In-Silico Identification and Functional Characterization of *Helicobacter pylori* gene using Sequence Analysis and Homology Based Annotation

With the availability of whole genome sequences, many genes have been identified, but several of them are still annotated as hypothetical genes because their functions are unknown. Studying these genes is important to understand cellular processes.

Helicobacter pylori is a pathogenic bacterium whose genome contains many uncharacterized genes. These hypothetical genes may play important roles in bacterial growth and regulation. In-silico analysis provides a simple and cost-effective way to study such genes.

Bio Python is a powerful bioinformatics tool that helps in retrieving biological sequences and performing sequence analysis such as length calculation, translation, and similarity analysis, which assists in predicting the probable function of a gene through homology-based annotation. In this study, a hypothetical gene from *Helicobacter pylori* was analyzed using Bio Python and other bioinformatics tools to predict its function.

Objectives:

1. To retrieve and analyze the quality of a hypothetical gene sequence of *Helicobacter pylori.*
2. To perform sequence filtering, validation, and homology search.
3. To carry out functional annotation and biological interpretation.

1. To retrieve and analyze the quality of a hypothetical gene sequence of *Helicobacter pylori.*

Step1: Sequence Retrieval hypothetical gene sequence of *Helicobacter pylori.*

Methodology-

Steps to retrieve the hypothetical gene sequence of *Helicobacter pylori* using Biopython

Go to https://www.ncbi.nlm.nih.gov/

Select the appropriate database (**Nucleotide** or **Protein**).

↓

Search for hypothetical gene along with organism name (*Helicobacter pylori)*

↓

Download the sequence in FASTA format

↓

Paste the sequence in Bio python file

↓

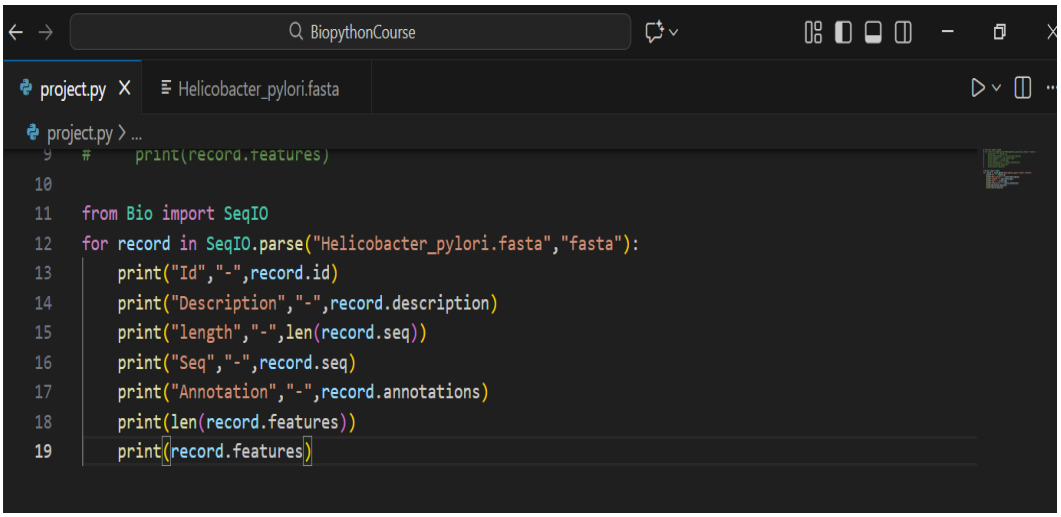
Open the sequence file in Visual Studio Code

↓

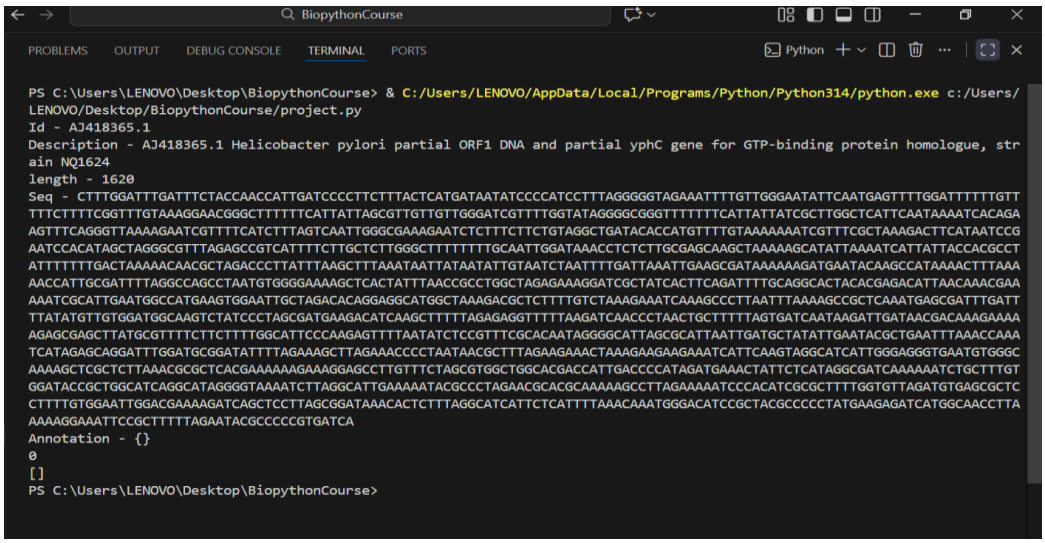
Import Bio python module (SeqIO) in VS Code

↓

Retrieve the sequence using Bio python.



```
#     print(record.features)

from Bio import SeqIO
for record in SeqIO.parse("Helicobacter_pylori.fasta","fasta"):
    print("Id","-",record.id)
    print("Description","-",record.description)
    print("length","-",len(record.seq))
    print("Seq","-",record.seq)
    print("Annotation","-",record.annotations)
    print(len(record.features))
    print(record.features)
```



```
PS C:\Users\LENOVO\Desktop\BiopythonCourse> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
LENOVO/Desktop/BiopythonCourse/project.py
Id - AJ418365.1
Description - AJ418365.1 Helicobacter pylori partial ORF1 DNA and partial yphC gene for GTP-binding protein homologue, str
ain NQ1624
length - 1620
Seq - CTTTGGATTTGATTTCTACCAACCATTGATCCCCTTCTTTACTCATGATAATATCCCCATCCTTTAGGGGGTAGAAATTTTGTTGGGAATATTCAATGAGTTTTGGATTTTTTGTT
TTTCTTTTCGGTTTGTAAAGGAACGGGCTTTTTTCATTATTAGCGTTGTTGTTGGGATCGTTTTGGTATAGGGGCGGGTTTTTTTCATTATTATCGCTTGGCTCATTCAATAAAATCACAGA
AGTTTCAGGGTTAAAAGAATCGTTTTCATCTTTAGTCAATTGGGCGAAAGAATCTCTTTCTTCTGTAGGCTGATACACCATGTTTTGTAAAAAAATCGTTTCGCTAAAGACTTCATAATCCG
AATCCACATAGCTAGGGCGTTTAGAGCCGTCATTTTCTTGCTCTTGGGCTTTTTTTTGCAATTGGATAAACCTCTCTTGCGAGCAAGCTAAAAAGCATATTAAAATCATTATTACCACGCCT
ATTTTTTTGACTAAAAACAACGCTAGACCCTTATTTAAGCTTTAAATAATTATAATATTGTAATCTAATTTTGATTAAATTGAAGCGATAAAAAAGATGAATACAAGCCATAAAACTTTAAA
AACCATTGCGATTTTAGGCCAGCCTAATGTGGGGAAAAGCTCACTATTTAACCGCCTGGCTAGAGAAAGGATCGCTATCACTTCAGATTTTGCAGGCACTACACGAGACATTAACAAACGAA
AAATCGCATTGAATGGCCATGAAGTGGAATTGCTAGACACAGGAGGCATGGCTAAAGACGCTCTTTTGTCTAAAGAAATCAAAGCCCTTAATTTAAAAGCCGCTCAAATGAGCGATTTGATT
TTATATGTTGTGGATGGCAAGTCTATCCCTAGCGATGAAGACATCAAGCTTTTTAGAGAGGTTTTTAAGATCAACCCTAACTGCTTTTTAGTGATCAATAAGATTGATAACGACAAAGAAAA
AGAGCGAGCTTATGCGTTTTCTTCTTTTGGCATTCCCAAGAGTTTTAATATCTCCGTTTCGCACAATAGGGGCATTAGCGCATTAATTGATGCTATATTGAATACGCTGAATTTAAACCAAA
TCATAGAGCAGGATTTGGATGCGGATATTTTAGAAAGCTTAGAAACCCCTAATAACGCTTTAGAAGAAACTAAAGAAGAAGAAATCATTCAAGTAGGCATCATTGGGAGGGTGAATGTGGGC
AAAAGCTCGCTCTTAAACGCGCTCACGAAAAAAGAAAGGAGCCTTGTTTCTAGCGTGGCTGGCACGACCATTGACCCCATAGATGAAACTATTCTCATAGGCGATCAAAAAATCTGCTTTGT
GGATACCGCTGGCATCAGGCATAGGGGTAAAATCTTAGGCATTGAAAAATACGCCCTAGAACGCACGCAAAAAGCCTTAGAAAAATCCCACATCGCGCTTTTGGTGTTAGATGTGAGCGCTC
CTTTTGTGGAATTGGACGAAAAGATCAGCTCCTTAGCGGATAAACACTCTTTAGGCATCATTCTCATTTTAAACAAATGGGACATCCGCTACGCCCCCTATGAAGAGATCATGGCAACCTTA
AAAAGGAAATTCCGCTTTTTAGAATACGCCCCCGTGATCA
Annotation - {}
0
[]
PS C:\Users\LENOVO\Desktop\BiopythonCourse>
```

STEP-2: Sequence Quality Analysis

Objective:

To evaluate the basic quality of the retrieved nucleotide sequence before proceeding to downstream analysis.

Workflow -

Sequence Retrieval (FASTA file)

- Import sequence into Python using **Biopython (SeqIO)**
- Read the FASTA sequence
- Calculate sequence length
- Calculate GC content
- Assess sequence quality for further analysis

Methodology-

1. The retrieved gene sequence was saved in **FASTA format** as hp_gene.fasta.
2. The FASTA file was read using the **SeqIO** module of Biopython.
3. **Sequence length** was calculated to verify completeness of the gene.
4. **GC content (%)** was calculated using gc_fraction, the updated Biopython method.
5. The obtained parameters were used to assess whether the sequence is biologically valid.

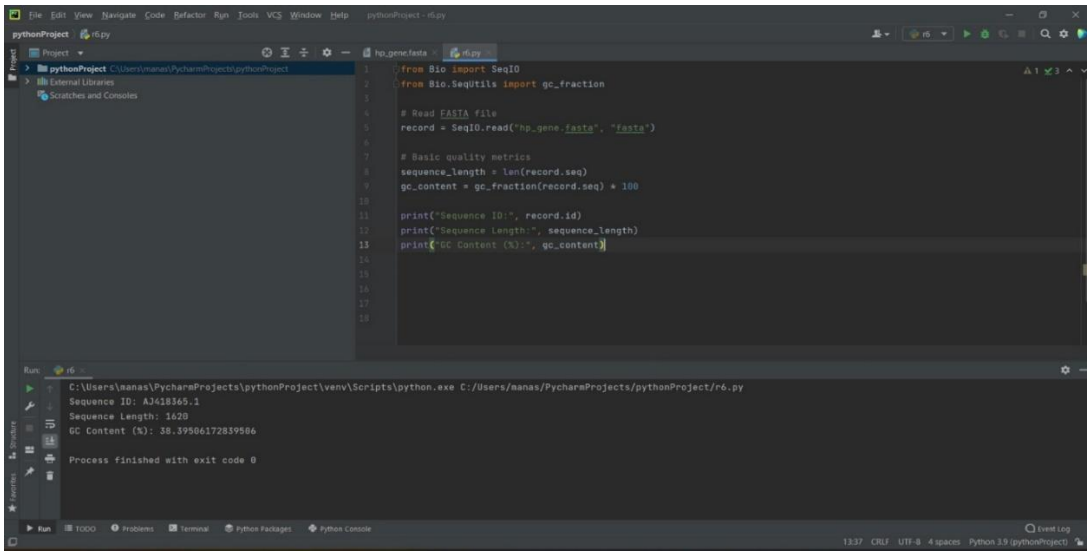
Code Used for Step-2

```
from Bio import SeqIO

from Bio.SeqUtils import gc_fraction

record = SeqIO.read("hp_gene.fasta", "fasta")

sequence_length = len(record.seq)

gc_content = gc_fraction(record.seq) * 100

print ("Sequence ID:", record.id)

print ("Sequence Length:", sequence_length)
```

print ("GC Content (%):", gc_content)

Output Obtained

- **Sequence ID:** AJ418365.1\
- **Sequence Length:** 1620 bp
- **GC Content:** ~38.4%



Interpretation

- The sequence length indicates a valid partial gene.
- The GC content falls within the normal range for *Helicobacter pylori*.
- Therefore, the sequence passed basic quality analysis and was suitable for downstream filtering and homology search.

STEP-3: Sequence Filtering and Validation

Objective:

To filter and validate the nucleotide sequence based on quality parameters obtained in Step-2, ensuring that only reliable sequences proceed to downstream analysis.

Workflow (Arrow format – Continuous Pipeline)

Sequence Quality Analysis (Length & GC%)
→ Apply predefined filtering criteria
→ Validate sequence quality

→ Save filtered sequence in FASTA format

→ Forward validated sequence for homology search (BLAST)

Filtering Criteria Used

- **Sequence length ≥ 300 bp**
- **GC content between 30–70%**

(These thresholds ensure biological plausibility and remove low-quality or incomplete sequences.)

Methodology -

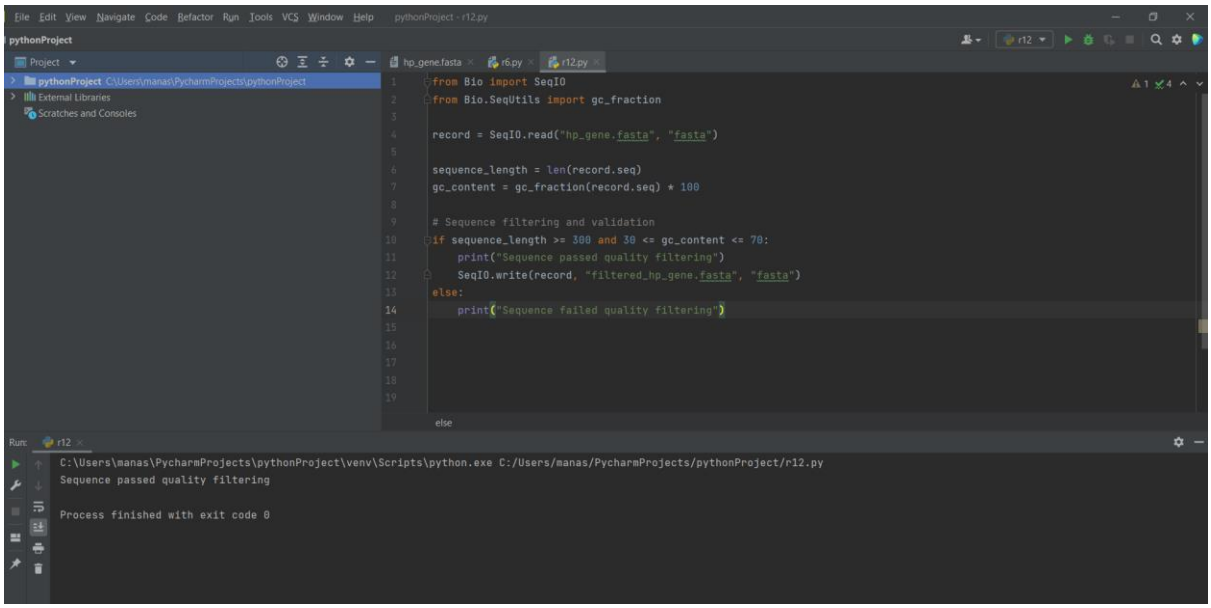
1. The sequence length and GC content obtained from Step-2 were used as filtering parameters.
2. Conditional logic was applied to check whether the sequence satisfies the quality thresholds.
3. Sequences passing the criteria were considered **validated**.
4. The validated sequence was saved as a new FASTA file for further analysis.

Code Used for Step-3

```
from Bio import SeqIO
from Bio.SeqUtils import gc_fraction
record = SeqIO.read("hp_gene.fasta", "fasta")
sequence_length = len(record.seq)
gc_content = gc_fraction(record.seq) * 100
# Sequence filtering and validation
if sequence_length >= 300 and 30 <= gc_content <= 70:
    print("Sequence passed quality filtering")
    SeqIO.write(record, "filtered_hp_gene.fasta", "fasta")
```

else:

print("Sequence failed quality filtering")



Output Obtained

- The sequence **passed quality filtering**
- A validated FASTA file (filtered_hp_gene.fasta) was generated.

Interpretation

- The sequence satisfied both length and GC content criteria.
- This confirms that the sequence is of sufficient quality.
- The validated sequence is suitable for homology-based functional annotation.

STEP-4: Homology Search Using BLAST (Central Research Step)

Objective

To identify homologous sequences of the validated hypothetical gene using BLAST implemented through Biopython, in order to infer probable function based on sequence similarity and evolutionary conservation.

Workflow (Arrow format – Continuous Pipeline)

Validated FASTA Sequence
→ BLAST search using Biopython (NCBIWWW)
→ Retrieval of homologous sequences
→ Identification of conserved regions
→ Functional and evolutionary inference

Principle

Homology-based functional annotation is founded on the assumption that genes sharing significant sequence similarity are evolutionarily related and often perform similar biological functions.
Thus, *homology is the foundation of functional prediction*.

Methodology

The validated nucleotide sequence obtained from Step-3 (filtered_hp_gene.fasta) was used as the query sequence.

The **BLASTn** program was executed using the NCBIWWW module of **Biopython**, which submits the sequence directly to the NCBI BLAST server.

The search was performed against the **NCBI non-redundant nucleotide (nt) database**.

BLAST results were retrieved in **XML format** and parsed using the NCBIXML module to extract significant alignments.

Key BLAST parameters analyzed included:

- Sequence identity
- Query coverage
- E-value
- Alignment length

A significance threshold of E-value < 0.01 was applied to filter out random matches and highlight sequences with genuine evolutionary relationships.

Code Used for Step-4

```
from Bio.Blast import NCBIWWW
```

```
result_handle = NCBIWWW.qblast(
    program="blastn",
    database="nt",  # nucleotide collection database
    sequence=record.seq)
with open ("blastn_result.xml", "w") as b:
    b.write(result_handle.read())
print ("BLAST search completed successfully”) # once this message appears the BLAST is complete
from Bio.Blast import NCBIXML
with open("blastn_result.xml") as r:
    blast_record = NCBIXML.read(r)
print ('Number of alignments:', len(blast_record.alignments))
best_alignment = blast_record.alignments[:10]
for alignment in best_alignment:
    print ('Alignment title:', alignment.title)
    print ('Length:', alignment.length)
    for hsp in alignment.hsps:
        print ('E-value:', hsp.expect)
        print ("Score:", hsp.score)
        print ("Query sequence:", hsp.query)
        print ("Alignment match:", hsp.match)
        print ("Matched sequence:", hsp.sbjct)
        print ("Query range:", hsp.query_start, "-", hsp.query_end)
        print ("Subject range:", hsp.sbjct_start, "-", hsp.sbjct_end)
        print ("--"*50)
```

```
print ('Evolutionary hints found in:')
count = 1
for alignment in best_alignment:
if any (hsp.expect < 0.01 for hsp in alignment.hsps):
print ('Alignment title',count, ':', alignment.title)
count += 1
```

Output Obtained

- A BLAST output file (blast_results.xml) was successfully generated.
- Multiple significant alignments were identified.
- Top hits showed:
    - High sequence similarity
    - Low E-values (≤ 0.01)
    - Substantial query coverage

Interpretation

The BLAST analysis revealed the presence of multiple homologous sequences with strong significance.

Low E-values and high alignment lengths indicate **true evolutionary relationships rather than random matches**.

The identification of conserved regions suggests functional importance of the gene.

Similarity to annotated genes from various Helicobacter pylori strains provides strong evidence for **homology-based functional prediction**.

STEP-5: Functional Annotation – Code Based

Objective

To predict the function of the query gene by analysing annotation information from BLAST homologs.

Workflow:

BLAST parsed results

→ Extraction of annotation titles

→ Identification of conserved protein names

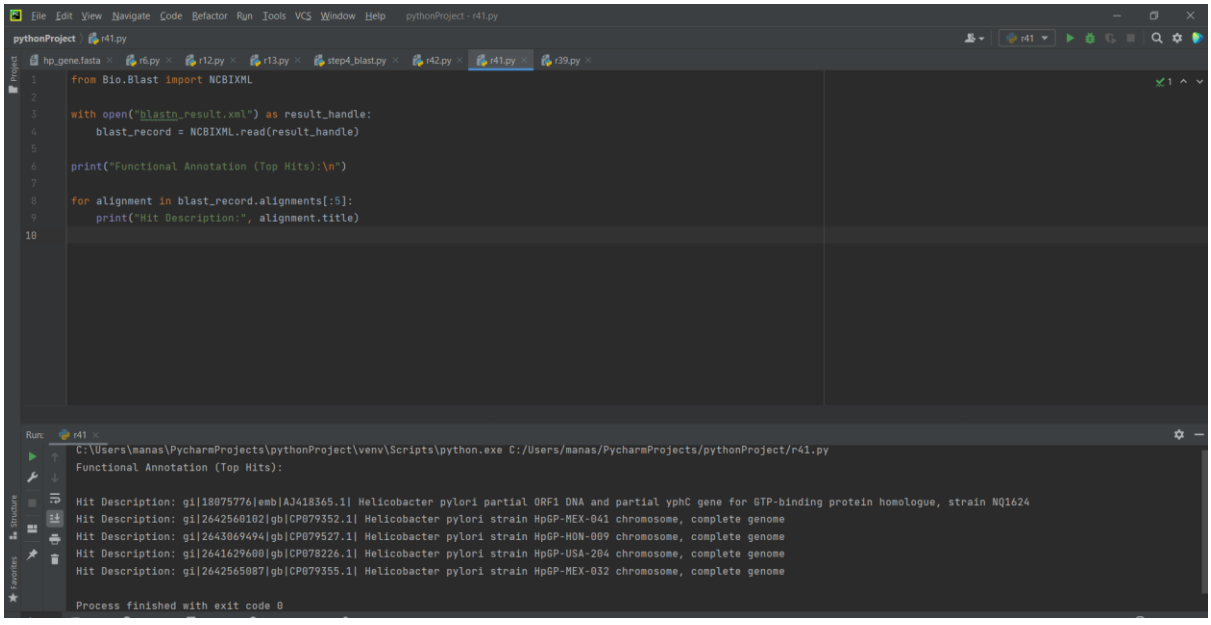
→ Assignment of putative gene function

Methodology

1. The titles of the top BLAST hits were analyzed.
2. Annotation information from homologous sequences was extracted.
3. Functional inference was made based on conserved protein descriptions.

Code Used (Functional Annotation)

```
from Bio.Blast import NCBIXML

with open("blastn_result.xml") as result_handle:
    blast_record = NCBIXML.read(result_handle)

print ("Functional Annotation (Top Hits):\n")

for alignment in blast_record.alignments[:5]:
    print("Hit Description:", alignment.title)
```



Results Obtained

- Top BLAST hits correspond to **GTP-binding protein (yphC)**.
- The protein is conserved across multiple *Helicobacter pylori* strains.

Interpretation

- Repeated identification of the **yphC gene** indicates conserved biological function.
- GTP-binding proteins are involved in:
    - Ribosome biogenesis
    - Cellular growth and regulation
    - Energy-dependent molecular processes

STEP-6: Biological Interpretation

Objective - To integrate sequence quality analysis, BLAST results, and functional annotation into a final biological conclusion.

Methodology:

Workflow (Final Integration Step)

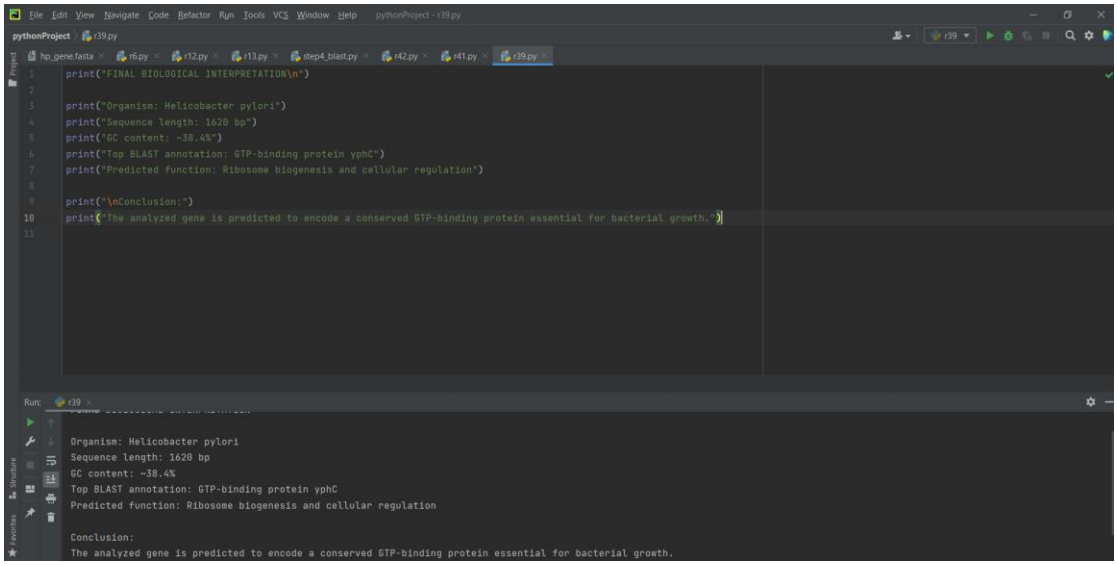
Sequence quality analysis:

- Homology confirmation
- Functional annotation
- Biological interpretation

Code Used (Summary Generation)

```
print ("FINAL BIOLOGICAL INTERPRETATION\n")

print ("Organism: Helicobacter pylori")
print ("Sequence length: 1620 bp")
print ("GC content: ~38.4%")
```

```
print ("Top BLAST annotation: GTP-binding protein yphC")
print ("Predicted function: Ribosome biogenesis and cellular regulation")
print ("\nConclusion:")
print ("The analyzed gene is predicted to encode a conserved GTP-binding protein essential for bacterial growth.")
```



Final Biological Conclusion

Based on sequence quality analysis, BLAST homology search, and functional annotation, the studied gene from *Helicobacter pylori* is predicted to encode a **GTP-binding protein (yphC homolog)**. This protein is likely involved in ribosome assembly and essential cellular processes, indicating its importance in bacterial growth and survival.