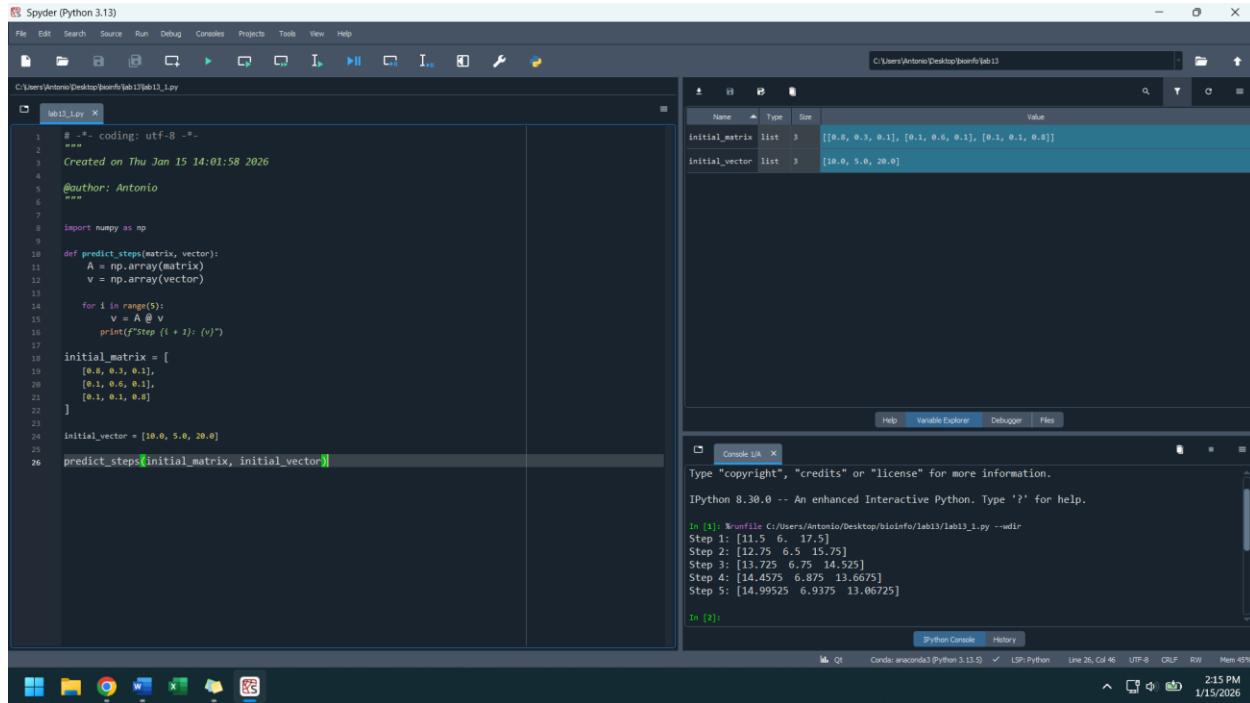


## LABORATORY REPORT #13

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

lab13\_1.py



```

# -*- coding: utf-8 -*-
"""
Created on Thu Jan 15 14:01:58 2026
@author: Antonio
"""

import numpy as np

def predict_steps(matrix, vector):
    A = np.array(matrix)
    v = np.array(vector)

    for i in range(5):
        v = A @ v
        print(f"Step ({i + 1}): {v}")

initial_matrix = [
    [0.8, 0.3, 0.1],
    [0.1, 0.6, 0.1],
    [0.1, 0.1, 0.8]
]

initial_vector = [10.0, 5.0, 20.0]

predict_steps(initial_matrix, initial_vector)

```

The Variable Explorer shows:

Name	Type	Size	Value
initial_matrix	list	3	[[0.8, 0.3, 0.1], [0.1, 0.6, 0.1], [0.1, 0.1, 0.8]]
initial_vector	list	3	[10.0, 5.0, 20.0]

The IPython Console shows:

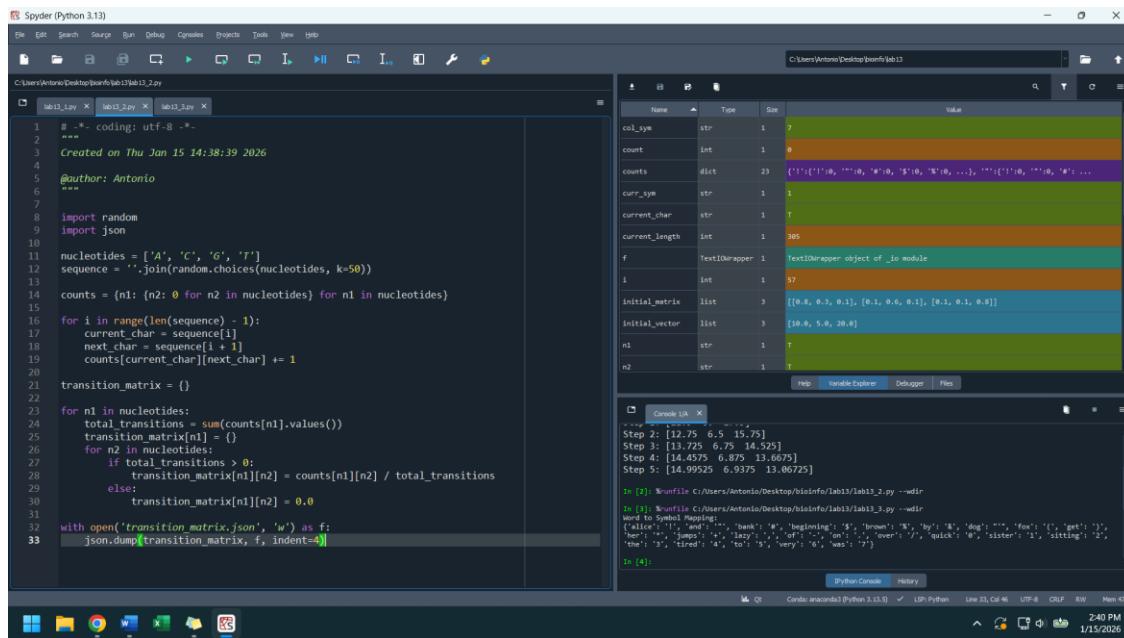
```

In [1]: runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_1.py --wdir
Step 1: [11.5 6. 17.5]
Step 2: [12.75 6.5 15.75]
Step 3: [13.725 6.75 14.525]
Step 4: [14.4575 6.875 13.6675]
Step 5: [14.99525 6.9375 13.06725]

In [2]:

```

lab13\_2.py



```

# -*- coding: utf-8 -*-
"""
Created on Thu Jan 15 14:38:39 2026
@author: Antonio
"""

import random
import json

nucleotides = ['A', 'C', 'G', 'T']
sequence = ''.join(random.choices(nucleotides, k=50))

counts = {n1: {n2: 0 for n2 in nucleotides} for n1 in nucleotides}

for i in range(len(sequence) - 1):
    current_char = sequence[i]
    next_char = sequence[i + 1]
    counts[current_char][next_char] += 1

transition_matrix = {}

for n1 in nucleotides:
    total_transitions = sum(counts[n1].values())
    transition_matrix[n1] = {}
    for n2 in nucleotides:
        if total_transitions > 0:
            transition_matrix[n1][n2] = counts[n1][n2] / total_transitions
        else:
            transition_matrix[n1][n2] = 0.0

with open('transition_matrix.json', 'w') as f:
    json.dump(transition_matrix, f, indent=4)

```

The Variable Explorer shows:

Name	Type	Size	Value
col_sym	str	1	'T'
count	int	1	0
counts	dict	23	{'A': {'A': 0, 'C': 0, 'G': 0, 'T': 0}, 'C': {'A': 0, 'C': 0, 'G': 0, 'T': 0}, 'G': {'A': 0, 'C': 0, 'G': 0, 'T': 0}, 'T': {'A': 0, 'C': 0, 'G': 0, 'T': 0}}
curr_sym	str	1	'T'
current_char	str	1	'T'
current_length	int	1	395
f	TextIOWrapper	1	TextIOWrapper object of '_io module'
i	int	1	57
initial_matrix	list	3	[[0.8, 0.3, 0.1], [0.1, 0.6, 0.1], [0.1, 0.1, 0.8]]
initial_vector	list	3	[10.0, 5.0, 20.0]
n1	str	1	'T'
n2	str	1	'T'

The IPython Console shows:

```

In [1]: runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_2.py --wdir
Step 2: [12.75 6.75 15.75]
Step 3: [13.725 6.75 14.525]
Step 4: [14.4575 6.875 13.6675]
Step 5: [14.99525 6.9375 13.06725]

In [2]: runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_3.py --wdir
Word to Symbol Mapping:
{'value': '1', 'word': 'one', 'key': '1'}, {'value': '2', 'word': 'two', 'key': '2'}, {'value': '3', 'word': 'three', 'key': '3'}, {'value': '4', 'word': 'four', 'key': '4'}, {'value': '5', 'word': 'five', 'key': '5'}, {'value': '6', 'word': 'six', 'key': '6'}, {'value': '7', 'word': 'seven', 'key': '7'}, {'value': '8', 'word': 'eight', 'key': '8'}, {'value': '9', 'word': 'nine', 'key': '9'}, {'value': '0', 'word': 'zero', 'key': '0'}, {'value': 'a', 'word': 'alpha', 'key': 'a'}, {'value': 'b', 'word': 'beta', 'key': 'b'}, {'value': 'c', 'word': 'gamma', 'key': 'c'}, {'value': 'd', 'word': 'delta', 'key': 'd'}, {'value': 'e', 'word': 'epsilon', 'key': 'e'}, {'value': 'f', 'word': 'phi', 'key': 'f'}, {'value': 'g', 'word': 'psi', 'key': 'g'}, {'value': 'h', 'word': 'eta', 'key': 'h'}, {'value': 'i', 'word': 'iota', 'key': 'i'}, {'value': 'j', 'word': 'jota', 'key': 'j'}, {'value': 'k', 'word': 'kappa', 'key': 'k'}, {'value': 'l', 'word': 'lambda', 'key': 'l'}, {'value': 'm', 'word': 'mu', 'key': 'm'}, {'value': 'n', 'word': 'nu', 'key': 'n'}, {'value': 'o', 'word': 'omicron', 'key': 'o'}, {'value': 'p', 'word': 'rho', 'key': 'p'}, {'value': 'r', 'word': 'tau', 'key': 'r'}, {'value': 's', 'word': 'sigma', 'key': 's'}, {'value': 't', 'word': 'tau', 'key': 't'}, {'value': 'u', 'word': 'upsilon', 'key': 'u'}, {'value': 'v', 'word': 'upsilon', 'key': 'v'}, {'value': 'w', 'word': 'omega', 'key': 'w'}, {'value': 'y', 'word': 'psi', 'key': 'y'}, {'value': 'z', 'word': 'zeta', 'key': 'z'}, {'value': 'x', 'word': 'xi', 'key': 'x'}, {'value': 'q', 'word': 'chi', 'key': 'q'}, {'value': 'hbar', 'word': 'h-bar', 'key': 'hbar'}, {'value': 'pi', 'word': 'pi', 'key': 'pi'}, {'value': 'e', 'word': 'e', 'key': 'e'}, {'value': 'h', 'word': 'h', 'key': 'h'}, {'value': 'm_psi', 'word': 'm-psi', 'key': 'm_psi'}, {'value': 'm_pi', 'word': 'm-pi', 'key': 'm_pi'}, {'value': 'm_h', 'word': 'm-h', 'key': 'm_h'}, {'value': 'm_e', 'word': 'm-e', 'key': 'm_e'}, {'value': 'm_chi', 'word': 'm-chi', 'key': 'm_chi'}, {'value': 'm_q', 'word': 'm-q', 'key': 'm_q'}, {'value': 'm_psi_bar', 'word': 'm-psi-bar', 'key': 'm_psi_bar'}, {'value': 'm_pi_bar', 'word': 'm-pi-bar', 'key': 'm_pi_bar'}, {'value': 'm_h_bar', 'word': 'm-h-bar', 'key': 'm_h_bar'}, {"value": "m_e_bar", "word": "m-e-bar", "key": "m_e_bar"}, {"value": "m_chi_bar", "word": "m-chi-bar", "key": "m_chi_bar"}, {"value": "m_q_bar", "word": "m-q-bar", "key": "m_q_bar"}}
Word to Symbol Mapping:
{'value': '1', 'word': 'one', 'key': '1'}, {"value': '2', 'word': 'two', 'key': '2'}, {"value': '3', 'word': 'three', 'key': '3'}, {"value': '4', 'word': 'four', 'key': '4'}, {"value': '5', 'word': 'five', 'key': '5'}, {"value': '6', 'word': 'six', 'key': '6'}, {"value': '7', 'word': 'seven', 'key': '7'}, {"value': '8', 'word': 'eight', 'key': '8'}, {"value': '9', 'word': 'nine', 'key': '9'}, {"value': '0', 'word': 'zero', 'key': '0'}, {"value': 'a', 'word': 'alpha', 'key': 'a'}, {"value': 'b', 'word': 'beta', 'key': 'b'}, {"value': 'c', 'word': 'gamma', 'key': 'c'}, {"value': 'd', 'word': 'delta', 'key': 'd"}, {"value': 'e', 'word': 'epsilon', 'key': 'e"}, {"value': 'f', 'word': 'phi', 'key': 'f"}, {"value': 'g', 'word': 'psi', 'key': 'g"}, {"value': 'h', 'word': 'eta', 'key': 'h"}, {"value': 'i', 'word': 'iota', 'key': 'i"}, {"value': 'j', 'word': 'jota', 'key': 'j"}, {"value': 'k', 'word': 'kappa', 'key': 'k"}, {"value': 'l', 'word': 'lambda', 'key': 'l"}, {"value': 'm', 'word': 'mu', 'key': 'm'}, {"value': 'n', 'word': 'nu', 'key': 'n'}, {"value': 'o', 'word': 'omicron', 'key': 'o'}, {"value': 'p', 'word': 'rho', 'key': 'p'}, {"value': 'r', 'word': 'tau', 'key': 'r'}, {"value': 's', 'word': 'sigma', 'key': 's'}, {"value': 't', 'word': 'tau', 'key': 't'}, {"value': 'u', 'word': 'upsilon', 'key': 'u"}, {"value': 'v', 'word': 'upsilon', 'key': 'v'}, {"value': 'w', 'word': 'omega', 'key': 'w'}, {"value": "y", "word": "psi", "key": "y"}, {"value": "z", "word": "zeta", "key": "z"}, {"value": "x", "word": "xi", "key": "x"}, {"value": "q", "word": "chi", "key": "q"}, {"value": "pi", "word": "pi", "key": "pi"}, {"value": "e", "word": "e", "key": "e"}, {"value": "h", "word": "h", "key": "h"}, {"value": "m_psi", "word": "m-psi", "key": "m_psi"}, {"value": "m_pi", "word": "m-pi", "key": "m_pi"}, {"value": "m_h", "word": "m-h", "key": "m_h"}, {"value": "m_e", "word": "m-e", "key": "m_e"}, {"value": "m_chi", "word": "m-chi", "key": "m_chi"}, {"value": "m_q", "word": "m-q", "key": "m_q"}, {"value": "m_psi_bar", "word": "m-psi-bar", "key": "m_psi_bar"}, {"value": "m_pi_bar", "word": "m-pi-bar", "key": "m_pi_bar"}, {"value": "m_h_bar", "word": "m-h-bar", "key": "m_h_bar"}, {"value": "m_e_bar", "word": "m-e-bar", "key": "m_e_bar"}, {"value": "m_chi_bar", "word": "m-chi-bar", "key": "m_chi_bar"}, {"value": "m_q_bar", "word": "m-q-bar", "key": "m_q_bar"}}

```

lab13\_3.py

The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The toolbar has icons for file operations like Open, Save, Run, Stop, and Help. The left sidebar lists three files: lab13\_1.py, lab13\_2.py, and lab13\_3.py. The main code editor window displays the contents of lab13\_3.py. The code generates a word matrix from a list of words, where each word is mapped to a unique symbol and its frequency is tracked. The right side features a Variable Explorer showing the state of variables:

Name	Type	Size	Value
col_sym	str	1	'7'
count	int	1	'8'
counts	dict	28	{'the': 1, 'quick': 1, 'brown': 1, 'fox': 1, 'jumps': 1, 'over': 1, 'Lazy': 1, 'dog': 1, 'alice': 1, 'was': 1, 'beginning': 1, 'to': 1, 'get': 1, 'very': 1, 'tired': 1, 'of': 1, 'sitting': 1, 'by': 1, 'her': 1, 'sister': 1, 'on': 1, 'bank': 1, 'and': 1}
curr_sym	str	1	'1'
current_char	str	1	'T'
current_length	int	1	'395'
f	TextIOWrapper	1	TextIOWrapper object of '_io module'
i	int	1	'57'
initial_matrix	list	3	[ [0.6, 0.3, 0.1], [0.1, 0.6, 0.1], [0.1, 0.1, 0.8] ]
initial_vector	list	3	[10.6, 5.6, 20.6]
n1	str	1	'T'
n2	str	1	'T'

The bottom section shows the Console and IPython Console outputs. The console shows steps of a script execution, and the IPython console shows command history and help information.

lab13\_4.py

The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The toolbar has icons for file operations like Open, Save, Run, Stop, and Help. The status bar at the bottom shows the path C:\Users\Antonio\Desktop\bioinfo\lab13, the Python version 3.13.0, and the current date and time 1/15/2022.

The left pane displays four tabs: lab13\_1.py, lab13\_2.py, lab13\_3.py, and lab13\_4.py. The lab13\_4.py tab is active, showing Python code for generating a sequence from a word transition matrix. The code uses json, random, and numpy modules to load a matrix, choose symbols, and generate a sequence of 20 symbols.

The right pane contains two panes: a Variable Explorer and a Console. The Variable Explorer lists variables with their types and values. The Console pane shows the output of running the code in the lab13\_2.py and lab13\_3.py files, and the generated sequence for lab13\_4.py.

Name	Type	Value
col_sym	str	'\$'
count	int	0
counts	dict	{'!': {'!': 0, '#': 0, '\$': 0, '%': 0, ...}, '*': {'!': 0, '#': 0, '\$': ...}
curr_sym	str	'\$'
current_char	str	'\$'
current_length	int	395
current_symbol	str	'\$'
f	TextIOWrapper	TextIOWrapper object of _io module
generated_sequence	list	['!', '#', '\$', '%', ...]
i	int	57
initial_matrix	list	[[0.8, 0.3, 0.1], [0.1, 0.6, 0.1], [0.1, 0.1, 0.8]]
initial_vector	list	[10.0, 5.0, 20.0]

Console output:

```
In [2]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_2.py --wdir
In [3]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_3.py --wdir
Word to Symbol Mapping:
{'alice': '!', 'and': '#', 'bank': '$', 'beginning': '%', 'brown': '&', 'by': '*', 'dog': '**', 'fox': '(*', 'get': ')',
'her': '&', 'home': '!', 'is': '<', 'lazy': '>', 'over': '/', 'quick': '(', 'sister': ')', 'sitting': '^',
'the': '!', 'there': '&', 'very': '*, 'was': '?'}

In [4]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab13/lab13_4.py --wdir
Generated Sequence (Symbols): & 1 - $ - 2 . * , ) ! . . ( & 1 / 5 " )
```

In [5]: