

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Import modules

import sys
from Bio import SeqIO

# Parse user input

fasta_file = sys.argv[1] # Input fasta file
output_file = sys.argv[2] # Output stat file

# Declare variables

nb_seq = 0
shortest_len = 999999999
shortest_seq = "Not found"
longest_len = 0
longest_seq = "Not found"
total_len = 0.0 # Nombre decimal
nb_A = 0
nb_C = 0
nb_G = 0
nb_T = 0

# Treat fasta file

fasta_sequences = SeqIO.parse(open(fasta_file), 'fasta')
for s in fasta_sequences:
    seq = s.seq.tostring()
    len_seq = len(seq)
    if seq.upper().count("N") / len_seq < 0.3:
        pass # exemple: print good sequences to file 1
    else:
        pass # print bad sequences to file 2
    nb_seq += 1
    total_len += len_seq
    nb_A += seq.upper().count("A")
    nb_C += seq.upper().count("C")
    nb_G += seq.upper().count("G")
    nb_T += seq.upper().count("T")
    if len(seq) < shortest_len:
        shortest_len = len_seq
        shortest_seq = s.id
    if len_seq > longest_len:
        longest_len = len_seq
        longest_seq = s.id

# Print statistics

print
print "Nb. seq: " + str(nb_seq)
print "Nb. seq: \t%06i\tin file\t%s" % (nb_seq, fasta_file)
print "Shortest len:\t" + str(shortest_len) + " bp"
print "Shortest seq:\t" + shortest_seq
print "Longest len:\t" + str(longest_len) + " bp"
print "Longest seq:\t" + longest_seq
print "Longueur moy:\t" + str(total_len / nb_seq)
print "% A:\t" + str(100 * nb_A / total_len)
print "% C:\t" + str(100 * nb_C / total_len)
print "% G:\t" + str(100 * nb_G / total_len)
print "% T:\t" + str(100 * nb_T / total_len)
print "% other:\t" + str(100 - 100 * sum([nb_A, nb_C, nb_G, nb_T]) / total_len)
print

with open(output_file, "w") as f:
    f.write("Shortest len:\t" + str(shortest_len) + " bp\n")
    f.write("deuxieme\tligne\n")

```