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

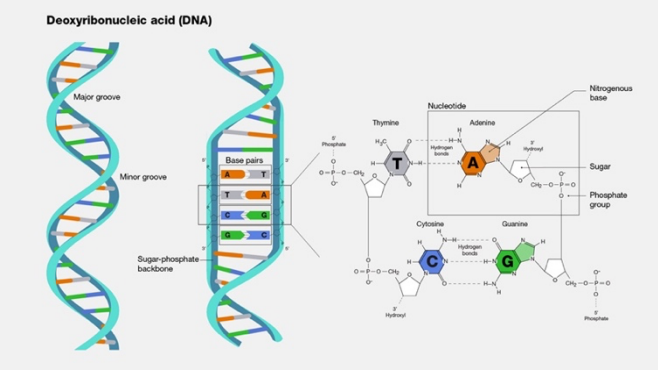
February 23, 2025

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 everyone, 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, analyzing 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 PyCharm, which includes Python and key libraries.

1. Installing Required Software and Libraries

Ensure you have Python installed (preferably **Python 3.10 or later**). Download Python from **Python.org** and complete the setup as directed for your operating 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

PyCharm is an interactive environment ideal for forensic bioinformatics, allowing you to run Python code step-by-step. The easiest and recommended way to install this application is through the official JetBrains PyCharm website and download the free "Community" edition: https://www.jetbrains.com/pycharm/

a. Downloading the Project Code

Before proceeding, you'll need to download the project code from GitHub. This code includes the Python scripts necessary for the forensic DNA analysis. Open the web browser and go to the project's GitHub repository: https://github.com/mjosewings/Forensic-DNA-Analysis-Tutorial/. To download the code, click on the "Code" button, select "Download ZIP," and extract the files to your working directory (the "Forensic DNA Analysis" folder you created earlier).

b. Launching PyCharm

To begin using PyCharm and open your project, click "Open" on the welcome screen, and select your "Forensic DNA Analysis" working directory.

Step-by-Step Guide to DNA Analysis

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

The synthetic crime scene FASTA file is included in the project code you downloaded from GitHub. Ensure that the file "crime_scene_dna.fasta" is located in your working directory.

    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 ensure that everything is installed correctly, run a simple test using Python and Biopython by running all the scripts in the following order:

- load_sequences.py
- analyze_results.py
- generate_report.py
- graphs.py

After running graph.py, a series of graphs should appear. If the scripts run without errors and the graphs are generated, your environment is set up correctly. If you encounter any errors, please refer to the troubleshooting section.

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

Add Python to your system's PATH. Search online for "add Python to PATH [Your Operating System]" for step-by-step instructions, then reopen terminal/command prompt.

    b. Library Installation Fails

Library installation failures can be due to network issues, outdated pip, or other conflicts. Ensure a stable internet connection and upgrade pip if necessary.

    c. PyCharm Won't Open

Verify that you downloaded the correct version for your operating system and that the installation process completed successfully. If issues persist, try reinstalling PyCharm.

2. File and Path Issues
    a. File Not Found (FASTA Files)

Double check the file path is in your code, the spelling, and ensure the FASTA file is in the correct directory.

3. Sequence Analysis Errors
    a. Invalid FASTA Format

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

Verify the input sequences, ensuring that they are in the same orientation, inputted correctly, and the alignment tool parameters are accurate.

4. Graph Generation Errors
    a. No Graph Appears or Graph is Empty/Incorrect

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.

5. General Debugging Tips

Read carefully for any error messages as they often provide valuable clues, use print() statements within your code to check the values of variables, and search online and solutions and consulting the documentation of the Python libraries you are using can be very helpful.

Practical Considerations and Challenges in Forensic DNA Analysis

Forensic DNA analysis, while powerful, faces practical challenges. The quality and quantity of the DNA samples (degraded or limited) from crime scenes that significantly impact results in creating incomplete or unreliable profiles. For instance, contamination is a constant concern in tracing amounts of foreign DNA that can skew analysis. Furthermore, the interpretation of complex DNA mixtures requires careful consideration, especially in cases that involves multiple contributors, which plays a crucial role in court for evidence, allowing for deciphering these complex profiles and distinguishing individuals' contributions can be difficult. The effective presentation of DNA evidence requires clear communication to non-scientific audience, bridging the gap between complex science and legal understanding.

Beyond technical aspects, ethical considerations are essential in terms of issues that surrounds around privacy, data storage, and potential for misuse of genetic information must be addressed. Ensuring the reliability and validity of DNA analysis methods is also an ongoing challenge, requiring continuous validation and quality control. Additionally, the increasing complexity of DNA analysis software and tools requires thorough training and expertise to avoid misinterpretations.

References