Local BLAST Protocol for finding target species in a set of DNA sequences generated from eDNA

This tutorial provides instructions on searching for target species in an eDNA data set generated using Nanopore sequencing. Basically, you isolate eDNA, PCR amplify a barcoding gene tagging individual samples with barcoding tags, and then sequence the resulting PCR products using Nanopore sequencing. Now you wish to detect a target species and what samples it occurred in.

To do this, you will download the BLAST program on to your computer, download the sequence of your target species in FASTA format from Genbank, and BLAST the target sequence against the reference database generated from the Nanopore eDNA sequences. For the reference database, you will need to filter the sequences by size and quality before using it.

This protocol is written to work on a Linux operating system and be run from the command line. I use Oracle's virtual machine box and run a Linux Ubuntu virtual machine on my PC.

***Basic Protocol (BLAST portion only):**

- Download the appropriate version of BLAST on to your computer
  - https://blast.ncbi.nlm.nih.gov/doc/blast-help/downloadblastdata.html
- Put the eDNA Nanopore sequences in FASTA format and the barcoding gene sequence of the target species from Genbank into the folder that you will work in
- Using a command line shell, navigate to the respective folder
- Convert the Nanopore sequence file into a local database by using the following command:
  - makeblastdb -in nanopore_sequence_file_name.fasta -dbtype nucl -out local_database
  - Note that the "nanopore_sequence_file_name.fasta" should be replaced with the actual name of the nanopore sequence file. Also note that this file should be in Fasta format.
  - This command is necessary for BLAST to know how to search through our sequences to find the target species we are looking for
- Use the following command to search for a particular focal species in the local database of sequences that we generated:
  - blastn -db local_database -query target_sequence.fasta –out results.fasta
  - What we are doing is telling BLAST to look for sequences similar to the target_sequence.fasta sequence downloaded from Genbank and put the results in a new file called "results.fasta". This can be repeated using other target species.

***Extended Protocol - Protocol that includes filtering and separating sequences based on Barcode Tags:**
This protocol provides more details on steps for manipulating the data files, filtering the data and identifying and separating sequences by the sample they came from.

Running Nanofilt: Filtering the Nanopore sequence data by size and quality
Nanofilt can be run from a Conda environment to avoid issues with software incompatibilities.

- #Navigate to folder with sequence data and activate conda environment

    -
      ```
      conda activate nanofilt_test1
      ```
    - #nanofilt test1 is the name of a previously created Conda environment for using the nanofilt program.
- Run Nanofilt:

    -
      ```
      NanoFilt --logfile LOGFILE~ -l 350 --maxlength 800 -q 10 input_file.fastq | gzip > filtered_output_file.fastq.gz
      ```
    - The command creates a log file, filters fragments by size including only fragments that are between 350 and 800 bp, and filters by quality, including only sequences that have a quality score of 10 (90% predicted accuracy).
- #Unzip the data by right clicking in explorer and asking to unzip
- #Count lines with nano. The number of sequences is the number of lines divided by 4:
- nano filtered_output_file.fastq

Converting data from fastq to fasta format:

There are multiple ways to convert files from fastq to fasta format. This is a very simple script that seems to work well.

- Convert from fastq to fasta format using sed script:

    -
      ```
      sed -n '1~4s/^@/>/p;2~4p' filtered_output_file.fastq > filtered_output_file.fasta
      ```

Using BLAST:

BLAST is a very popular bioinformatics tool that can be run online or downloaded on your computer for local use. Online, you can upload a sequence you generated and get the top matches from the entire Genbank database. We will download the program to our computer and use it to transform the Nanopore eDNA sequences into a reference database. We can then BLAST a target species downloaded from Genbank against our Nanopore sequences to see if we get any fits. By tagging samples with barcodes, we can also multiplex several samples. We can then bioinformatically separate the samples by putting the sequences with different barcodes in different files.

- Create the BLAST reference database form the Nanopore sequences:

    -
      ```
      makeblastdb -in filtered_output_file.fasta -dbtype nucl -out blastdb
      ```
    - This makes a local database out of our filtered Nanopore sequences. We can now search for any species in this database using BLAST.
- Run basic BLAST (this is a command for a basic BLAST):

    -
      ```
      blastn -db blastdb -query target_sequence.fasta
      ```
    - BLAST searches the reference database that we created in the previous step (blastdb) for our target species (target_sequence.fasta). The target species sequence is in FASTA format and is a sequence downloaded from Genbank or that you generated.

- Run BLAST with a minimum percent identity and saving the output in a table format:

    -
      ```
      blastn -db blastdb1 -query target_sequence.fasta -outfmt 6 -perc_identity 90 -out blast_results_table
      ```
    - The outfmt 6 is what makes the output be in a table format and the -perc_identity sets the minimum similarity threshold for selecting sequences.
    - Saving in a table format will allow us to extract the sequence IDs as a column and use these to separate the matching sequences from our filtered FASTA file.

Separate the sequences that are matches for the target into a new file:

- First, we have to extract the IDs column from the table file that was output from the BLAST search. We can do this using the **awk** command:

    -
      ```
      awk '{print $2 > " matches_ID_column "}' blast_results_table
      ```
- #This prints column 2, the IDs column (matches_ID_column), from the blast_results_table file generated as output from the BLAST search.
- We can then use this list to separate the matches into a new file in FASTA format with the **grep** command:

    -
      ```
      grep -A 1 -f matches_ID_column filtered_output_file.fasta > target_match_sequences.fasta
      ```
    - -A 1 is pulling the actual DNA sequence along with the line with the string match.
    - -f indicates that the motif grep is searching for is in a **file**. In this case, that file is named “matches_ID_column”
    - filtered_output_file.fasta is the original FASTA file with the filtered sequences
    - The output will be stored in a file called target_match_sequences.fasta
- Unfortunately, grep puts in “—” (two dashes) and an extra line between the records pulled, so we have to fix this at the command line using the **sed** command:

    -
      ```
      sed 's/--//g' target_match_sequences.fasta > target_match_sequences_ed.fasta
      ```

    -
      ```
      sed '/^$/d' target_match_sequences_ed > target_match_sequences_ed_b.fasta
      ```
    - The first line of code removes the “- -“, while the second line of code removes the empty line between records.
    - Now we have a FASTA file with all the sequences that matched our target.
- Finally, we need to demultiplex the matches in FASTA format, if we performed a multiplexed sequencing reaction, which we usually do. This will tell us which of our samples had the target species, and also how abundant the target was in each sample, or at least how many target PCR products were generated from each sample.

    -
      ```
      grep -B 1 -i -s 'aactgactaacc' target_match_sequences_ed_b.fasta > target_matches_in_sample_with_Barcode_F1
      ```
    - This puts all sequences with that specific barcode tag in a file. We repeat for all barcodes or all forward and reverse barcode combinations.