#!/bin/usr/python3

# Requires Biopython,BCBio-GFF

# Note that you must move the results of interest

# to their own folder

from Bio import SeqIO

from Bio.Seq import Seq

from Bio.SeqRecord import SeqRecord

from Bio.SeqFeature import SeqFeature, FeatureLocation

from BCBio import GFF

from glob import glob

import os

prodigal\_folder="../prodigal-M/Mackay\_MAGs/"

gff\_ext="gff"

#results\_folder="/monfs00/scratch/lwoo0007/WoodsL/MS6-5\_hybrid\_seq\_sample/metabolic\_gene\_characterisation/MS6-5\_MAGs/test\_diamond\_Greening\_genes\_combined/"

results\_folder="Mackay\_MAGs/Mackay\_MAGs\_diamond\_Greening\_metabolic\_marker\_genes\_pident\_cutoffs\_for\_gff\_conversion"

op\_folder="Mackay\_MAGs\_Greening2021\_metabolic\_w\_diamond\_gffs/"

# Feature of interest

foi="metabolic\_gene\_DIAMOND\_hit"

# Check that input extension is okay and fix otherwise

if gff\_ext[0]!=".":

gff\_ext=".%s" % gff\_ext

# Make the ouput folder, if needed

if os.path.exists(op\_folder) == False:

os.mkdir(op\_folder)

for gff\_file in glob("%s/\*%s" % (prodigal\_folder, gff\_ext)):

sample\_name=gff\_file.split("/")[-1].replace(gff\_ext, "")

#blastp\_results="../metabolic\_gene\_characterisation/MS6-5\_MAGs/bin10029a\_diamond\_Greening\_metabolic\_marker\_genes/bin10029a\_metabolic\_markers\_DIAMOND\_ID\_cutoff\_50\_diamondDB\_result\_pident\_50.tsv"

blastp\_results=glob("%s/\*%s\*" % (results\_folder, sample\_name))[0]

op\_file="%s/%s.gff" % (op\_folder, sample\_name)

#op\_file="./test\_bin10029a\_metabolic\_DIAMOND\_hit\_filtered.gff"

# 1-based indexing

qseqid\_field="1"

sseqid\_field="3"

pident\_field="4"

qseqid\_field=int(qseqid\_field)

sseqid\_field=int(sseqid\_field)

pident\_field=int(pident\_field)

inp\_handle = open(gff\_file)

op\_handle = open(op\_file, 'w')

blastp\_data=[line.strip('\n') for line in open(blastp\_results)]

#print(blastp\_data)

new\_data=[]

records=[rec for rec in GFF.parse(inp\_handle)]

inp\_handle.close()

# SeqFeature(FeatureLocation(ExactPosition(400247), ExactPosition(400724), strand=-1), type='CDS', id='2\_402')]

for record in records:

new\_features=[]

#print(record)

for feat in record.features:

orf=str(feat.id).split("\_")[1]

#orf=feat.qualifiers["note"].split(";")[0].replace("\"ID=\*\_","")

#print(orf)

feature\_coord="%s\_%s" % (record.id, orf)

#print(feature\_coord)

for line in blastp\_data:

#print(line)

if line.split('\t')[(qseqid\_field-1)]==feature\_coord:

print("A MATCH!")

start=feat.location.start

end=feat.location.end

strand=feat.location.strand

hit=line.split('\t')[(sseqid\_field-1)]

pident=line.split('\t')[(pident\_field-1)]

new\_qualifiers={

"subject":[hit],

"score":[pident]

}

new\_feature=SeqFeature(FeatureLocation(start,end,strand),type=foi,qualifiers=new\_qualifiers)

new\_features.append(new\_feature)

new\_record=SeqRecord(id=str(record.id), seq=str(record.seq),features=new\_features)

new\_record.annotations={}

new\_data.append(new\_record)

GFF.write(new\_data, op\_handle)

inp\_handle.close()