#!/bin/usr/python3

# Requires Biopython, BCBio-GFF

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="./Mackay\_MAGs/"

hmm\_folder\_suffix="\_Resfams\_hmmsearch.tsv"

op\_folder="./Mackay\_MAGs\_resfams\_gffs"

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

os.mkdir(op\_folder)

# Feature of interest

foi="resfams\_HMMER\_hit"

# Check that input extension is okay and fix otherwise

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

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

qseqid\_field="1"

hmmhit\_field="3"

score\_field="8"

hmmacc\_field="4"

qseqid\_field=int(qseqid\_field)

hmmhit\_field=int(hmmhit\_field)

score\_field=int(score\_field)

hmmacc\_field=int(hmmacc\_field)

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

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

print("Working on %s" % (sample\_name))

#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"

results\_file="%s/%s%s" % (results\_folder, sample\_name,hmm\_folder\_suffix)

#print(results\_folder)

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

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

inp\_handle = open(gff\_file)

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

results\_data=['\t'.join(line.strip('\n').split()) for line in open(results\_file) if line[0]!="#"]

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 results\_data:

#print(line)

#print("results line: %s" % (line))

#print(line.split('\t')[(qseqid\_field-1)])

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

#print("A MATCH!")

start=feat.location.start

end=feat.location.end

strand=feat.location.strand

hmm\_hit=line.split('\t')[(hmmhit\_field-1)]

hmm\_accession=line.split('\t')[(hmmacc\_field-1)]

score=line.split('\t')[(score\_field)]

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

new\_qualifiers={

"hit":[hmm\_hit],

"hmm\_accession":[hmm\_accession],

"score":[score]

}

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()