Data Preparation Documentation

As the Staphopia API does not provide information regarding antimicrobial resistance (AMR) genetics of *Staphylococcus Aureus (S.Aureus)* so the Webscape has decided to prepare AMR genes. This documentation dedicates to how to install necessary tools and use them.

Source to the documentation of Staphopia API: https://staphopia.emory.edu/docs/api/

1. Install AMRFinderPlus

AMRFinderPlus is a tool to find AMR genes in the bacterial. This tool can generate results so the front-end side can visualize the AMR genes for the website.

Note: this tool is only available on Linux and not on Window and installation via putting command lines in the terminal.

Step 1: Install Miniconda for Linux by using these commands in the terminal:

```
curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
bash ./Miniconda3-latest-Linux-x86_64.sh
```

Step 2: Install AMRFinder with bioconda

Make sure the conda environment you just created is activated:

```
source ~/miniconda3/bin/activate
```

Install AMRFinder and all of the prerequisites:

```
conda install -y -c bioconda ncbi-amrfinderplus
```

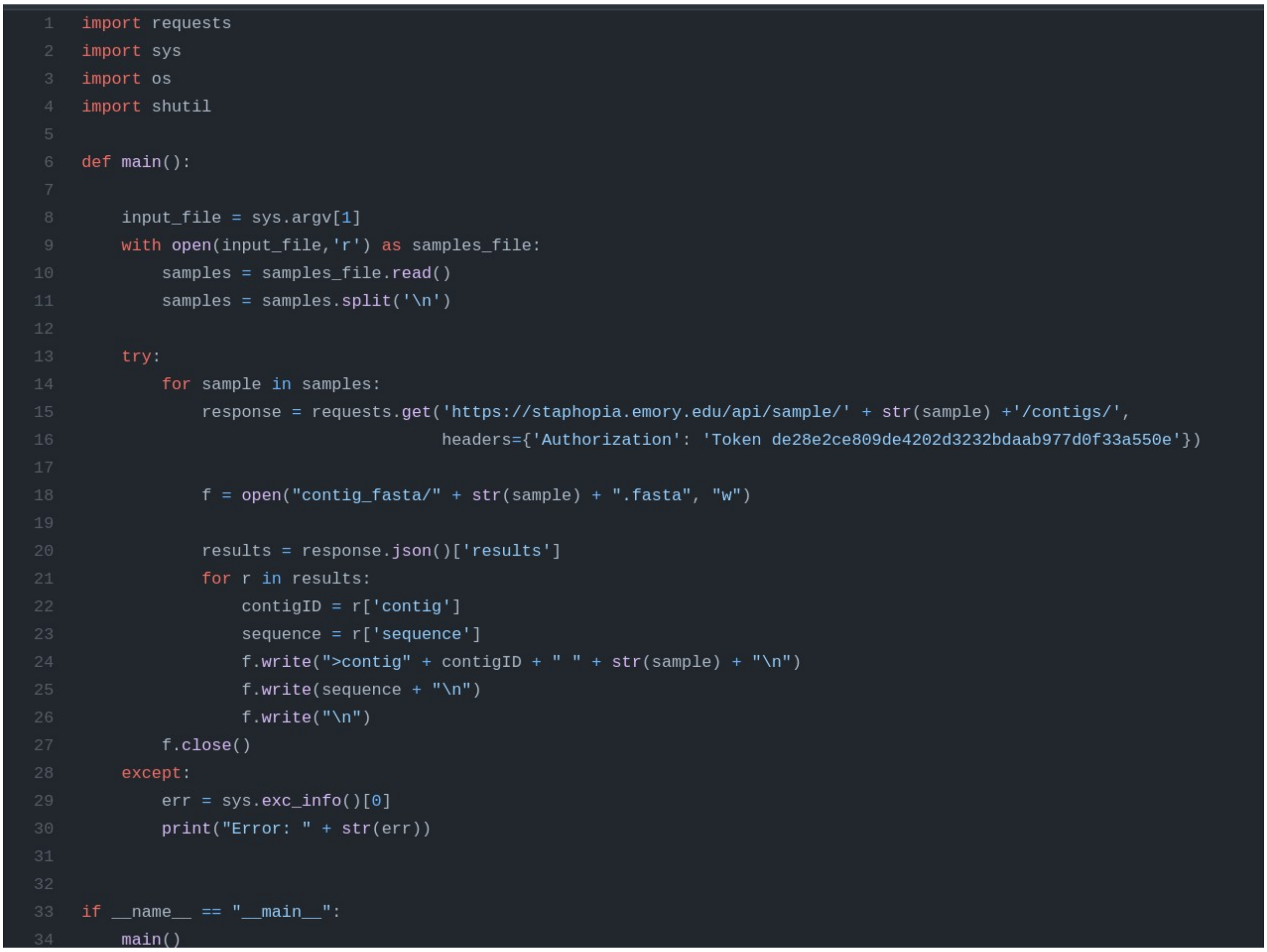
Step 3: Update the tool:

```
conda update -y -c bioconda ncbi-amrfinderplus
```

2. Use AMRFinderPlus

This section show how to use the tool correctly.

Step 1: Create a python file called fasta_generator.py with the code below:



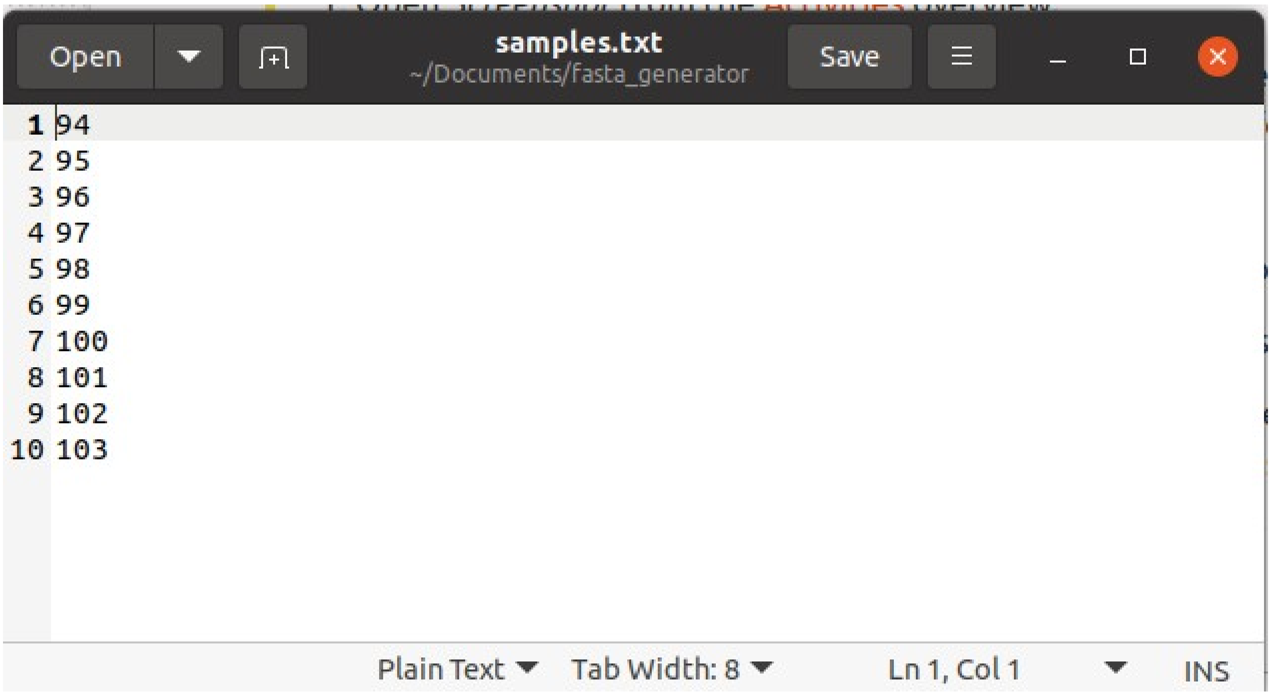
```python
import requests
import sys
import os
import shutil

def main():

    input_file = sys.argv[1]
    with open(input_file,'r') as samples_file:
        samples = samples_file.read()
        samples = samples.split('\n')

    try:
        for sample in samples:
            response = requests.get('https://staphopia.emory.edu/api/sample/' + str(sample) +'/contigs/',
                                    headers={'Authorization': 'Token de28e2ce809de4202d3232bdaab977d0f33a550e'})

            f = open("contig_fasta/" + str(sample) + ".fasta", "w")

            results = response.json()['results']
            for r in results:
                contigID = r['contig']
                sequence = r['sequence']
                f.write(">contig" + contigID + " " + str(sample) + "\n")
                f.write(sequence + "\n")
                f.write("\n")
        f.close()
    except:
        err = sys.exc_info()[0]
        print("Error: " + str(err))


if __name__ == "__main__":
    main()
```

Alternative: There is a python file called fasta_generatot.py inside of ‘datasets’ folder, which is located in ‘code’ folder.

The python file create a file in fasta format (a file only contains genetics sequence) by getting the Assembled Contigs of each *S.Aureus* sample from Staphopia API.

Step 2: Create a samples.txt and each line put the wanted sample ID of *S.Aureus* available in the Staphopia API.

Example of Sample.txt :



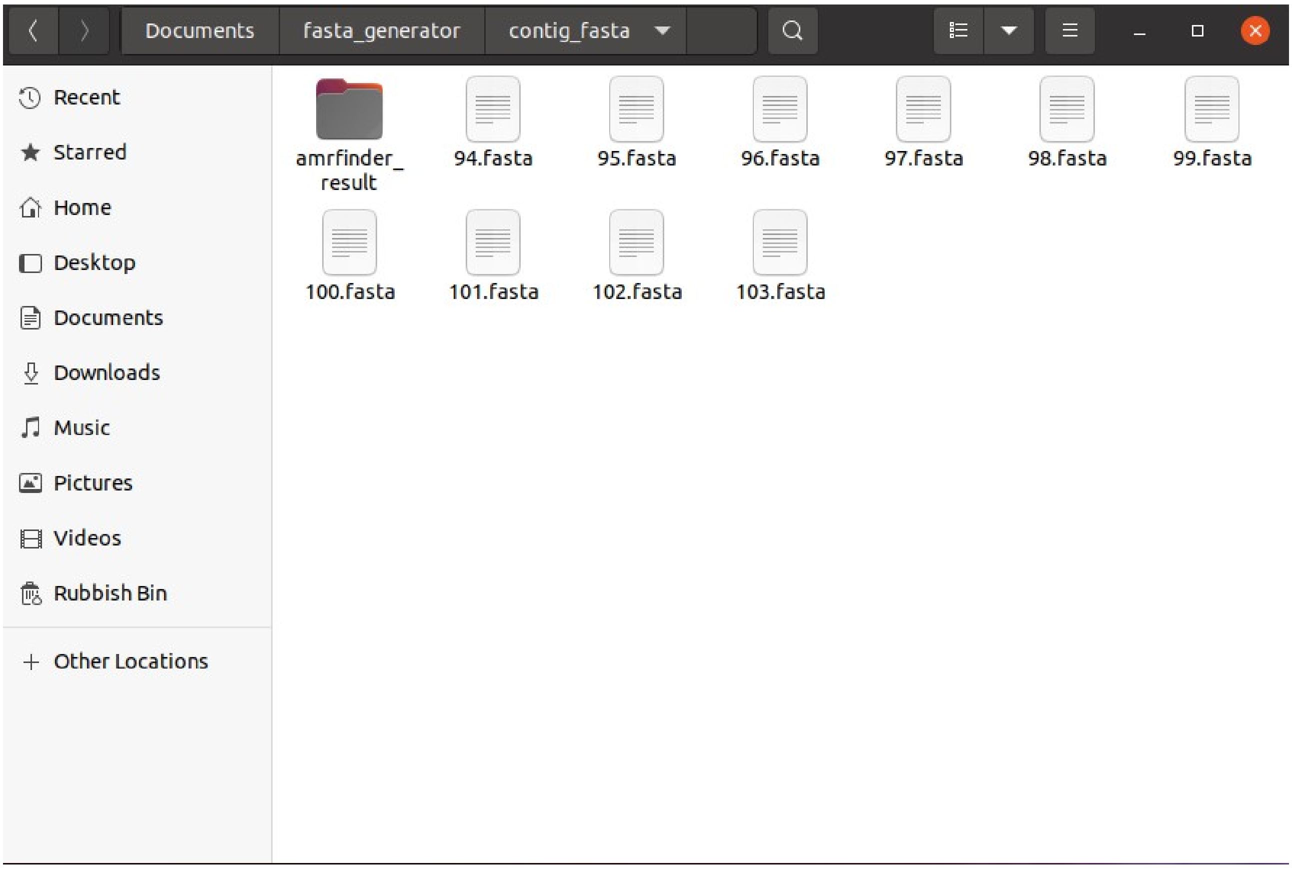
```
94
95
96
97
98
99
100
101
102
103
```

Step 3: Using the python file to generate fasta file for each sample inserted in samples.txt by putting this command line:

```
python fasta_generator.py samples.txt
```

All the results is stored in "contig_fasta" folder.

Example of input:



Each fasta is name to each sample ID.

Step 4: Go to “contig_fasta” folder and use AMRFinderPlus for each sample:

Example for file 95.fasta:

```
amrfinder -n 95.fasta -O Staphylococcus_aureus -o amrfinderresult_95.csv
```

Explanation for the commandline:

‘amrfinder’ : this is indicating that the tool is used for generating the AMR result is AMRFinderPlus

‘-n’ : this is option command indicating the input fasta file is Nucleotide Fasta file

‘95.fasta’ : the input fasta file

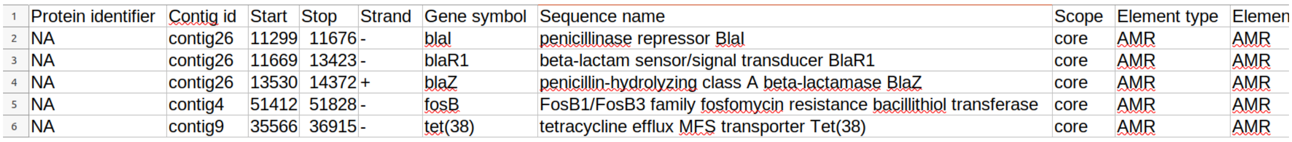
‘-O’ : Taxon used for screening known resistance causing point mutations and blacklisting of common, non-informative genes

‘ Staphylococcus_aureus ‘: indicating that only find AMR genes related to *S.Aureus*

‘-o’: output option

‘amrfinderresult_95.csv’ : name of the output file in csv format.

Example of an output file ‘amrfinderresult_95.csv’:



1	Protein identifier	Contig id	Start	Stop	Strand	Gene symbol	Sequence name	Scope	Element type	Elemen
2	NA	contig26	11299	11676	-	blaI	penicillinase repressor BlaI	core	AMR	AMR
3	NA	contig26	11669	13423	-	blaR1	beta-lactam sensor/signal transducer BlaR1	core	AMR	AMR
4	NA	contig26	13530	14372	+	blaZ	penicillin-hydrolyzing class A beta-lactamase BlaZ	core	AMR	AMR
5	NA	contig4	51412	51828	-	fosB	FosB1/FosB3 family fosfomycin resistance bacillithiol transferase	core	AMR	AMR
6	NA	contig9	35566	36915	-	tet(38)	tetracycline efflux MFS transporter Tet(38)	core	AMR	AMR

Note: This is not the whole result of amrfinderresult_95.csv as there are much more columns in the csv file.

Output fields in the output format:

Protein Identifier - This is from the FASTA defline for the protein or DNA sequence.

- Contig id - Contig name.
- Start - 1-based coordinate of first nucleotide coding for protein in DNA sequence on contig.
- Stop - 1-based coordinate of last nucleotide coding for protein in DNA sequence on contig. Note that for protein hits (where the Method is HMM or ends in P) the coordinates are taken from the GFF, which means that for circular contigs when the protein spans the contig break the stop coordinate may be larger than the contig size
- Gene symbol - Gene or gene-family symbol for protein or nucleotide hit. For point mutations it is a combination of the gene symbol and the SNP definition separated by "_"
- Sequence name - Full-text name for the protein, RNA, or point mutation.

- Scope - The AMRFinderPlus database is split into 'core' AMR proteins that are expected to have an effect on resistance and 'plus' proteins of interest added with less stringent inclusion criteria. These may or may not be expected to have an effect on phenotype.
- Element type - AMRFinder+ genes are placed into functional categories based on predominant function AMR, STRESS, or VIRULENCE.
- Element subtype - Further elaboration of functional category into (ANTIGEN, BIOCIDE, HEAT, METAL, PORIN) if more specific category is available, otherwise he element is repeated.
- Class - For AMR genes this is the class of drugs that this gene is known to contribute to resistance of.
- Subclass - If more specificity about drugs within the drug class is known it is elaborated here.
- Method - Type of hit found by AMRFinder. A suffix of 'P' or 'X' is appended to "Methods" that could be found by protein or nucleotide.
    - ALLELE - 100% sequence match over 100% of length to a protein named at the allele level in the AMRFinderPlus database.
    - EXACT - 100% sequence match over 100% of length to a protein in the database that is not a named allele.
    - BLAST - BLAST alignment is > 90% of length and > 90% identity to a protein in the AMRFinderPlus database.
    - PARTIAL - BLAST alignment is > 50% of length, but < 90% of length and > 90% identity to the reference, and does not end at a contig boundary.
    - PARTIAL_CONTIG_END - BLAST alignment is > 50% of length, but < 90% of length and > 90% identity to the reference, and the break occurrs at a contig boundary indicating that this gene is more likely to have been split by an assembly issue.
    - HMM - HMM was hit above the cutoff, but there was not a BLAST hit that met standards for BLAST or PARTIAL. This does not have a suffix because only protein sequences are searched by HMM.
    - INTERNAL_STOP - Translated blast reveals a stop codon that occurred before the end of the protein. This can only be assessed if the `-n <nucleotide_fasta>` option is used.
    - POINT - Point mutation identified by blast.
- Target length - The length of the query protein or gene. The length will be in amino-acids if the reference sequence is a protein, but nucleotide if the reference sequence is nucleotide.
- Reference sequence length - The length of the Reference protein or nucleotide in the database (NA if HMM-only hit).
- % Coverage of reference sequence - % of reference covered by blast hit (NA if HMM-only hit).
- % Identity to reference sequence - % amino-acid identity to reference protein or nucleotide identity for nucleotide reference. (NA if HMM-only hit).
- Alignment length - Length of BLAST alignment in amino-acids or nucleotides if nucleotide reference (NA if HMM-only hit).
- Accession of closest protein - RefSeq accession for reference hit by BLAST (NA if HMM-only hit). Note that only one reference will be chosen if the blast hit is equidistant from

multiple references. For point mutations the reference is the sensitive "wild-type" allele, and the element symbol describes the specific mutation.

- Name of closest protein - Full name assigned to the closest reference hit (NA if HMM-only hit).
- HMM id - Accession for the HMM, NA if none.
- HMM description - The family name associated with the HMM, NA if none.

Note: full documetation of AMRFinderPlus can be found in: https://github.com/ncbi/amr/wiki/Running-AMRFinderPlus

Final Note: This documentation only documents the process of data preparation process of Capstone Project Phase 1 and there will be more added into this documentation.