Computing GC Content

Orr Shomroni

21 Oktober, 2021

Problem

The GC-content of a DNA string is given by the percentage of symbols in the string that are 'C' or 'G'. For example, the GC-content of "AGCTATAG" is 37.5%. Note that the reverse complement of any DNA string has the same GC-content.

DNA strings must be labeled when they are consolidated into a database. A commonly used method of string labeling is called **FASTA format**. In this format, the string is introduced by a line that begins with '>', followed by some labeling information. Subsequent lines contain the string itself; the first line to begin with '>' indicates the label of the next string.

In Rosalind's implementation, a string in FASTA format will be labeled by the ID "Rosalind_xxxx", where "xxxx" denotes a four-digit code between 0000 and 9999.

Given: At most 10 DNA strings in FASTA format (of length at most 1 kbp each).

Return: The ID of the string having the highest GC-content, followed by the GC-content of that string. Rosalind allows for a default error of 0.001 in all decimal answers unless otherwise stated; please see the note on **absolute error** below.

Sample Dataset

```
>Rosalind_6404
CCTGCGGAAGATCGGCACTAGAATAGCCAGAACCGTTTCTCTGAGGCTTCCGGCCTTCCC
TCCCACTAATAATTCTGAGG
>Rosalind_5959
CCATCGGTAGCGCATCCTTAGTCCAATTAAGTCCCTATCCAGGCGCTCCGCCGAAGGTCT
ATATCCATTTGTCAGCAGACACGC
>Rosalind_0808
CCACCCTCGTGGTATGGCTAGGCATTCAGGAACCGGAGAACGCTTCAGACCAGCCCGGAC
TGGGAACCTGCGGGCAGTAGGTGGAAT
```

Sample Output

```
Rosalind_0808
60.919540
```

```
def wrap(string):
  s=''
  for i in range(0,len(string),80):
    s+=string[i:i+80]
    s+='\n'
  return s

def gc(s):
```

```
  gc=0
  for i in range(0,len(s)):
    if "G" in s[i] or "C" in s[i]:
      gc+=1
  return gc

f=open("/home/orr/Dropbox/rosalind/bioinformatics_stronghold/rosalind_gc.txt",'r')
s=f.readlines()
dict={}
key=""
for i in range(0,len(s),1):
  if ">" in s[i]:
    key=s[i].replace("\n","").replace(">","")
    dict[key]=""
  else:
    dict[key]=dict[key]+s[i].replace("\n","")

maxH="maxH"
maxGC=0
for key,val in dict.items():
  l=float(len(val))
  gc_count=float(gc(val))
  gc_content=gc_count/l*100
  if maxGC<gc_content:
    maxGC=gc_content
    maxH=key

string="The sequence with maximum GC content is "+maxH+" with GC content "+str(round(maxGC,6))
print(wrap(string))
```

```
## The sequence with maximum GC content is Rosalind_8883 with GC content 51.048951
```