Practical: cgMLST clustering

Overview

[How to use this document 1](#_Toc160195905)

[cgMLST 2](#_Toc160195906)

[The cgMLST scheme 2](#_Toc160195907)

[Run the chewBBACA allele calling: 2](#_Toc160195908)

# How to use this document

Software commands are highlighted in grey. Arguments must be filled in manually (so don’t copy-paste the commands!); all arguments are indicated by square brackets. For example, if you need to use the following command:

bgzip -k [input vcf-file]

on a vcf-file named “exciting.vcf”, you would type the following in your terminal:

bgzip -k exciting.vcf

# cgMLST

Core Genome Multi locus Sequence Type (cgMLST) is used by many public health institutes as typing method for surveillance and outbreak investigations. SeqSphere and BioNumerics are two major commercial software used for cgMLST analysis. Also, widely used are the web-based Enterobase and the open-source chewBBACA. We will use chewBBACA in today's exercise.

## The cgMLST scheme

The cgMLST scheme is a very important part of any cgMLST calling. Schemes can be found on Enterobase, pubMLST, BIGSdb (Pasteur) with more. In this exercise, we are using the *Listeria monocytogenes* scheme from BIGSdb.

Start by moving into the documents folder on the terminal and create a folder for the results

 cd Documents

mkdir cgMLST

cd cgMLST

Then activate the correct mamba environment

 mamba activate BTG\_genEpi\_cgMLST\_snp

The scheme used to call alleles needs to be prepared before running chewBBACA the first time. We have already done this for you. The code is below, but we will not run it here.

chewBBACA.py PrepExternalSchema -g name\_of\_scheme\_folder -o output\_name\_of\_scheme\_folder --ptf scheme.trn --cpu X --st X

When and if you prepare your own schemes and test the results in your lab, especially the size threshold (--st) option is very important. It decides when an allele is inferred as a new allele. If the size threshold is too big, an allele which is just a subsection of a previous allele could be called a new allele, thereby giving a larger distance than there is between the genomes.

Run the chewBBACA allele calling:

In this exercise, we're using 4 threads - if you have more available on your lab's cluster when you'll possibly run it yourself later, this can always be increased.

Make a folder called assemblies and link to the assemblies you haven’t made yourself earlier in the week, they are all here: /home/gebt/BTG\_2024/precomputed\_data/day7/assemblies

And also include the ones you made yourself. If you want to include everything in the directory you can just use \* instead of sampleid. You can do it like this:

 mkdir assemblies

cd assemblies

ln -s [path to assemblies]/<sampleid>.fa .

cd ..

Then call the alleles of all the samples using the cgMLST scheme below.

**cgMLST scheme:** /home/gebt/BTG\_2024/data/databases/Listeria\_monocytogenes\_Pasteur\_cgMLST\_2021-05-31T15\_chewBBACA3

chewBBACA.py AlleleCall -i [paths to assemblies folder with all included] -g [path to cgMLST scheme] -o [output folder] --cpu 4

Move into the output folder and have a look at the output files

 cd [output]

The actual allele calls are in the file **results\_alleles.tsv**. Check the file out with less (exit by pressing *q*) or just with a text viewer.

less -S results\_alleles.tsv

Check out the two other output files as well:

**results\_contigsInfo.tsv**: Where the position of the allele is on the contig.

**results\_statistics.tsv**: Overview of how many exact matches, how many loci have not been found and so on.

The abbreviations and descriptions can be found here, for example, check what LNF stands for. <https://chewbbaca.readthedocs.io/en/latest/user/modules/AlleleCall.html>

Do the results look as they should? Is enough of the genomes covered to make a proper decision? (the statistics file can be of help here)

To compare/cluster the allele calls we first have to calculate the distance between all genomes. We will do that using a tool Torsten Seemann created called cgmlst-dists. You can check out how the distances are calculated on the github page:
<https://github.com/tseemann/cgmlst-dists>

Calculate the distances:

 cgmlst-dists results\_alleles.tsv > dist.tsv

This distance matrix can be used to visualise the clustering. Here we will use an R script, where we will use hierarchical clustering with single linkage and write the plot to a pdf.

Furthermore, the plot will also plot the snp tree you made yesterday with the cgMLST tree you’re making now.

Rscript /home/gebt/BTG\_2024/scripts/automatic\_chewbbaca\_dendrograms\_and\_compTree\_genepi.R --cgmlst\_dist dist.tsv --cgmlst\_out cgmlst\_tree.pdf --snp\_out snp\_cgmlst\_tree\_comp.pdf --snp\_tree [path to snp tree from yesterday] --snp\_align [path to snp alignment (core.aln) from yesterday]

Take a look at the new file (**cgmlst\_tree.pdf**) showing the cgMLST clustering:

Is the sample you assembled yourself part of a genetic cluster? How many differences are the isolates approximately apart? Are there any clusters?

Also take a look at the other generated file (**snp\_cgmlst\_tree\_comp.pdf**) that compares the snp-tree you got yesterday using snippy to the new cgMLST tree.

Do you get the same results with the two methods? The same genetic clusters?