.paired.trimmed12.fastq,MaleC2\_R1.paired.trimmed12.fastq,MaleC4\_R1.paired.trimmed12.fastq,MaleH1\_R1.paired.trimmed12.fastq,MaleH2\_R1.paired.trimmed12.fastq,MaleH3\_R1.paired.trimmed12.fastq,MaleH4\_R1.paired.trimmed12.fastq,MaleL1\_R1.paired.trimmed12.fastq,MaleL2\_R1.paired.trimmed12.fastq,MaleL3\_R1.paired.trimmed12.fastq,MaleL4\_R1.paired.trimmed12.fastq,MaleM1\_R1.paired.trimmed12.fastq,MaleM2\_R1.paired.trimmed12.fastq,MaleM3\_R1.paired.trimmed12.fastq,MaleM4\_R1.paired.trimmed12.fastq,MaleT1\_R1.paired.trimmed12.fastq,MaleT2\_R1.paired.trimmed12.fastq,v7EB\_R1.paired.trimmed12.fastq,v8EB\_R1.paired.trimmed12.fastq,v12EB\_R1.paired.trimmed12.fastq,v17EB\_R1.paired.trimmed12.fastq,v7MB\_R1.paired.trimmed12.fastq,v8MB\_R1.paired.trimmed12.fastq,v12MB\_R1.paired.trimmed12.fastq,v17MB\_R1.paired.trimmed12.fastq,v7LB\_R1.paired.trimmed12.fastq,v8LB\_R1.paired.trimmed12.fastq,v12LB\_R1.paired.trimmed12.fastq,v17LB\_R1.paired.trimmed12.fastq,v7WB\_R1.paired.trimmed12.fastq,v8WB\_R1.paired.trimmed12.fastq,v12WB\_R1.paired.trimmed12.fastq,v17WB\_R1.paired.trimmed12.fastq,v7EH\_R1.paired.trimmed12.fastq,v8EH\_R1.paired.trimmed12.fastq,v12EH\_R1.paired.trimmed12.fastq,v17EH\_R1.paired.trimmed12.fastq,v7MH\_R1.paired.trimmed12.fastq,v8MH\_R1.paired.trimmed12.fastq,v12MH\_R1.paired.trimmed12.fastq,v17MH\_R1.paired.trimmed12.fastq,v7LH\_R1.paired.trimmed12.fastq,v8LH\_R1.paired.trimmed12.fastq,v12LH\_R1.paired.trimmed12.fastq,v17LH\_R1.paired.trimmed12.fastq,v7WH\_R1.paired.trimmed12.fastq,v8WH\_R1.paired.trimmed12.fastq,v12WH\_R1.paired.trimmed12.fastq,v17WH\_R1.paired.trimmed12.fastq FemaleA1\_R2.paired.trimmed12.fastq,FemaleA2\_R2.paired.trimmed12.fastq,FemaleA3\_R2.paired.trimmed12.fastq,FemaleA4\_R2.paired.trimmed12.fastq,FemaleO1\_R2.paired.trimmed12.fastq,FemaleO2\_R2.paired.trimmed12.fastq,FemaleO3\_R2.paired.trimmed12.fastq,FemaleO4\_R2.paired.trimmed12.fastq,FemaleH1\_R2.paired.trimmed12.fastq,FemaleH2\_R2.paired.trimmed12.fastq,FemaleH3\_R2.paired.trimmed12.fastq,FemaleH4\_R2.paired.trimmed12.fastq,FemaleL1\_R2.paired.trimmed12.fastq,FemaleL2\_R2.paired.trimmed12.fastq,FemaleL3\_R2.paired.trimmed12.fastq,FemaleL4\_R2.paired.trimmed12.fastq,FemaleM1\_R2.paired.trimmed12.fastq,FemaleM2\_R2.paired.trimmed12.fastq,FemaleM3\_R2.paired.trimmed12.fastq,FemaleM4\_R2.paired.trimmed12.fastq,FemaleT1\_R2.paired.trimmed12.fastq,FemaleT2\_R2.paired.trimmed12.fastq,FemaleT3\_R2.paired.trimmed12.fastq,FemaleT4\_R2.paired.trimmed12.fastq,MaleA1\_R2.paired.trimmed12.fastq,MaleA2\_R2.paired.trimmed12.fastq,MaleA3\_R2.paired.trimmed12.fastq,MaleA4\_R2.paired.trimmed12.fastq,MaleC1\_R2.paired.trimmed12.fastq,MaleC2\_R2.paired.trimmed12.fastq,MaleC4\_R2.paired.trimmed12.fastq,MaleH1\_R2.paired.trimmed12.fastq,MaleH2\_R2.paired.trimmed12.fastq,MaleH3\_R2.paired.trimmed12.fastq,MaleH4\_R2.paired.trimmed12.fastq,MaleL1\_R2.paired.trimmed12.fastq,MaleL2\_R2.paired.trimmed12.fastq,MaleL3\_R2.paired.trimmed12.fastq,MaleL4\_R2.paired.trimmed12.fastq,MaleM1\_R2.paired.trimmed12.fastq,MaleM2\_R2.paired.trimmed12.fastq,MaleM3\_R2.paired.trimmed12.fastq,MaleM4\_R2.paired.trimmed12.fastq,MaleT1\_R2.paired.trimmed12.fastq,MaleT2\_R2.paired.trimmed12.fastq,v7EB\_R2.paired.trimmed12.fastq,v8EB\_R2.paired.trimmed12.fastq,v12EB\_R2.paired.trimmed12.fastq,v17EB\_R2.paired.trimmed12.fastq,v7MB\_R2.paired.trimmed12.fastq,v8MB\_R2.paired.trimmed12.fastq,v12MB\_R2.paired.trimmed12.fastq,v17MB\_R2.paired.trimmed12.fastq,v7LB\_R2.paired.trimmed12.fastq,v8LB\_R2.paired.trimmed12.fastq,v12LB\_R2.paired.trimmed12.fastq,v17LB\_R2.paired.trimmed12.fastq,v7WB\_R2.paired.trimmed12.fastq,v8WB\_R2.paired.trimmed12.fastq,v12WB\_R2.paired.trimmed12.fastq,v17WB\_R2.paired.trimmed12.fastq,v7EH\_R2.paired.trimmed12.fastq,v8EH\_R2.paired.trimmed12.fastq,v12EH\_R2.paired.trimmed12.fastq,v17EH\_R2.paired.trimmed12.fastq,v7MH\_R2.paired.trimmed12.fastq,v8MH\_R2.paired.trimmed12.fastq,v12MH\_R2.paired.trimmed12.fastq,v17MH\_R2.paired.trimmed12.fastq,v7LH\_R2.paired.trimmed12.fastq,v8LH\_R2.paired.trimmed12.fastq,v12LH\_R2.paired.trimmed12.fastq,v17LH\_R2.paired.trimmed12.fastq,v7WH\_R2.paired.trimmed12.fastq,v8WH\_R2.paired.trimmed12.fastq,v12WH\_R2.paired.trimmed12.fastq,v17WH\_R2.paired.trimmed12.fastq

#creates tophat\_out which contains directory:

#accepted\_hits.bam  align\_summary.txt  deletions.bed  insertions.bed  junctions.bed  **logs**  prep\_reads.info  unmapped.bam

## #Use accepted\_hits.bam as input for Trinity (but sort it first)

#TrinGG\_all.job

echo 'sorting'

samtools sort accepted\_hits.bam > accepted\_hits.coorsort.bam

echo 'done sorting'

Trinity --genome\_guided\_bam accepted\_hits.coorsort.bam --genome\_guided\_max\_intron 10000  --max\_memory 20G --CPU 24

#Outputs Trinity-GG.fasta (contains 132243 Trinity contigs)

## #Cluster reads to reduce redundancy

#oneGG\_cdhit.job

cd-hit -i Trinity-GG.fasta -o Trinity-GG\_clustered.fasta

#Outputs Trinity-GG\_clustered.fasta (contains 109375 Trinity contigs)

## #Align and Estimate Abundance for the 77 samples against the clustered transcriptome

#base\_aea.job

align\_and\_estimate\_abundance.pl --transcripts Trinity-GG\_clustered.fasta --seqType fq --single base1\_R1.paired.trimmed12.fastq --SS\_lib\_type F --est\_method RSEM --aln\_method bowtie2 --output\_dir base1R1

align\_and\_estimate\_abundance.pl --transcripts Trinity-GG\_clustered.fasta --seqType fq --single base2\_R1.paired.trimmed12.fastq --SS\_lib\_type F --est\_method RSEM --aln\_method bowtie2 --output\_dir base2R1

align\_and\_estimate\_abundance.pl --transcripts Trinity-GG\_clustered.fasta --seqType fq --single base3\_R1.paired.trimmed12.fastq --SS\_lib\_type F --est\_method RSEM --aln\_method bowtie2 --output\_dir base3R1

align\_and\_estimate\_abundance.pl --transcripts Trinity-GG\_clustered.fasta --seqType fq --single base4\_R1.paired.trimmed12.fastq --SS\_lib\_type F --est\_method RSEM --aln\_method bowtie2 --output\_dir base4R1

#aeajobs.py (script makes a job file for each sample type (Female Heads, Male Heads, Early Larval Heads, etc)

infilename='base\_aea.job'

infile=open(infilename, 'r')

content=infile.read()

infile.close()

larval=['EB','MB','LB','WB','EH','MH','LH','WH']

adult=['A','H','L','M','T']

sexes=['Female','Male']

for stagetiss in larval:

    towrite=content

    base1='v7'+stagetiss

    base2='v8'+stagetiss

    base3='v12'+stagetiss

    base4='v17'+stagetiss

    towrite=towrite.replace('base1',base1)

    towrite=towrite.replace('base2',base2)

    towrite=towrite.replace('base3',base3)

    towrite=towrite.replace('base4',base4)

    outfilename=infilename

    outfilename=outfilename.replace('base',stagetiss)

    outfile=open(outfilename, 'w')

    outfilename=outfile.write(towrite)

    outfile.close()

for sex in sexes:

    if sex=='Female':

        abb='F'

        tissues=adult

        tissues.append('O')

    else:

        abb='M'

        tissues=adult

        tissues.append('C')

    for tiss in tissues:

        towrite=content

        base=sex+tiss

        towrite=towrite.replace('base',base)

        outfilename=infilename

        outfilename=outfilename.replace('base',base)

        outfile=open(outfilename, 'w')

        outfilename=outfile.write(towrite)

        outfile.close()

#prepref.job (creates all job files, preps transcriptome for alignment, and submits job files)

python aeajobs.py

align\_and\_estimate\_abundance.pl --transcripts Trinity-GG\_clustered.fasta --seqType fq --single v7EB\_R1.paired.trimmed12.fastq --SS\_lib\_type F --est\_method RSEM --aln\_method bowtie2 --output\_dir prep --prep\_reference

./submitaea

#Identify isoforms with at least 1 transcript per million in at least 2 samples (because of low mapping of Male Thorax 2 to the genome, this sample was excluded when meeting this criteria)

#TPM1\_isoforms\_secondTPM1sample.py

import time

import re

from re import findall

Pre='>'+'[\S]+'

tabre='[^\t]+'

spacere='[\S]+'

speciesre='>'+'[^?]+'

partre='[^\_]+'

TPM1list=[]

TPM1dict={}

infilename='Trinity-GG\_clustered\_noMT2\_genes.TPM.not\_cross\_norm'

infile=open(infilename, 'r')

for line in infile:

    if 'TRINITY' in line:

        tabs=findall(tabre,line)

#print(tabs[0])

        num=1

        TPM1='no'

        second='no'

        while num < len(tabs):

            #print(num)

    if float(tabs[num]) >= 1:

                if TPM1 == 'yes':

  second='yes'

                else:

                   TPM1='yes'

            num=num+1

        if second=='yes':

            TPM1list.append(tabs[0])

            TPM1dict[tabs[0]]=tabs[0]

print(len(TPM1list))

print(TPM1list[0])

print('making new fasta')

infilename='Trinity-GG\_clustered.fasta'

infile=open(infilename, 'r')

outfilename='Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta'

infile=open(infilename, 'r')

outfile=open(outfilename, 'w')

counter=0

for line in infile:

    if '>' in line:

        TPM1='false'

        tabs=findall(spacere,line)

        trinity=tabs[0].replace('>','')

        #parts=findall(partre,trinity)

#trinity=parts[0]+'\_'+parts[1]+'\_'+parts[2]+'\_'+parts[3]

#print(trinity)

if trinity in TPM1list:

            #print(trinity,TPM1dict[trinity])

            TPM1='true'

    counter=counter+1

    if TPM1=='true':

        outfile.write(line)

infile.close()

outfile.close()

print(counter)

#58,106 contigs have at least 1 TPM in at least 2 samples (not including MT2)

## #Using the 16,603 contigs (Trinity-GG\_clustered\_isoformsTPM1\_2samples\_blastx-3.fasta)

#>TPM1 in >1 sample (not including MT2) AND blastx-3 hits to lecontei or nr\_insects for alignment and abundance estimates

#lec.job

#!/bin/bash

#SBATCH -N 1

#SBATCH -n 16

#SBATCH -p Long

#SBATCH -t 2-00:00:00

#SBATCH --mail-type ALL

#SBATCH --no-requeue

#makeblastdb -in GCF\_001263575.1\_Nlec1.0\_protein.faa -parse\_seqids -dbtype prot

blastx -query Trinity-GG\_clustered\_isoformsTPM1.fasta -db GCF\_001263575.1\_Nlec1.0\_protein.faa -out Trinity-GG\_clustered\_isoTPM1\_GCFlec-3.outfmt6 -evalue 1e-3 -num\_threads 6 -max\_target\_seqs 1 -outfmt 6

#idtrinity\_lechits.py

import time

import re

from re import findall

Pre='>'+'[\S]+'

tabre='[^\t]+'

spacere='[\S]+'

speciesre='>'+'[^?]+'

trinitygenere='TRINITY\_GG\_'+'[0-9]+'+'\_c'+'[0-9]+'+'\_g'+'[0-9]+'

matchname={}

infilename='Trinity-GG\_clustered\_isoTPM1\_GCFlec-3.outfmt6'

infile=open(infilename, 'r')

for line in infile:

    tabs=findall(spacere,line)

    trinitygene=findall(trinitygenere,tabs[0])

    trinitylength=abs(int(tabs[9])-int(tabs[8]))

    percentcov=float(tabs[3])/trinitylength

    print(percentcov,line)

    if percentcov > 0.5:

        matchname[trinitygene[0]]=tabs[1]

infile.close()

print(len(matchname.keys()))

matchset=set(matchname.keys())

infilename='Trinity-GG\_clustered\_isoformsTPM1.fasta'

outfilename1='Trinity-GG\_clustered\_isoformsTPM1\_notGCFlec-3.fasta'

outfilename2='Trinity-GG\_clustered\_isoformsTPM1\_GCFlec-3.fasta'

infile=open(infilename, 'r')

outfile1=open(outfilename1, 'w')

outfile2=open(outfilename2, 'w')

for line in infile:

    if '>' in line:

        tabs=findall(spacere,line)

        trinitygene=findall(trinitygenere,tabs[0])

        trinitygene=trinitygene[0].replace('>','')

        if trinitygene not in matchset:

            nomatch='true'

        else:

            nomatch='false'

    if nomatch=='true':

        outfile1.write(line)

    else:

        if '>' in line:

            towrite='>'+matchname[trinitygene]

            outfile2.write(towrite)

        outfile2.write(line)

infile.close()

outfile1.close()

outfile1.close()

#insectbase.job

blastx -query TrinityGGclTPM1notGCFlec-3.fasta -db nr\_insects.txt -out Trinity-GG\_isoTPM1notGCFlec-3\_insect-3.outfmt6 -evalue 1e-3 -num\_threads 6 -max\_target\_seqs 1 -outfmt 6

cat Trinity-GG\_clustered\_isoTPM1\_GCFlec-3.outfmt6 Trinity-GG\_isoTPM1notGCFlec-3\_insect-3.> Trinity-GG\_clustered\_isoTPM1\_GCFlec-3\_insect-3.outfmt6

#idtrinity\_blastx.py

import time

import re

from re import findall

Pre='>'+'[\S]+'

tabre='[^\t]+'

spacere='[\S]+'

speciesre='>'+'[^?]+'

trinityisore='TRINITY\_GG\_'+'[0-9]+'+'\_c'+'[0-9]+'+'\_g'+'[0-9]+'+'\_i'+'[0-9]+'

matchname={}

infilename='Trinity-GG\_clustered\_isoTPM1\_GCFlec-3\_insect-3.outfmt6'

infile=open(infilename, 'r')

for line in infile:

    tabs=findall(spacere,line)

    trinityiso=findall(trinityisore,tabs[0])

    trinitylength=abs(int(tabs[9])-int(tabs[8]))

    percentcov=float(tabs[3])/trinitylength

    print(percentcov,line)

    if percentcov > 0.5:

        matchname[trinityiso[0]]=tabs[1]

infile.close()

print(len(matchname.keys()))

matchset=set(matchname.keys())

infilename='Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta'

outfilename1='Trinity-GG\_clustered\_isoformsTPM1\_2samples\_notblastx-3.fasta'

outfilename2='Trinity-GG\_clustered\_isoformsTPM1\_2samples\_blastx-3.fasta'

infile=open(infilename, 'r')

outfile1=open(outfilename1, 'w')

outfile2=open(outfilename2, 'w')

for line in infile:

    if '>' in line:

        tabs=findall(spacere,line)

        trinityiso=findall(trinityisore,tabs[0])

        trinityiso=trinityiso[0].replace('>','')

        if trinityiso not in matchset:

            nomatch='true'

        else:

            nomatch='false'

    if nomatch=='true':

        outfile1.write(line)

    else:

        if '>' in line:

            towrite='>'+matchname[trinityiso]

            outfile2.write(towrite)

        outfile2.write(line)

infile.close()

outfile1.close()

outfile1.close()

## #Using the X contigs that have >TPM1 in >1 sample (not including MT2) AND blastx-3 hits to lecontei or nr\_insects for alignment and abundance estimates AND contigs that are retained when removing reads that map well to contigs with blastx-3 hits

#makesams.job

#also submit jobs with 2nd, 3rd, and 4th replicates for the aea directory and the blastx-3 directory

samtools view -h MaleA1R1/bowtie2.bam > MaleA1R1/bowtie2.sam

samtools view -h MaleC1R1/bowtie2.bam > MaleC1R1/bowtie2.sam

samtools view -h MaleH1R1/bowtie2.bam > MaleH1R1/bowtie2.sam

samtools view -h MaleL1R1/bowtie2.bam > MaleL1R1/bowtie2.sam

samtools view -h MaleM1R1/bowtie2.bam > MaleM1R1/bowtie2.sam

samtools view -h MaleT1R1/bowtie2.bam > MaleT1R1/bowtie2.sam

samtools view -h FemaleA1R1/bowtie2.bam > FemaleA1R1/bowtie2.sam

samtools view -h FemaleO1R1/bowtie2.bam > FemaleO1R1/bowtie2.sam

samtools view -h FemaleH1R1/bowtie2.bam > FemaleH1R1/bowtie2.sam

samtools view -h FemaleL1R1/bowtie2.bam > FemaleL1R1/bowtie2.sam

samtools view -h FemaleM1R1/bowtie2.bam > FemaleM1R1/bowtie2.sam

samtools view -h FemaleT1R1/bowtie2.bam > FemaleT1R1/bowtie2.sam

samtools view -h v12EBR1/bowtie2.bam > v12EBR1/bowtie2.sam

samtools view -h v12MBR1/bowtie2.bam > v12MBR1/bowtie2.sam

samtools view -h v12LBR1/bowtie2.bam > v12LBR1/bowtie2.sam

samtools view -h v12WBR1/bowtie2.bam > v12WBR1/bowtie2.sam

samtools view -h v12EHR1/bowtie2.bam > v12EHR1/bowtie2.sam

samtools view -h v12MHR1/bowtie2.bam > v12MHR1/bowtie2.sam

samtools view -h v12LHR1/bowtie2.bam > v12LHR1/bowtie2.sam

samtools view -h v12WHR1/bowtie2.bam > v12WHR1/bowtie2.sam

#FindUniqueAndNew.py

import re

from re import findall

larval=['EB','MB','LB','WB','EH','MH','LH','WH']

adult=['A','H','L','M','T']

sexes=['Female','Male']

larvalnames=['v7','v8','v12','v17']

tabre='[\S]+'

def runcomparisons(basename):

        All={}

        Blasted={}

        infilename='aea/'+basename

        logfilename=logfile.write(infilename+'\n')

        infile=open(infilename, 'r')

        for line in infile:

            if 'K00363:' in line:

                tabs=findall(tabre,line)

                All[tabs[0]]=tabs[2]

        infile.close()

        infilename='blastx-3/'+basename

        logfilename=logfile.write(infilename+'\n')

        infile=open(infilename, 'r')

        for line in infile:

            if 'K00363:' in line:

                tabs=findall(tabre,line)

                Blasted[tabs[0]]=tabs[2]

        infile.close()

        logfilename=logfile.write(infilename+'\n')

        AllReads=set(All.keys())

        BlastedReads=set(Blasted.keys())

        OnlyAllReads=AllReads-BlastedReads

        logfilename=logfile.write(str(len(OnlyAllReads))+'\n')

        for Read in OnlyAllReads:

            new.append(All[Read])

logfilename='log'

logfile=open(logfilename, 'w')

new=[]

for stagetiss in larval:

    for name in larvalnames:

        basename=name+stagetiss+'R1/bowtie2.sam'

        runcomparisons(basename)

for sex in sexes:

    if sex=='Female':

        abb='F'

        tissues=adult

        tissues.append('O')

    else:

        abb='M'

        tissues=adult

        tissues.append('C')

    for tiss in tissues:

        counter=0

        while counter < 4:

            counter=counter+1

            basename=sex+tiss+str(counter)+'R1/bowtie2.sam'

            runcomparisons(basename)

logfilename=logfile.write(len(new)+'\n')

new=set(new)

logfilename=logfile.write(len(new)+'\n')

infilename='Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta'

logfilename=logfile.write(infilename+'\n')

logfile.close()

infile=open(infilename, 'r')

outfilename='uniqueTOnonblasted.fasta'

outfile=open(outfilename, 'w')

for line in infile:

    if 'TRINITY' in line:

        keep='no'

        tabs=findall(tabre,line)

        contig=tabs[0].replace('>','')

        if contig in new:

            keep='yes'

    if keep == 'yes':

        outfilename=outfile.write(line)

infile.close()

outfile.close()

## Blast to known lecontei and insect genes

makeblastdb -in AAmanAAGCF\_remove90pidtomanual.fasta -parse\_seqids -dbtype prot

blastx -query Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta -db AAmanAAGCF\_remove90pidtomanual.fasta -out Trinity-GG\_clustered\_isoTPM1\_blastxmanGCF.outfmt6 -evalue 1e-3 -num\_threads 6 -max\_target\_seqs 1 -outfmt 6

makeblastdb -in nr\_insects.txt -parse\_seqids -dbtype prot

blastx -query Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta -db nr\_insects.txt -out Trinity-GG\_isoTPM1\_insects-3.outfmt6 -evalue 1e-3 -num\_threads 6 -max\_target\_seqs 1 -outfmt 6

idblastx\_GCF90pid\_bestmatch\_blastinfo.py

pid=90

import time

import re

from re import findall

Pre='>'+'[\S]+'

tabre='[^\t]+'

spacere='[\S]+'

speciesre='>'+'[^?]+'

trinityisore='TRINITY\_GG\_'+'[0-9]+'+'\_c'+'[0-9]+'+'\_g'+'[0-9]+'+'\_i'+'[0-9]+'

lecdict={}

lec90={}

infilename='Trinity-GG\_clustered\_isoTPM1\_blastxmanGCF.outfmt6'

infile=open(infilename, 'r')

for line in infile:

    tabs=findall(spacere,line)

    trinityiso=findall(trinityisore,tabs[0])

    #print(line)

    if float(tabs[2]) >= pid:

        lec90[trinityiso[0]]=tabs[1]

    lecdict[trinityiso[0]]=line

infile.close()

print('lec pid',len(lec90.keys()))

lec90set=set(lec90.keys())

print('lec hits',len(lecdict.keys()))

lecset=set(lecdict.keys())

infilename='Trinity-GG\_clustered\_isoformsTPM1\_2samples.fasta'

outfilename1='Trinity-GG\_clustered\_isoformsTPM1in2samples\_iso\_notGCFlec'+str(pid)+'.fasta'

outfilename2='Trinity-GG\_clustered\_isoformsTPM1in2samples\_iso\_GCFlec'+str(pid)+'\_blastinfo.fasta'

infile=open(infilename, 'r')

outfile1=open(outfilename1, 'w')

outfile2=open(outfilename2, 'w')

for line in infile:

    if '>' in line:

        tabs=findall(spacere,line)

        trinityiso=findall(trinityisore,tabs[0])

        trinityiso=trinityiso[0].replace('>','')

        if trinityiso not in lec90set:

            nomatch='true'

        else:

            nomatch='false'

    if nomatch=='true':

        outfile1.write(line)

    else:

        if '>' in line:

            tempinfo=lecdict[trinityiso].replace('\n','')

            towrite='>lec90'+lec90[trinityiso]+'|'+tempinfo+'|'

            outfile2.write(towrite)

        outfile2.write(line)

infile.close()

outfile1.close()

outfile2.close()

insectdict={}

infilename='Trinity-GG\_isoTPM1\_insects-3.outfmt6'

infile=open(infilename, 'r')

for line in infile:

    tabs=findall(spacere,line)

    trinityiso=findall(trinityisore,tabs[0])

    insectdict[trinityiso[0]]=line

infile.close()

insectset=set(insectdict.keys())

print('insect',len(insectdict.keys()))

insectset=set(insectdict.keys())

print(insectdict[tabs[0]])

matchname={}

infilename='Trinity-GG\_clustered\_isoformsTPM1in2samples\_iso\_notGCFlec'+str(pid)+'.fasta'

outfilename1='Trinity-GG\_clustered\_isoformsTPM1in2samples\_iso\_notGCFlec'+str(pid)+'\_unmatched.fasta'

outfilename2='Trinity-GG\_clustered\_isoformsTPM1in2samples\_iso\_notGCFlec'+str(pid)+'\_bestmatch\_blastinfo.fasta'

infile=open(infilename, 'r')

outfile1=open(outfilename1, 'w')

outfile2=open(outfilename2, 'w')

for line in infile:

    if '>' in line:

        tabs=findall(spacere,line)

        trinityiso=findall(trinityisore,tabs[0])

        trinityiso=trinityiso[0].replace('>','')

        #print(trinityiso)

        if trinityiso in lecset and trinityiso in insectset:

            nomatch='false'

            lectabs=findall(tabre,lecdict[trinityiso])

            leceval=float(lectabs[10])

            insecttabs=findall(tabre,insectdict[trinityiso])

            insecteval=float(insecttabs[10])

            print(leceval,insecteval)

            if leceval <= insecteval:

                matchname[trinityiso]=lectabs[1]

                tempinfo=lecdict[trinityiso].replace('\n','')

                towrite='>lec'+matchname[trinityiso]+'|'+tempinfo+'|'

                #print('lec better',line)

            elif insecteval < leceval:

                matchname[trinityiso]=insecttabs[1]

                tempinfo=insectdict[trinityiso].replace('\n','')

                towrite='>insect'+matchname[trinityiso]+'|'+tempinfo+'|'

                #print('insect better',line)

        elif trinityiso in lecset:

            nomatch='false'

            lectabs=findall(tabre,lecdict[trinityiso])

            matchname[trinityiso]=lectabs[1]

            tempinfo=lecdict[trinityiso].replace('\n','')

            towrite='>lec'+matchname[trinityiso]+'|'+tempinfo+'|'

            #print('lec only',line)

        elif trinityiso in insectset:

            nomatch='false'

            insecttabs=findall(tabre,insectdict[trinityiso])

            matchname[trinityiso]=insecttabs[1]

            tempinfo=insectdict[trinityiso].replace('\n','')

            towrite='>insect'+matchname[trinityiso]+'|'+tempinfo+'|'

            #print('insect only',line)

        else:

            nomatch='true'

    if nomatch=='true':

        outfile1.write(line)

    else:

        if '>' in line:

            outfile2.write(towrite)

        outfile2.write(line)

infile.close()

outfile1.close()

outfile1.close()