

CIAAlign - Clean and Interpret Alignments

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

Introduction

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

1. Citation
2. Functionality
3. Mailing List

Citation

If you found CIAAlign useful, please cite:

Tumescheit C, Firth AE, Brown K. 2022. CIAAlign: A highly customisable command line tool to clean, interpret and visualise multiple sequence alignments. PeerJ 10:e12983 <https://doi.org/10.7717/peerj.12983>

Functionality

CIAAlign allows the user to:

- Cleaning functions - remove sources of noise from an MSA
 - Remove sequences above a threshold level percentage of divergence from the majority
 - Remove insertions which are not present in the majority of sequences
 - Crop poorly aligned sequence ends
 - Remove short sequences below a threshold number of bases or amino acids
 - Remove columns containing only gaps
 - Remove either end of an alignment where columns don't meet a minimum identity threshold and coverage level
- Generate consensus sequences
- Visualise alignments
 - Generate image files showing the alignment before and after using CIAAlign cleaning functions and showing which columns and rows have been removed
 - Draw sequence logos
 - Visualise coverage and conservation at each position in the alignment
 - Generate position frequency, position probability and position weight matrices based on the alignment and produce output formatted to be used as input for the BLAMM and MEME motif analysis tools.
- Analyse alignment statistics
 - Generate a similarity matrix showing the percentage identity between each sequence pair
- Make changes to the alignment
 - Extract a section of the alignment
 - Unalign the alignment
 - Replace U with T, or T with U in a nucleotide alignment

CIAAlign 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.

Mailing List

Sign up here for updates when a new feature is added to CIAAlign ## Installation ### Requirements * python >= 3.6 * matplotlib >= 2.1.1 * numpy >= 1.16.3 * scipy >= 1.3.0

Recommendations

The easiest way to install CIAAlign is using conda or pip3.

Conda

```
conda install -c bioconda cialign
```

[link](#)

pip3

```
pip3 install cialign
```

[link](#)

Download

The current release of CIAAlign can also be downloaded directly using this [link](#),

If you download the package directly, you will also need to add the CIAAlign directory to your PATH environment variable as described here## Usage ### Basic Usage CIAAlign --infile INFILE --outfile_stem STEM --infile 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 CIAAlign/templates/ini_template.ini - edit this file and provide the path to the -infile 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 CIAAlign --help

Parameter	Description	Default
--infile	Path to input alignment file in FASTA format	None
--infile	Path to config file	None
--outfile_stem	Prefix for output files, including the path to the output directory	CIAAlign
--all	Use all available functions with default parameters. Does not currently include crop_divergent	False
--clean	Use all available cleaning functions (except crop_divergent) with default parameters	False
--visualise	Use all available mini alignment visualisation functions with default parameters	False
--interpret	Use all available interpretation functions (except sequence logos) with default parameters	False
--silent	Do not print progress to the screen	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.

CIAAlign always produces a log file, specifying which functions have been run with which 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 (if specified) 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`, `crop_ends` and `crop_divergent` require three or more sequences in the alignment, `remove_short` and `remove_gap_only` require two or more sequences.

The **retain** functions allow the user to specify sequences to keep regardless of the CIAAlign results.

Remove Divergent Removes divergent sequences from the alignment - sequences with \leq `remove_divergent_minperc` positions at which the most common residue in the alignment is present

Remove Divergent

Parameter	Description	Default Value	Min	Max
--remove_divergent	Remove sequences with \leq <code>remove_divergent_minperc</code> 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_divergent_retain	Do not remove sequences with this name when running the remove divergent function	None	NA	NA
--remove_divergent_retain_char	Do not remove sequences with names containing this character string when running the remove divergent function	None	NA	NA
--remove_divergent_retain_list	Do not remove sequences with names listed in this file when running the remove divergent function	None	NA	NA

Remove Insertions Removes insertions from the alignment which are found in \leq `insertion_min_perc` of the sequences.

Remove Insertions

Parameter	Description	Default Value	Min	Max
--remove_insertions	Remove insertions found in \leq <code>insertion_min_perc</code> of sequences from the alignment	False	NA	NA
--insertion_min_size	Only remove insertions \geq this number of residues	3	1	n_col
--insertion_max_size	Only remove insertions \leq 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

Parameter	Description	Default Value	Min	Max
<code>--insertion_min_perc</code>	Remove insertions which are present in less than this proportion of sequences	0.5	0	1

Crop Ends Crops the ends of individual sequences if they contain a high proportion of gaps relative to the rest of the alignment.

Crop Ends

Parameter	Description	Default Value	Min	Max
<code>--crop_ends</code>	Crop the ends of sequences if they are poorly aligned	False	NA	NA
<code>--crop_ends_mingap_perc</code>	Minimum proportion of the sequence length (excluding gaps) that is the threshold for change in gap numbers.	0.05	0	0.6
<code>--crop_ends_redefine_perc</code>	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
<code>--crop_ends_retain</code>	Do not crop sequences with this name when running the crop ends function	None	NA	NA
<code>--crop_ends_retain_str</code>	Do not crop sequences with names containing this character string when running the crop ends function	None	NA	NA
<code>--crop_ends_retain_list</code>	Do not crop sequences with names listed in this file when running the crop ends function	None	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.

Remove Short Removes sequences below a threshold length.

Parameter	Description	Default Value	Min	Max
<code>--remove_short</code>	Remove sequences <= <code>remove_min_length</code> amino acids from the alignment	False	NA	NA
<code>--remove_min_length</code>	Sequences are removed if they are shorter than this minimum length, excluding gaps.	50	0	n_col
<code>--remove_short_retain</code>	Do not remove sequences with this name when running the remove short function	None	NA	NA
<code>--remove_short_retain_str</code>	Do not remove sequences with names containing this character string when running the remove short function	None	NA	NA
<code>--remove_short_retain_list</code>	Do not remove sequences with names listed in this file when running the remove short function	None	NA	NA

Keep Gap Only Removes columns containing only gaps.

Parameter	Description	Default Value	Min	Max
<code>--keep_gaponly</code>	Keep gap only columns in the alignment	False	NA	NA

Crop Divergent Crops columns from the sides of alignment to leave only a single conserved section, based on a threshold percentage of identical residues and percentage of gaps in each column.

Parameter	Description	Default Value	Min	Max
--crop_divergent	Crop either end of the alignment until > crop_divergent_min_prop_ident residues in a column are identical and > crop_divergent_min_prop_nongap residues are not gaps, over buffer_size consecutive columns	False	NA	NA
--crop_divergent_min_prop_ident	Minumum proportion of identical residues in a column to be retained by crop_divergent	0.5	0.01	1
--crop_divergent_min_prop_nongap	Minumum proportion of non gap residues in a column to be retained by crop_divergent	0.5	0.01	1
--crop_divergent_buffer_size	Minumum number of consecutive columns which must meet the criteria for crop_divergent to be retained	5	1	n_col

Retain These parameters allow the user to specify sequences to not edit with any of the rowwise functions, regardless of the CIALign results. The rowwise functions are currently remove_divergent, crop_ends and remove_short.

Parameter	Description	Default Value	Min	Max
--retain	Do not edit or remove sequences with this name when running any rowwise function (currently remove divergent, crop ends and remove short)	None	NA	NA
--retain_str	Do not edit or remove sequences with names containing this character string when running any rowwise function	None	NA	NA
--retain_list	Do not edit or remove sequences with names listed in this file when running any rowwise function	None	NA	NA

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

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
<code>--plot_input</code>	Plot a mini alignment - an image representing the input alignment	False
<code>--plot_output</code>	Plot a mini alignment - an image representing the output alignment	False
<code>--plot_markup</code>	Draws the input alignment but with the columns and rows which have been removed by each function marked up in corresponding colours	False
<code>-plot_dpi</code>	DPI for mini alignments	300
<code>-plot_format</code>	Image format for mini alignments - can be png, svg, tiff or jpg	png
<code>-plot_width</code>	Mini alignment width in inches	5
<code>-plot_height</code>	Mini alignment height in inches	3
<code>-plot_keep_numbers</code>	Label rows in mini alignments based on input alignment, rather than renumbering	False
<code>-plot_force_numbers</code>	Force all rows in mini alignments to be numbered rather than labelling e.g. every 10th row for larger plots Will cause labels to overlap in larger plots	False

Sequence logos These functions draw sequence logos representing output (cleaned) alignment. You can also specify a subsection of the alignment using the `logo_start` and `logo_end` arguments, positions should be relative to the input 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

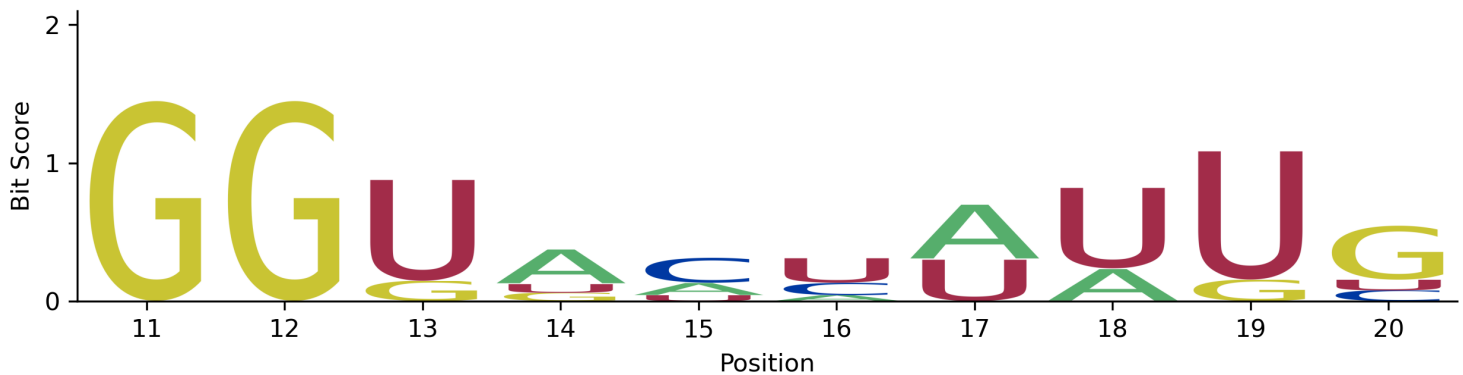


Figure 1: Sequence logo text

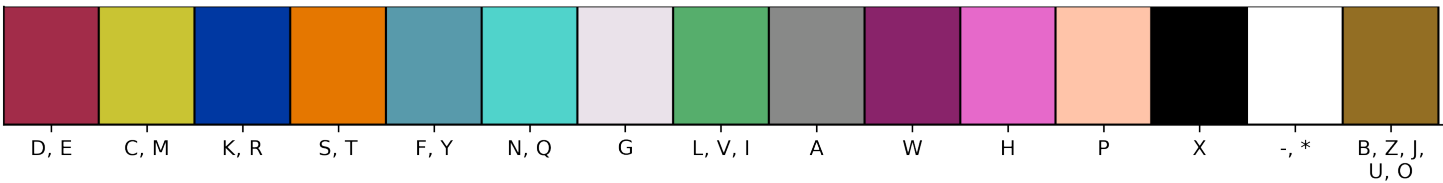
Parameter	Description	Default
<code>--make_sequence_logo</code>	Draw a sequence logo	False
<code>--sequence_logo_type</code>	Type of sequence logo - bar/text/both	bar
<code>--sequence_logo_dpi</code>	DPI for sequence logo	300
<code>--sequence_logo_font</code>	Font (see NB below) for bases / amino acids in a text based sequence logo	monospace
<code>--sequence_logo_nt_per_row</code>	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
<code>--sequence_logo_filetype</code>	Image file type to use for the sequence logo - can be png, svg, tiff or jpg	png
<code>-logo_start</code>	Start sequence logo	0
<code>-logo_end</code>	End of sequence logo	MSA length

NB: to see available fonts on your system, run `CIAAlign -list_fonts_only` and view `CIAAlign_fonts.png`

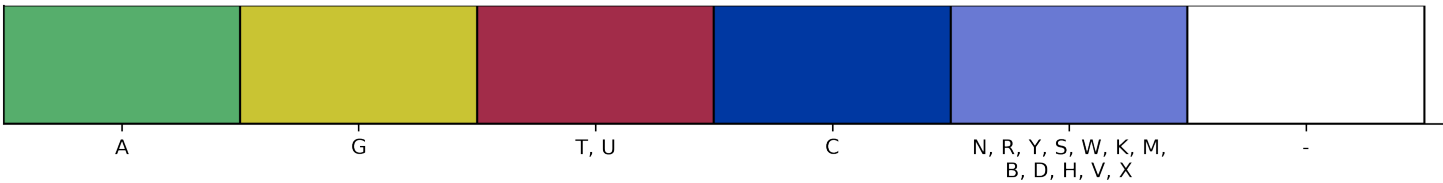
Palettes This function sets the colour palette for the mini alignments. Currently available palettes are colour blind safe (CBS) and bright.

CBS palette

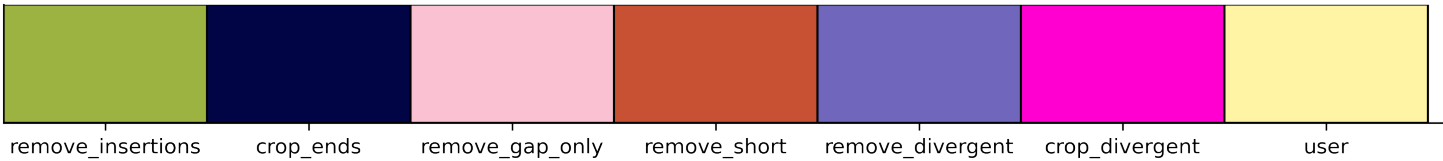
amino acids



nucleotides

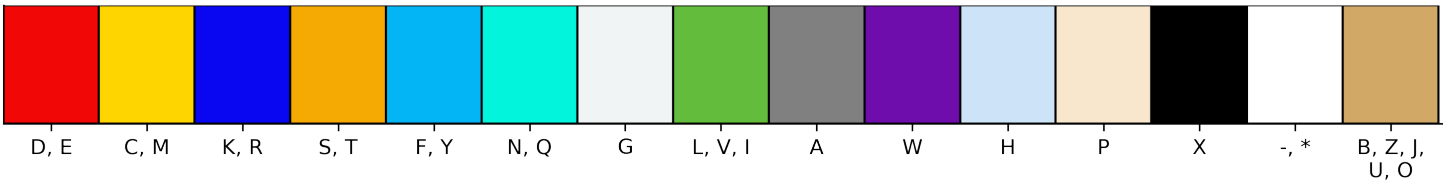


markup

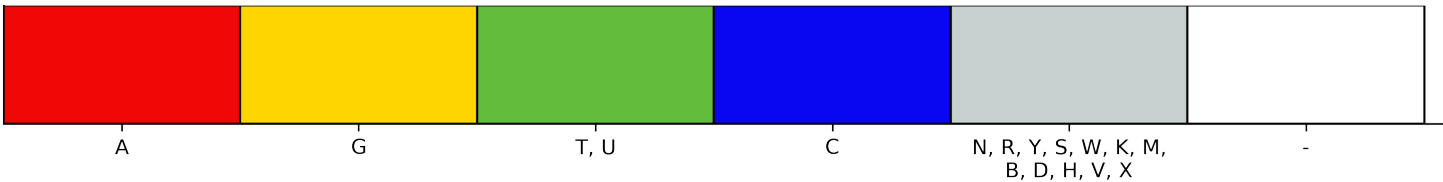


bright palette

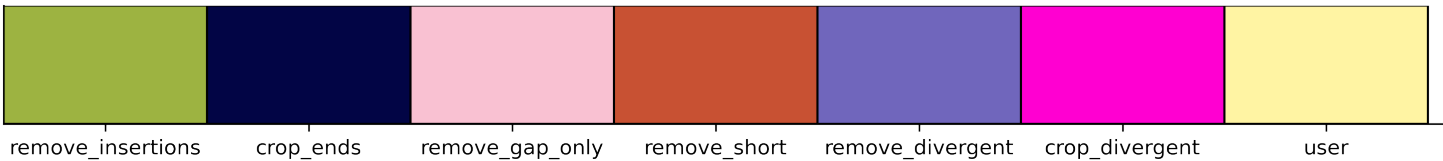
amino acids



nucleotides



markup



Parameter	Description	Default Value
<code>--palette</code>	Colour palette. Currently implemented CBS (colourblind safe) and bright	CBS

Analysing Alignment Statistics

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

Position Frequency, Probability and Weight Matrices These functions are used to create a position weight matrix, position frequency matrix or position probability matrix for your input or output (cleaned) alignment. These are numerical representations of the alignment which can be used as input for various other software, for example to find regions of another sequence resembling part of your alignment. PFMs, PPMs and PWMs are described well in the Wikipedia article [here](#).

You can also specify a subsection of the alignment using the `pwm_start` and `pwm_end` arguments, positions should be relative to the input alignment.

Output_files:

- `OUTFILE_STEM_pwm_(input/output).txt` - position weight matrix representing the alignment (or part of the alignment)
- `OUTFILE_STEM_ppm_(input/output).txt` - position probability matrix representing the alignment (or part of the alignment)
- `OUTFILE_STEM_pfm_(input/output).txt` - position frequency matrix representing the alignment (or part of the alignment)
- `OUTFILE_STEM_ppm_meme_(input/output).txt` - position probability matrix representing the alignment (or part of the alignment) in the format used by the MEME software suite.
- `OUTFILE_STEM_blamm_(input/output).png` - position probability matrix representing the alignment (or part of the alignment) in the format used by the BLAMM software tool.

Parameter	Description	Default
<code>--pwm_input</code>	Generate a position frequency matrix, position probability matrix and position weight matrix based on the input alignment	False
<code>--pwm_output</code>	Generate a position frequency matrix, position probability matrix and position weight matrix based on the cleaned output alignment	False
<code>--pwm_start</code>	Start the PWM and other matrices from this column of the input alignment	None
<code>--pwm_end</code>	Start the PWM and other matrices from this column of the input alignment	None
<code>--pwm_freqtype</code>	Type of background frequency matrix to use when generating the PWM. Should be 'equal', 'calc', 'calc2' or 'user'. 'equal', assume all residues are equally common, 'calc', frequency is calculated using the PFM, 'calc2', frequency is calculated using the full alignment (same as calc if <code>pwm_start</code> and <code>pwm_end</code> are not specified).	equal
<code>--pwm_alphatype</code>	Alpha value to use as a pseudocount to avoid zero values in the PPM. Should be 'calc' or 'user'. If <code>alphatype</code> is 'calc', alpha is calculated as $\text{frequency}(\text{base}) * (\text{square root}(\text{n rows in alignment}))$, as described in Dave Tang's blog here , which recreates the method used in Wasserman & Sandelin 2004. If <code>alphatype</code> is 'user' the user provides the value of alpha as <code>pwm_alphatype</code> . To run without pseudocounts set <code>pwm_alphatype</code> as user and <code>pwm_alphaval</code> as 0	calc
<code>--pwm_alphaval</code>	User defined value of the alpha parameter to use as a pseudocount in the PPM.	1
<code>--pwm_output_blamm</code>	Output PPM formatted for BLAMM software	False
<code>--pwm_output_meme</code>	Output PPM formatted for MEME software	False

Statistics Plots For each position in the alignment, these functions plot: * Coverage (the number of non-gap residues) * Information content * Shannon entropy

Output files:

- `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
- `OUTFILE_STEM_input_information_content.png` (or `svg`, `tiff`, `jpg`) - image showing the input alignment information content
- `OUTFILE_STEM_output_information_content.png` (or `svg`, `tiff`, `jpg`) - image showing the output alignment information content

- **OUTFILE_STEM_input_shannon_entropy.png (or svg, tiff, jpg)** - image showing the input alignment Shannon entropy
- **OUTFILE_STEM_output_shannon_entropy.png (or svg, tiff, jpg)** - image showing the output alignment Shannon entropy

Parameter	Description	Default
--plot_stats_input	Plot the statistics for the input MSA	False
--plot_stats_output	Plot the statistics for the output MSA	False
-plot_stats_dpi	DPI for coverage plot	300
-plot_stats_height	Height for coverage plot (inches)	3
-plot_stats_width	Width for coverage plot (inches)	5
-plot_stats_colour	Colour for coverage plot (hex code or name)	#007bf5
-plot_stats_filetype	File type for coverage plot (png, svg, tiff, jpg)	png

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

Additional Functions

Extracting sections from an alignment

Replacing U or T This function replaces the U nucleotides with T nucleotides or vice versa without otherwise changing the alignment.

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

or

- **OUTFILE_STEM_U_input.fasta** - input alignment with U's instead of T's
- **OUTFILE_STEM_U_output.fasta** - output alignment with U's instead of T's

Parameter	Description	Default
--replace_input_tu	Generates a copy of the input alignment with T's instead of U's	False
--replace_output_tu	Generates a copy of the output alignment with T's instead of U's	False
--replace_input_ut	Generates a copy of the input alignment with U's instead of T's	False
--replace_output_ut	Generates a copy of the output alignment with U's instead of T's	False

Unaligning (removing gaps) 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
<code>--unalign_input</code>	Generates a copy of the input alignment with no gaps	False
<code>--unalign_output</code>	Generates a copy of the output alignment with no gaps	False