**Determining the presence or absence of nucleotide sequences: a data science approach.**

**Bristol Student No. 2208260**

**Introduction**

NCBI provides a useful BLAST feature which allows the user to screen a public database for entries which contain and/or are highly similar to the query. But when interested in a lab strain not yet entered into the database, the BLAST tool cannot be used to search its genome for operons, genes, etc. To address this problem, here we have created a script which will screen multiple loaded files for a sequence of interest.

To demonstrate its functionality, five plasmids from classic Cystic Fibrosis (CF) pathogens were chosen at random to be screened for the antimicrobial resistance gene (AMR), *ampR.* For context, CF is a genetic condition caused by mutations of the CF transmembrane conductance regulator gene. Common CF pathogens include *Staphylococcus aureus* (in younger patients), *Pseudomonas* *aeruginosa*, *Achromobacter* spp., *Stenotrophomonas* *maltophilia*, and those defined in the *Burkholderia* *cepacia* complex (Blanchard, A. C. and Waters, V. J., 2019). Antimicrobial strategies are employed to treat infections however the rapid increase in AMR strains renders them ineffective in some cases (Chmiel, J. F., et al., 2014)). *ampR* is an example of an AMR gene, a transcriptional regulator which confers β-lactam resistance (Kuga A., et al., 2000).

Briefly, the python script loads pre-saved FASTA files into the workspace using ‘SeqIO’ from Biopython, extracts and assigns the nucleotide sequence from SeqRecord to a usable variable, and finally loops over a dictionary of these sequences and aligns them with the gene of interest using the PairwiseAligner within ‘Align’ from Biopython. Ultimately, the output will inform the user whether the AMR gene of interest is present in the sequences loaded.

**Methods**

The script was written in the JupyterLab 3.2.1 application supported by the Anaconda Navigator software. It can be found at, <https://github.com/NaomiH98/data_science_assessment>

*Preparing FASTA files*

To use this alignment script, the sequences of interest must be prepared and accessible to the JupyterLab file directory as FASTA files. The script written used the files as detailed in Table 1.

|  |  |  |
| --- | --- | --- |
| **Table** **1** |  |  |
| NCBI Accession Number | Details | URL |
| X03274 | *Pseudomonas* *aeruginosa* plasmid pMG7 | <https://www.ncbi.nlm.nih.gov/nuccore/X03274.1?report=fasta> |
| NC\_005564.1 | *Staphylococcus* *aureus* plasmid pS194 | <https://www.ncbi.nlm.nih.gov/nuccore/X03274.1?report=fasta> |
| NZ\_KJ588780 | *Achromobacter* *xylosoxidans* strain A22732 plasmid PA22732-IMP | <https://www.ncbi.nlm.nih.gov/nuccore/NZ_KJ588780.1?report=fasta> |
| NC\_010464.1 | *Stenotrophomonas* *maltophilia* plasmid pSM76 | <https://www.ncbi.nlm.nih.gov/nuccore/NC_010464.1?report=fasta> |
| NZ\_LR890526.1 | *Burkholderia* *cepacia* 4Asc2280433 | <https://www.ncbi.nlm.nih.gov/nuccore/NZ_LR890526.1?report=fasta> |
| NC\_002516.2 | *Pseudomonas aeruginosa* PAO1 | <https://www.ncbi.nlm.nih.gov/nuccore/NC_002516.2> |
| NC\_002516.2 | *ampR* | <https://www.ncbi.nlm.nih.gov/nuccore/NC_002516.2?report=fasta&from=4592990&to=4593880&strand=true> |

*Loading prepared FASTA files and extracting sequence information*

SeqIO from Biopython is used to import the prepared FASTA files. Nucleotide sequences were extracted and assigned to unique sequence variable names.

*Creating a dictionary*

Once extracted, the sequence variables were used to create a dictionary. This is later included in a loop to align the plasmid sequences with the *ampR* gene and report on the presence or absence of *ampR*.

*Alignment*

The Align module was imported from BioPython in order to use the PairwiseAligner() submodule. The alignment type to be conducted is a local alignment and this is set using the .mode function. The alignment method being used is checked using .algorithm. The default setting for the alignment conducted by PairwiseAligner() is to award the alignments one point for a match, and zero points for both gaps and mismatches. Here, we changed the gap score to a penalty of -0.1 using .gap\_score. .score was used to calculate an alignment score for *ampR* to the plasmid sequences previously mentioned. Test scores were compared against the score of our control alignment of *ampR* against its origin genome (*P. aeruginosa* PAO1*).*

Relevant pieces of the code presented here was adapted from Matt Williams Beginning Python and Intermediate Python (Williams, M.) and the Biopython Tutorial and Cookbook (Chang, J., et al., 2021).

**Results**

To demonstrate the functionality of the code written, four commonly occurring CF pathogen plasmids (Table 1) were chosen at random from the NCBI database and screened for a well characterised AMR gene known as ampR. A control was also included to compare the alignment scores of our test alignments against. Within the methodology, there is a variable stringency and the user can determine how similar the test alignment scores must be to the control score for the gene to be present. In our script, two stringencies (low, 60% and high, 95%) were tested. At the lower stringency threshold, all plasmids were found to include the AMR gene of interest. At the higher stringency, the absence of *ampR* *Staphylococcus* *aureus* plasmid pS194 was determined. It should be noted that this alignment method is not the same as that used by the NCBI blastn tool and so alignment scores can differ.

**Conclusion**

At the higher stringency threshold, all pathogen plasmids contain the *ampR* gene except *S. aureus* plasmid pS194 This script can be used to determine the absence and presence of a gene of interest in bacterial nucleotide sequences such as plasmids.

**References**

Blanchard, A. C., & Waters, V. J. (2019) ‘Cystic Fibrosis: Advances in Understanding and Treatment: Microbiology of Cystic Fibrosis Airway Disease’, *Seminars in Respiratory and Critical Care Medicine*, 40(6), pp. 727. <https://doi.org/10.1055/S-0039-1698464>

Chang, J, Chapman, B., Friedberg, I., Hamelryck, T., de Hoon, M., Cock, P., Antao, T., Talevich, e., Wilczynski B. (2021) *Biopython Tutorial and Cookbook: 6.6.2 PairwiseAligner.* Available at: [Biopython Tutorial and Cookbook](http://biopython.org/DIST/docs/tutorial/Tutorial.html#sec101) (10/01/2021)

Chmiel, J. F., Aksamit, T. R., Chotirmall, S. H., Dasenbrook, E. C., Elborn, J. S., LiPuma, J. J., Ranganathan, S. C., Waters, V. J., & Ratjen, F. A. (2014) ‘Antibiotic management of lung infections in cystic fibrosis: I. The microbiome, methicillin-resistant Staphylococcus aureus, gram-negative bacteria, and multiple infections’, *Annals of the American Thoracic Society*, 11(7), pp. 1120–1129. <https://doi.org/10.1513/ANNALSATS.201402-050AS/SUPPL_FILE/DISCLOSURES.PDF>

Kuga, A., Okamoto, R., & Inoue, M. (2000) ‘ampR Gene Mutations That Greatly Increase Class C  β-Lactamase Activity in Enterobacter cloacae’, *Antimicrobial Agents and Chemotherapy*, 44(3), pp. 561. <https://doi.org/10.1128/AAC.44.3.561-567.2000>

Williams, M. *Beginning Python: Dictionaries*. Available at: [Introduction - Beginning Python (milliams.gitlab.io)](https://milliams.gitlab.io/beginning_python/) (10/01/2021)

Williams, M. *Intermediate* *Python: Using functions*. Available at: [Using functions - Intermediate Python (milliams.com)](https://milliams.com/courses/intermediate_python/Using%20functions.html) (10/01/2021)

Williams, M. *Intermediate* *Python: Writing functions*. Available at: [Using functions - Intermediate Python (milliams.com)](https://milliams.com/courses/intermediate_python/Using%20functions.html) (10/01/2021)