# CIAlign - Clean and Interpret Alignments

Charlotte Tumescheit, Dr. Katherine Brown, Dr. Andrew E. Firth

CIAlign is a command line tool which performs various functions to clean and analyse a multiple sequence alignment (MSA).

The tool is designed to be highly customisable, allowing users to specify exactly which functions to run and which settings to use. It is also transparent, generating a clear log file and alignment markup showing exactly how the alignment has changed and what has been removed by which function.

This allows the user to:

- Remove sources of noise from their MSA
  - Remove insertions which are not present in the majority of sequences
  - Remove sequences below a threshold number of bases or amino acids
  - Crop poorly aligned sequence ends
  - Remove columns containing only gaps
  - Remove sequences above a threshold level percentage of divergence from the majority
- Generate consensus sequences
- Visualise alignments
  - Generate image files showing the alignment before and after using CIAlign cleaning functions and showing which columns and rows have been removed
  - Draw sequence logos
  - Visualise coverage at each position in the alignment
- Analyse alignment statistics
  - Generate a similarity matrix showing the percentage identity between each sequence pair
- Unalign the alignment
- Replace U's by T's

## Requirements

- python >= 3.6
- matplotlib >= 2.1.1
- numpy >= 1.16.3
- scipy >= 1.3.0

## Installation

The easiest way to install CIAlign is using pip3:

```
pip3 install cialign
```

The current release of CIAlign can also be downloaded directly using this link,

If you download the package directly, you will also need to add the CIAlign directory to your PATH environment variable as described here

## Usage

### Basic Usage

```
CIAlign --infile INFILE --outfile_stem STEM --inifile my_config.ini
```

**Parameters** Parameters can be specified in the command line or in a config file using the naming system below.

A template config file is provided in CIAlign/templates/ini_template.txt - edit this file and provide the path to the –inifile argument. If this argument is not provided command line arguments and defaults will be used.

Parameters passed in the command line will take precedence over config file parameters, which take precedence over defaults.

Command help can be accessed by typing `CIAlign --help`

| Parameter | Description | Default |
|---|---|---|
| `--infile` | Path to input alignment file in FASTA format | None |
| `--inifile` | Path to config file | None |
| `--outfile_stem` | Prefix for output files, including the path to the output directory | CIAlign |
| `--silent` | Do not print progress to the screen | False |
| `--all` | Use all available functions with default parameters | False |
| `--help` | Show all available parameters with an explanation | None |
| `--version` | Show the version | None |

Beside these main parameters, the use of every function and corresponding thresholds can be specified by the user by adding parameters to the command line or by setting them in the configuration file. Available functions and their parameters will be specified in the following section.

CIAlign always produces a log file, specifying which functions have been run with witch parameters and what has been removed. It also outputs a file that only specifies what has been removed with the original column positions and the sequence names.

Output files:

- **`OUTFILE_STEM_log.txt`** - general log file
- **`OUTFILE_STEM_removed.txt`** - removed columns positions and sequence names text file

## Cleaning an MSA

Each of these steps will be performed sequentially in the order specified in the table below.

The "cleaned" alignment after all steps have been performed will be saved as **`OUTFILE_STEM_cleaned.fasta`**

remove_divergent, remove_insertions and crop_ends require three or more sequences in the alignment, remove_short and remove_gap_only require two or more sequences.

| Parameter | Description | Default Value | Min | Max |
|---|---|---|---|---|
| **`--remove_divergent`** | Remove sequences with <= N proportion of positions at which the most common base / amino acid in the alignment is present | False | NA | NA |
| *`--remove_divergent_minperc`* | Minimum proportion of positions which should be identical to the most common base / amino acid in order to be preserved | 0.65 | 0 | 1 |
| **`--remove_insertions`** | Remove insertions found in <= 50% of sequences from the alignment | False | NA | NA |
| *`--insertion_min_size`* | Only remove insertions >= this number of residues | 3 | 1 | n_col |
| *`--insertion_max_size`* | Only remove insertions <= this number of residues | 200 | 1 | 1000 |
| *`--insertion_min_flank`* | Minimum number of bases on either side of an insertion to classify it as an insertion | 5 | 0 | n_col/2 |
| **`--crop_ends`** | Crop the ends of sequences if they are poorly aligned | False | NA | NA |

| Parameter | Description | Default Value | Min | Max |
|---|---|---|---|---|
| *--crop_ends_mingap_perc* | Minimum proportion of the sequence length (excluding gaps) that is the threshold for change in gap numbers. | 0.05 | 0 | 0.5 |
| *--crop_ends_redefine_perc* | Proportion of the sequence length (excluding gaps) that is being checked for change in gap numbers to redefine start/end. | 0.1 | 0 | 0.5 |
| **--remove_short** | Remove sequences <= N bases / amino acids from the alignment | False | NA | NA |
| *--remove_min_length* | Sequences are removed if they are shorter than this minimum length, excluding gaps. | 50 | 0 | n_col |
| **--keep_gaponly** | Keep gap only columns in the alignment | False | NA | NA |

Note: if the sequences are short (e.g. < 100), a low crop_ends_mingap_perc (e.g. 0.01) will result in a change of gap numbers that is too low (e.g. 0). If this happens, the change in gap numbers will be set to 2 and a warning will be printed.

## Generating a Consensus Sequence

This step generates a consensus sequence based on the cleaned alignment. If no cleaning functions are performed, the consensus will be based on the input alignment. For the "majority" based consensus sequences, where the two most frequent characters are equally common a random character is selected.

Output files:

- **OUTFILE_STEM_consensus.fasta** - the consensus sequence only
- **OUTFILE_STEM_with_consensus.fasta** - the cleaned alignment plus the consensus

| Parameter | Description | Default |
|---|---|---|
| **--make_consensus** | Make a consensus sequence based on the cleaned alignment | False |
| *--consensus_type* | Type of consensus sequence to make - can be majority, to use the most common character at each position in the consensus, even if this is a gap, or majority_nongap, to use the most common non-gap character at each position | majority |
| *--consensus_keep_gaps* | If there are gaps in the consensus (if majority_nongap is used as consensus_type), should these be included in the consensus (True) or should this position in the consensus be deleted (False) | False |
| *--consensus_name* | Name to use for the consensus sequence in the output fasta file | consensus |

## Unaligning the Alignment

This function simply removes the gaps from the input or output alignment and creates and unaligned file of the sequences.

Output files:

- **OUTFILE_STEM_unaligned_input.fasta** - unaligned sequences of input alignment
- **OUTFILE_STEM_unaligned_output.fasta** - unaligned sequences of output alignment

| Parameter | Description | Default |
|---|---|---|
| --unalign_input | Generates a copy of the input alignment with no gaps | False |
| --unalign_output | Generates a copy of the output alignment with no gaps | False |

## Replacing U's by T's

This function replaces the U nucleotides by T nucleotides without disturbing the sequence names.

Output files:

- **OUTFILE_STEM_T_input.fasta** - input alignment with T's instead of U's
- **OUTFILE_STEM_T_output.fasta** - output alignment with T's instead of U's

| Parameter | Description | Default |
|---|---|---|
| --replace_input | Generates a copy of the input alignment with T's instead of U's | False |
| --replace_output | Generates a copy of the output alignment with T's instead of U's | False |

## Visualising Alignments

Each of these functions produces some kind of visualisation of your alignment.

### Mini Alignments

These functions produce "mini alignments" - images showing a small representation of your whole alignment, so that gaps and poorly aligned regions are clearly visible.

Output files:

- **OUTFILE_STEM_input.png (or svg, tiff, jpg)** - the input alignment
- **OUTFILE_STEM_output.png (or svg, tiff, jpg)** - the cleaned output alignment
- **OUTFILE_STEM_markup.png (or svg, tiff, jpg)** - the input alignment with deleted rows and columns marked

| Parameter | Description | Default |
|---|---|---|
| **--plot_input** | Plot a mini alignment - an image representing the input alignment | False |
| **--plot_output** | Plot a mini alignment - an image representing the output alignment | False |
| **--plot_markup** | Draws the input alignment but with the columns and rows which have been removed by each function marked up in corresponding colours | False |
| *--plot_dpi* | DPI for mini alignments | 300 |
| *--plot_format* | Image format for mini alignments - can be png, svg, tiff or jpg | png |
| *--plot_width* | Mini alignment width in inches | 5 |
| *--plot_height* | Mini alignment height in inches | 3 |

### Sequence logos

These functions draw sequence logos representing your output (cleaned) alignment. If no cleaning functions are specified, the logo will be based on your input alignment.

Output_files:

- **OUTFILE_STEM_logo_bar.png (or svg, tiff, jpg)** - the alignment represented as a bar chart
- **OUTFILE_STEM_logo_text.png (or svg, tiff, jpg)** - the alignment represented as a standard sequence logo using text

| Parameter | Description | Default |
|---|---|---|
| **--make_sequence_logo** | Draw a sequence logo | False |
| *--sequence_logo_type* | Type of sequence logo - bar/text/both | bar |
| *--sequence_logo_dpi* | DPI for sequence logo | 300 |
| *--sequence_logo_font* | Font (see NB below) for bases / amino acids in a text based sequence logo | monospace |
| *--sequence_logo_nt_per_row* | Number of bases / amino acids to show per row in the sequence logo, where the logo is too large to show on a single line | 50 |
| *--sequence_logo_filetype* | Image file type to use for the sequence logo - can be png, svg, tiff or jpg | png |

NB: to see available fonts on your system, run CIAlign –list_fonts_only and view CIAlign_fonts.png

### Coverage Plots

This function plots the number of non-gap residues at each position in the alignment.

Output file:

- `OUTFILE_STEM_input_coverage.png (or svg, tiff, jpg)` - image showing the input alignment coverage
- `OUTFILE_STEM_output_coverage.png (or svg, tiff, jpg)` - image showing the output alignment coverage

| Parameter | Description | Default |
|---|---|---|
| `--plot_coverage_input` | Plot the coverage of the input MSA | False |
| `--plot_coverage_output` | Plot the coverage of the output MSA | False |
| *`--plot_coverage_dpi`* | DPI for coverage plot | 300 |
| *`--plot_coverage_height`* | Height for coverage plot (inches) | 3 |
| *`--plot_coverage_width`* | Width for coverage plot (inches) | 5 |
| *`--plot_coverage_colour`* | Colour for coverage plot (hex code or name) | #007bf5 |
| *`--plot_coverage_filetype`* | File type for coverage plot (png, svg, tiff, jpg) | png |

## Analysing Alignment Statistics

These functions provide additional analyses you may wish to perform on your alignment.

### Similarity Matrices

Generates a matrix showing the proportion of identical bases / amino acids between each pair of sequences in the MSA.

Output file:

- `OUTFILE_STEM_input_similarity.tsv` - similarity matrix for the input file
- `OUTFILE_STEM_output_similarity.tsv` - similarity matrix for the output file

| Parameter | Description | Default |
|---|---|---|
| `--make_similarity_matrix_input` | Make a similarity matrix for the input alignment | False |
| `--make_similarity_matrix_output` | Make a similarity matrix for the output alignment | False |
| *`--make_simmatrix_keepgaps`* | 0 - exclude positions which are gaps in either or both sequences from similarity calculations, 1 - exclude positions which are gaps in both sequences, 2 - include all positions | 0 |
| *`--make_simmatrix_dp`* | Number of decimal places to display in the similarity matrix output file | 4 |
| *`--make_simmatrix_minoverlap`* | Minimum overlap between two sequences to have non-zero similarity in the similarity matrix | 1 |