#!/bin/python3

# Requires Biopython, pandas

from Bio import SeqIO

import os

from glob import glob

import pandas as pd

# Folder with mob recon outputs

mob\_dir="./"

# Mob output folder suffix if anything other than “sample/”

mob\_folder\_suffix="\_recon"

op\_dir="./Mackay\_MAGs\_mob\_suite\_gffs/"

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

os.mkdir(op\_dir)

for folder in glob("%s/\*%s/" % (mob\_dir,mob\_folder\_suffix)):

sample\_name=folder.split('/')[-2].replace(mob\_folder\_suffix,'')

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

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

plasmid\_df=pd.DataFrame(columns=['file\_id','sequenceid','attributes'])

file\_ids=[]

sequenceids=[]

for plasmid\_fasta in glob("%s/\*plasmid\*fasta" % folder):

#print(plasmid\_fasta)

plasmid\_id=plasmid\_fasta.split('/')[-1]

record=SeqIO.read(plasmid\_fasta,"fasta")

contig=record.id.split("|")[1]

#print(contig)

file\_ids.append(plasmid\_id)

sequenceids.append(contig)

if len(file\_ids)==0:

print("No plasmids found for this sample:%s" % (sample\_name))

continue

plasmid\_df['file\_id']=file\_ids

plasmid\_df['sequenceid']=sequenceids

mob\_agg\_df=pd.read\_csv("%s/mobtyper\_aggregate\_report.txt" % (folder), sep='\t')

merged\_df=plasmid\_df.merge(mob\_agg\_df)

merged\_df['source']=["MOB\_suite\_recon" for i in range(len(merged\_df))]

merged\_df['feature']=["plasmid" for i in range(len(merged\_df))]

merged\_df['start']=[1 for i in range(len(merged\_df))]

merged\_df['end']=merged\_df['total\_length']

merged\_df['score']=["." for i in range(len(merged\_df))]

merged\_df['strand']=["." for i in range(len(merged\_df))]

merged\_df['phase']=["." for i in range(len(merged\_df))]

att\_cols=['rep\_type(s)','rep\_type\_accession(s)','relaxase\_type(s)','relaxase\_type\_accession(s)','mpf\_type','mpf\_type\_accession(s)','orit\_type$

i=0

att\_list=[]

for col in att\_cols:

if i==0:

merged\_df['attributes'] = ("%s=" % (col)) + merged\_df[col].astype(str)

#merged\_df[col] = ("%s=" % (col)) + merged\_df[col].astype(str)

merged\_df=merged\_df.drop(col, axis=1)

else:

merged\_df['attributes'] = merged\_df['attributes'].astype(str) + (";%s=" % (col)) + merged\_df[col].astype(str)

#merged\_df[col] = (";%s=" % (col)) + merged\_df[col].astype(str)

merged\_df=merged\_df.drop(col, axis=1)

i+=1

gff\_cols=['sequenceid','source','feature','start','end','score','strand','phase','attributes']

merged\_df=merged\_df[gff\_cols]

#print(merged\_df)

#print(op\_file)

#merged\_df.to\_csv("MS6-5\_MAGs\_mob\_suite\_gffs/test.gff", sep="\t", index=False, header=False)

merged\_df.to\_csv(op\_file, sep="\t", index=False, header=False)