# CHIIMP User Guide

*Jesse Connell*

*2018/02/19*

## 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.

### Installation

Most dependencies are provided by installation of R and RStudio. Once these are installed, follow the specific instructions below for your operating system.

### Windows

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

### Linux

On Linux, run the `install_linux.sh` shell script to automatically install the package along with R dependencies. A symbolic link to the program is created at `$HOME/Desktop/chiimp`.

### Mac OS

## 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 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 | 1 | 100 | A |
| 100-2-A.fastq | 2 | 100 | A |
| 100-1-B.fastq | 1 | 100 | B |
| 100-2-B.fastq | 2 | 100 | B |
| 100-1-1.fastq | 1 | 100 | 1 |
| 100-2-1.fastq | 2 | 100 | 1 |
| 100-1-2.fastq | 1 | 100 | 2 |
| 100-2-2.fastq | 2 | 100 | 2 |
| 101-1-A.fastq | 1 | 101 | A |
| 101-2-A.fastq | 2 | 101 | A |

| Filename | Replicate | Sample | Locus |
|---|---|---|---|
| 101-1-B.fastq | 1 | 101 | B |
| 101-2-B.fastq | 2 | 101 | B |
| 101-1-1.fastq | 1 | 101 | 1 |
| 101-2-1.fastq | 2 | 101 | 1 |
| 101-1-2.fastq | 1 | 101 | 2 |
| 101-2-2.fastq | 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.

### 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.

## 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 shortcut 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]+)
# "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 inforamtion. (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_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-samples`: directory of spreadsheets for each sample. Each spreadsheet contains one unique sequence per row with attributes on columns. These represent the intermediate 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, 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.

## Walkthrough

## Full Configuration Options List

- fp_dataset:
- fp_locus_attrs: locus_attrs.csv
- fp_allele_names:
- fp_genotypes_known:
- dataset_opts
  - dp: str-data
  - pattern: (+)-(+)-([A-Za-z0-9]+).fastaq
  - 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:
  - dp_histograms: histograms
  - dp_alignments: alignments
  - dp_alignment_images: alignment-images
  - dp_processed_samples: processed-samples
  - dp_allele_seqs: allele-sequences
- dataset_analysis
  - ncores: 0
- sample_analysis
  - nrepeats: 3
- sample_summary_func: summarize_sample
- sample_summary
  - fraction.min: 0.05
  - counts.min: 500
- report: TRUE
- report.echo: FALSE
- report.title: Microsatellite Report
- report.author:
- report.hash_len: 6
- report.locus_chunks:
- report.group_samples: FALSE
- report.na.replicates:
- report.dist_range: 2
- report.dist_max: 3
- report.sections
  - genotypes: TRUE
  - identifications: TRUE
  - distances: TRUE
  - flags: TRUE
  - alignments: TRUE
  - contamination: TRUE
- verbose: TRUE