Metatranscriptomic screening for genes of interest

Helena Pound[1], Steven Wilhelm[2]

[1]University of Tennessee, Knoxville, [2]The University of Tennessee, Knoxville

Nov 11, 2019

1 Works for me

dx.doi.org/10.17504/protocols.io.7vyhn7w

The Aquatic Microbial Ecology Research Group - AMERG (The Buchan, Zinser and Wilhelm labs)

Great Lakes Center for Fresh Waters and Human Health

Helena Pound
University of Tennessee, Knoxville

ABSTRACT

Find genes of interest in assembled metatranscriptomic library using blast.

EXTERNAL LINK

http://wilhelmlab.utk.edu/

GUIDELINES

You will need:

1. Metatranscriptomic assembly (fasta file) - recommended that you pool samples from individual experiments and assemble them together

(Called "assembly.fasta" in this protocol)

2. List of genes of interest (fasta file) - Protein sequences of genes; commonly genes of similar function with representatives from many species

(Called "reference.fasta" in this protocol)

MATERIALS TEXT

- CLC Workbench
- command line blast
- command line diamond
- python

Establishing Candidates

1 Make a database from the list of genes of interest (referred to as the reference.fasta) in command line window.

Protein sequences are required for phylogenetic placement in later steps.

1.1

>makeblastdb -dbtype prot -in reference.fasta -parse_seqids

>makeblastdb -dbtype prot -in reference.fasta -parse_seqids

2 Blast assembly against reference database you just created

- e-value can be changed to your preference
- final.txt will produce list of contigs from assembly that had blast hits, to be referred to as "candidates"

2.1

>blastx -query assembly.fasta -db reference.fasta -out final.txt -outfmt "6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qframe" -e-value 1E-10

>blastx -query assembly.fasta -db reference.fasta -out final.txt -outfmt "6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qframe" -e-value 1E-10

3 Extract only the aligned portion of candidate contigs.

- Sometimes, contigs from assembly may be longer than a single gene. To accurately estimate transcript expression, we want to trim candidates to the exact length of the reference gene, by trimming it according to the start and finish locations of the blast alignment. Custom python script performs this trimming. Script relies on column order, so double-check that your .txt file contains all columns listed in 2.1. If a candidate contig hits to more than one reference sequence, the lowest start point and the highest end point from all hits will be used to capture the entire candidate sequence.

3.1 Custom python script found at https://github.com/Wilhelmlab/general-scripts/blob/master/Pound2019_Extract_aligned.py

4 **Convert** .txt file generated from step 3.1 to a .csv file

-Open .txt file in Microsoft Excel and save as .csv

5 **Import** .csv file into CLC Workbench as "sequences in .csv format"

6 Sort sequences by length and delete sequences smaller than 150 nucleic acids or personal preference

This length cutoff can be changed based on personal preference.

7 **Export** sequence list as .fasta file and .csv file (named trimmed.fasta/csv or something similar).

8 Perform Diamond blastx against NCBI non-redundant database.

-This step is a quality control mechanism to identify sequences that may not be of viral origin.
- Diamond is used to for expediting the process (faster than traditional blast)

8.1 **Download** non-redundant database from NCBI in .fasta format

-rename nonredundant.fasta

8.2 **Make diamond database**

>diamond makedb --in non-redundant.fasta -d non-redundant

8.3 **Blast**

>diamond blastx -d nonredundant -q trimmed.fasta -o candidates.txt -f 6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qframe stitle -k 1

>diamond blastx -d nonredundant -q trimmed.fasta -o candidates.txt -f 6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qframe stitle -k 1

9 **Open** candidates.txt file and trimmed.csv file in Excel. Copy sequences from trimmed.csv to a new column in candidates.txt file in excel (make sure sequence names match up).

10 **Delete** sequences that are not viral in origin (refer to diamond blast hit in the candidates.txt file).

Be sure to delete the entire row.

11 **Create** a new .csv file with viral sequences and query frames from candidates.txt file

- 3 columns: sequence name, sequence, query frame

12 **Sort** by query frame. Create a new .csv file for each query frame containing respective sequences

Ex. CandidatesQF+1.csv

13 **Import** each query frame.csv file into CLC workbench as separate sequence lists

13.1 **Create** new sequence list and combine all query frame sequence lists

- rename "FinalCandidatesNucleicAcid"

14 Translate individual query frame sequence list to protein based on the query frame.

14.1 **Create** new sequence list and combine all protein sequences

- rename "FinalCandidatesProtein"

15 **Export** FinalCandidatesProtein sequence list as a .fasta file