Laboratory Pipelines

Cody Glickman

cody.glickman@ucdenver.edu

2018

Contents

BLAST Usage

Make BLAST Database
-dbtype [prot/nucl] protein or nuclotide, use

```
-parse_seqids
```

to enable retrieval of sequences by sequence identifiers.

```
makeblastdb -in [input_seq] -dbtype prot -out Combined_VF -title "Combined_VF"

blastp -db [Path_to_Database] -out results.txt -outfmt 6 -query [input_seqs]
-max_target_seqs 2 -perc_identity 95
```

Output format Useful outfmt link on biostars. Formats of interest include 5 XML Blast output, 6 tabular output, 10 comma-seperated values
Standard columns in outfmt 6:

```
1.  qseqid  query (e.g., gene) sequence id
2.  sseqid  subject (e.g., reference genome) sequence id
3.  pident  percentage of identical matches
4.  length  alignment length
5.  mismatch  number of mismatches
6.  gapopen  number of gap openings
7.  qstart  start of alignment in query
8.  qend  end of alignment in query
9.  sstart  start of alignment in subject
10.  send  end of alignment in subject
11.  evalue  expect value
12.  bitscore  bit score

Additional columns can be specific by -outfmt "6 std qlen"
```

Custom Output Format
-outfmt ”6 qseqid sseqid pident length mismatch evalue bitscore stitle”
Columns in Tab-seperated format
qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore

Microbiome Analysis

1 QIIME2 Microbiome

Cite vHMM Earth Virome Project.

```
## Load required libraries
module load qiime2
source activate qiime2-2017.12

## Create demux sequences
qiime tools import --type 'SampleData[PairedEndSequencesWithQuality]'
--input-path manifest_file.csv --output-path [paired-end-demux]
--source-format PairedEndFastqManifestPhred33

## Perform Dada2 Processing
bsub qiime dada2 denoise-paired --i-demultiplexed-seqs [paired-end-demux]
--p-trunc-len-f 280 --p-trunc-len-r 279 --p-max-ee Inf --p-chimera-method
pooled --p-n-threads 0 --o-representative-sequences [rep-seqs]
--o-table [table]

## Summarize Table
bsub qiime feature-table summarize --i-table table.qza --o-visualization table.qzv

## Assign Taxonomy
bsub qiime feature-classifier classify-sklearn --i-classifier [Database File]
--i-reads [rep-seqs] --o-classification [taxonomy]

## Visualize Taxonomy
qiime metadata tabulate --m-input-file ./Qiime_demo/taxonomy.qza --o-visualization [taxo
```

Gene Prediction Methodolgy

Utilize Prodigal to predict genes in prokaryotes.

1 Genome Gene Prediction

```
prodigal -i [input_seqs] -o my_genes -a my_proteins.faa
```

2 Metagenomic Gene Prediction

```
prodigal -i [input_seqs] -o my_genes -a my_proteins.faa -p meta
```

```
## Hello World
print(x)
```

BASHING

PHASTER Summary to Table Parser Input File is summary.txt from Phaster and the output table is output.txt
The following workflow is wrapped into table_parsing_workflow.sh the folder with the summary file shoud be in the same directory as the workflow and python script.

```
sed -i '' -n '/REGION/,$p' $table

sed -i '' -e 's/-//g' $table

sed -i '' -e 's/REGION_POSITION/ASSEMBLY_CONTIG_NUMBER
CONTIG_LENGTH      CONTIG_DEPTH/g' $table

awk -v OFS="\t" '$1=$1' $table > output.txt

./Table_Parsing.py output.txt $table
```