

LABORATORY REPORT #6

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

lab6_1.py

The screenshot displays the Spyder Python IDE interface. The left pane shows the code for lab6_1.py, which is a script for parsing a FASTA file and generating a gel electrophoresis simulation. The right pane shows the Variable Explorer with a table of variables and their values. The bottom pane shows the IPython console with the execution output.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Nov 6 09:56:54 2025
4
5 @author: Antonio
6 """
7
8 import random
9 import matplotlib.pyplot as plt
10 import numpy as np
11 import re
12
13 # --- Step 1: Parse the FASTA file ---
14 # Assumes the file 'covid.fasta' is in the same directory
15 file_path = 'covid.fasta'
16
17 sequence_lines = []
18 full_genome = ""
19 genome_length = 0
20
21 try:
22     with open(file_path, 'r') as f:
23         for line in f:
24             # Skip the header line
25             if not line.startswith('>'):
26                 sequence_lines.append(line.strip())
27
28     # Join all sequence lines into one large string
29     full_genome = "".join(sequence_lines)
30     genome_length = len(full_genome)
31
32     if genome_length == 0:
33         print("Warning: Genome sequence is empty. Check FASTA file.")
34         # Use a fallback length if parsing failed
35         genome_length = 30000
36
37 except Exception as e:
38     print(f"Error reading file: {e}. Using a dummy length.")
39     # Fallback length in case of file read error
40     genome_length = 30000
```

Name	Type	Size	Value
ax	axes._axes.Axes	1	Axes object of matplotlib.axes._axes module
band_size	int	1	500
f	TextIOWrapper	1	TextIOWrapper object of _io module
fig	Figure.Figure	1	Figure object of matplotlib.figure module
file_path	str	11	covid.fasta
frag_len	int	1	151
fragment_lengths	list	10	[2184, 2058, 1684, 1488, 1172, 1110, 541, 329, 211, 151]
full_genome	str	29903	ATTAAAGGTTTATACCTCCGAGGTACAAACCAACCACTTTCGATCTCTGTAGATCTGCTCTTAA ...
genome_length	int	1	29903
label_text	str	8	500 bp -
ladder_bands	list	14	[10000, 6000, 6000, 5000, 4000, 3000, 2500, 2000, 1500, 1000, ...]
ladder_labels	dict	3	{3000: '3000 bp -', 1500: '1500 bp -', 500: '500 bp -'}

Help Variable Explorer Debugger Files

```
(AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.30.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %runfile C:/Users/Antonio/Desktop/bioinfo/lab6/lab6_1.py --wdir
Parsed genome. Total length: 29903 bp
Generated 10 random fragment lengths (bp): [2184, 2058, 1684, 1488, 1172, 1110, 541, 329, 211, 151]
Generated image: gel_electrophoresis_simulation.png

In [2]:
```

Python Console History

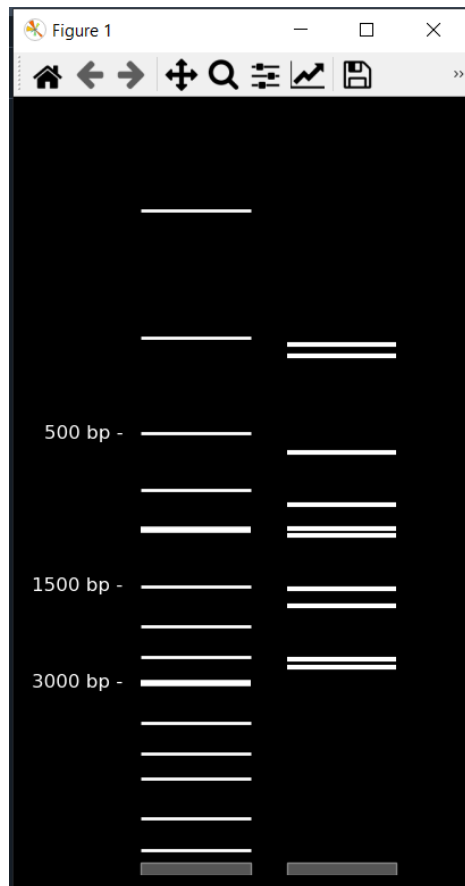
The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named `lab6_1.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Nov 6 09:56:54 2025
4
5 @author: Antonio
6 """
7
8 import random
9 import matplotlib.pyplot as plt
10 import numpy as np
11 import re
12
13 # --- Step 1: Parse the FASTA file ---
14 # Assumes the file 'covid.fasta' is in the same directory
15 file_path = 'covid.fasta'
16
17 sequence_lines = []
18 full_genome = ""
19 genome_length = 0
20
21 try:
22     with open(file_path, 'r') as f:
23         for line in f:
24             # Skip the header line
25             if not line.startswith('>'):
26                 sequence_lines.append(line.strip())
27
28 # Join all sequence lines into one large string
29 full_genome = "".join(sequence_lines)
30 genome_length = len(full_genome)
31
32 if genome_length == 0:
33     print("Warning: Genome sequence is empty. Check FASTA file.")
34     # Use a fallback length if parsing failed
35     genome_length = 30000
36
37 except Exception as e:
38     print(f"Error reading file: {e}. Using a dummy length.")
39     # Fallback length in case of file read error
40     genome_length = 30000
```

The right pane shows a variable explorer with the following data:

Type	Size	Value
axes._axes.Axes	1	Axes object of matplotlib.axes._axes module
int	1	500
TextWrapper	1	TextWrapper object of _io module
Figure.Figure	1	Figure object of matplotlib.figure module
str	11	covid.fasta
int	1	263
list	10	[2688, 2546, 1729, 1522, 1041, 987, 834, 572, 286, 263]
str	29903	ATTAAAGGTTTATACCTTCCCAAGTAAACCAACCACTTTCGATCTCTGTAGATCTCTCTCTAA ...
int	1	29903
str	8	500 bp -
list	14	[10000, 8000, 6000, 5000, 4000, 3000, 2500, 2000, 1500, 1000, ...]
dict	3	{'3000': '3000 bp -', '1500': '1500 bp -', '500': '500 bp -'}

The bottom pane shows the output of the script, including the total length of the genome (29903 bp) and the random fragment lengths (bp): [2184, 2058, 1684, 1488, 1172, 1110, 541, 329, 211, ...]. The generated image is `gel_electrophoresis_simulation.png`.



lab6_2.py

The screenshot shows the Spyder Python IDE with the file `lab6_2.py` open. The script defines a function `parse_multi_fasta` that reads a multi-FASTA file and returns a dictionary of sequences. The output console shows the execution of the script, which successfully generated a multi-gel simulation image.

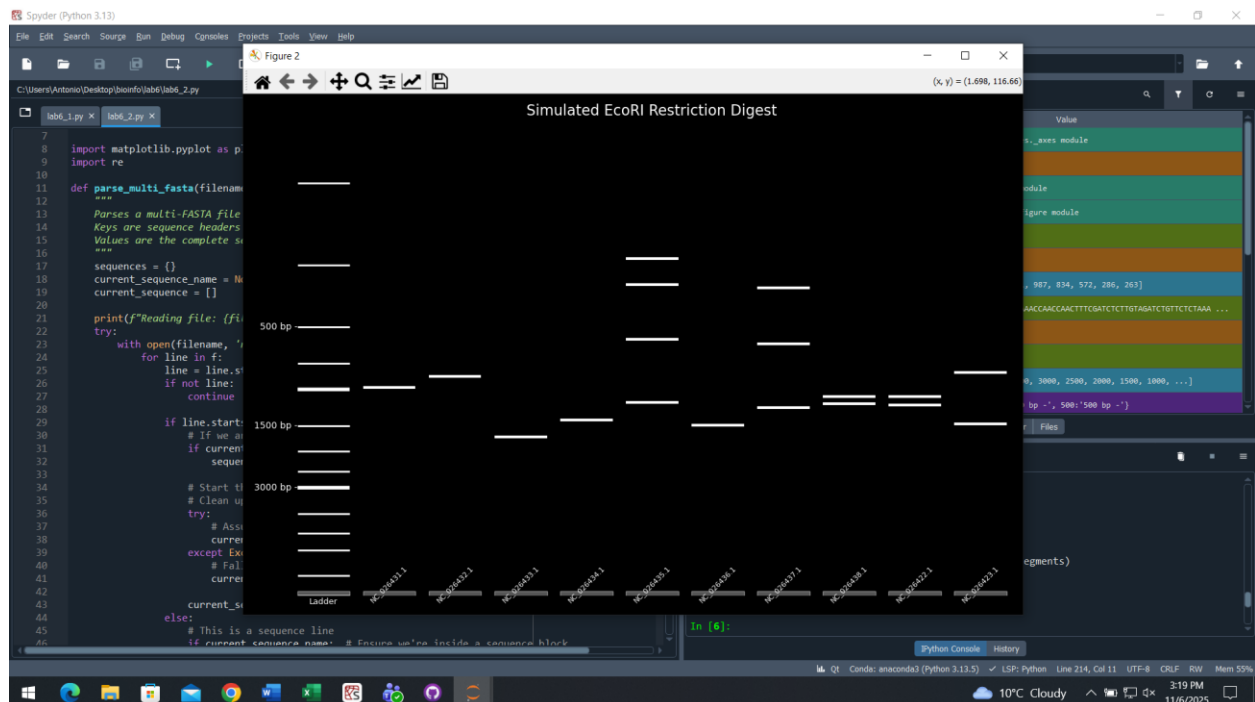
```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Nov 6 14:40:36 2025
4
5 @author: Antonio
6
7 import matplotlib.pyplot as plt
8 import re
9
10 def parse_multi_fasta(filename):
11     """
12     Parses a multi-FASTA file and returns a dictionary of sequences.
13     Keys are sequence headers (e.g., 'NC_026431.1')
14     Values are the complete sequence strings.
15     """
16     sequences = {}
17     current_sequence_name = None
18     current_sequence = []
19
20     print(f"Reading file: {filename}")
21     try:
22         with open(filename, 'r') as f:
23             for line in f:
24                 line = line.strip()
25                 if not line:
26                     continue # Skip empty lines
27
28                 if line.startswith('>'):
29                     # If we are already building a sequence, save it first
30                     if current_sequence_name:
31                         sequences[current_sequence_name] = "".join(current_sequence)
32
33                     # Start the new sequence
34                     # Clean up the header to get a short, usable name
35                     try:
36                         # Assumes format like >ID | description
37                         current_sequence_name = line.split('/')[0].lstrip('>').strip()
38                     except Exception:
39                         pass
40                     current_sequence = []
41                 else:
42                     current_sequence.append(line)
43
44     except Exception:
45         pass
46
47     # This is a sequence line
48     if current_sequence_name:
49         # Ensure we've incide a sequence block
```

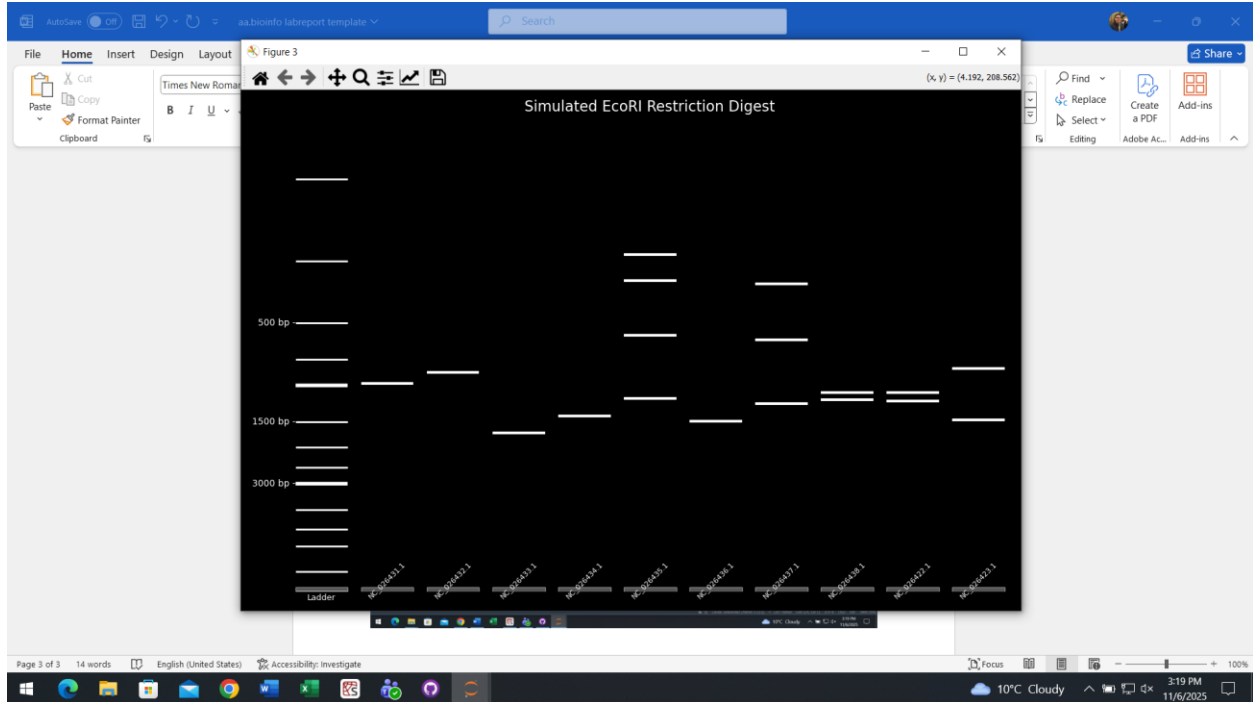
Variable Explorer:

Name	Type	Size	Value
ax	axes._axes.Axes	1	Axes object of matplotlib.axes._axes module
band_size	int	1	500
f	TextIOWrapper	1	TextIOWrapper object of _io module
fig	Figure.Figure	1	Figure object of matplotlib.figure module
file_path	str	11	covid.fasta
frag_len	int	1	263
fragment_lengths	list	10	[2688, 2546, 1729, 1522, 1041, 987, 834, 572, 285, 263]
full_genome	str	29983	ATTAAAGGTTTATACCTCCCAAGTACCAACCAACCACTTCGATCTCTGTAGATCTGTCTCTAA ...
genome_length	int	1	29983
label_text	str	8	500 bp -
ladder_bands	list	14	[18000, 8000, 6000, 5000, 4000, 3000, 2500, 2000, 1500, 1000, ...]
ladder_labels	dict	3	{3000: '3000 bp -', 1500: '1500 bp -', 500: '500 bp -'}

Console 2/A:

```
NC_026436.1: 1 segments
NC_026437.1: 3 segments
NC_026438.1: 3 segments
NC_026422.1: 2 segments
NC_026423.1: 2 segments
**Sequence with the most DNA segments:** NC_026435.1 (4 segments)
Successfully generated image: multi_gel_simulation.png
In [6]:
```





NC_026435.1: 3 bands

NC_026437.1: 2 bands

NC_026438.1: 2 bands

NC_026422.1: 2 bands

NC_026423.1: 2 bands

NC_026431.1: 1 band

NC_026432.1: 1 band

NC_026433.1: 1 band

NC_026434.1: 1 band

NC_026436.1: 1 band