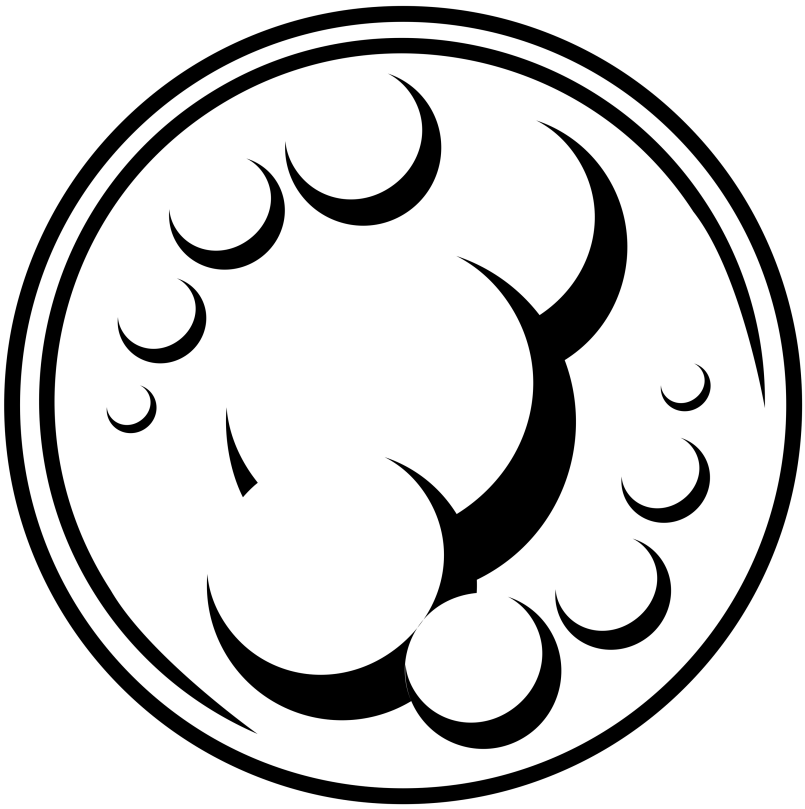

CYCLONE

CYbernetic Clustering Of Non-clonal Eukaryotyes

Installation and usage guide for CDCs *Cyclospora cayetanensis* typing workflow:

CYCLONE

Complete description of the workflow

Version of this document was saved April 2, 2021.

Author: Joel Barratt[1,2]

[1] Parasitic Diseases Branch, Division of Parasitic Diseases and Malaria, Center for Global Health, Centers for Disease Control and Prevention, Atlanta, GA, USA

[2] Oak Ridge Institute for Science and Education, Oak ridge, TN, USA

*To whom correspondence should be addressed. Email: jbarratt@cdc.gov Alternative email: joelbarratt43@gmail.com

Table of Contents

Dependencies

This workflow was developed and tested by Joel Barratt using a Mac running OSX Catalina 10.15.3. The subsequent instructions are provided only for installing this workflow on an OSX system. The R code provided with this software should function correctly with R versions 3.6.1 to 3.6.3. The following R libraries must be installed: filesstrings, dplyr, stringr, gtools, parallel, tidyverse, cluster, quantmod, reshape, dbscan and spaa. Next, to install other third-party software required to run this workflow (e.g. CD-HIT) you will need to install GCC – the GNU Compiler Collection (e.g. **`brew install gcc`**). Other third-party software required to run this workflow (e.g. cutadapt v.3.0) require python (i.e. python 3.9) and pysam. The user must navigate to the BBMAP and CD-HIT directories included in this workflow (See Figure S1) and these software must be extracted from the tar.gz files and ***compiled in place***, in the correct directory without modifying names of extracted directories. The user must navigate to the BLAST directory (See Figure S1), and download this version of BLAST ***precisely***: ncbi-blast-2.9.0+-x64-macosx.tar.gz. The user must extract this file in the BLAST directory and extract this tar.gz file in place (See Figure S1) without changing the name of the uncompressed folder. In addition, EMBOSS must be installed (seqret which is part of EMBOSS, is required by this workflow). The user must have SAMTOOLS and BCFTOOLS installed correctly and included in the PATH variable. Seqtk must be installed and added to the PATH variable as well. Users must install gsed and add it to their PATH variable. Installation of samjs[1], pcrclipreads[2], and biostar84452[3] is required. Users must install these tools and then move the samjs.jar, pcrclipreads.jar, and biostar84452.jar files to the **`/Library/Java/Extensions/`** directory on their Mac OS machine. Alternatively, users can

[1] http://lindenb.github.io/jvarkit/SamJavascript.html
[2] http://lindenb.github.io/jvarkit/PcrClipReads.html
[3] http://lindenb.github.io/jvarkit/Biostar84452.html

modify the source code to add the full path to these files in the shell script: **`Finding_New_Haps_SNP_Based_Module.sh`** located in the **`HAPLOTYPE_CALLER_CYCLO_V2 directory`** (see Figure S1). Each time these jar files are executed in that script (i.e lines 98, 104 and 109), add the full path to wherever these jar files have been placed (eg. change from **`/Library/Java/Extensions/samjs.jar`** to **`/Path/to/the/files/samjs.jar`**). Other third-party software provided with this workflow (discussed next) must be installed correctly by the user who should refer to the installation instructions for each of these software to ensure all dependencies and pre-requisites are met. Third party software that must be unzipped and (sometimes) compiled in place include: BBMAP – provided with this workflow (**`./HAPLOTYPE_CALLER_CYCLO_V2/BBMAP/`**), BLAST – discussed above – specific version must be downloaded and installed in a specific directory (**`./HAPLOTYPE_CALLER_CYCLO_V2/BLAST/`**), Bowtie2 – provided with this workflow (**`./HAPLOTYPE_CALLER_CYCLO_V2/BOWTIE/`**), Cap3 – provided with this workflow (**`./HAPLOTYPE_CALLER_CYCLO_V2/cap3/`**), CD-HIT – discussed above – provided with this workflow and must be compiled in place (**`./HAPLOTYPE_CALLER_CYCLO_V2/CD-HIT/`**), and Mira4 – provided with this workflow (**`./HAPLOTYPE_CALLER_CYCLO_V2/MIRA/`**). The complete workflow is available for download at the following GitHub repository: https://github.com/Joel-Barratt/CDC-Complete-Cyclospora-typing-workflow

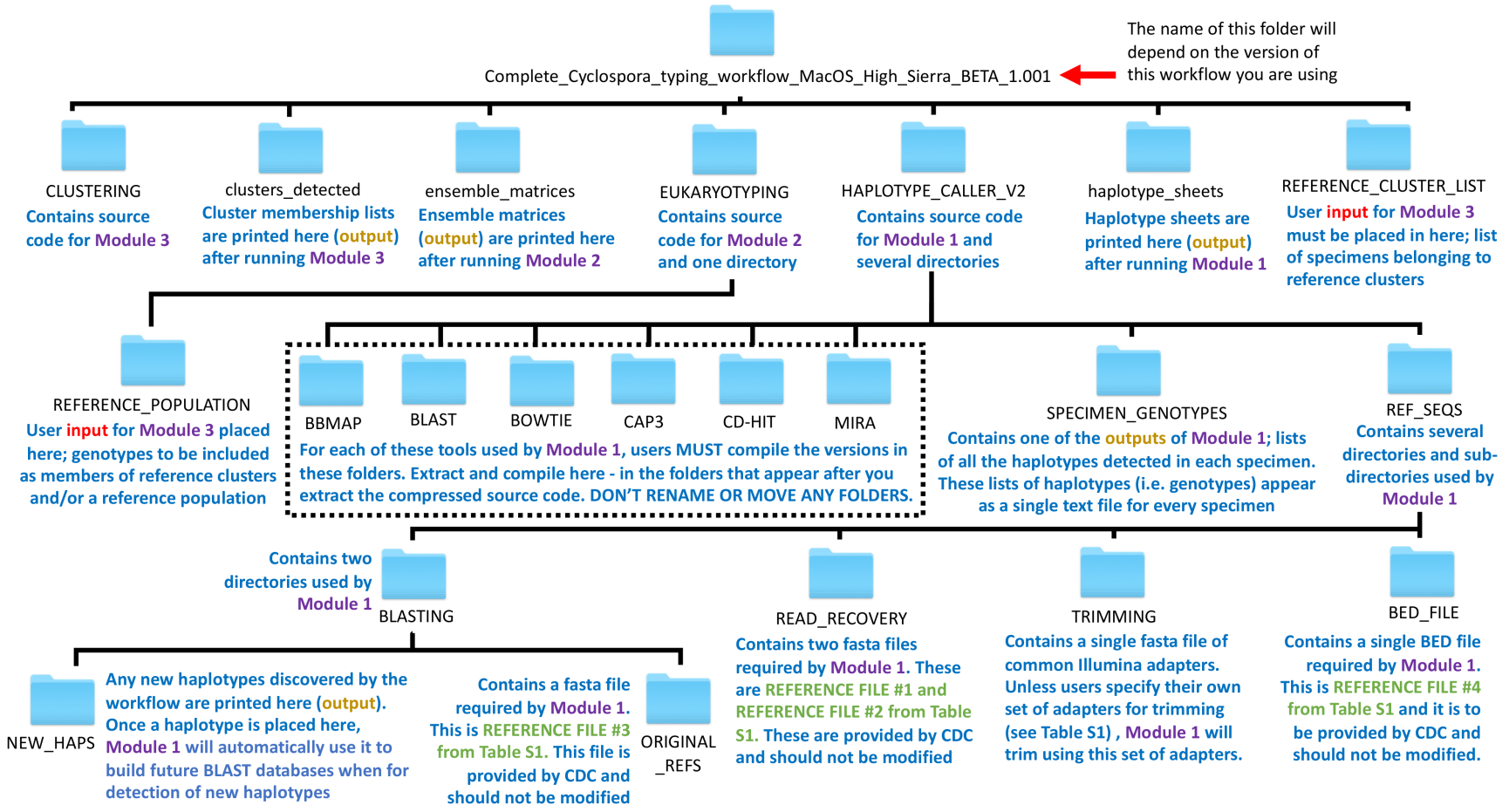

Figure 1 – Complete directory structure of the CDC *Cyclospora cayetanensis* genotyping workflow

This figure shows the location of all files and folders referred to in this usage guide. It also tells users where specific inputs must be provided by the user, and where specific outputs will be written by this software.

Brief description of the CDC *Cyclospora cayetanensis* workflow

This workflow is comprised of three modules that perform three major tasks required by CDC to identify genetically related clusters of Cyclospora cayetanensis infections.

1. **MODULE 1** - Assign haplotypes to each specimen in accordance with the CDC typing method discussed here. This module will also detect and validate novel haplotypes that have not been encountered previously and write them to a local database.

2. **MODULE 2** - Examine the genotype information generated by MODULE 1 and assess the relationship between each possible pair of specimens using this information. This second module is based on an updated version of the CDCs Eukaryotyping ensemble described here.

3. **MODULE 3** - Will predict the most appropriate number of clusters in the population under analysis using a set of reference specimens of known genetic linkage.

Usage instructions for each Module

Module 1 - Input data for haplotype calling

In order to execute Module 1 correctly, several inputs must be provided by the user including several user-generated files (fasta files and BED files), as well as several user-defined parameters supplied as arguments to the following script: **`MODULE_1_hap_caller.sh`**

To remain compatible with the CDC *Cyclospora* genotyping program, the most up-to-date copies of these files must be provided by CDC upon request by the user.

The **`MODULE_1_hap_caller.sh`** script can be found in the following location:

`/Your_home_directory/place_you_extracted_cyclone/CYCLONE_MacOS_High_Sierra_BETA_Cyclospora/MODULE_1_hap_caller.sh`

Specific details on how to run the haplotype calling module of this workflow please refer to Table 1 below. To print a brief help menu containing a complete list of arguments to your terminal window, execute the following code:

```
/Your_home_directory/place_you_extracted_cyclone/CYCLONE_MacOS_High_Sierra_BETA_Cyclospora/MODULE_1_hap_caller.sh -h
```

Module 1 - Notes on Reference files 1 through 4

For the workflow to function correctly all reference sequences require that they are named according to the appropriate naming convention as set by CDC. This is because the anatomy of the reference sequence name informs the workflow of how to treat the data at various stages. The importance of correct file naming is discussed below.

REFERENCE FILE #1

REFERENCE FILE #1 which is described in Table S1 below, contains a single set of reference sequences for each marker (except for marker 7 – the Mt Junction – discussed later). **CDC will provide specific instructions on what this file must be named and the sequences contained within this file**. The purpose of this fasta file is to serve as a single representative reference sequence for mapping reads to. The name of each reference sequence in this file is as follows: **Nu_360i2, Nu_378, Nu_CDS1, Nu_CDS1, Nu_CDS1, Nu_CDS1, Mt_MSR.** The naming anatomy of these sequences **is important** in terms of how the workflow treats the data. In the case of the sequence with the name **Nu_360i2**, the string "Nu" indicates to the workflow that this marker is nuclear in origin which has implications for how distances are calculated by Module 2. The underscore "_" serves as a spacer and is required in this location. Next the string "360i2" is the base name of that specific locus as defined by CDC. For the sequence **Mt_MSR**, the string

"Mt" indicates that the locus is mitochondrial in origin which again serves to inform Module 2 of how to treat this locus when calculating distances. Again, the underscore "_" serves as a spacer and "MSR" is the base name of that specific locus. **Modifying the names of sequences in this file will cause the workflow to not function correctly.**

This file will be in the following directory:

/Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/READ_RECOVERY

REFERENCE FILE #2

This file serves a similar function to REFERENCE FILE #1 but contains references that are specific to the Mitochondrial Junction repeat locus. **CDC will provide specific instructions on what this file must be named and the sequences contained within this file.** Take the sequence named **Mt_Cmt109.A_Junction_Hap_1** for example. The string "Mt" tells the workflow that this is a mitochondrial locus which is important to inform Module 2 of how to treat this locus when calculating distances. The underscores "_" serve as spacers. The next string "Cmt109.A_Junction" is the base name for the locus. The number "109" tells you the length of the repeat, which includes the primer sequence as described here. The string "Hap_1" tells the user that this is the first haplotype described by CDC for this locus. **This number must never be changed as this will impact the calculation of distances by Module 2 resulting in erroneous links being identified.**

This file will be in the following directory:

/Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/READ_RECOVERY

REFERENCE FILE #3

Full-length amplicons for each locus analyzed as part of our *Cyclospora* genotyping workflow (which vary in length from about 400 to 700 base pairs) are dissected into segments of about 100 base pairs and haplotypes are defined separately for each 100 base-pair segment (i.e. our haplotypes are all ~100 bases long). **This has been done for a very good reason.** As described in our previous works here and here, PCR-induced chimera formation is a universal problem when PCR amplicons are deep sequenced. By dividing markers into small sub-segments, we greatly mitigate this issue. Importantly, the Eukaryotyping algorithm (described here) is not impacted much at all by this practice, yet the impact of identifying PCR-induced chimeras as novel haplotypes would be far more deleterious to the final analysis. Again, the naming anatomy of these sequences **is important** in terms of how the workflow treats the data, and the naming of sequences in this file is related to the name of sequences in REFERENCE FILE #1. In the case of the sequence with the name **Nu_360i2_PART_E_Hap_1**, the string "Nu" indicates to the workflow that this marker is nuclear in origin which has implications for how distances are calculated by Module 2. The underscore "_" serves as a spacer and is required in this location. Next the string "360i2" is the base name of that specific locus as defined by CDC. The string "PART_E" tells the workflow that this is segment "E" (i.e. the fifth 100 base pair section) of the locus 360i2. The anatomy of the string "PART_E" should not be changed – the text should be capitalized and the underscore must also come before the segment letter (a single letter). **CDC will provide specific instructions on what this file must be named and the sequences contained within this file.**

This file will be in the following directory:

`/Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/BLASTING/ORIGINAL_REFS`

REFERENCE FILE #4

This is a BED file that defines for the haplotype caller the location of the sub-segments of the full-length sequences in REFERENCE FILE #1 at which we would like to define our separate haplotypes. The BED file is prepared in the following format:

Nu_360i2	20	120	Nu_360i2_PART_A
Nu_360i2	120	220	Nu_360i2_PART_B
Nu_360i2	220	320	Nu_360i2_PART_C
Nu_360i2	320	420	Nu_360i2_PART_D
Nu_360i2	420	520	Nu_360i2_PART_E
Nu_360i2	520	630	Nu_360i2_PART_F
Nu_378	22	136	Nu_378_PART_A
Nu_378	136	236	Nu_378_PART_B
Nu_378	236	336	Nu_378_PART_C
Nu_378	336	450	Nu_378_PART_D
Nu_CDS1	20	87	Nu_CDS1_PART_A
Nu_CDS1	87	155	Nu_CDS1_PART_B
Nu_CDS2	20	123	Nu_CDS2_PART_A
Nu_CDS2	123	226	Nu_CDS2_PART_B
Nu_CDS3	22	111	Nu_CDS3_PART_A
Nu_CDS3	111	200	Nu_CDS3_PART_B
Nu_CDS4	21	90	Nu_CDS4_PART_A
Nu_CDS4	90	159	Nu_CDS4_PART_B
Mt_MSR	21	143	Mt_MSR_PART_A
Mt_MSR	143	243	Mt_MSR_PART_B
Mt_MSR	243	343	Mt_MSR_PART_C
Mt_MSR	343	443	Mt_MSR_PART_D
Mt_MSR	443	543	Mt_MSR_PART_E
Mt_MSR	543	665	Mt_MSR_PART_F

This is a tab-separated text file with four columns and now column headers. Column 1 contains the name of each locus being analyzed, and the names listed in this column are identical to those in REFERENCE FILE #1. The second column defines where the specific segment of this locus begins. So actually, for Nu_360i2_PART_A the number 20 shown in the first row of the second column, indicates that PART_A of this locus (as per the name in column 4) begins at the next base (i.e. base 21) and ends at (and includes) base 120 (as per row 1, column 3) relative to the full-length sequence of Nu_360i2 in REFERENCE FILE #1. Column 4 of this bed file follows precisely the same naming convention as sequences in REFERENCE FILE #3. **For consistency**

across all users, CDC will provide specific instructions on what this BED file must be named and the named sequences contained within this file.

This file will be in the following directory:

`/Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/BED_FILE`

TIP: If this BED file was made using Microsoft excel, you will need to run DOS2UNIX to convert it to the correct text file encoding or Module 1 will not run correctly.

Table S1. Arguments and data required as input for the CYCLONE haplotype calling workflow for *Cyclospora cayetanensis*

Input file name or argument [essential or optional]	**Description**
Argument used to print short help menu **[optional]** **-h**	This will simply print a short help menu containing a list of arguments. **Example:** **~ Username$ bash MODULE_1_hap_caller.sh -h**
Argument to indicate where the workflow was installed **[essential]** **-C**	Argument tells the bash script the location that you have installed (unzipped) the workflow. I use this flag to greatly simplify installation. DO NOT include the workflow folder in this path (e.g. **CDC_Complete_Cyclospora_typing_workflow_[version]/**). Only include the directory in which the workflow was unzipped. **Example:** **~ Username$ bash MODULE_1_hap_caller.sh ** **-C /user/home/path/to/where/I/unzipped/workflow ** **-*some more arguments***
name_of_folder_containing_all_of_your_Illumina_fastq.gz_files **[essential]** **-D**	Allows the user to specify where their paired-end fastq.gz files for analysis are located. The software will search this folder for your pairs of fastq.qz files and begin processing only the reads placed in this folder upon execution. Users must supply the *explicit* path to this folder using this flag. All files MUST be paired end and they MUST be provided in **fastq.gz** format. **No exceptions.** Don't place anything else in this folder – just your data. If there is an issue with the files provided in your folder, the workflow will throw an error. **Example:** **~ Username$ bash MODULE_1_hap_caller.sh -C /user/home/path/to/where/I/unzipped/workflow ** **-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files ** **-*some more arguments***
list_of_Illumina_adpater_sequences.fasta **[optional]** **-I**	Allows the user to specify their own set of Illumina adapter sequences for BBTOOLS to use during adapter trimming. If no value is provided automatically, a default set will be used. A set of default Illumina adapters is provided with this software. **Example:** **~ Username$ bash CYCLOSPORA_haplotype_caller.sh -C /user/home/path_to_where_I_unzipped_CYCLONE ** **-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -I /user/home/path/Illumina_adapters.fasta ** **-*some more arguments*** **Default adapter file location if flag is not used:** **/user/home/path/to/where/I/unzipped/workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/TRIMMING/Illumina_adapters.fasta**

REFERENCE FILE #1 [essential] **Note: to remain compatible with the network users must request a copy of the most up-to-date file from CDC. Users SHOULD NOT modify this file** A multi fasta sequence file.	This file includes a single (one) reference sequence for each of genotyping markers 1 to 6, and 8, from the current panel of 8 CDC genotyping markers. Specifically, a single (one) full length reference (consensus) sequence is included in this file for each locus; Nu_360i2, Nu_378, Nu_CDS1, Nu_CDS2, Nu_CDS3, Nu_CDS4, and Mt_MSR. **This file is curated at CDC to ensure consistency in the reference sequences for all users of this software.** The workflow uses these references as part of an early QC process where reads are mapped to each reference using highly relaxed mapping parameters, and reads that map are retained for downstream haplotype calling. These references also correspond to the sequence names referenced in REFERENCE FILE #3 and REFERENCE FILE #4 (see below). **Location of this file:** **`/user/home/path/to/where/I/unzipped/workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/READ_RECOVERY/MAPPING_NON_JUNCTION_REFERENCES_WITH_PRIMER_FEB_2020.fasta`** **To remain compatible with the CDC genotyping program DO NOT provide your own references.**
REFERENCE FILE #2 [essential] **Note: to remain compatible with the network users must request a copy of the most up-to-date file from CDC. Users SHOULD NOT modify this file** A multi fasta sequence file.	This file includes the sequence of numerous haplotypes (20 at the time this was written) for the mitochondrial junction (marker 7). **This file is curated at CDC to ensure consistency in the reference sequences for all users of this software.** The workflow uses these references as part of an early QC process where reads are mapped to each reference using highly relaxed mapping parameters, and reads that map are retained for downstream haplotype calling. As this mitochondrial junction represents a repeat sequence, haplotypes for this marker are called using a separate workflow to the other markers which possess polymorphisms that are SNP and/or indel based. For this reason, the fasta sequences in this reference file are not referenced in the BED file. **`/user/home/path/to/where/I/unzipped/workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/READ_RECOVERY/MAPPING_JUNCTION_WITH_PRIMERS_FEB_2020.fasta`** **To remain compatible with the CDC genotyping program DO NOT provide your own references.**
REFERENCE FILE #3 [essential] **Note: to remain compatible with the network users must request a copy of the most up-to-date file from CDC. Users SHOULD NOT modify this file** A multi fasta sequence file.	This fasta file contains the reference sequence of various haplotypes that have already been divided into segments based on the REFERENCE FILE #4. This does not have to be a complete list of all known haplotypes because the workflow searches for novel haplotypes de novo and writes these to an alternative location: the NEW_HAPS directory (see Figure 1). This file primes the workflow with known haplotypes. **`/user/home/path/to/where/I/unzipped/workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/BLASTING/ORIGINAL_REFS/CYCLOSPORA_REFERENCES_SHORTENED_FEB_2020.fasta`** **To remain compatible with the CDC genotyping program DO NOT provide your own references.**
REFERENCE FILE #4 [essential] **Note: to remain compatible with the network users must request a copy of the most up-to-date file from CDC. Users SHOULD NOT modify this file** A BED file.	A file in Browser Extensible Data (BED) format. Recall that as part of this workflow, each of the markers (excluding the Mt Junction) are divided into sub-segments of approximately 100 bases (though this may change), where each 100 base segment is considered a separate locus. This BED file defines the specific location of each sub-segment relative to all fasta sequences in the REFERENCE FILE #1 so that the haplotype caller can identify the haplotypes within each segment. The sequences referenced in this BED file must be identical to the sequences provided in REFERENCE FILE #1. **`/user/home/path/to/where/I/unzipped/workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/BED_FILE/CYCLOSPORA_FEB_11_2020.bed`** Not applicable. The version of the software is set up so that CDC provides the most up-to-date reference sequences to its users to ensure consistency across the *Cyclospora cayetanensis* surveillance network. **We advise that users DO NOT provide their own references at this step for the sake of consistency.** A newer version of this software is under development where users can provide their own references, though this will be released at a later time.

Argument to specify required depth in order to assign a haplotype to a specimen **[optional]**
Note: to remain compatible with the network users must use 20 as their default.
-A
	This flag allows the user to specify the depth of sequencing required before a haplotype that is already in the haplotype database is assigned to a specimen. A default of 20 is used if no value is specified. Note that this is **not the same** as the depth required to write a newly discovered haplotype to the haplotype database (i.e. in the NEW_HAPS directory). If the haplotype caller identifies a haplotype within a given specimen that achieves a sequencing depth of this value or greater, the haplotype caller will consider this haplotype to be present in the specimen. It will then compare the sequence of this haplotype to all haplotypes in REFERENCE FILE #3 and any haplotypes written to the NEW_HAPS directory to identify which haplotype was found in the specimen. Once the haplotype is identified, the workflow will write it to the text file containing this specimens' complete genotype (i.e. in the SPECIMEN_GENOTYPES folder – see Figure 1). Also note that depth is calculated separately for each sub-segment of the markers listed in REFERENCE FILE #1 as defined in the REFERENCE FILE #4 BED file (not as an average across the full length of any marker). For the junction repeat, an average across the entire marker is used though haplotype calling for this marker is slightly different and uses an entirely separate script to the other markers. **IMPORTANT:** The workflow is dynamic in that the depth required to assign a haplotype to a specimen changes depending on the coverage obtained. If a particular locus obtains excessive coverage, the haplotype will need to be supported by 10% of all reads that map to this specific site before it is recorded as being part of this specimens' genotype. If this amount of coverage is not obtained for a given haplotype, then this haplotype will not be assigned to that specimen – even if a coverage of 20 is exceeded. Therefore, the cutoff is dynamic, and 20 reads represents the minimum requirement.
	Example:
	`~ Username$ bash MODULE_1_hap_caller.sh -C /path/to/workflow/ \`
	`-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -I /user/home/path/Illumina_adapters.fasta -A 20 \`
	`-*some more arguments*`

Argument used to indicate if you would like to generate a haplotype data sheet or not **[optional]**
-H
	This flag allows the user to indicate whether they want a haplotype data sheet to be generated after running MODULE 1 on a set of specimens. If no value is provided, a default of NO will be used. Users can specify 'yes' or 'no' in multiple ways; eg – NO, no No, N, YES, yes, Yes, Y
	Example:
	`~ Username$ bash MODULE_1_hap_caller.sh -C /path/to/workflow/ \`
	`-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -I /user/home/path/Illumina_adapters.fasta -A 20 -H Y`
	`-*some more arguments*`

Argument used to indicate the name length of your specimens. **[optional]**
-L
	This flag will define the length of the specimen names (it assumes name lengths are consistent for every specimen) - i.e. after how many characters should the haplotype caller truncate the name of the fastq.gz files provided? Default is 11. You must use 11 to be compatible with the CDC specimen naming conventions.
	Example:
	`~ Username$ bash CYCLOSPORA_haplotype_caller.sh -C /path/to/workflow/ \`
	`-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -H Y -L 11 -*some more arguments*`

Argument used to allocate the amount of RAM required during clustering. **[optional]**
-R
Value must be provided in MB.
	Default is 1000 MB. Depending on the number of reads generated for each specimen, you may want to allocate more. An insufficient amount of RAM can cause problems during the read clustering stages of the workflow (i.e. when CD-HIT is executed). The workflow may appear to have run to completion, but you will notice that several markers may be missing in the final genotype of specimens for which large amounts of data were generated (i.e when the paired fastq.qz files for a specimen are more than 100 MB each in size 1000 MB of RAM or more may be needed).
	Example:
	`~ Username$ bash MODULE_1_hap_caller.sh -C /path/to/workflow/ \`
	`-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -H Y -L 11 -R 2000 -*some more arguments*`

Argument used to indicate the number of threads you would like to use **[optional]** **-T**	Default is 10, noting that only certain functions are multi-threaded. However, running various functions on multiple threads will drastically speed up the workflow. **Example:** **~ Username$ bash MODULE_1_hap_caller.sh -C /user/home/path/to/workflow/ ** **-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -H Y -L 11 -R 2000 ** **-T 28 ** **-*some more arguments***
Argument used to indicate the depth of sequencing required to add a novel haplotype to your database (i.e. to the NEW_HAPS directory – see Figure 1) **[optional]** **-N**	Default is 100. This coverage requirement **is not the same** as the depth required to assign a haplotype to a specimen; a specimen will only be assigned a haplotype if that haplotype is already in the reference database. Therefore, the search for novel haplotypes occurs on each specimen as the first step of MODULE 1 so that novel haplotypes can be written to file before they are assigned to a specimen. After performing read quality control, the workflow checks all specimens in the current batch for novel haplotypes at each locus. First, reads are mapped to REFERENCE FILE #1. The BED file (REFERENCE FILE #4) is then used to divide each of the reference sequence in REFERENCE FILE #1 into segments, where each segment is considered a separate locus when haplotypes are defined. The workflow extracts the number of reads that span each of these segments entirely, trims the bases outside of each segment defined in the BED file, and then counts the number of remaining reads. This represents the sequencing depth at that segment. Next, all these reads spanning each segment are clustered into bins of 100% identity and the number of reads present in each bin represents the depth of coverage for a given haplotype. If this coverage exceeds the value of -N, then one read from this bin will be BLASTed against the haplotypes in REFERENCE FILE #3 plus any haplotypes previously written to the NEW_HAPS directory. If any of the bins with enough coverage (coverage greater than or equal to -N) do not obtain a 100% identity BLAST hit to anything in REFERENCE FILE #3 or the NEW_HAPS directory, the sequence is considered novel and will be written to the NEW_HAPS directory. From then on, it will be included as part the list of reference haplotypes in all future runs of Module 1. At this time, a name and number is assigned automatically to a haplotype based on the sequence in which it was discovered. **Never remove files from the NEW_HAPS folder** without first consulting CDC. This will throw out the sequence of numbering/naming. This process just described applies to all markers except for marker 7 (i.e. the Mt Junction). The procedure for the Mt Junction sequence is similar (though slightly different), requiring a separate and slightly more complicated workflow. This is namely because its polymorphisms are based on repeat copy number, where for all other markers the polymorphisms are based on SNPs and indels. **IMPORTANT:** The workflow is dynamic in that the depth required to detect a novel haplotype and then write it to the reference database depends on the coverage obtained. If a particular locus obtains excessive coverage, the haplotype will need to be supported by 25% of all reads that map to this specific site before the novel haplotype is retained for future reference. If this amount of coverage is not obtained for a given haplotype, then this haplotype will not be written to the database – even if a coverage of 100 is exceeded. Therefore, the cutoff of 100 reads represents the minimum depth of coverage required to write a novel haplotype to file for future reference. **Example:** **~ Username$ bash MODULE_1_hap_caller.sh -C /user/home/path/to/workflow/ ** **-D /user/home/path/folder_where_I_pasted_my_fastq.gz_files -H Y -L 11 -R 2000 -T 28 -N 100 -*some more arguments***

Note: Figure 2 provides a detailed schematic of whole Module 1 of this workflow functions.

Module 1 - Outputs

There are three main outputs of Module 1: (1) the genotype of each specimen, (2) any haplotypes that were discovered *de novo* by the haplotype caller, and (3) a haplotype data sheet (if the user selected -H Y when running the haplotype calling module).

Specimen genotypes

The genotype of each specimen will be printed to a single text file. This text file contains the list of all the haplotypes detected in a specimen along with some of the BLAST results obtained fLor these haplotypes. The sequence of each haplotype detected is also included in this text file. The name of each text file corresponds to the name of the paired fastq.gz files that you provided to the haplotype caller. The haplotype caller truncates the name of the paired end fastq.gz files provided by 11 characters (11 by default, though the name length can be modified using the -L flag) and the resulting truncated string will become the name of the text file containing that specimens' genotype. Note that any dashes in the original fastq.gz file names will be replaced for underscores when the text file name is printed. If the specimen contains an insufficient amount of data to detect any haplotypes, a text file ***should*** still be printed in the fashion described here, but this file will be empty. All specimen genotypes are printed to this location:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/SPECIMEN_GENOTYPES

Novel haplotypes

Any novel haplotypes detected will be printed to a fasta file. When the haplotype calling workflow discovers a new haplotype for any locus, it automatically assigns a new name and haplotype number to this novel sequence. To do this, the workflow first counts the number of

haplotypes that already exist for this locus and then assigns the newly discovered haplotype to the next available number in sequence. **For this reason, it is important that this software be maintained at a central location and that every copy of this software reference the same database of haplotypes. Also, if any files are moved or deleted from this folder this will cause problems downstream with the automatic naming of newly discovered haplotypes.** For example, if two different haplotypes are detected at the same time in different laboratories and they are assigned the same haplotype name and number, the results in each of these laboratories will not be comparable, and the Module 2 will assume that these two laboratories have detected the same sequence – even though they may be different. For this reason, CDC will not compare genotypes compiled in external laboratories using this software to genotypes detected at CDC or in other laboratories. The fasta files generated will have a name that begins with the name of the specimen in which the haplotype was detected, followed by a decimal point, and then the name of that haplotype. For example: ***STX10034_25.Nu_Locusname_PART_A_Hap_55.fasta***. This will allow investigators to validate the accuracy of this new haplotype in the future by going back to this specimen and re-sequencing it at this locus if this is deemed necessary or desired. The contents of this fasta file will be a fasta sequence where the header line (the first line) contains the name of the newly discovered haplotype – eg: ***>Nu_Locusname_PART_A_Hap_55.fasta***. The second line contains the sequence of the new haplotype. All newly discovered haplotypes will be written to the following location:

`/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/REF_SEQS/BLASTING/NEW_HAPS`

Haplotype sheets

If the user has indicated that they would like a haplotype data sheet printed (i.e. the user has indicated -H Y) then the workflow will also generate a haplotype sheet. For a full explanation of what a haplotype data sheet is, how it should be interpreted, and how it is formatted, please refer to the following GitHub repository: https://github.com/Joel-Barratt/Eukaryotyping. Essentially, the left most column of this sheet contains the sequencing ID (or Seq_ID); this ID is identical to the filename of each file containing a list of each specimens' genotype. The remaining column headers list the name of each haplotype that has ever been detected since installing the workflow along with those references provided with the workflow. If a particular haplotype was detected for a given specimen, an X will appear in the cell at the intersection of the row containing the specimen name (the Seq_ID) and the column where the header is the name of that haplotype. When the user indicates -H Y, this executes the script "START_haplotype_sheet_generator.R" which can be found in the following directory:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/HAPLOTYPE_CALLER_CYCLO_V2/

This script examines the contents of every list in the SPECIMEN_GENOTYPES directory (see Figure 1) and uses this information to build a haplotype data sheet as described above. The resulting sheet will contain only the specimen names pulled directly from the names of the files present in this folder and the REFERENCE_POPULATION folder (see Figure 1 and below. Also see: Input data for Module 3). The REFERENCE_POPULATION folder is found at this location:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/EUKARYOTYPING/REFERENCE_POPULATION

Users analyzing specimens from the present cyclosporiasis season will wish to include specimens from previous years as part of a reference population, for direct comparison to the

current population. **This reference population will be curated at CDC.** Users can add additional haplotype lists from specimens of interest to this folder at any time and any haplotype lists present in this folder will also be added to the haplotype data sheet generated when the user runs Module 1 when indicating -H Y. Please be aware that the specimens added to this REFERENCE_POPULATION directory must have been genotyped using precisely the same reference haplotypes (i.e. the same version of this software) or the relationships detected will be incorrect and the haplotype sheet will contain errors. Haplotype data sheets will always be printed to the following location:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/haplotype_sheets

Haplotype sheets are tab delimited text files. The file name begins with the date that the file was generated in the format: YYYY-MM-DD, followed by an underscore "_", and then the text "Cyclospora_haplotype_data_sheet.txt". If you generate two haplotype data sheets on the same day, note that your previous one **will be overwritten** so if this is a problem, you may wish to remove the first file from this folder and place it elsewhere. The resulting haplotype data sheets are used as the direct input for the Eukaryotyping ensemble of algorithms included as part of Module 2 of this workflow. Note that when you execute Module 2 (discussed below), the code will automatically analyze the haplotype data sheet possessing the most recent date stamp in its filename. If you wish to analyze a haplotype data sheet from an earlier date, you can do this using the scripts and instructions found here: https://github.com/Joel-Barratt/Eukaryotyping

Module 2 - Input data for distance matrix calculation - Eukaryotyping

Module 2 is simpler than the haplotype calling module in terms of executing the code, and it requires fewer user inputs.

Before running this module, the user must provide three inputs:

1) The directory where you unzipped/installed the bioinformatic workflow – e.g:
   `/Your_home_directory/place_you_extracted_cyclone/`
2) A value for epsilon (between 0 and 1)
3) The number of threads you would like to use to perform the analysis.

To supply these inputs, the user must manually modify the following script:

`/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/MODULE_2_eukaryotyping.sh`

The user must modify this script using their preferred code/text/script. Edit only the lines indicated in the script. Only modify text after the equals "=" sign. There should be no empty spaces after the equal's sign.

First argument (installation directory):

`working_directory=/Your_home_directory/place_you_extracted_cyclone/`

This tells the script that the `CDC_Complete_Cyclospora_typing_workflow_[version]` directory should be found here: `/Your_home_directory/place_you_extracted_cyclone/`

Second argument (number of threads to use):

`number_of_threads=10`

Third argument (value of epsilon):

```
epsilon=0.3072
```

Provide a numeric value between 0 and 1 for epsilon. For a description of how to establish a rational value for epsilon, please refer to the following GitHub repository: https://github.com/Joel-Barratt/Eukaryotyping

Executing Module 2:

Once you have correctly provided the three arguments as described above, execute Module 2 as follows from within the CYCLONE directory in your terminal window:

```
Username$ bash MODULE_2_eukaryotyping.sh
```

The code will automatically import the haplotype data sheet in the `haplotype_sheets` directory that possesses the most recent date stamp at the beginning of its name. For this reason, it is not necessary to specify a specific haplotype sheet as input. The script will examine the haplotype data sheet and exclude markers where there only one haplotype was detected in the current population being analyzed (i.e loci with no diversity). The code will also apply the minimum data availability requirements discussed here and below, excluding all specimens in the haplotype data sheet that do not meet these requirements. After this filtration step is complete, the scripts will run and the progress of the Bayesian algorithms calculation will be printed to the terminal window first. Once this calculation is complete, the terminal window will print the progress for the heuristic algorithm. The heuristic algorithm can take substantially longer to run than the Bayesian algorithm depending on the number of specimens in the most recently generated haplotype data sheet.

Module 2 – Outputs and additional notes

Module 2 will produce a single output; a pairwise matrix of distances calculated using the Barratt-Plucinski Eukaryotyping ensemble of algorithms described here: https://github.com/Joel-Barratt/Eukaryotyping. Distance matrices will be printed to the following directory: **/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/ensemble_matrices**

This file is printed in CSV format and is one of the inputs of Module 3. The file name begins with the date that the file was generated in the format: YYYY-MM-DD, followed by an underscore "_", and then the text "pairwisedistancematrix_ensemble.csv". If you generate two ensemble matrices on the same day, note that your previous one **will be over-written** so if this is a problem, you may wish to remove the first file from this folder and place it elsewhere. This matrix can be clustered for visualization as a tree or dendrogram using external software. However, compliance with the CDCs *Cyclospora* genotyping program requires that the data be analyzed using Module 3 which facilitates detection of genetic clusters using a curated list of internal reference genotypes with known genetic and epidemiologic links.

Minimum data requirements for Module 2:

As described here: https://github.com/Joel-Barratt/Eukaryotyping the "Barratt-Plucinski" ensemble does not require that data be available for every locus in a panel of markers to generate a set of distances. However, minimum data availability requirements must still be met. The R script **import_data_V2.r** can be found within the following location and filters the specimens listed in a haplotype data sheet based on these minimum data requirements:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/EUKARYOTYPING/import_data_V2.r

Line 96 of this script is responsible for excluding specimens with an insufficient number of markers available based on these minimum data requirements. These minimum data requirements were set empirically based on the entropy of each locus and to be considered in the calculation of an ensemble matrix, one or both of the following requirements must be met: (1) specimens must have sequence data available for any three of marker numbers 5 to 8 plus one additional marker, or (2) a specimen must have sequence data available for any five markers. If the panel of markers used for *Cyclospora* genotyping is ever modified in the future, these minimum data requirements will have to be modified accordingly at this line of the code.

Module 3 – Input data for delineating clusters

This module was designed to select the most appropriate number of clusters from a hierarchical tree (a dendrogram) generated from an ensemble distance matrix calculated by Module 2. Heirarchical clustering is a nested clustering approach meaning that all smaller clusters are nested, falling within larger clusters. Therefore, manual (human) assignment of a discrete number of clusters to a hierarchical tree can drastically bias the interpretation of which specimens are related. Under these circumstances, assuming a smaller number of clusters results in a larger number of specimens being considered genetically similar and could culminate in the superficial linkage of specimens that may not be related to the same point of exposure. Alternatively, assuming many discrete clusters may lead to the ultra-fine separation of specimens that are closely related but not identical. Module 3 was developed to reduce these biases. It utilizes an internal set of reference specimens to inform an approximate discrete cluster number that would reflect groups of genetically linked specimens (i.e. specimens of the same strain) that likely come from a common source of exposure. First, these reference specimens must be included in the calculation of an

ensemble matrix alongside the test population (eg. Specimens from the current cyclosporiasis season). Module 3 then explores a range of possible cluster numbers (provided by the user) to determine the cluster number which results in all specimens within each cluster being as similar to each other (on average) as the internal reference set of closely related specimens. In this way, a cluster number is selected based on the distance observed within that specific matrix between a set of reference specimens that are known to be linked genetically and epidemiologically.

Before running this module, the user must provide 6 (or 7) inputs/arguments:

1) The directory where you unzipped/installed this workflow – e.g:
   /Your_home_directory/place_you_extracted_this_workflow/
2) A level of stringency for cluster delineation (must be a value between 1 and 100).
3) The smallest number of clusters to evaluate for fitness (any whole number – **must** be smaller than the largest cluster number of clusters to test).
4) The largest number of clusters to evaluate for fitness (any whole number – **must** be larger than the smallest cluster number to test).
5) A set of reference specimens – these **must** have been included in the calculation of the distance matrix you are about to cluster.
6) A list (a text file) defining which of the reference specimens are closely related to each other.
7) Optional argument – number of threads to use when running this code.

To supply inputs/arguments 1 through 6, the user must manually modify the following script:

/Your_home_directory/place_you_extracted_this_workflow/
CDC_Complete_Cyclospora_typing_workflow_[version]/MODULE_3_clustering.sh

The user must modify this script using their preferred code/text/script editing software by editing the following lines at the beginning of the script (ONLY THESE LINES). Only modify text after the equals "=" sign. There should be no empty spaces after the equal's sign.

First argument:

`cyclone_location=/Your_home_directory/place_you_extracted_cyclone/`

Tells the script that you placed the `CDC_Complete_Cyclospora_typing_workflow_[version]` in this folder: `/Your_home_directory/place_you_extracted_cyclone/`

Second argument:

`stringency=95`

A value between 1 and 100. **For best results a stringency of 95 is recommended**. Only modify text after the equals "=" sign. There should be no empty spaces after the equal's sign. In practice, a stringency level set to 95 means that the most appropriate cluster number will be the one where 95% of within-cluster distances (on average, across all clusters; not the average for each individual cluster) will be less than or equal to the average distance between our known-to-be-linked reference specimens plus three standard deviations.

Third argument:

`cluster_min=5`

Module 3 will begin by testing this number of clusters (5 in the above example) and work its way up to the maximum cluster number designated by the user, until it hits a cluster number that satisfies the requirement provided previously under the heading "Second argument". I currently have this set to start testing at 5 clusters. This **must** be smaller than the largest cluster number to

test (i.e., smaller than **cluster_max**). **For small populations (i.e., less than 500 specimens or so) you should set this to 1.**

Fourth argument:

cluster_max=50

The maximum number of clusters to test when trying to find a cluster number that satisfies the requirement provided previously under the heading **Second argument**. This **must** be larger than the smallest cluster number to test (i.e., larger than **cluster_min**). The module will print on the screen the cluster number currently being investigated until it reaches a cluster number that reaches the most appropriate cluster number, at which point it will notify the user and pring the cluster memberships to the **clusters_detected** directory. If **cluster_max** is reached without an appropriate number of clusters being printed, you should try providing a larger range of cluster numbers (i.e., provide a higher number for **cluster_max**) and then re-run the module.

The fifth requirement:

This module will automatically attempt to cluster the ensemble distance matrix within the **ensemble_matrices** directory with the most recent date stamp in its name. However, for this module to work correctly it is important that prior to calculation of this matrix using module 2, that a set of reference specimens with known genetic and epidemiologic links must have been included in the calculation of this matrix. To ensure these specimens are included in the calculation, the haplotype lists for these specimens **must** be added to the following directory:

/Your_home_directory/place_you_extracted_this_workflow/CDC_Complete_Cyclospora_typing_workflow_[version]/EUKARYOTYPING/REFERENCE_POPULATION

These reference population haplotype lists must have been copied to the **REFERENCE_POPULATION** directory prior to generation of the most recent haplotype data sheet, to ensure that they were included in the calculation of the most recent distance matrix. **Again, this module expects a group of specimens with known epidemiologic links to have been placed in the REFERENCE_POPULATION directory prior to running MODULE 1 and selecting H -Y to ensure that a haplotype sheet containing this reference population was generated. In other words, this reference population of specimens must have been included in the calculation of the most recent ensemble distance matrix for MODULE 3 to be relevant. If no specimens were included as part of this reference population, running this module is a waste of time.**

The sixth requirement:

your_list_of_reference_clusters=2018_gold_clusters.txt

This file must be placed by the user in the following directory **prior to running this module**:

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/REFERENCE_CLUSTER_LIST

This is a tab delimited text file with two columns. The first column lists the sequencing ID (Seq_ID) and the second column provides the cluster membership of that specimen. That cluster must be given an alias (e.g., a name of some kind) containing no spaces (**if spaces/gaps are required please use underscores in the cluster alias name**). The purpose of this text file is to inform MODULE 3 of which specimens represent known links that can be used to infer where to cut the hierarchical tree to delineate a discrete set of clusters. These are essentially a set of "gold standard" specimens that have been linked previously, both epidemiologically and genetically. Note that the specimens in this list should have been included in the calculation of the most recent distance matrix by MODULE 2 or this list will be irrelevant. Please adhere to the following format

– no changes – no exceptions – column headings should be exactly as shown – table should be in delimited format:

Seq_ID	Cluster_alias
C_IA025_18	Vendor_A
C_IA058_18	Vendor_A
C_IA013_18	Vendor_B
C_IA018_18	Vendor_B

Optional seventh argument:

Parts of this code are multi-threaded and can be run in parallel. Just modify the number of threads in the MODULE_3 source code (set to 11 currently).

```
number_of_threads=11
```

Running the code to start Module 3:

Once you have correctly provided the three arguments as described above, execute Module 2 as follows from within the CYCLONE directory in your terminal window:

```
Username$ bash MODULE_3_clustering.sh
```

Module 3 – Outputs and additional notes

The main output is the number of clusters predicted from the current distance matrix; this number will be printed to the terminal window. Additionally, a table will be printed to the following location (in tab delimited text format):

/Your_home_directory/place_you_extracted_cyclone/CDC_Complete_Cyclospora_typing_workflow_[version]/clusters_detected

This text file will contain a date stamp in its name as well as the number of clusters predicted from the most recent matrix at the time this file was generated. The document provides a list of the cluster membership of each specimen in the dataset. This information is important for inferring which specimens share a close genetic relationship, and which are distantly related. As one might expect, if MODULE 3 has run correctly, **most, or all** the specimens within your reference population list should have been assigned to the same cluster number as their epidemiologically linked partners. It is possible that not all specimens in your reference list receive a cluster assignment. This can happen depending on the version of the workflow you are using (i.e., if the requirements for inclusion in downstream clustering analysis change – see minimum data requirements – MODULE 2 – some specimens may not be present for example). However, the vast majority of specimens in your reference list should be assigned to the same cluster if the software is functioning correctly.

END OF DOCUMENT