Manual

This script automates the process of downloading reference genomes, searching for and fetching assembly data, running quality assessment (QUAST), and genome annotation (Prokka) tools, and organizing the output for a specified organism. It starts by loading configuration settings from a JSON file, which includes details such as the user's email, the organism, location, URLs for the reference genome and GFF file, and assembly ID. The script downloads the reference genome and GFF file from provided URLs if they don't already exist locally.

Next, it searches for genome assemblies related to the specified organism and location using the Entrez API, fetching their summaries to obtain their accession numbers. These accessions are then separated into GenBank (GCA) and RefSeq (GCF) categories, identifying and removing common IDs between them. The script attempts to download these genome assemblies using the ncbi-genome-download tool. If it encounters any errors, it retries the download with modified accessions.

Once the assemblies are downloaded, the script decompresses and renames the files for consistency. For quality assessment, it runs QUAST on the downloaded genomes, comparing them against the reference genome and the GFF file. Additionally, it runs Prokka for genome annotation. The results of these analyses are organized into designated output directories, making it easier for users to access and interpret the data. This script simplifies the workflow for genomic data processing, ensuring reproducibility and efficiency in handling large-scale genomic datasets.

Requirement :-

The script requires several tools and libraries to function correctly. Below is a list of these tools along with their purposes:

Python Libraries :-

subprocess: Used to run shell commands from within the Python script.

json: Used to load and parse configuration files in JSON format.

os: Used for interacting with the operating system, such as creating directories and renaming files.

Bio.Entrez: Part of the Biopython library, used to interact with the NCBI Entrez database for searching and fetching genomic data.

External Tools :-

wget: Used to download files from the internet, such as reference genomes and GFF files.

ncbi-genome-download: A command-line tool used to download genome assemblies from NCBI. This tool must be installed separately.

gunzip: Used to decompress .gz files.

quast.py: A tool for evaluating genome assemblies. QUAST provides various metrics and visualizations to assess the quality of assemblies.

Prokka: A software tool used for the rapid annotation of prokaryotic genomes

Maker: It is a popular genome annotation pipeline used primarily for annotating eukaryotic genomes

Tool	version
Biopython	1.83
Ncbi-genome-download	0.3.3
QUAST	5.2.0
Prokka	1.14.6
Maker	2.31.9

Installation :-

Biopython: pip3 install biopython

Ncbi-genome download: pip3 install ncbi-genome-download

Quast: pip3 install quast

###install prokka in a different environment if its conflicting with any tool but make sure to change the env name in the “def run_prokka_with_env” function

prokka: conda install -c bioconda prokka ()

Maker: conda install bioconda::maker

Running command :-

Python3 annotation_script.py

Input file :- config.yaml (configuration file)

Configuration file (contains):-

This is the query for the organism Mycobacterium tuberculosis in NCBI databases

organism_query: "Mycobacterium[Organism]"

Your email address for notifications

email: "your@email.com"

organism: "Mycobacterium tuberculosis" #goes into query to fetch assemblies

The location where the organism is found, in this case, India

location: "India"

URL to the reference genome file for Mycobacterium tuberculosis

reference_genome_url: "https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/195/955/GCF_000195955.2_ASM19595v2/GCF_000195955.2_ASM19595v2_genomic.fna.gz"

URL to the GFF file for Mycobacterium tuberculosis

gff_url: "https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/195/955/GCF_000195955.2_ASM19595v2/GCF_000195955.2_ASM19595v2_genomic.gff.gz"

#NCBI genome download parameter

The NCBI taxonomic groups to download (default: all). A comma-separated list of taxonomic groups is also possible.

#For example: "bacteria,viral"Choose from: ['all', 'archaea','bacteria', 'fungi', 'invertebrate', 'metagenomes', 'plant', 'protozoa', 'vertebrate_mammalian', 'vertebrate_other', 'viral']

group: "bacteria" # deafult: all

organism_type: "PK" #PK for prokaryotes, EK for eukaryotes

#ouput folder name

output_folder: "Mycobacterium"

Commands inside the script :-

1. wget Command

Command: wget -P {output_dir1} {url}

Purpose: Downloads the reference genome (or other files) from the provided URL into the specified output directory (output_dir1).

2. os.makedirs Command

Command: os.makedirs(output_dir1, exist_ok=True)

Purpose: Creates the output_dir1 directory (e.g., genomes). The exist_ok=True parameter prevents an error if the directory already exists.

Command: os.makedirs(output_dir2, exist_ok=True)

Purpose: Creates the output_dir2 directory (e.g., assembly).

Command: os.makedirs(output_dir3, exist_ok=True)

Purpose: Creates the output_dir3 directory (e.g., quast_output).

3. ncbi-genome-download Command

Command: ncbi-genome-download -l all -F fasta -o {output_dir2} bacteria --assembly-accessions {gcf_accession}

Purpose: Downloads the genome sequences in FASTA format for the specified GCF accession from the NCBI database.

Command: ncbi-genome-download -s genbank -l all -F fasta -o {output_dir2} bacteria --assembly-accessions {gca_accession}

Purpose: Downloads the genome sequences in FASTA format for the specified GCA accession from the GenBank database.

Command: ncbi-genome-download -s genbank -l all -F fasta -o {output_dir2} bacteria --assembly-accessions {','.join(gca_accessions)}

Purpose: Downloads genome sequences for multiple GCA accessions at once.

4. find and mv Commands

Command: find {output_dir2}/*/* -type f -name '*.fna.gz' -exec mv {} {output_dir2} \;;

Purpose: Moves all .fna.gz files from their current locations within subdirectories to the main output_dir2 directory.

5. gunzip Command

Command: gunzip {output_dir2}/*.gz {output_dir1}/*.gz && find {output_dir2} -type d -empty -delete

Purpose: Decompresses all .gz files in output_dir1 and output_dir2 and removes any empty directories afterward.

6. quast.py Command

Command: quast.py {assembly_path}/{accession}*.fna -r {reference_genome} -g{gff_genome} -o quast_output/{accession}

Purpose: Runs QUAST (Quality Assessment Tool) to evaluate the quality of the downloaded genome assemblies against the reference genome and gene annotations. The results are saved in the quast_output/{accession} directory.

7. prokka Command

Command: prokka assembly/{asm_id}.fna -outdir prokka_output/{asm_id} --prefix {asm_id}

Purpose: Runs Prokka for genome annotation on the specified assembly, saving the results in the prokka_output/{asm_id} directory with the given prefix.

8. os.rename Command

Command: os.rename(old_path, new_path)

Purpose: Renames files in the output_dir2 directory based on a specific naming convention (e.g., joining parts of the filename and adding .fna as the extension).