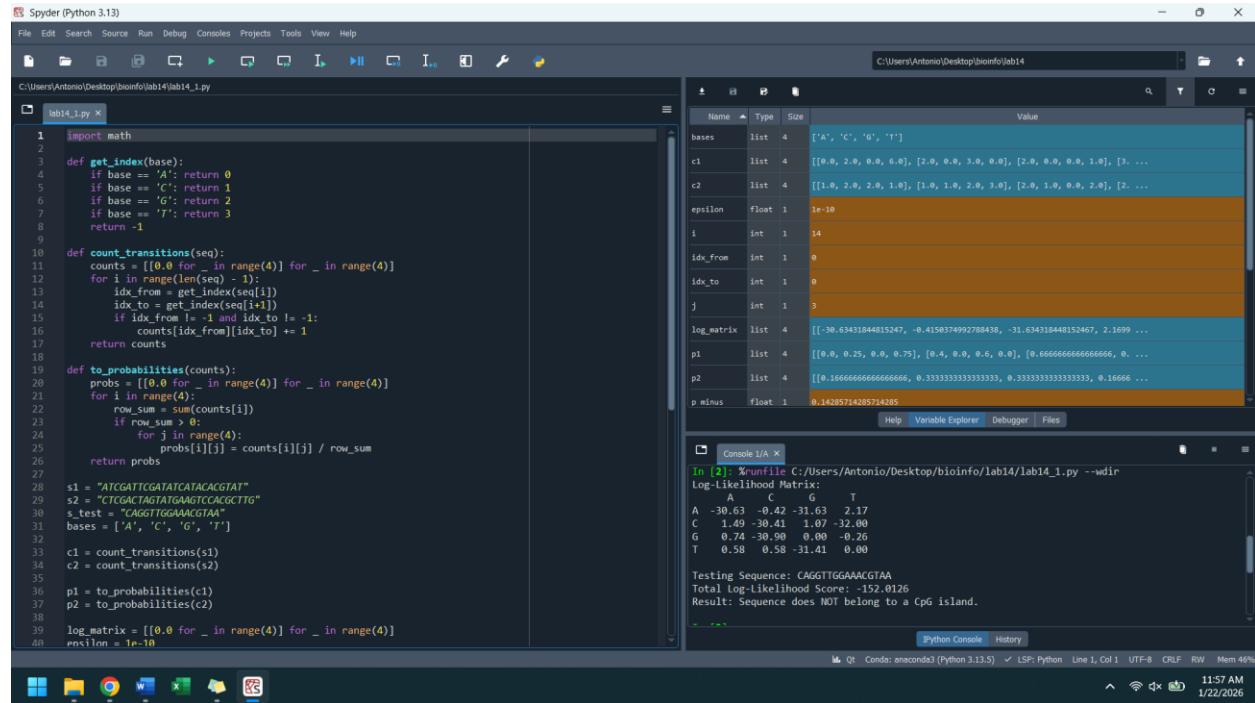


LABORATORY REPORT #14

GEORGESCU Mihai-Antonio, 1242EA
 Bioinformatics, 4th year 1st semester, 2025-2026

lab14_1.py



```

1 import math
2
3 def get_index(base):
4     if base == 'A': return 0
5     if base == 'C': return 1
6     if base == 'G': return 2
7     if base == 'T': return 3
8     return -1
9
10 def count_transitions(seq):
11     counts = [[0.0 for _ in range(4)] for _ in range(4)]
12     for i in range(len(seq) - 1):
13         idx_from = get_index(seq[i])
14         idx_to = get_index(seq[i+1])
15         if idx_from != -1 and idx_to != -1:
16             counts[idx_from][idx_to] += 1
17     return counts
18
19 def to_probabilities(counts):
20     probs = [[0.0 for _ in range(4)] for _ in range(4)]
21     for i in range(4):
22         row_sum = sum(counts[i])
23         if row_sum > 0:
24             for j in range(4):
25                 probs[i][j] = counts[i][j] / row_sum
26     return probs
27
28 s1 = "ATCGATTCGATATCATACAGCTAT"
29 s2 = "CTCGACTAGTAGTGTGNGTCACGGCTTG"
30 s3 = "CAAGTTGGAAACGTA"
31 bases = ['A', 'C', 'G', 'T']
32
33 c1 = count_transitions(s1)
34 c2 = count_transitions(s2)
35
36 p1 = to_probabilities(c1)
37 p2 = to_probabilities(c2)
38
39 log_matrix = [[0.0 for _ in range(4)] for _ in range(4)]
40 epsilon = 1e-10

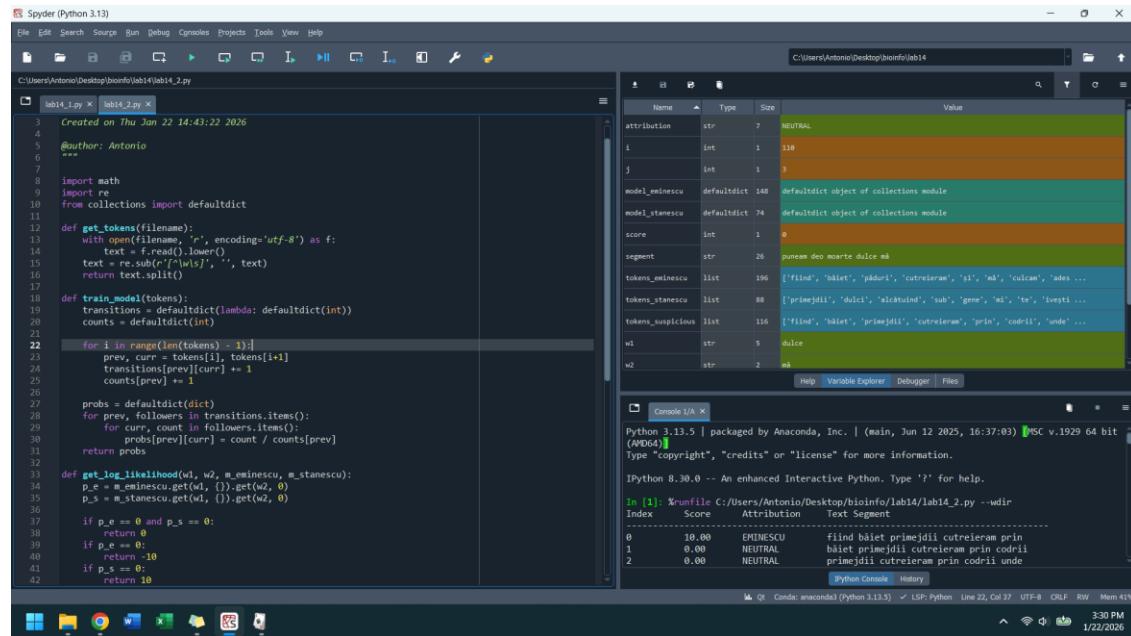
```

In [2]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab14/lab14_1.py --wdir
 Log-Likelihood Matrix:

	A	C	G	T
A	-30.63	-0.42	-31.63	2.17
C	1.49	-30.41	1.07	-32.02
G	0.74	-30.99	0.00	-0.26
T	0.58	0.58	-31.41	

 Testing Sequence: CAGTTGGAAACGTA
 Total Log-Likelihood Score: -152.0126
 Result: Sequence does NOT belong to a CpG island.

lab15_2.py



```

3 Created on Thu Jan 22 14:43:22 2026
4
5 #author: Antonio
6 """
7
8 import math
9 import re
10 from collections import defaultdict
11
12 def get_tokens(filename):
13     with open(filename, 'r', encoding='utf-8') as f:
14         text = f.read().lower()
15         text = re.sub(r'[\w\s]', ' ', text)
16     return text.split()
17
18 def train_model(tokens):
19     transitions = defaultdict(lambda: defaultdict(int))
20     counts = defaultdict(int)
21
22     for i in range(len(tokens) - 1):
23         prev, curr = tokens[i], tokens[i+1]
24         transitions[prev][curr] += 1
25         counts[prev] += 1
26
27     probs = defaultdict(dict)
28     for prev, followers in transitions.items():
29         for curr, count in followers.items():
30             probs[prev][curr] = count / counts[prev]
31     return probs
32
33 def get_log_likelihood(w1, w2, m_elinescu, m_stanescu):
34     p_e = m_elinescu.get(w1, {}).get(w2, 0)
35     p_s = m_stanescu.get(w1, {}).get(w2, 0)
36
37     if p_e == 0 and p_s == 0:
38         return 0
39     if p_e == 0:
40         return -10
41     if p_s == 0:
42         return 10

```

In [1]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab14/lab15_2.py --wdir
 Index Score Attribution Text Segment

0 10.00 EMINESCU filind bălet primejdii cutreieram prin
 bălet primejdii cutreieram prin codrii
1 0.00 NEUTRAL bălet primejdii cutreieram prin codrii
2 0.00 NEUTRAL primejdii cutreieram prin codrii unde

