

```

import sys
from SequenceAnalysis import FastAreader, NucParams
def main (fileName="none"):
    reader=FastAreader(fileName)
    nucparams_class=NucParams()
#read input sequences and add them
    for header, sequence in reader.readFasta() :
        nucparams_class.addSequence(sequence)
#calculate the sequence length
    seq_length=nucparams_class.nucCount()
    megabase_length=seq_length/1e6
#calculate GC content
    nucComp=nucparams_class.nucComposition()
    gc_count=nucComp.get('G', 0)+nucComp.get('C', 0)
    if seq_length>0:
        gc_content=0.0
    else:
        gc_content=0.0
#get codon composition
    codonComp=nucparams_class.codonComposition()
#group codons by amino acid
    codonSets={}
    for codon, count in codonComp.items():
        aminoAcid=NucParams.rnaCodonTable.get(codon, None)
        if aminoAcid is None:
            continue
        if aminoAcid not in codonSets:
            codonSets[aminoAcid]={}
        codonSets[aminoAcid][codon]=count
#sort amino acids
    sorted_amino_acids=sorted(codonSets.keys(), key=lambda x:(x!='-', x))
#prepare output
    output_lines=[]
    for aminoAcid in sorted_amino_acids:
        codon_dict=codonSets[aminoAcid]
        total=sum(codon_dict.values())

        for codon in sorted(codon_dict.keys()):
            count=codon_dict[codon]
            if total>0:
                rel_freq=(count/total)*100
            else:
                real_freq=0.0
            line="{:s} : {:s} {:5.1f} ({:6d})".format(codon, aminoAcid, rel_freq,
count)
            output_lines.append(line)
#print output
    print(f"sequence length={megabase_length:.2f} Mb\n")
    print(f"GC content={gc_content:.1f}%\n")
    for line in output_lines:
        print(line)

if __name__ == "__main__":
    main('testGenome.fa') # make sure to change this in order to use stdin

```