Jun 26, 2024

Core Genome Multilocous Sequence Typing Development Workflow

DOI
dx.doi.org/10.17504/protocols.io.4r3l22o64l1y/v1

made.krisna[1,2], William Monteith[1,3], odile.harrison[4], Martin C. J. Maiden[1]

[1]Biology Department, University of Oxford, UK; [2]Nuffield Department of Clinical Medicine, University of Oxford, UK; [3]Biology Department, University of Bath, UK.; [4]Nuffield Department of Population Health, University of Oxford, UK

Made Krisna
University of Oxford

DOI: dx.doi.org/10.17504/protocols.io.4r3l22o64l1y/v1

Protocol Citation: made.krisna, William Monteith, odile.harrison, Martin C. J. Maiden 2024. Core Genome Multilocous Sequence Typing Development Workflow. **protocols.io** https://dx.doi.org/10.17504/protocols.io.4r3l22o64l1y/v1

Manuscript citation:
MA Krisna, KA Jolley, W Monteith, A Boubour, RL Hamers, AB Brueggemann, OB Harrison, MCJ Maiden. Development and Implementation of a Core Genome Multilocus Sequence Typing (cgMLST) scheme for Haemophilus influenzae. bioRxiv 2024.04.15.589521

Protocol status: Working
We use this protocol and it's working

Created: July 25, 2023

Last Modified: June 26, 2024

Protocol Integer ID: 85467

Keywords: population genetics, typing scheme, core genome, cgMLST

Abstract

Core genome MLST (cgMLST) is a high-resolution typing approach that enables the evaluation of bacterial population structure for research and public health-related purposes, such as outbreak confirmation and routine surveillance. Currently, many pangenome analysis tools are available, with each tool employing distinct strategies for clustering orthologous genes and identifying paralogs. This protocol aims to describe a step-by-step process of developing the computational pipeline for pangenome analyses utilising a combination of software, namely PIRATE, Panaroo, chewBBACA, PEPPAN, and Genome Comparator from BIGSdb.

Developing a computational pipeline for pangenome analysis tools using complete, reference genomes

1 Annotation of complete, reference genomes

1.1

- Input file FASTA files of complete, reference genomes
- The dataset of these genomes, FASTA files and metadata included, can be accessed through PubMLST H. influenzae Isolate Database below:

Dataset	
Reference genomes for cgMLST development	NAME
https://pubmlst.org/bigsdb?db=pubmlst_hinfluenzae_isolates&l=1&page=query	LINK

- The list of IDs:

Note

1
495
2222
5068
5069
5082
5083
5084
5230
5257
5284
5559
5571
19836

1.2 ▪ Annotation was done with Prokka software:

Software	
Prokka	NAME
CentOS 8.1	OS
Torsten Seemann	DEVELOPER
https://github.com/tseemann/prokka	SOURCE LINK

▪ Command:

Command

Genome annotation with Prokka (CentOS 8.1)

```
prokka  --outdir        [Output directory]      --cpus [number of
core/CPU] --compliant --prefix  [Genome ID]     [Path to FASTA file]
```

▪ Important output file: GFF file, one per genome

2 **Pangenome analysis 1: PIRATE (Pangenome Iterative Refinement and Threshold Evaluation)**

▪ Sample GFF file from annotation step using Prokka

▪ Software:

Software

PIRATE	NAME
CentOS 8.1	OS
Sion Bayliss	DEVELOPER
https://github.com/SionBayliss/PIRATE	SOURCE LINK

- Command:

Command

Pangenome analysis using PIRATE (CentOS 8.1)

```
PIRATE    -i    [Path to GFF files]    -o    [Output directory]    -a
   -r    -t    [number of core/CPU]
```

- Key output file:

FILE PIRATE.gene_families.ordered.tsv

Also available as csv file.

3 **Pangenome analysis 2: Panaroo**

- Sample GFF file from annotation step using Prokka

- Software:

Software

PIRATE	NAME
CentOS 8.1	OS
Sion Bayliss	DEVELOPER
https://github.com/SionBayliss/PIRATE	SOURCE LINK

- Command:

Command

Pangenome analysis using Panaroo (CentOS 8.1)

```
panaroo    -i    [path to gff files]/*.gff    -o    [output
directory]   -t    16    --clean-mode strict    -a    core    --
search_radius    1000    --refind_prop_match    0.7
```

- Key output file:

FILE gene_presence_absence.csv

4 **Pangenome analysis 3: PEPPAN (Phylogeny Enhanced Pipeline for PAN-genome)**

- Sample GFF file from annotation step using Prokka OR separate FASTA files and annotation file (i.e. annotation by BIGSdb)

- Software:

Software

PIRATE	NAME
CentOS 8.1	OS
Sion Bayliss	DEVELOPER
https://github.com/SionBayliss/PIRATE	SOURCE LINK

- Command:

Command

Pangenome analysis using PEPPAN (CentOS 8.1)

```
PEPPAN    --match_identity    0.7    --orthology    ml    [Path to
gff files]/*.gff    -t
   48
#output directory is automatically set up as current/working directory

PEPPAN_parser    -g    [Path to PEPPAN output file]/PEPPAN.PEPPAN.gff
   -s
  [output directory]    -t    -c    -a    95
```

- Key output file:

PEPPAN.PEPPAN.gene_content.csv

5 **Pangenome analysis 5: chewBBACA**

- Sample FASTA files

- Software:

Software	
chewBBACA	NAME
CentOS 8.1	OS
Instituto de Microbiologia, Instituto de Medicina Molecular, Faculdade de Medicina, Universidade de Lisboa	DEVELOPER
https://github.com/B-UMMI/chewBBACA	SOURCE LINK

- Command:

Command

Pangenome analysis using chewBBACA ver 2.8.5 (CentOS 8.1)

```
#Make training file from reference genome (NC_016809.1)

prodigal    -i    [Path to reference genome FASTA file]    -t
hinf_training_file.trn
 -p    single

#Create schema

chewBBACA.py    CreateSchema    -i    [Path to FASTA files]    -o
   [Output directory for schema]    --n    [New directory name for
schema]    --ptf
  [Path to prodigal training file]    --st    0.1    --cpu 4
#st = size threshold = CDS size variation threshold. Added to the
schema's config file and used to identify alleles with a length value
that deviates from the locus length mode during the allele calling
process (default: 0.2)

#Allele calling

chewBBACA.py    AlleleCall    -i    [Path to schema directory]    -o
  [Output directory for allele calling]    --ptf    [Path to prodigal
training file]    --fc

#Test genome quality

chewBBACA.py    TestGenomeQuality    -i
  [Path to allele calling result]/results_alleles.tsv    -n    12    -
t    300    -s    5
#Results of this module will be available in the working directory.
Main output files: RepeatedLoci.txt and RemovedGenomes.txt


#Determine cgMLST

chewBBACA.py    ExtractCgMLST    -i    [Path to allele calling
result]/results_alleles.tsv    -o
  [Output directory]    --r    [Path to TestGenomeQuality
result]/RepeatedLoci.txt
  --g    [Path to TestGenomeQuality result]/RemovedGenomes.txt    --t
   0.95
```

- Key output files:

results_statistics.tsv cgMLST.tsv Presence_Absence.tsv

Comparison of pangenome analysis from different tools

6 - **Table 1. Summary of pangenome analysis results (N = 14)**

	A	B	C	D	E
	Group	PEPPAN	PIRATE	chewBBACA	Panaroo
	Core genes	1413	1328	849	1318
	Shell genes	1452	746	1029	819
	Cloud genes	387	876	967	1325
	Total	3252	2950	2845	3462

Putative paralogous loci have been excluded from each gene group classification.
Core genes: genes present in >= 95% of the genomes.
Shell genes: gene present in 15 >= but < 95% of the genomes.
Cloud genes: genes present in < 15% of the genomes.

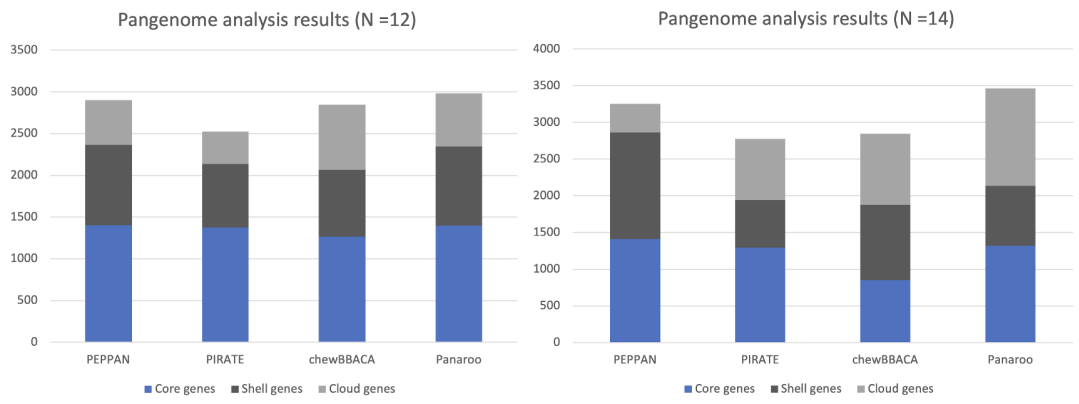
- Based on chewBBACA pangenome analysis in step 5 above, two genomes were suggested to be excluded: 10P129H1 and 84P36H1 (PubMLST id: 2222 and 19836, respectively) due to a lower number of annotated genes. Please refer to this output file of the TestGenomeQuality step: removedGenomes.txt 1KB

- The pangenome analysis (Steps 1-5) was redone with the 12 genomes.

- **Table 2. Summary of pangenome analysis results (N = 12)**

	A	B	C	D	E
	Group	PEPPAN	PIRATE	chewBBACA	Panaroo
	Core genes	1400	1376	1265	1397
	Shell genes	964	761	803	950
	Cloud genes	539	388	777	636
	Total	2903	2525	2845	2983

- Bar graph comparison of results from Table 1 and 2:



Core genes: genes present in >= 95% of the genomes.
Shell genes: gene present in 15 >= but < 95% of the genomes.
Cloud genes: genes present in < 15% of the genomes.

- Based on this comparison: Although they were still affected by the quality of genomes included in the analysis, **PIRATE's pan- and core genome analysis showed the most stable results**, with the total number of pan genes approximating the number discovered in previous literature.

Core genome multilocus sequence typing (cgMLST) development

7 Compiling dataset for cgMLST development and validation

The dataset of these genomes, FASTA files and metadata included, can be accessed through PubMLST H. influenzae Isolate Database, with the following IDs:

Development dataset PubMLST IDs:

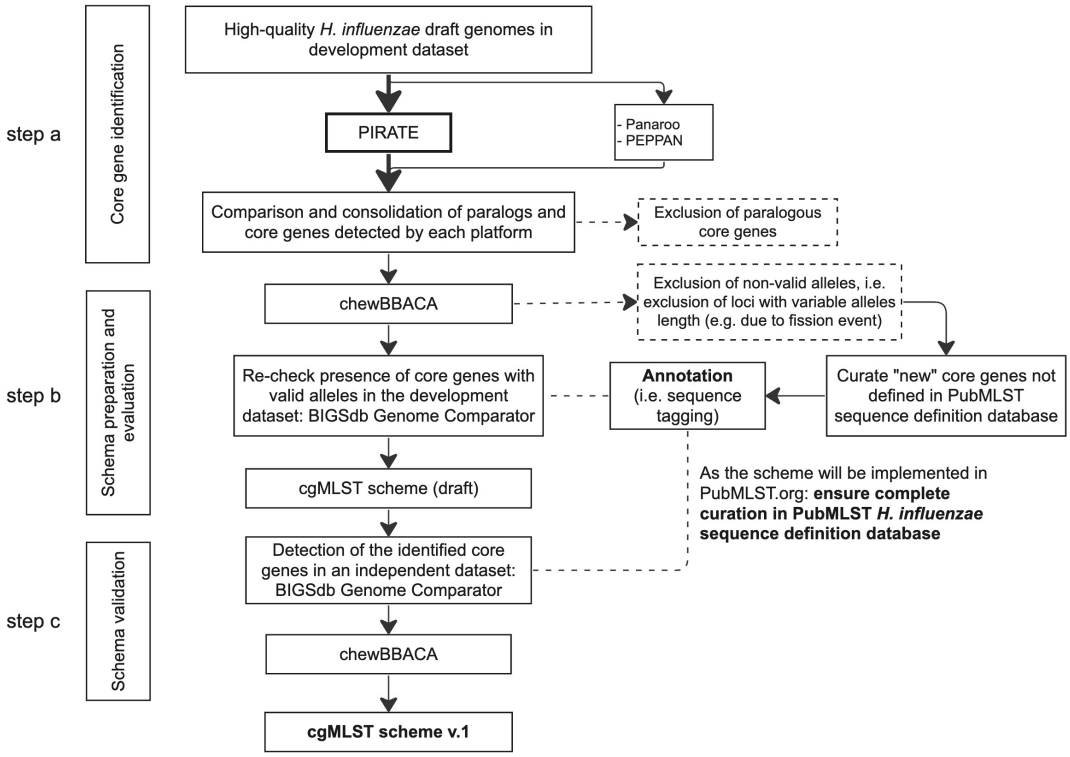
development dataset_id_public.txt 4KB

Validation dataset PubMLST IDs:

validation dataset_id_public.txt 7KB

8 Combining multiple pangenome analysis pipelines

The final workflow for core genome identification combining multiple pangenome analysis pipelines, based on the results obtained from the Part 1-6 above, is as follows:



PIRATE was the primary tool used for core gene identification. At the same time, analysis with Panaroo and PEPPAN was done to account for possible annotation errors and increase the sensitivity of detecting paralogous loci, respectively.

9 Consolidating results from multiple pangenome analysis pipelines

Step by step on how to identify core genes, combining automatic and manual curation:

1. Choosing several "model" genomes based on PIRATE key output
2. Preparing input file format
3. Python script to generate output file
4. Repeat step 1-3 for different pangenome analysis tools
5. Compare each output file from step 4 in Excel with VLOOKUP function.
6. If needed, compare the output file from step 4 with PubMLST annotation to identify the core gene(s) not defined in the database yet.

Step 1-4 will be detailed further below.

9.1 Choosing several "model genomes" based on PIRATE key output*

1. Excluding genes (or "gene family", a term in PIRATE) which:
   - were not present in at least 95% of the genomes.
   - underwent duplication event ("genomes_containing_duplication" column > 0)

This will be the temporary core gene lists.

2. Find genome(s) with the most core genes annotated and with the least "fission events". The fission event for each locus (if there are any) will be notable as loci with more than one allele (usually two) whose sequences do not overlap with each other.

These genomes will be our "model genomes" and their annotation results will cover the total core genes in our temporary core gene lists.

**See Part 2 above for an example of PIRATE key output*

9.2 Preparing input file format

There will be two types of input files: CSV file and the modified gff files from the pangenome analysis.

1. CSV file

Template, with example: input_file_format-consolidation_step.... 0B
Each row represents 1 (temporary) core loci/core gene, as distinguished by the "gene_family" column and 1 core gene is represented at least once.
Note on each column:

- gene_family: unique ID from the pangenome analysis output for each core gene.
- consensus_gene_name: the gene name of the core gene, when available. This is not a unique identification measure for the core gene.
- isolate_id: the ID of one of the model genomes that "covers" the corresponding core gene. It is possible to have one core gene covered by more than one model genome. See examples below.
- locus_id: locus id from the corresponding model genome and core gene it represents. See examples below.

Example 1

	consensus_gene_name	gene_family	isolate_id	locus_id
	rpoD	g02211	isolate1	isolate1_00341
	dnaG	g01840	isolate2	isolate2_00831
	rpsU	g01156	isolate3	isolate3_12943
	tsaD	g00998	isolate4	isolate4_27381

There were 4 core genes, each represented by one different model genome.

Example 2

	consensus_gene_name	gene_family	isolate_id	locus_id
	rpoD	g02211	isolate1	isolate1_00341
	rpoD	g02211	isolate2	isolate2_00712
	dnaG	g01840	isolate2	isolate2_00831
	rpsU	g01156	isolate3	isolate3_12943
	tsaD	g00998	isolate3	isolate3_13007

There were 4 core genes, with 1) one core gene (g02211) represented by 2 different model genomes (isolate1 and isolate2) and 1) two core genes (g01156 and g00998) represented by the same model genome (isolate3)

2. Modified gff files
Copy modified gff files (one of the outputs of pangenome analysis) for all the model genomes to a new folder.

9.3 Python script to generate output file

This script was run via Jupyter Notebook and structured according to PIRATE pangenome analysis output files structure. There are slight modifications of the script to process output from other pangenome analysis tools (i.e. Panaroo and PEPPAN).

What the script does: Look up modified gff files to find the start and end location of each core gene in the corresponding model genomes. This information allows for comparison with any pangenome analysis and/or genome annotation tools.

```
##by Krisna, M & Monteith, W 2023

import pandas as pd
import os

df_p2 = pd.read_csv('/path/to/input_file.csv')
core_gene_list_p2 = df_p2['locus_id'].to_list()

os.chdir('/path/to/selected/modified/gff/files')
files_p2 = os.listdir()

result_p2 = {}
for locus in core_gene_list_p2:
    for filename in files_p2:
        with open(filename, 'r') as f:
            content = f.readlines()
        for line in content:
            if locus in line:
                if locus not in result_p2:
                    result_p2[locus] = line

if len(result_p2) != len(core_gene_list_p2):
    for locus in core_gene_list_p2:
        if locus not in result_p2:
            print(locus, 'not found in gffs')

start_loc_p2 = []
end_loc_p2 = []
note = []
for locus in core_gene_list_p2:
    result = result_p2[locus]
    result = result.split('\t')
    start_loc_p2.append(result[3])
    end_loc_p2.append(result[4])
    note.append(result[8])

note2 = []
prev_locus = []
for data in note:
    split_data = data.split('prev_locus=',1)[1]
    note2.append(split_data)

for data in note2:
    split_data1 = data.split(';')
```

```
    prev_locus.append(split_data1[0])

pirate_output = pd.DataFrame (
    {
        'gene_family' : df_p2['gene_family'],
        'locus_id' : df_p2['locus_id'],
        'isolate_id' : df_p2['isolate_id'],
        'start_loc' : start_loc_p2,
        'end_loc' : end_loc_p2,
        'prokka_locus_name' : prev_locus
    }
)

pirate_output.to_csv('/path/to/output_file.csv')
```

Output file structure, with example: output_file_format-consolidation_step… 0B

Protocol references

Seemann T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*. 2014;30(14):2068-2069. doi:10.1093/bioinformatics/btu153

Sion C Bayliss and others, PIRATE: A fast and scalable pangenomics toolbox for clustering diverged orthologues in bacteria, *GigaScience*, Volume 8, Issue 10, October 2019, giz119, https://doi.org/10.1093/gigascience/giz119

Silva M, Machado MP, Silva DN, Rossi M, Moran-Gilad J, Santos S, Ramirez M, Carriço JA. chewBBACA: A complete suite for gene-by-gene schema creation and strain identification. Microb Genom. 2018 Mar;4(3):e000166. doi: 10.1099/mgen.0.000166. Epub 2018 Mar 15. PMID: 29543149; PMCID: PMC5885018.

Tonkin-Hill, G., MacAlasdair, N., Ruis, C. *et al.* Producing polished prokaryotic pangenomes with the Panaroo pipeline. *Genome Biol* **21**, 180 (2020). https://doi.org/10.1186/s13059-020-02090-4

Zhou Z, Charlesworth J, Achtman M. Accurate reconstruction of bacterial pan- and core genomes with PEPPAN. *Genome Res*. 2020;30(11):1667-1679. doi:10.1101/gr.260828.120