Python Practice and Workshop 1

Printing data

Practice printing out data

Reading from a file and counting information from a file

Let's count up Exon number and length.

Use rice_random_exons.bed

Write python code to do the following:

1. Open the file
2. Loop through each line in the file
3. Print the length of each exon
4. Summarize the total length of all the exons

```
import os
riceurl="https://raw.githubusercontent.com/biodataprog/GEN220/master/data/rice_random_exons.
ricefile=os.path.basename(riceurl)
print("Rice filename is {}".format(ricefile))
if not os.path.exists(ricefile):
    os.system("curl -O {}".format(riceurl))
total_exon_length = 0
with open(ricefile,"r") as fh:
    for bedline in fh:
#        bedline = bedline.strip()
        bedarray = bedline.strip().split("\t")
        exonlen = abs(int(bedarray[2])-int(bedarray[1]))
#        print("exon length is {}".format(exonlen))
        total_exon_length += exonlen
print("Total length of exons is {} bp ({} kb)".format(total_exon_length,total_exon_length/10
```

Reading from a comma delimited file Use the data file from this

site: https://datacarpentry.org/2015-03-09-ISI-CODATA/data/biology/species.csv for this example.

Print out the names of all the genera (genus column) and the counts of each.

```
import os, csv, sys
speciesurl="https://datacarpentry.org/2015-03-09-ISI-CODATA/data/biology/species.csv"
spfile=os.path.basename(speciesurl)
```

```
print("Species filename is {}".format(spfile))
if not os.path.exists(spfile):
    os.system("curl -O {}".format(speciesurl))

genera = {} # this is a dictionary
with open(spfile,"r") as fh:
    speciesreader = csv.reader(fh, delimiter=',')
    header = next(speciesreader)
    for row in speciesreader:
        genus = row[1]
        if len(genus) == 0:
            continue
        if genus in genera:
            genera[genus] += 1
        else:
            genera[genus] = 1

for genus in sorted(genera):
    generaout = csv.writer(sys.stdout,delimiter="\t")
    generaout.writerow([genus,genera[genus]])
    #print("{}\t{}".format(genus,genera[genus]))

print("===")
print("now ordered by abundance using the value stored in the dictionary")
for genus in reversed(sorted(genera.keys(), key=lambda x: genera[x])):
    generaout = csv.writer(sys.stdout,delimiter="\t")
    generaout.writerow([genus,genera[genus]])
```

FASTA file processing

Let's read in a dataset of Genes from a FASTA formatted file and print out: 1. The number of sequences in the file. 2. The first and last codons (first 3 bases and last 3 bases) 3. Count the number of genes on each strand (this is coded by the last character in the gene name).

```
import itertools, sys, re

# define what a header looks like in FASTA format
def isheader(line):
    return line[0] == '>'


def aspairs(f):
    seq_id = ''
    sequence = ''
    for header,group in itertools.groupby(f, isheader):
        if header:
            line = next(group)
            seq_id = line[1:].split()[0]
        else:
            sequence = ''.join(line.strip() for line in group)
            yield seq_id, sequence

# you could use this or you can download this from here instead
# https://github.com/biodataprog/GEN220_data/raw/main/data/S_cerevisiae_ORFs.fasta
yeastorfs="https://github.com/biodataprog/GEN220_data/raw/main/data/S_cerevisiae_ORFs.fasta"
filename=os.path.basename(yeastorfs)
if not os.path.exists(filename):
    os.system("curl -L -O {}".format(yeastorfs))
#filename="/bigdata/gen220/shared/data/S_cerevisiae_ORFs.fasta"
#filename="S_cerevisiae_ORFs.fasta"
print("filename is {}".format(filename))
with open(filename,"r") as f:
    seqs = dict(aspairs(f))
    first_codon = {}
    last_codon  = {}
    sequence_count = 0
    strand_count = {}
    i = 0
    for seqname in seqs:
#        if i == 0:
#            print("seqname is {}".format(seqname))
#            i += 1
        sequence_count += 1
        firstcodon = seqs[seqname][0:3]
        #print("length of sequence = ",len(seqs[seqname]))
        lastcodon = seqs[seqname][-3:]
        if firstcodon in first_codon:
            first_codon[firstcodon] +=1
        else:
            first_codon[firstcodon] = 1

        if lastcodon in last_codon:
            last_codon[lastcodon] +=1
        else:
            last_codon[lastcodon] = 1

#       print(firstcodon,lastcodon)
        last_char = seqname[-1]
#       print(last_char, " in ", seqname)
        if last_char in strand_count:
            strand_count[last_char] += 1
        else:
            strand_count[last_char] = 1
print("1.")
print("There are %d sequences in the file"%(sequence_count))

print("2.")
print("The distribution of first codons is:")

for codon in first_codon:
    print("%s => %d (%.1f%%)" % (codon, first_codon[codon],
                                 100.0 * first_codon[codon] / sequence_count))


print("The distribution of last codons is:")
for codon in last_codon:
    print("%s => %d (%.1f%%)" % (codon, last_codon[codon],
                                 100.0 * last_codon[codon] / sequence_count))

print("3.")
print("There are %s genes on the Watson (+) strand"%(strand_count['W']))
print("          %s genes on the Crick  (-) strand"%(strand_count['C']))
```