Notebook Documenting the Development Process & Results of a Python Tool for the Purpose of Understanding HNH Endonuclease in Bacteriophage

Table of Contents:

1. Installing BLAST+ locally
2. Adding BLAST+ to the system PATH
3. Installing Python
4. Installing Biopython
5. Configuring your IDE to work with Python
6. Using Biopython modules
7. Creating a file structure
8. Writing the code
9. Using the tool
10. Results

**Installing BLAST+**

To build and utilize the HNH endonuclease gene search function, BLAST+ must first be downloaded and extracted properly. Normally, BLAST+ functions are called via a command line environment – in this instance however, BLAST+ functions will be used by Biopython modules. The BLAST+ source code and executables can be found at <https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/> for download. The proper version for your system(s) must be downloaded, in this instance I will be using the Linux version as a ‘tar.gz’ archive. Additional breakdown on how to install BLAST+ can be found at <https://www.ncbi.nlm.nih.gov/books/NBK279671/>.

**Adding BLAST+ to the system PATH**

To allow the Python program to interface with BLAST+ functions, we must point the operating system PATH to the proper system directory where the BLAST+ bin folder is located.

This can be accomplished in UNIX systems through bash:

**$ export PATH=$PATH:$HOME/ncbi-blast-2.10.0+/bin**

**Installing Python**

We must ensure that Python 3x. is installed on the system. To find the current version of Python or check if Python is installed run the following command in bash:

**$ python -V**

If python is not installed or up to date, run:

**$ apt-get update**

**$ apt-get install python3**

**Installing Biopython**

We must also ensure that Biopython is installed so that the Python program can properly import the needed modules.

Running the following command will use the Python package management tool to quickly install Biopython on any platform:

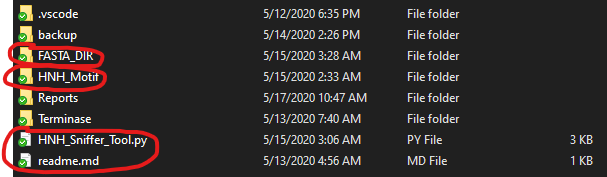
**$ pip install biopython**

For updating an older version of Biopython, use:

**$ pip install biopython --upgrade**

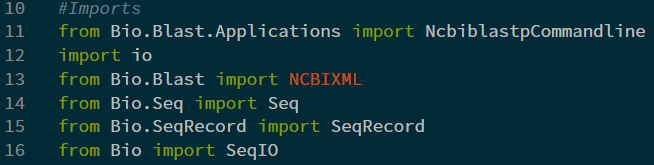
**Creating a file structure**

To keep the project organized and allow for easy access to FASTA or FAA files; a file structure must be created. This file structure consists of a folder for HNH endonuclease consensus sequences and a folder for FAA genomes for comparison. The Python program and the markdown readme file will exist in the primary directory.



**Writing the Code**

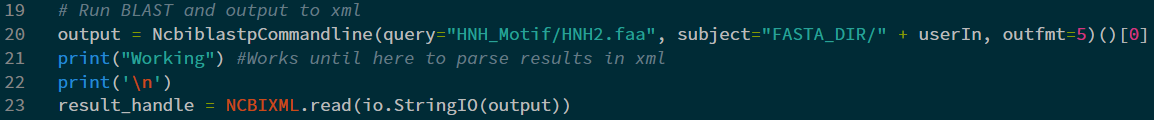
The program must first import the necessary modules from the Biopython library like the ‘NcbiblastpCommandline’ function, and other dependencies like the ‘io’ library.



The next step is writing a line that allows for user input and will determine which FASTA/FAA file should be analyzed for HNH endonuclease consensus similarities.



Once we have the ‘userIn’ variable, we must pass this string on to the ‘NcbiblastpCommandline()’ function so it blasts the correct file for comparison. After the blastp function completes, the ‘output’ variable must take the output and pass it to a handler, in this case we will use the ‘NCBIXML.read()’ function before passing it on for alignment.



The handled information will then be passed on and parsed via ‘alignment’. The parsed information will then be displayed on the output console only if the alignment has an E-value less than 0.04.



Using this code (lines 24 – 48), the title, e-value, alignment length, position, and alignment visualization will be printed out in a clean and ordered fashion.

**Using the Tool**

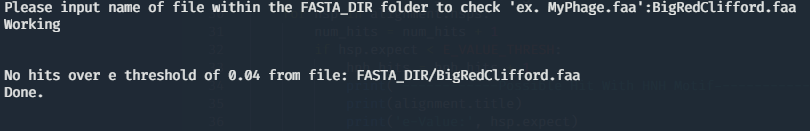
Using the tool is a simple matter of invoking the python script:

**(directory where the project is located)~$ python HNH\_Sniffer\_Tool.py**

The program then prompts the user.

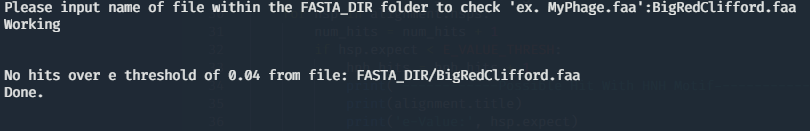


Giving the program a valid input, ‘BigRedClifford.faa’ for example – will compare this file to the HNH endonuclease consensus sequence.

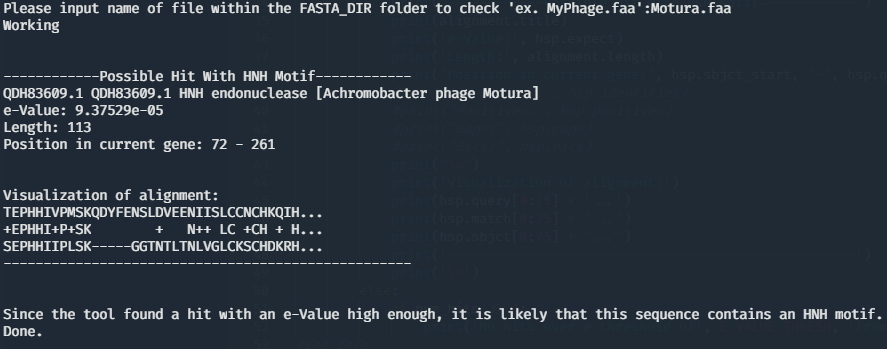


**Results**

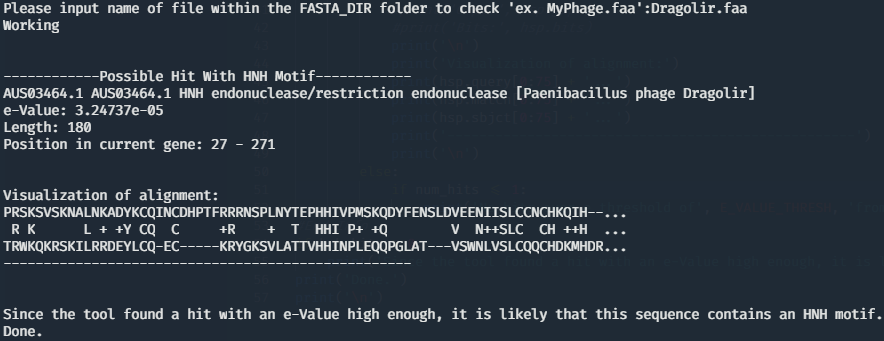
When testing the program with a genome file known not to contain HNH endonuclease, the results were as expected – running ‘BigRedClifford.faa’ through the program returned no hits over an e threshold of 0.04.



Testing a file that is known and annotated to contain HNH endonuclease also yielded results as expected – running the Achromobacter phage Motura, complete genome (QDH83728.1 accession) yielded the following result in the program’s output.



Testing an arbitrary file that was not known at the time to contain HNH endonuclease allowed for a quick search of the genome. Running the Paenibacillus phage Dragolir, complete genome (AUS03409.1 accession) yielded this result in the program output.



This output lends itself to indicating the presence of HNH endonuclease within the Dragolir genome. The e-value falls within the 0.04 threshold and is clearly annotated within the file as, “HNH endonuclease/restriction endonuclease”. With this output, we can be confident concluding that HNH endonuclease exists within this genome.

Any amino acid sequence can be input through the program. The tool has shown to be reliable at quickly identifying HNH endonuclease within bacteriophage genomes and is ready to be developed into a more mature tool for enhancing the study and research of HNH endonuclease as a whole.