# BIFS 617 Final Project Instructions: ORF Finder

## Learning Objective

This final project will demonstrate your ability to apply Python to a biological data analysis problem.

## Team Requirement

You will work as a team of 2–3 students to design and implement a program that identifies open reading frames (ORFs) in DNA sequences provided in FASTA format in both forward and reverse. Students will be assigned during week 2 of the course. If you wish to choose your team, you must notify me via email before week 4 begins, with all members' names, emails, and consent.

## Project Objective

Your task is to write a Python program that identifies all ORFs in all 6 reading frames of each DNA sequence in the input file. The output should be in FASTA format and include metadata on frame, position, and length. Your program must be modular, commented, and include authorship on each function.

## ORF Code Requirements

- **One Team Member:** Fork the class repository to your personal GitHub account.
    a. https://github.com/slsevilla/2025_BINF_UMGC
- **All team members:** Clone the **FORKED** GitHub repository that contains example files and starter code to your local system.
- **Select team members:** Write your ORF-finding code using the provided \`starter_code/orf_finder_skeleton.py\`.
    a. Create a directory called /final_code/ where your complete code should reside
    b. Your code should accept the fasta file as an input, as well as ask for a minimum ORF length, with the default set to 5.
    c. Results should both be printed to the command line AND printed to a file
        - Create a directory called /output/orfs/
        - Final file should be named "orf_output.fasta".
    d. ORF's must include a stop codon, otherwise they should be skipped
    e. The output for each ORF should be

    - > seqID | FRAME COUNT | POSITION | LENGTH | DIRECTION (FOR,REV)
    - ORF SEQUENCE

  f. Example Output
    - > seq1 | FRAME = 1 | POS = 10 | LEN = 9 | REV
    - ABC DEF GHI

- **Select team members:** Use the `example_input/orfs_test_input.fasta` file as your test case.
  a. Use the ‘example_input/expected_results.txt’ to help check if your code is outputting ORF numbers, as expected
- **Select team members:** Add at least one visualization to your output (e.g., ORF length bar chart or position diagram).
  a. Create a directory called /output/visualization/
  b. The final file should be named orf_visualization.<extension>
- **All team members:** Commit and push your changes back to your forked GitHub repo.
- **One team member:** Submit your GitHub repo URL within the final_project document.

## Team Report Requirements

### Admin

- Length: 1–2 pages
- Format: PDF or Word (.docx)
- Submission: One student from the team must upload this report to Blackboard as team_report_group<number>.<extension>

### Purpose

- This report provides a high-level summary of your team's project development process. It highlights how your group collaborated, who contributed what, and includes the GitHub repository that contains your working code and visualizations.

### Required Sections

#### *Project Summary*

- Briefly describe the goal of your project: identifying ORFs across all 6 reading frames.
- Mention the tools you used (e.g., Python, BioPython, Matplotlib).
- Include one or two sentences summarizing your final output.

#### *Workflow and Task Division*

- Describe how your team structured the work.
- Describe how you decided to accomplish tasks between members, and how you communicated updates

#### *GitHub Repository*

- Provide the full URL to your team's forked GitHub repository.
- Make sure your code and comments are pushed and visible.
- Describe the benefits of using this tool when working on a team

#### *Challenges and Resolutions*

- Briefly describe any **technical** challenges you faced.
- Explain how your team resolved them or adapted.
- Future Improvements (Optional)
- Mention any features or enhancements you would implement with more time.
- 
- Examples: GUI interface, multi-threading, file upload, ORF translation to protein.

## Contributor Report Requirements

### Admin

- Length: 1–2 pages
- Format: PDF or Word (.docx)
- Submission: Each student must upload this report to Blackboard as contributor_report_group<number>_LastName_FirstName.<extension>

### Purpose

- This report allows you to reflect on your individual contributions, team dynamics, challenges, and suggestions for improving the project or your approach in the future. It also helps ensure fair and transparent collaboration grading.

### Required Sections

#### *1. Your Contribution*

- Describe the parts of the project you worked on in detail. Note whether you participated in testing, file organization, GitHub commits, or report writing.
    - Example: "I wrote and tested the find_orfs() function, handled reverse strand logic, and helped build the visualization."
- Mention any code files or functions you authored or helped debug.

#### *2. Team Collaboration*

- Reflect on how your team worked together. Was the collaboration successful? Were tasks distributed fairly?
- Mention if responsibilities shifted during the project or if you had to step up in any area.

#### *3. Challenges*

- Discuss any personal or technical difficulties you encountered. If relevant, comment on any interpersonal or scheduling challenges within the team.

#### *4. Future Improvements*

- If you had more time or resources, what would you improve?
    - Examples: more advanced error handling, protein translation, GUI/streamlit interface, using argparse, additional input validation.

#### *5. Project Feedback*

- Provide feedback on this project, for the course. Include things that you've learned or skills this project helped you develop. Provide feedback on the instructions, the timeline, and supporting files. Any other constructive feedback can go in this section as well.

## Deliverables

- Within the forked GitHub repository, the following should be included:
    a. Final code with documentation and author credits in a directory called /final_code/.
    b. At least one output visualization in a directory called /visualizations/.
- Submission to Blackboard:
    a. **One student submits for the team:** FASTA-formatted output of the provided input, with annotated ORFs, with the default length settings.
    b. **One student submits for the team:** A 1-to-2-page team report
    c. **Each member of the team submits:** A 1-page contributor report

## Grading

| Category | Excellent (Full) | Partial Credit | No Credit |
|---|---|---|---|
| | **8** | **4** | **0** |
| **Correct parsing of FASTA input** | Sequences and headers read and cleaned properly | Minor errors or assumptions | Incomplete or broken input handling |
| | **8** | **4** | **0** |
| **ORF logic: forward strands (1–3)** | All frames searched, ORFs found accurately | Some ORFs missed or frame errors | No or incorrect ORF identification |
| | **8** | **4** | **0** |
| **ORF logic: reverse strands (4–6)** | Reverse complement used properly, correct ORFs | Reverse strand implemented but buggy | Reverse strand not implemented |
| | **12** | **6** | **0** |
| **FASTA-formatted output** | Headers and sequences match spec | Output close but with formatting issues | Output missing or unreadable |
| | **12** | **6** | **0** |
| **Code structure & modularity** | Functions well-organized with docstrings | Some use of functions | Entirely procedural code |
| | **12** | **6** | **0** |
| **Visualization included** | Clear, informative, relevant | Basic or unclear | Missing |
| | **8** | **4** | **0** |
| **Comments & authorship** | Each function has author and purpose listed | Partial commenting or unclear credits | No comments or authorship |
| | **8** | **4** | **0** |
| **GitHub usage** | Repository is properly forked, committed, and URL submitted | Repository is properly forked but limited commits | Repository is not properly forked OR URL is not submitted OR commits are not included |
| | **12** | **6** | **0** |
| **Team Report** | Team workflow, contributions are equal and defined | Some imbalance noted OR workflow includes errors | Team report shows lack of collaboration OR workflow design |
| | **12** | **6** | **0** |
| **Individual Report** | Clear problem division, issues, future work | Partial or minimal summary | Missing or unclear report |