CHIIMP User Guide

Jesse Connell

2022/04/25

Contents

Introduction

CHIIMP (Computational, High-throughput Individual Identification through Microsatellite Profiling) is a program to analyze microsatellite (short tandem repeat) DNA sequence data, producing genotypes from raw data and automating some typical analysis tasks.

CHIIMP runs as a standalone tool, but is built as an R language package. All functionality of the standalone program can be accessed from functions within R, and the reporting and visualization functions are designed to integrate well with RStudio and R Markdown.

This document mostly focuses on CHIIMP as a standalone tool. For more information on the use of specific functions within R, also see the built-in package documentation.

Installation

First install R and RStudio, which will supply most software dependencies for CHIIMP. Once these are installed, follow the specific instructions below for your operating system. In all three cases CHIIMP performs an analysis when a configuration file is dragged and dropped onto the desktop icon; there is no interactive interface via the icon, though the R package can be used interactively. See the Usage section for more information.

Windows

On Windows, double-click the `install_windows.cmd` script. This will install the package and R dependencies, and create a desktop shortcut.

Mac OS

On Mac OS, right-click (control+click) the `install_mac.command` shell script, select "Open," and also click "Open" in the window that appears to confirm that really do want to open it. (Apple has specific instructions about these security precautions here.) This will automatically install the package along with R dependencies and create a desktop alias.

If a window appears recommending installation of the Mac OS command-line developer tools, go ahead and install them. After that you'll probably need to re-run the CHIIMP installer again to finish the install.

Linux

On Linux, run the `install_linux.sh` shell script to automatically install the package along with R dependencies. An icon for the program is created at `$HOME/Desktop/CHIIMP.desktop`. Specific usage of the desktop icon will depend on the desktop environment in use. (The `CHIIMP.desktop` text file references the installed chiimp executable, and supplies the config file as a command-line argument when dragged and dropped onto the icon.)

Input Data Organization

The information CHIIMP uses during analysis is:

- Sequence files containing complete microsatellite sequences (FASTA or FASTQ, plain text or gzip-compressed)
- Spreadsheet of dataset sample attributes
- Spreadsheet of locus attributes
- Spreadsheet of known individuals (optional)
- Spreadsheet of named alleles (optional)

The spreadsheets are in comma-separated (CSV) format. Column names are important but not column order. Extra columns are imported as-is but are otherwise ignored.

Sequence Files

The sequence files must contain sequences that span complete microsatellites. No assembly is performed to handle fragments of microsatellites, and the lengths of sequences identified as alleles are reported as-is. (An implicit assumption throughout the analysis is that any candidate allele sequence begins and ends with conserved regions corresponding to the PCR primers used, and the forward primer sequence is one of the filtering criteria during analysis.)

Dataset Sample Attributes

The description of the samples to be analyzed can be provided in a spreadsheet, or automatically loaded from the data file names. An example spreadsheet:

Filename	Replicate	Sample	Locus
100-1-A.fastq.gz	1	100	A
100-2-A.fastq.gz	2	100	A
100-1-B.fastq.gz	1	100	B
100-2-B.fastq.gz	2	100	B
100-1-1.fastq.gz	1	100	1
100-2-1.fastq.gz	2	100	1
100-1-2.fastq.gz	1	100	2
100-2-2.fastq.gz	2	100	2
101-1-A.fastq.gz	1	101	A
101-2-A.fastq.gz	2	101	A
101-1-B.fastq.gz	1	101	B
101-2-B.fastq.gz	2	101	B
101-1-1.fastq.gz	1	101	1
101-2-1.fastq.gz	2	101	1
101-1-2.fastq.gz	1	101	2
101-2-2.fastq.gz	2	101	2

These columns are required for each entry:

- Filename: The name of the sequence file to analyze. Note that if samples were multiplexed on the sequencer by pooling PCR products for multiple loci, filenames can be repeated here; just vary the text in the Locus column across rows. The analysis will use the forward PCR primer (see Locus Attributes below) to select just the sequences matching each locus as needed.
- Replicate: An identifier for a repeated case of the same biological sample. If not applicable, the column may be left blank, but is still required.
- Sample: An identifier for a particular biological sample.
- Locus: The identifier for the locus being genotyped with. These must match the identifiers in Locus Attributes below.

For simple cases that have a one-to-one match between sequence files and sample/locus combinations, and with descriptive filenames following a consistent pattern, the dataset table can be created automatically at run-time. See the Usage section for more information.

Locus Attributes

The description of the loci should be given in a spreadsheet with loci on rows and attributes on columns. For example:

Locus	LengthMin	LengthMax	LengthBuffer	Motif	Primer	ReversePrimer
A	131	179	20	TAGA	TATCACTGGTGT...	CACAGTTGTGTG...
B	194	235	20	TAGA	AGTCTCTCTTTC...	TAGGAGCCTGTG...
1	232	270	20	TATC	ACAGTCAAGAAT...	CTGTGGCTCAAA...
2	218	337	20	TCCA	TTGTCTCCCCAG...	TCTGTCATAAAC...

These columns are required:

- Locus: A short unique identifier.
- LengthMin: The minimum expected sequence length in bases.
- LengthMax: The maximum expected sequence length in bases.
- LengthBuffer: An additional length below LengthMin or above LengthMax to accept for candidate allele sequences. (If the length range of alleles for a given locus is uncertain or unknown, this may be set very high to effectively disable the length range requirement.)
- Motif: The short sequence repeating in tandem.
- Primer: The forward PCR primer used in preparing the sequencing library. This is used as one of the checks for candidate allele sequences.
- ReversePrimer: The reverse PCR primer used in preparing the sequencing library. This is not currently used unless `use_reverse_primers` is enabled in the configuration.

Known Individuals (Optional)

If a spreadsheet of genotypes for known individuals is supplied, the analysis can attempt to match samples with the known genotypes automatically. For example:

Name	Locus	Allele1Seq	Allele2Seq
CH001	A	ATTATCACTGG...	ATTATCACTGG...
CH001	B	TCAGTCTCTCT...	
CH001	1	AGACAGTCAAG...	AGACAGTCAAG...
CH001	2	CTTTGTCTCCC...	CTTTGTCTCCC...
CH002	A	ATTATCACTGG...	ATTATCACTGG...
CH002	B	TCAGTCTCTCT...	TCAGTCTCTCT...
CH002	1	AGACAGTCAAG...	
CH002	2	CTTTGTCTCCC...	CTTTGTCTCCC...

The order of the alleles given is not important, and homozygous individuals may have Allele2Seq either left blank or set to a copy of Allele1Seq. The sequences should contain any conserved region before and after the repeats including that used for the PCR primers described above.

Named Alleles (Optional)

If a spreadsheet of allele names and sequences is supplied, the analysis will use those names in summary tables in the output report. For example:

Locus	Name	Seq
A	200-a	ATTATCACTGG...
A	180-a	ATTATCACTGG...
A	180-b	ATTATCACTGG...
B	300-a	ATTATCACTGG...
B	305-a	ATTATCACTGG...
B	290-a	ATTATCACTGG...

The software will automatically create short allele names for any identified allele not listed in the allele spreadsheet (or for all alleles if no spreadsheet is given).

The automatic names are the sequence length and a sequence-specific suffix separated by a hyphen, for example, “180-fdd1c6” for a 180 bp sequence with no assigned name and particular sequence content. Any other 180 bp sequence would receive a different suffix when the name is assigned.

Algorithm

CHIIMP breaks the genotyping process into two parts. First a sample file is de-replicated and a table of unique sequences is created, with no filtering yet applied. Second the table is filtered to just candidate allele sequences, and up to two sequences are reported as the genotype. Both the per-sequence table and the final genotypes are saved in the final output, as spreadsheets in the `processed-files` directory and as the `summary.csv` spreadsheet.

Sample Processing

The table of unique sequences includes basic information for each case: sequence content, length, and read counts observed. These are the Seq, Count, and Length columns. The sequences are ordered by count with the most abundant first.

Additional columns associate each sequence with a particular locus, using the locus attributes described above. First each locus' forward primer is compared with the sequence and the matching locus name is stored in a MatchingLocus column. The sequence is then checked for several tandem repeats of the motif for that locus, and compared to the length range expected for that locus. TRUE/FALSE values for these are stored in MotifMatch and LengthMatch columns respectively. The Ambiguous column marks any sequences containing bases outside of A, C, T, and G (such as N).

PCR artifacts can obscure real allele sequences with incorrect sequences. There are extra filters to attempt to remove these if possible or highlight cases that may require further attention.

The sample data tables include "Stutter" and "Artifact" columns to mark entries that look like possible polymerase stutter or other artifacts of another sequence present at higher counts. For cases of potential polymerase stutter, the higher-count sequence is one motif repeat longer. For insertion/deletion/substitution artifacts the higher-count sequence is within 1 bp of the same length. In both cases the supposed artifact sequence will be marked if the read counts are 1/3 or lower than the higher-count sequence. (This represents a trade-off in sensitivity and specificity since genuine allele sequences may differ in length by one or even zero repeats, and read counts for pairs of alleles in a given sample can vary considerably.) Both of these columns store row numbers for the higher-count sequence that an artifact may have originated in, if found. Note that relative sequence lengths and counts determine the outcome here, since sequence content for the artifacts is largely indistinguishable from real allele sequences.

Lastly, the ratio of read counts for each sequence to the total reads in the sample and the reads with the same MatchingLocus value is stored in FractionOfTotal and FractionOfLocus columns respectively.

This is the `analyze_seqs` function in the R package.

Genotype Calling

In the previous stage every single unique sequence for each data file was described in a table, but no filtering or genotyping occurred. Now just one or two candidate allele sequences are extracted from each table and reported as the genotype.

First, the table rows are restricted to just those matching the expected locus' primer, motif, and length range (using the MatchingLocus, MotifMatch, and LengthMatch columns). If the resulting total read count is below a minimum value (by default , customizable via the `sample_summary: counts.min` setting) no genotyping will be attempted. Next only those sequences accounting for at least a minimum fraction of the remaining reads are considered. (The default value is . This can be changed via the `sample_analysis: fraction.min` setting.) Sequences that are marked as potential stutter or other artifacts (via the Stutter and Artifact columns of the table) or contain ambiguous sequence content (via the Ambiguous column) are excluded next.

After these filters are applied, the top one or two remaining sequences are labeled as the alleles. (If only one sequence remains, the sample is labeled homozygous; if two or more, heterozygous.) The final details kept for each sample are:

- the sequence content, length, and counts for the one or two alleles
- the zygosity of the sample
- whether the ambiguous-sequence filter removed a potential allele
- whether the stutter and/or artifact filter removed a potential allele
- The read counts of the entire sample before any filtering
- The read counts of just those sequences matching the locus primer, motif, and length range

These tasks (the filtering and categorizing of each sequence in the table and the short genotype summary) are the `analyze_sample` and `summarize_sample` functions in the R package.

Summary and Reporting

The genotype and details identified in the previous step for each sample are aggregated into a spreadsheet with a row for each sample. This summary spreadsheet and the more detailed per-file and per-sample tables are all saved in the final output.

For inter-sample comparisons, the alleles identified across samples for each locus are aligned to one another. The genotypes for each sample are clustered by number of matching alleles, showing similarity between samples. If a spreadsheet of known genotypes was given, the sample genotypes are also compared to the known genotypes, with any close matches reported. If a Name column was provided with the sample definition table as well as a known genotypes spreadsheet, the known-correct genotypes will be paired with applicable samples and a column tracking the result of the genotyping (Correct, Incorrect, Blank, or Dropped Allele) will be added. A single report document summarizes the genotyping and these other details. See the Output Data Organization section below for more information on the output.

These steps are handled by the `full_analysis` function in the R package.

Usage

CHIIMP takes a configuration file as input and saves all output to a folder. The configuration file points to all of the input data described above, and specifies options for the analysis and output. All options have defaults, so the file may be very brief or even empty. The file format is YAML, with a simple text layout using nested lists.

For example, a configuration file might have just two entries, showing the spreadsheets to use for the samples and loci to analyze:

```
fp_dataset: samples.csv
fp_locus_attrs: locus_attrs.csv
```

The configuration file can be dragged and dropped onto the desktop icon created during installation.

For command-line usage, the configuration file can be given as the first argument to the R script installed with the package. (The location of the script can be shown in R with `system.file("bin", "chiimp", package="chiimp")`.) To run the same analysis within R, pass a list of configuration options to the `chiimp::full_analysis()` function.

Example Configuration File

The text in the example configuration file included here shows a slightly more complex case:

```
---
# This is an example configuration file for a CHIIMP analysis.  These lines
# starting with a "#" are comments.  See the below lines with a keyword
# followed by a colon for example settings.  Sub-sections are indented with two
# spaces.

# "dataset_opts" defines options related to the input data.
#   "dp" defines the directory containing sequence files.
#   "pattern" defines the how the Replicate, Sample, and Locus fields are
#   positioned within the file names.
#   An example file name matching this pattern:
#     "Replicate1-Sample30-A.fastq.gz"
dataset_opts:
  dp: str-dataset
  pattern: Replicate(\d+)-Sample(\d+)-([A-Za-z0-9]+)
# "dataset_analysis" defines options passed to analyze_dataset() by
# full_analysis().  Here we'll just ensure that any example analysis run with
# this configuration is single-threaded.
dataset_analysis:
  ncores: 1
# "output" defines options related to analysis output.
#   "dp" defines the directory that will contain all output files.
output:
  dp: str-results
# This is the location of the spreadsheet defining locus attributes (lengths,
# primers, etc.)  See example_locus_attrs.csv for an example.
fp_locus_attrs: locus_attrs.csv

# There are many more options available than are shown here.  For the full list
# and all default values, see R/chiimp.R.  This file format is YAML
# (http://yaml.org/).
```

Common Options

Below is a list of commonly-customized options. Nested lists imply nested options in the configuration file; see the "Example Configuration File" section above for more information. (See also the end of this document for a full list with all default settings.) For more information on the format of the spreadsheets listed here, see the "Input Data Organization" section above.

- `fp_dataset`: file path to table of sample attributes to use (rather than detecting sample attributes via dataset_opts)
- `fp_locus_attrs`: file path to locus attributes CSV file
- `fp_allele_names`: file path to known alleles CSV file
- `fp_genotypes_known`: file path to known genotypes CSV file
- `dataset_opts`: Options related to automatically detecting sample attributes for a dataset (rather than loading a spreadsheet via fp_dataset)
  - `dp`: directory path to input sequence files
  - `pattern`: regular expression for the input filename pattern
  - `ord`: order of fields in the input filename pattern
- `output`: Options related to how program output is saved
  - `dp`: directory path for saving output data

Output Data Organization

A the end of an analysis CHIIMP creates a directory of files with all results.

- `summary.csv`: spreadsheet of the called genotypes and additional attributes for each sample. Each sample is on a separate row, and each column corresponds to a separate attribute in the results. This includes all columns in the input dataset spreadsheet including locus, replicate, and sample identifiers, the sequences, sequence lengths, and counts of the identified allele(s), and several additional attributes.
- `processed-files`: directory of spreadsheets for each input data file. Each spreadsheet contains one unique sequence per row with attributes on columns. At this stage no filtering for sample/locus-specific attributes has been applied. (This is particularly relevant for sequencer-multiplexed samples as one input data file may contain data for multiple samples.)
- `processed-samples`: directory of spreadsheets for each sample. As for `processed-files`, each spreadsheet contains one unique sequence per row with attributes on columns. These represent the intermediate sample-specific data CHIIMP uses to call a genotype for each sample, and each spreadsheet here corresponds to a single row in the `summary.csv` file.
- `histograms`: directory of counts-versus-length histograms for each sample. Counts are tallied on a by-sequence basis rather than by-length for alleles, so the bars for called alleles (in red) are generally shorter than the bars for unfiltered sequences (in black) or the matching-locus sequences (in pink).
- `allele-sequences`: directory of FASTA files for each sample, giving just the sequence content also shown in `summary.csv`. (This is a convenience feature to make the called alleles easily usable in a standard format, but the same information is available in `summary.csv`.)
- `alignments`: directory of FASTA files for each locus, giving a multiple alignment of all identified alleles per locus.
- `alignment-images`: directory of visualization images of the per-locus alignments. These are also included in the report document.
- `report.html`: report document summarizing the genotyping results, inter-sample comparisons, and (if known genotypes were provided), a comparison of samples with known individuals.

An additional file will be created if `fp_rds` is defined in the `output` setting of the configuration. This file contains all analysis results in a single R object using R's native data serialization format for easy post-analysis in R if desired.

These directory and file names are customizable in the `output` section of the configuration.

Full Configuration Options List

Configuration options list for CHIIMP version 0.4.1.

- `fp_dataset`: `NULL`
- `fp_locus_attrs`: `locus_attrs.csv`
- `fp_allele_names`: `NULL`
- `fp_genotypes_known`: `NULL`
- `dataset_opts`
  - `dp`: `str-data`
  - `pattern`: `(\d+)-(\d+)-([A-Za-z0-9]+).fast[aq](?:\.gz)`
  - `ord`: `1 2 3`
  - `autorep`: `FALSE`
- `output`
  - `dp`: `str-results`
  - `fp_summary`: `summary.csv`
  - `fp_report`: `report.html`
  - `fp_dist_mat`: `sample-distances.csv`
  - `fp_rds`: `NULL`
  - `dp_histograms`: `histograms`

  - dp_alignments: alignments
  - dp_alignment_images: alignment-images
  - dp_processed_files: processed-files
  - dp_processed_samples: processed-samples
  - dp_allele_seqs: allele-sequences
- dataset_analysis
  - ncores: 0
  - name_args
    - hash_len: 6
- seq_analysis
  - nrepeats: 3
  - stutter.count.ratio_max: 0.333333333333333
  - artifact.count.ratio_max: 0.333333333333333
  - use_reverse_primers: FALSE
  - reverse_primer_r1: TRUE
- sample_analysis_func: analyze_sample
- sample_analysis_opts
  - fraction.min: 0.05
- sample_summary_func: summarize_sample
- sample_summary_opts
  - counts.min: 500
- report: TRUE
- report.echo: FALSE
- report.title: Microsatellite Report
- report.author: NULL
- report.locus_chunks: NULL
- report.group_samples: FALSE
- report.na.replicates: ‘‘
- report.na.alleles: ‘‘
- report.dist_range: 2
- report.dist_max: 3
- report.sections
  - genotypes: TRUE
  - identifications: TRUE
  - distances: TRUE
  - flags: TRUE
  - alignments: TRUE
  - contamination: TRUE
- verbose: TRUE