ProkGenomics

Laura Perlaza-Jiménez (Ph.D.)

Compiled: May 02, 2024

Contents

Chapter 1

START HERE

I don't know where to start

I know command-line and SSH

ProkGenomics is already installed

Chapter 2

Introduction

This **nextflow** pipeline allows you to run several programs using a one-line command. It wraps programs to assemble, annotate, taxonomically identify, genotypic characterize and perform simple comparative genomics on prokaryotic sequencing data. This pipeline is self-contained and is NOT meant to be run in modules (not independent parts of it). This with the intention of simplifying the user interaction and the user knowledge about the bioinformatics behind this processes. However, making yourself familiar with the programs behind the scenes helps you to make informed decisions. Please go to resources to read more about what each program is doing.

2.1 Overview of the pipeline

- **Assembly, Annotation and Classification**
  - Quality control of reads (short, long, or both)
  - Cleaning reads
  - De-novo Assembly (short, long or hybrid)
  - Quality Control of the assembly and detection of contamination
  - Identification of the species in the sample
  - Identification of plasmids, phages and prophages
  - Genes Annotation
- **Comparative genomics (when genome reference provided)**
  - Coverage of reference genome
  - Identification of core genome
  - Identification of Single Nucleotide Variants (SNV)
  - Identification of rearrangements and larger deletions
- **Characterization of genes of interest (when genes sequences provided)**
  - Alignments to genes of interest
  - Summary table to identy presence, absence and truncated genes

  - Generation peptides sequences from the denovo assembly

Chapter 3

Getting Started

3.1 Make sure you have the required dependencies

Before you get started with the pipeline, make sure you have installed all necessary dependencies for the program to run. The ProkGenomics pipeline allows you to run several tools without the need of installing each tool by yourself. It uses **conda**, **docker**, **singularity** or **Apptainer** to create environments or containers with the necessary dependencies and tools. At the moment, the option of running this pipeline with conda is susceptible to break in some conda versions or conda with customised channels in its configuration. Therefore the ***best option for now is to run it with singularity***.

To install dependencies the easiest manner is to use **conda**. If you are using a **cluster** there are high chances that **conda** is a module already installed. Please call the module. Type the following command and use `tab` after the word `conda` to see if you have different versions available. Versions successfully tested are: `conda 4.14.0`, `conda 24.1.2`.

```
# if you are in a cluster
module add anaconda
# or
module add miniconda
```

If you have **conda** installed in your system, either in your **computer** or your **server**, you can type `conda --version` to verify you have the right version to work with.

3.1.1 Create a conda environment to run ProkGenomics

Create a **conda** environment with the required dependencies. Use the following commands. Versions successfully tested are: `nextflow 21.10.6`, `singularity-ce version 3.9.9-focal`, `singularity version 3.8.6`

```
# if you can create conda env, this could work for a cluster, server or your own compu
conda create -n ProkGenomics_ENV
conda activate ProkGenomics_ENV
conda install -c bioconda nextflow=21.10.6
conda install conda-forge::singularity
conda install apptainer
```

Use `conda activate ProkGenomics_ENV` every time you want to use nextflow, and the singularity profile

Alternatively, you can install **nextflow** and **singularity** without using **conda**.

- Install nextflow using the nextflow installation instructions.
- Install **singularity** following these instructions, make sure you are following the right instructions for your system (Linux, Windows or Mac)

3.2 Download the ProkGenomics

Run the following command in your terminal

```
# when private
git clone git@github.com:Grinter-Lab/ProkGenomics.git

# when public
git clone https://github.com/Grinter-Lab/ProkGenomics.git
```

If you have issues consult the troubleshoot section for help

A successful download of the repository should look like this:

```
Cloning into 'ProkGenomics'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

3.2.1 Execute ProkGenomics from any path

To be able to run the program from any location without using the complete path, run the following commands

1. Find out the path where your program is located

```
# move to the program folder
cd ProkGenomics/

# print your working directory
pwd

# This will print your location something like: /path/to/dir/program/
```

2. add this path to the default executable folder.

```
# IF you are NOT using a conda environment
# export the path of the folder into your PATH if your install singularity and nextflow in bin
export PATH="$PATH:/path/to/dir/program/"

# IF you are working with a conda env

#find your conda env path and add the content of
conda env list

cd /path/to/conda/env/bin
for name in /path/to/dir/program/ProkGenomics/*;
do
  ln -s name
done
```

Chapter 4

Set up your project

Create a working directory for your project

```
mkdir Project1
cd Project1
```

Create a folder for your raw data with in that working directory

```
mkdir rawdata
cd rawdata
# copy your rawdata for that project inside this folder
```

Please run one ProkGenomics pipeline per working directory. You can run multiple ProkGenomics at a time if the working directories are different. Several runs in the same working directory cause conflict with the *.nextflow.logs* and your pipeline will fail.

4.1 How to transfer raw data to cluster/server

Open a tab in your terminal from your local computer

```
# from tab in your local computer
# scp <location in the server, notice structue as serve:path> <location in your computer where yo
# notice that the wild card allows you to move all files ending in fq.gz. if your files have a di

scp *fq.gz <username>@<cluster_name>:/srv/home/username/folder/
```

Chapter 5

Run Pipeline

Remember to put the program in your $PATH if you haven't done it. This step has to be done every time you start a new terminal session. If you want to make this change permanent you could modify your bash profile (don't play around with it if you don't feel confident about it)

If you added the program to your `$PATH` successfully you should be able to run

```
#remember to activate the conda env if you are using one
conda activate ProkGenomics_ENV

#run the pipeline
ProkGenomics
```

If ProkGenomics starts correctly, you will see something like:

If you don't see a version of that go to troubleshoot to look for possible solutions

5.1 Simple run

```
#remember to activate the conda env if you are using one, and if you haven't activated it yet
conda activate ProkGenomics_ENV

#run the pipeline
ProkGenomics -profile singularity --sample_name '1-77321'
```

5.2 Include comparative genomcis steps

If you have a reference genome put the path using the `--reference` flag. This will activate all the comparative genomics steps. This file can be formatted as FASTA or GENBANK. If you provide a GENBANK file your Single Nucleotide

Variant file will be annotated (tell you what gene has the mutations). The use of GENBANK files work only when the file has also the fasta sequence at the end of it. In the NCBI download the `Genbank (full)` option.

```
#remember to activate the conda env if you are using one, and if you haven't activated
conda activate ProkGenomics_ENV

#run the pipeline
ProkGenomics -profile singularity --sample_name '1-77321' --reference reference.fasta
```

5.3 Include characterization of genes of interest

To activate the genes characterization steps use the flag `--genes_interest`. Use the path to a folder that contains all genes of interest. The correct formatting is ONE gene per file in FASTA format. This folder can have any name, just make sure that it doesn't contain spaces in the name. Do not store additional files in this folder that are not genes of interest

```
#remember to activate the conda env if you are using one, and if you haven't activated
conda activate ProkGenomics_ENV

#run the pipeline
ProkGenomics -profile singularity --sample_name '1-77321' --genes_interest GenesDB
```

5.4 Run all steps

```
#remember to activate the conda env if you are using one, and if you haven't activated
conda activate ProkGenomics_ENV

#run the pipeline
ProkGenomics -profile singularity --sample_name '1-77321'--reference reference.fasta -
```

Parameters you can use:

Command	Description
`--sample_path` *./rawdata/*	The default path for the reads is the folder rawdata in the working directory (please follow the instructions for setting up the working folder). if you have your reads somewhere else you should set this parameter to that path.

Command	Description
`--sample_name` *1-77321*	The sample name is the prefix of your samples files. it doesn't have a default because I don't know your sample names. Please don't use sample names with spaces in them. Best approach is to use the name of the file as it comes from the sequencing facility
`--assembly_type` *short*	This parameter can be short long or hybrid. The default is 'short'. if you have short reads you don't have to specify this parameter. If you pick the argument long or hybrid the longreads parameter should be specify. For hybrid make sure to give a path for short and long reads.
`--longreads` *./rawdata/longreads/*	Path to the long reads.
`--threads` *16*	Number of threats to use. More threats faster your processing. Make sure you know what is available for you.
`--outdir` *1-77321*	The results will be in a folder in the working directory with the same sample name and _results ex. 1-77321_results.
`--reference` *ReferenceGenome.fasta or ReferenceGenome.gbk*	If you have a reference genome put the path here. This will activate all the comparative genomics steps. This file can be formatted as FASTA or GENBANK. If you provide a GENBANK file your Single Nucleotide Variant file will be annotated (tell you what gene has the mutations). The use of GENBANK files work only when the file has also the fasta sequence at the end of it. In the NCBI download the `Genbank (full)` option
`--adapter_file` *TruSeq3-PE.fa*	To trim your short reads you need to specify what adaptors where used when sequencing. Arguments are `TruSeq2-SE.fa`, `TruSeq2-PE.fa`, `TruSeq3-PE.fa`. The default is `TruSeq3-PE.fa`.

Command	Description
`--genes_interest` *GenesDB/*	Path to a folder that contains all genes of interest. The correct formatting is ONE gene per file in FASTA format. This folder can have any name, just make sure that it doesn't contain spaces in the name. Do not store additional files in this folder that are not genes of interest
`--assembly` *genome_assembly.fasta*	If you already have an assembly you can set this parameter and the pipeline will skip all the steps of assembly
`--run_classification` *FALSE*	The taxonomical classifications is based on a database of ~90G. This step will take several hours to complete downloading the database. The default argument is `TRUE`. When `--run_classification FALSE` is not set, ProkGenomics will download the database for taxonomical classification and stored in its base-directory. All runs after that will search in this location for the database avoiding the lengthy download again. The download of the database depends on the available disk space, this step will be skipped if there is not enough disk space
`--db_gtdb_path` */path/gtdb/existing/database*	If you already download the gtdbtk database, indicate the path with this option. This parameter is only necessary if you have the gtdbtk database in a different location from where ProkGenomics search. Default is ProkGenomics/DB/db_gtdb/, so if you have your db_gtdb in that location, you don't have to specify any path
`--keep_intermediate_files`	Keep all intermediate files, including mapping files
`--cleanup FALSE`	The pipeline cleans-up by defaults. If you are debugging you can set the cleanup to FALSE and keep your work folders. This will allow you to use `-resume` and troubleshoot errors

Chapter 6

Understand Outputs

This pipeline produce a folder per program run and several main outputs. The main results are described in the final report named `<sample_name>_ProkGenomics_report.html` (click hyperlink to see)

6.1 How to explore files

6.2 Additional outputs description

Folder structure

Chapter 7

Write Methods

1. Input short reads paired ends from whole genome bacteria sequencing
2. Fastqc is used to determined quality control of rawdata
3. Trimmomatic is used for trimming adaptor in the sequencing data
4.
5.
6.
7.
8.

Chapter 8

Troubleshooting

8.1 git@github.com: Permission denied

You may see this error:

```
Cloning into 'ProkGenomics'...
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Note that this is a private repository, you may required to log in using your github details. Github now requires for you to setup a token key to access private repositories, please follow the github instructions to set up one

Chapter 9

Resources

Programs description and citations

Tool	Description	Paper
FastQC	Sequence quality controls	Andrews S. (2010). FastQC: a quality control tool for high throughput sequence data. Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc
Trimmomatic	Trim primer adaptor from reads	Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014 Aug 1;30(15):2114-20. doi: 10.1093/bioinformatics/btu170. Epub 2014 Apr 1. PMID: 24695404; PMCID: PMC4103590. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4103590/

Tool	Description	Paper
Unicycler	De novo assembly	Wick RR, Judd LM, Gorrie CL, Holt KE (2017) Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLOS Computational Biology 13(6): e1005595. https://doi.org/10.1371/journal.pcbi.1005595
Plasclass	Plasmids prediction	Pellow D, Mizrahi I, Shamir R (2020) PlasClass improves plasmid sequence classification. PLOS Computational Biology 16(4): e1007781. https://doi.org/10.1371/journal.pcbi.1007781
CheckV	Phage prediction	Nayfach, S., Camargo, A.P., Schulz, F. et al. CheckV assesses the quality and completeness of metagenome-assembled viral genomes. Nat Biotechnol 39, 578–585 (2021). https://doi.org/10.1038/s41587-020-00774-7
Prokka	Prokaryotic gene annotation	Torsten Seemann, Prokka: rapid prokaryotic genome annotation, Bioinformatics, Volume 30, Issue 14, July 2014, Pages 2068–2069, https://doi.org/10.1093/bioinformatics/btu153

Tool	Description	Paper
Pharokka	Phage gene annotation	Bouras G, Nepal R, Houtak G, Psaltis AJ, Wormald PJ, Vreugde S. Pharokka: a fast scalable bacteriophage annotation tool. Bioinformatics. 2023 Jan 1;39(1):btac776. doi: 10.1093/bioinformatics/btac776. PMID: 36453861; PMCID: PMC9805569. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9805569/
CheckM	Assembly quality controls	Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015 Jul;25(7):1043-55. doi: 10.1101/gr.186072.114. Epub 2015 May 14. PMID: 25977477; PMCID: PMC4484387. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4484387/
Snippy	Single Nucleotide Variant (SNV) detection	

Other programs and pipelines

https://bactopia.github.io/v3.0.0/

https://proksee.ca/

https://genome.usegalaxy.org.au/

Chapter 10

Command Line Intro

If you are completely new to working with command line the following short introduction should be useful to get you started. Please read this section carefully, it will help you to understand instructions in later sections.

10.1 General syntax and conventions:

• Code or command are instructions directly given to the computer through a console or terminal window. Code or command lines in this tutorial are written with `this style` or

```
in this boxes
```

• If a string is written between < > it means that you have to type what that means in your case. For example: Login as: `<your username>` this means you have to type your user name in that space without the `< >`

For example, the following instructions should look like:

```
cp <file_name> <file_destination>
```

My file name os `myfile.txt` and my file destination is `newfolder`

```
cp myfile.txt newfolder
```

• When asterisk `*` is used it means all of that kind. For example: `ls *.fasta` will print a list of all files that have the extension .fasta

• Every line starting with `#` is a comment. There lines are not interpreted by your computer, there are there only to give you additional information.

10.1.1 Programs:

Command lines for executing programs usually looks like:

```
program --input <inputfile>
```

where
`program` is the program in question
`--input` is the option or parameter
`inputfile` is the argument

- Options/parameters for a program are denotated by a dash and a letter as:`-f` or a double dash and a string as: `--file`. If an option is not required but optional is often explained using `[ ]`, for example: `[-t 8]`

- Arguments are the input to the options/parameters. For example `-f myfile.txt`. -f is the option to input your file and myfile.txt is the argument for that option, the name of your file. The arguments are often explained using < >. When several arguments are possible for an option pipes are used to show the different possibilities, for example `[-f sam|bam]`. This means the option `-f` allows sam or bam formats

10.2 Basic commands:

When you enter your terminal your prompt consists of: `HOST_NAME:MACHINE CURRENT_DIRECTORY $` everything after $ is your command line. You can use the following basic commands to access information or perform tasks in your computer.

- **c**hange **d**irectory

```
cd <name of directory you want to change to>
```

`cd` or `cd ~` move you to your home directory

- **p**rint **w**orking **d**irectory

```
pwd
```

- **l**i**s**t your files

```
ls
```

- **m**a**k**e **dir***ectory

```
mkdir <new folder name>
```

- **c**o**p**y (needs file to be copied and destination).

```
cp <path of file to be copy> <destination path>
```

10.3 Files system

Please note that directories are structured in a hierarchical system. You have to know where you are standing to ask the computer to move to the correct folder.

Example of folder structure:

```
           | subfolder_1
           |
main_folder
           |
           | subfolder_2
                        |
                        |subfolder_2.1 (YOU ARE HERE)

#where am I?

pwd

#shows this path: /main_folder/subfolder_2/subfolder_2.1

# I want to go to the folder conteining this folder

cd ..

# moves to /main_folder/subfolder_2/


# I want to go to the folder conteining this folder and change to a folder that is there


cd ../subfolder_1

#moves to subfolder_1
```

Chapter 11

Connect to the cluster

11.0.1 Macs

If you are working on Mac you can directly open the terminal from applications or click the Launchpad icon in the Dock, type `Terminal` in the search field, then click Terminal. You will see a version of this:

type the following command

```
ssh <username>@<cluster_name>
```

where `<username>` is your authcate and the `<cluster_name>` is the cluster you are connecting to. Click enter, you will be asked for a password. Enter your password and click enter. Note you will not see the characters as they are typed. You are now in your home directory on the cluster.

11.0.2 Windows

If you are on a windows-based PC, you will need to download PuTTY.

In the hostname (or IP address) box, enter the hostname that you were provided, ie. `<username>@<cluster_name>`, where `<username>` where `<username>` is your authcate and the `<cluster_name>` is the cluster you are connecting to. Ensure the connection type is SSH. Click open. You will be prompted to enter your username (authcate) and password in the terminal window. Enter your credentials and click enter. Note you will not see the characters as they are typed. You are now in your home directory on the cluster.

Now you are ready to go Let's get started