Example steps for running metaSNV analysis

Author: Enrique Doster (enriquedoster@tamu.edu)

Last modified: 2023.09.14

Requirements:

- AMR++ installation
- Reduced database of MEGARes to serve as "representative" database. As of 9/14/2023, we are using a database made from clustering by 90% sequence identity, "meg_90id_rep_seqs.fasta".
- metaSNV installation (conda)
- metaSNV_organizer.py

Overview of steps

1. Run "align" pipeline workflow with AMR++ using the "representative" database
2. Use metaSNV to analyze the bam files and identify SNVs for all samples
3. Run custom script to parse metaSNV results and create a count table.

Step 1 - Run AMR++

Run the "align" pipeline from AMR++ on your target reads. Use the "--amr" flag to use the "representative" MEGARes database. Here's an example of what your command might look like:

```
nextflow run main_AMR++.nf --reads "/path/to/nonhost/reads/*R{1,2}.fastq.gz" -
profile local --pipeline align --output AMR++_rep_db_alignment --amr
"meg_90id_rep_seqs.fasta"
```

Step 2 - Run metaSNV

For this step you'll need to have metaSNV installed. Conda is likely the easiest installation option. We'll be using the bam alignment files created in step 1.

```
# Make a text file with the path to each of your bam alignment files. Here's what
I used:
ls -d /path/to/AMR++_rep_db_alignment/Alignment/BAM_files/Deduped/* >
metaSNV_sample_names.txt

# Call SNVs using sample name file and representative database
metaSNV.py Output_metaSNV_results/ metaSNV_sample_names.txt
meg_90id_rep_seqs.fasta
```

Step 3 - Parse metaSNV results

Download the script "metaSNV_organizer.py" from our github page and place it in your working directory.

In the step above, we made metaSNV output into this folder "Output_metaSNV_results/" and inside we'll find another directory snpCaller with the results split into various files, depending on how many threads you used (silly). We can run the following command to parse the results and convert it into a more user-friendly count table format.

```
python metaSNV_organizer.py -i "called_SNPs.best_split_*" -s
"metaSNV_sample_names.txt" -o "metaSNV_AMR_analytic_matrix.csv"
```

You'll now have the "metaSNV_AMR_analytic_matrix.csv" file which contains all of the results for each sample. You'll need to make an annotation file using the first column of the count table, but this shouldn't be too difficult if you split the names using the "|".