Forensic DNA Analysis Using Bioinformatics: A Hands-On Introduction

Maria Joseph

Department of Science and Engineering, Penn State Abington

ENGL 202C: Effective Writing

Dr. Charles Archer

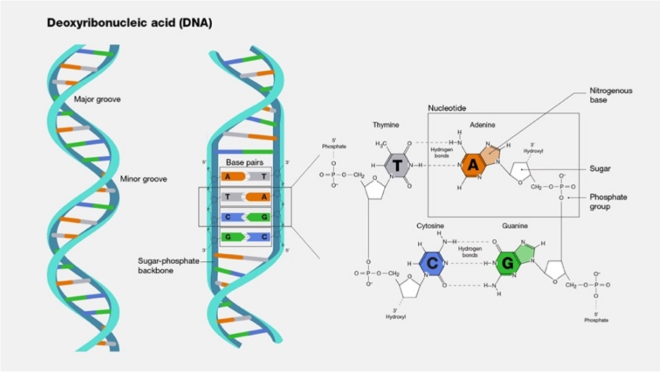
February 23, 2025

Table of Contents

Fundamentals of Forensic DNA Analysis and Bioinformatics

Forensic science uses DNA analysis—a powerful tool relying on STRs, mtDNA, and other genetic markers--to identify suspects and prove the innocence of those wrongly accused by comparing biological evidence from crime scenes to known samples. *DNA (deoxyribonucleic acid)* is a unique genetic blueprint found in nearly all living organisms, making it a powerful tool for forensic investigations. Because everyone's DNA (except in identical twins) is unique to each individual, forensic scientists can accurately compare samples from crime scenes to those of potential suspects. Analysis involves extracting, amplifying, and comparing genetic material to match evidence with known individuals. This process is commonly used in criminal investigations, missing persons cases, and disaster victim identification.

DNA consists of four nucleotide bases: adenine (A), thymine (T), cytosine (C), and guanine (G) arranged in a double-helix structure. The sequence of these bases creates the genetic code that determine an organism's traits. STRs are short sequences of DNA (2-6 base pairs long) repeated multiple times at specific genome locations. The number of repeats varies between individuals, making STRs a reliable profiling tool, which involves polymerase chain reaction (PCR) amplification and capillary electrophoresis.



Mitochondrial DNA (mtDNA), inherited exclusively from the mother, is found in multiple cell copies, making it useful for analyzing degraded samples. Unlike STRs (nuclear DNA), mtDNA sequencing is often used for unidentified human remains, degraded samples (e.g. hair, bones, and teeth), and maternal lineage investigations.

Bioinformatics uses computational tools to analyze biological data, including DNA sequences.

By the end of this manual, you will gain an understanding of:

- The role of bioinformatics in forensic DNA analysis,
- How to process and compare DNA sequences using FASTA files,
- The significance of STR analysis in forensic investigations, and
- How to generate statistical models and visualizations to interpret forensic DNA evidence.

Intended Audience

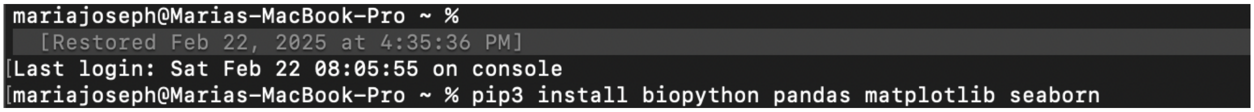
These introductory manual guides beginners and those who are interested in learning how to apply computational tools to forensic DNA profiling. Basic Python programming knowledge is helpful but not required. Topics covered include processing and comparing DNA sequences, analyze STR patterns, and visualizing data using Python-based bioinformatics tools.

Setting Up the Environment for Forensic Bioinformatics

This section guides you through setting up the necessary software and resources for forensic bioinformatics analysis. We'll use Python and key libraries.

1. Installing Required Software and Libraries

Download and install **Python 3.10 or later** from Python.org and follow the installation instructions for your operation system. Open **Terminal (macOS/Linux) or Command Prompt (Windows)** and run the following commands to install the necessary packages:



```
mariajoseph@Marias-MacBook-Pro ~ %
   [Restored Feb 22, 2025 at 4:35:36 PM]
[Last login: Sat Feb 22 08:05:55 on console
[mariajoseph@Marias-MacBook-Pro ~ % pip3 install biopython pandas matplotlib seaborn
```

- Biopython: Handles DNA sequence processing
- Pandas: Manages and analyzes data
- Matplotlib & Seaborn: Creates graphs and visualizations

2. Setting Up a Working Directory

It is highly recommended to create a dedicated directory on your computer for your forensic bioinformatics projects. This will help you keep your files organized and prevent conflicts with other projects. For example, you could create a folder named "Forensic DNA Analysis" on your desktop.

3. Downloading and Setting Up an IDE

Jupyter Notebook is an interactive environment ideal for forensic bioinformatics, allowing you to run Python code step-by-step. The easiest way to install this application is through Anaconda. Open your web browser and go to the official Anaconda download page and download and install the application based on your operating system, following the on-screen instructions:
[insert image of Jupyer

   a. Launching Jupyter Notebook

Step-by-Step Guide to DNA Analysis

4. Downloading Sample DNA Sequences
   a. Crime Scene DNA Sample

We will use a synthetic crime scene file simulating an unknown DNA sample. You can download a sample FASTA file from the project's GitHub repository: https://github.com/mjosewings/Forensic-DNA-Analysis-Tutorial/

b. Suspect DNA Profiles

To compare crime scene DNA to suspect DNA, download 3-5 different multiple human mitochondrial or STR sequences in FASTA format from GenBank (NCBI) or STRBase. Genbank is generally recommended for beginners due to its extensive FASTA sequence collection, simplifying search, download, and direct use in bioinformatics tools. STRBase, while an excellent resource for STR information, often provides allele data, requiring consultation of additional resources for conversion.

When searching, use terms such as:

- mtDNA: "human mtDNA control region," "human mitochondrial DNA D-loop," "Homo sapiens mitochondrial DNA," specific haplogroup designations (e.g., "human mtDNA haplogroup H")
- STRs: Specific STR marker name (e.g., "D13S317," "TH01", "vWA"), "human STR alleles" or "human STR sequences," specific allele (e.g. "D13S317 allele 14")

Be sure to name the downloaded files descriptively (e.g., "mtDNA_haplogroup_H.fasta," "D13S317.fasta") to avoid confusion and save them to a "suspects" subfolder in your working directory.

5. Verifying the Environment Setup

To test that everything is correctly installed, run the following Python script to check Biopython.

If this script runs without errors and prints the first 50 bases of your sequence, you are ready to proceed with forensic DNA analysis!

Troubleshooting and Common Issues

This section provides solutions to common problems you might encounter while setting up your environment and performing forensic bioinformatics analysis.

1. Installation Issues
    a. Python or Pip Not Recognized

Problem: Python's Scripts directory is not in your system's PATH environment variable.

Solution: Search online for "add Python to PATH [Your Operating System]" (e.g., "add Python to PATH Windows," "add Python to PATH macOS"). Reopen terminal/command prompt after adding to PATH.

b. Library Installation Fails

Problem: Network issues, outdated pip, or other conflicts.

Solution: Ensure that you have a stable internet connection

- Ensure you have a stable internet connection.
- Upgrade pip:

c. Jupyter Notebook Won't Open or 'jupyter' Not Found

Problem: Jupyter Notebook may fail to open.

Solution:

- If installed via Anaconda, launch Jupyter Notebook from Anaconda Navigator.
- If installed via pip, ensure Jupyter is in your PATH. Try reinstalling:

[insert screenshot of the command code]

2. File and Path Issues

    a. File Not Found (FASTA Files)

Problem: You get a "File not found" error when trying to open your FASTA files.

Solution: Double check the file path is in your code with the correct spelling and the FASTA file is in the correct directory.

3. Sequence Analysis Errors

    a. Invalid FASTA Format

Problem: Biopython reports an error about an invalid FASTA format.

Solution: Verify that the FASTA file starts with a ">" and a sequence identifier; check for valid nucleotide characters (A, T, C, G), and with hidden characters using a text editor.

b. Alignment Issues

Problem: If your sequence alignments are incorrect or not working.

Solution: Verify the input sequences by making sure that they are in the same orientation, inputted correctly, and the alignment tool parameters.

4. Graph Generation Errors

    a. No Graph Appears or Graph is Empty/Incorrect

Problem: Your Python script generates graphs, but nothing appears, or the graph is empty or incorrect.

Solution: Ensure that Matplotlib and Seaborn are correctly installed, check the data being used to generate the graph, and review the code for any logical errors. Be sure to

5. General Debugging Tips

- **Read Error Messages Carefully:** Error messages provide clues about the problem.
- **Use Print Statements:** Insert print() statements in your code to check the values of variables and identify where errors occur.
- **Search Online:** Use search engines to find solutions to specific error messages.
- **Check Documentation:** Refer to the documentation for the Python libraries you're using.
- **Restart Jupyter Notebook:** Sometimes restarting the kernel in Jupyter Notebook can resolve issues.

Practical Considerations and Challenges in Forensic DNA Analysis