#!/usr/bin/env p3

# -\*- coding: utf-8 -\*-

##

# gencos

##

# Utility for visualizing NGS data along gene models

##

import re, os, sys, logging, time, datetime

from optparse import OptionParser, OptionGroup

# reload(sys)

# sys.setdefaultencoding('utf-8')

import matplotlib

from numpy import \*

# Use PDF backend

matplotlib.use("pdf")

sys.path.insert(1, os.path.split(os.path.realpath(\_\_file\_\_))[0] + "/./")

from lib import Configuration,Tracks,Plot

# from .lib.Plot import plottracks

matplotlib.rcParams['xtick.direction'] = 'out'

matplotlib.rcParams['ytick.direction'] = 'out'

def run(conf\_filename, region, output\_dir, plot\_title=None, plot\_label=None):

"""

plot tracks between region

"""

plotconfig = Configuration.loadconfiguration(conf\_filename, region)

(tracks,chrome,tx\_start, tx\_end, graphcoords, graphToGene) = Tracks.maketracks(plotconfig)

# print(tracks)

fig = Plot.initFig(plotconfig,tracks)

Plot.plottracks(fig,plotconfig,tracks,chrome,tx\_start, tx\_end, graphcoords, graphToGene, plot\_title ,plot\_label)

def greeting():

print("Ablife Genome View: Visualize spliced RNA-Seq reads along gene models. ")

print("See --help for usage.\n")

print("Manual available at: \n")

def main():

parser = OptionParser()

parser.add\_option('-c', '--conf', dest='conf', action='store', type='string', help='config file')

parser.add\_option('-r', "--plot-region", dest="plot\_region", nargs=1, default=None, help="Plot read densities for a given region. "

"format: chr:start-end")

parser.add\_option('-t', "--plot-title", dest="plot\_title", default=None, nargs=1, help="Title of plot: a string that will be displayed at top of plot. Example: "

"--plot-title \"My favorite gene\".")

parser.add\_option('-l', "--plot-label", dest="plot\_label", default=None, nargs=1, help="Plot label. If given, plot will be saved in the output directory as "

"the plot label ending in the relevant extension, e.g. <plot\_label>.pdf. "

"Example: --plot-label my\_gene")

parser.add\_option('-o', "--output-dir", dest="output\_dir", nargs=1, default="./", help="Output directory.")

(options, args) = parser.parse\_args()

if options.conf is None:

greeting()

sys.exit(1)

output\_dir = os.path.abspath(os.path.expanduser(options.output\_dir))

if not os.path.isdir(output\_dir):

os.makedirs(output\_dir)

plot\_title = options.plot\_title

plot\_label = options.plot\_label

if options.conf != None:

region = options.plot\_region

conf\_filename = os.path.abspath(os.path.expanduser(options.conf))

run(conf\_filename, region, output\_dir, plot\_title=plot\_title, plot\_label=plot\_label)

if \_\_name\_\_ == '\_\_main\_\_':

main()