Manual

Align-m v2.3

Ivo Van Walle

January 6, 2005

ivwalle@vub.ac.be

Contents

Copyright notice

1 Align-m

Align-m is a program for multiple alignment of biological sequences. It can align protein sequences, DNA sequences or any other type, including those that cannot be represented as a character sequence. It currently contains 3 modules: S2P, P2P and P2M. The data flow is given in figure 1.

Its strengths are:

- Alignment can be restricted to columns of high confidence. Align-m is by far the most accurate algorithm in terms of number of incorrectly aligned residues, compared to ClustalW and T-Coffee.[3, 2, 5]
- Among the more accurate algorithms in terms of number of correctly aligned residues
- Highly customisable: extra information can be added to improve the alignment, modules can be ran separately and/or iteratively for specific purposes (see section 3 on use cases).

In contrast, the modules are computationally demanding, putting a practical limit on the number and length of the sequences. The complexity of each module is given in §1.1 through §1.3. Align-m is therefore not very suitable for large-scale applications. Rather, it is intended for specific alignment problems that possibly require a lot of user intervention.

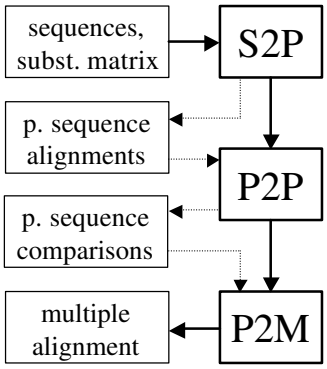

Figure 1: Flow of data (left column) through the Align-m modules (right column). Each module can be executed separately or in sequence. P2P produces pairwise sequence comparisons in the form of either a set of CMs or a set of PSAs.

1.1 S2P

The S2P module converts sequences with a character representation into a set of pairwise alignments, using a substitution matrix. In a first step, high scoring columns through all N sequences (average length L) are calculated using the FASTER algorithm.[1]. Then, this information is used to produce one or more 'guided' alignments per pair. The procedure is described in detail in Van Walle et al. (2004).[5] In addition, it is possible to include extra information into the calculations by replacing the similarity scores of some residue pairs by custom values. Also, it is possible to switch off the guided alignments, in which case standard Needleman-Wunsch alignments (overhangs not penalised) are produced, and which drastically reduces the running time and memory requirements.

Complexity: $O(N^2L^2)$ time and memory for guided alignments, $O(L^2)$ memory for unguided alignments.

1.2 P2P

The P2P module detects the consistencies within a set of pairwise alignments, while allowing for small shift errors. The alignments may contain multiple solutions per pair. It is basically an improved version of the Relaxed Transitive Alignment algorithm described in Van Walle et al.(2003), which allows less paths between residues.[4] The output can be either a set of consistency matrices (CMs) or a new set of 'most consistent' pairwise alignments (MCAs) derived from them.

Complexity: $O(N^3L)$ time and memory.

1.3 P2M

The P2M module converts a set of pairwise sequence comparisons, i.e. not necessarily alignments, into a set of multiple alignment columns.[6] By default, this set of columns is reduced until no residue is aligned more than once and the set represents a colinear multiple alignment. Only for the last case is it possible to output a FASTA or CLUSTAL file, if all sequences have a character representation.

This module has 2 important characteristics. Firstly, the output multiple alignment can be tuned to include only aligned columns of a given minimum confidence, allowing one to focus e.g. on highly conserved regions only. Secondly, and related to this, the module does not assume that all sequences have to be aligned. Thus, if it is clear that some input sequences are unrelated, they will not be aligned.

Complexity: $O(N^2L^2)$ time and $O(N^2L)$ memory.

2 Command line options

All of the command line arguments for Align-m are optional, except those that provide necessary input data. They are of the form `-parameter_name value`, except for a few that take no value. Tables 1-4 give an overview of all arguments.

3 Use cases

In order to easily keep track of the exact parameter settings and input files used to create an alignment, it is recommended to put all command line options in a single file, and then run the program with:

```
align_m -optfile options_file
```

The format of this options file is very simple and described in section 4. It also allows quick manipulation of the input, if a rerun with modified parameters is necessary.

3.1 Multiple sequence alignment

A standard multiple sequence alignment is produced by:

```
align_m -i seqfile -o outfile
```

This will run all 3 modules consecutively. The output format can be set to ALIGNM, FASTA, or CLUSTAL by the `-otype` option. If not set explicitly, the format is guessed from the outfile's extension, with FASTA as the default. Multiple sequence files can be provided by adding more `-i seqfile`. To speed up the alignment somewhat, losing some accuracy, set `-p2p off`. A drastic speed-up and reduction in memory requirements, losing some sensitivity but generally not specificity, can be obtained by switching off the guided alignments with `-s2p_guided_aln off`.

Higher settings of the `-p2m_Fmin` parameter will result in smaller alignments, in which only the more confident regions kept. Similarly, the `-p2m_Nmin` parameter determines the minimum number of aligned residues in each column, so a higher setting will also keep only the more confident and common regions. If feasible, it is recommended that several combinations of these parameter values are tested to see which yields the best compromise between accuracy in terms of correctly aligned residues, and alignment size.

The default parameters of Align-m are for protein sequences. For aligning DNA or RNA (or other) sequences, some should be adjusted, such as the gap opening and extension parameters, and perhaps also the ungapped segment width over which to average similarity scores. Also, if these sequences contain some ambiguity symbols, it is necessary to explicitly specify the substitution matrix, so that Align-m cannot make a mistake in determining the type of sequence in order to select a default substitution matrix itself. We have not yet exhaustively tested DNA and RNA alignment, but example settings could be:

```
-m DNA2 -s2p_w 23 -s2p_go 8 -s2p_ge 0.5
-m RNA2 -s2p_w 23 -s2p_go 8 -s2p_ge 0.5
```

Frequently, Align-m will output an alignment in which, for some region, each aligned residue is separated by one or more gaps in between. For example:

```
A-A-A-A-BBBBCCCC
A-A-A-A-BBB-CCCC
---D-D-DBBBB----
--dD-D-DBBBBeeee
```

Clearly, the first 2 sequences have the 'A' part in common, whereas the last 2 have, in that same region, a common 'D' part. All 4 share the 'B' part though, which is aligned as such. The many gaps in between the A's and the D's should not be interpreted as (separate) indels: they are merely there to be able to put the alignments of the 2 clusters underneath each other. An alternative would be to align each cluster to a single long gap, but Align-m does not take this kind of formatting issues into account. Finally, the lowercase letters are simply not aligned to any other residue (see also section 4).

3.2 Including extra information in a sequence alignment

As mentioned before, extra information can be injected into the multiple alignment process at any of the 3 levels S2P, P2P and P2M. For example, if conserved residues in some or all sequences are known, an ALIGNM formatted file can be prepared that contains their alignment, together with a score for each residue pair or column. This file can then be given to S2P via the `-s2p_i` option:

```
align_m -i seqfile -s2p_i alnfile1 ...  -s2p_i alnfileN -o outfile
```

If the provided scores are sufficiently high compared to the substitution matrix's scores, the pairwise alignments created by S2P will, insofar possible, be constrained to contain these residue pairs. If instead of an ALIGNM file, standard alignments are provided through the FASTA or CLUSTAL format, each aligned residue pair in these will be given a score equal to the maximum similarity score of the substitution matrix.

3.3 Multiple structure alignment

A multiple structure alignment can be derived from separately calculated pairwise structure alignments, by switching off the S2P module and running only P2P and P2M. Many solutions can be provided for each pair but currently, the structure information itself can however not be used during this process. The use of the P2P module is also optional, but recommended as a filter that will remove completely inconsistent residue pairs, or entire alignments, prior to running P2M. Alignments for P2P can only be provided in FASTA and CLUSTAL format, whereas P2M also accepts ALIGNM format files:

```
align_m -s2p off -p2p_i alnfile1 ...  -p2p_i alnfileN -o outfile
```
or
```
align_m -s2p off -p2p off -p2m_i alnfile1 ...  -p2m_i alnfileN -o outfile
```

Again, the `-p2m_Fmin` and `-p2m_Nmin` parameters can be used to fine tune the alignment. To determine e.g. the common core of the proteins, set N_{min} (close) to the total number of sequences.

P2M does not require that the input alignments be colinear, so it is possible to provide through the ALIGNM format an alignment that has e.g. a circular permutation or a different domain order. The output alignment however will not contain topology differences unless `-p2m_allow_topdiff` is set. The output format will then also be forced to ALIGNM, since neither FASTA nor CLUSTAL can represent such alignments. Depending on F_{min} and N_{min}, setting this option increases the chance that short, spurious alignments will be included into the final alignment, as a result of which some post-processing may be needed.

3.4 Homology modelling

Homology modelling mainly consists of mapping one or more sequences onto one or more structures. Align-m does not contain any algorithms that will directly take into account structure information. However, because extra information can be added at almost any stage (see previous sections), it is possible to use it as an easy way of generating an alignment from bits and pieces of knowledge in the form of equivalent residues and their score. This process can be executed iteratively as follows:

1. Generate an initial alignment of only the very high confidence regions, e.g. from structure alignment and biochemical data
2. Manually (or otherwise) edit the alignment by removing parts that are likely to be incorrect, and include extra, confident parts
3. Run Align-m with the new alignment as a constraint for S2P and/or P2M, by giving these residue pairs an appropriate high score. The program will then try to align the regions in between.
4. Repeat steps 2-3 until no more confident common regions can be found

The `-p2m_pre_cluster_columns` option can be handy here to improve alignment of sequences with clear subsets.

3.5 Filtering Blast alignments

To test if a number of sequences are related to each other, pairwise alignments can be generated between them and given to P2M. P2M will turn them into a multiple alignment, keeping only the sufficiently consistent parts, and thereby removing most of the incorrect alignments. The resulting 'filtered' multiple alignment can then be used to more clearly determine which sequences are related. This was succesfully tested on bl2seq (Blast 2 sequences) pairwise alignments of sequences of which only half were related, taken from the SABmark database, which covers the entire known fold space.[7] If the F_{min} and N_{min} parameters are set sufficiently strict, nearly all of the incorrect alignments were removed, along with slightly less than half of the correct alignments. The command line used was:

```
align_m -s2p off -p2p off -p2m_Fmin 0.7 -p2m_Nmin 5
   -p2m_i alnfile1 ...  -p2m_i alnfileN -o outfile
```

3.6 Combining alignments into a consensus

Similarly to filtering Blast alignments, alignment data from various sources can be combined into 1 single consensus alignment. When FASTA or CLUSTAL alignments are given as input to P2M, each residue pair gets a score of 1, so if a residue pair occurs for example in 2 different alignments, it will get a total score of 2. In this respect, the `-p2m_Sbad_max` parameter is useful: it sets the maximum score for a residue pair that Align-m still considers as insignificant. For example, setting this value to 1 will restrict the consensus multiple alignment to contain only columns in which enough residue pairs (as determined by F_{min}) have a score greater than 1. So, if a consensus of N alignments is to be generated, in which each column has enough residue pairs that were aligned at least twice, the following command line can be used:

```
align_m -s2p off -p2p off -p2m_Sbad_max 1
   -p2m_i alnfile1 ...  -p2m_i alnfileN -o outfile
```

3.7 Multiple genome alignment

This use case has only been tested on simulated data, so this paragraph is largely theoretical. Genome alignments usually contain topology differences due to rearrangements. Similar to the multiple structure alignment however, Align-m could produce a multiple genome alignment from pairwise genome alignments (or fragments of them), by setting the `-p2m_allow_topdiff` flag:

```
align_m -s2p off -p2p off -p2m_i alnfile1 ...  -p2m_i alnfileN
   -o outfile -p2m_allow_topdiff
```

In addition, P2M (and P2P, though not yet implemented) do not require that the sequences have a symbol representation, as is the case for amino and nucleic acid sequences. To increase speed and perhaps also efficiency, genomes could be represented as sequences of 'features' (putative genes, promotor binding regions, marker sites, etc.), detected by some program or some experiment. Between genomes, similarity scores between these features can be calculated and represented in ALIGNM format, and finally used by P2M to derive a multiple genome alignment from.

4 File formats

Align-m currently supports 3 sequence alignment file formats for both input (depending also on settings) and output: FASTA, CLUSTAL and its native format, ALIGNM. There are 2 additional requirements imposed on these:

1. Sequence names may not contain whitespace characters
2. Any **lowercase letters in an alignment are not considered as aligned**, even if they are in the same column as other (uppercase) letters.

In addition, a substitution matrix can be read from file, as well as additional command line arguments.

4.1 FASTA and CLUSTAL

The FASTA format can describe both sequences and sequence alignments. Align-m allows an extension of the latter, so as to be able to describe multiple solutions for a single alignments.

Example:

```
>seq1
AcDE--G
>seq2
A-DEF-G
>seq2
A-D-EfG
```

This FASTA file is interpreted by Align-m as containing 2 solutions for the alignment of seq1 to seq2. The 'alignment' between the first and the second seq2 is ignored. This may also be extended to more than 2 sequences, and the CLUSTAL format is interpreted in the same way.

4.2 ALIGNM native format

Align-m uses a native format that allows to represent data that cannot be represented by either FASTA and CLUSTAL, such as:

- Sequences that cannot be represented as a character sequence.
- Separate alignments between subsets of sequences.
- More general forms of comparisons between sequences: non-colinear alignments, and alignments of a single residue to more than 1 residue of the other sequence
- A score associated with each residue pair

The format is kept as simple as possible:

1. The first line starts with 'ALIGNM'.
2. Empty lines or lines starting with '#' are ignored.
3. The other lines contain alignment records for a subset of the sequences, where each record is spread across multiple lines. Multiple records for the same sequence subset are allowed. A record has the following properties:
   (a) The first line starts with '$>$' immediately followed by $nseq$ sequence names separated by whitespace. As such, sequence names may not contain whitespace. A total of $nseq + 1$ data lines must follow this line.

(b) The first $nseq$ data lines are indices, separated by whitespace, into the resp. sequences described in the first line. The indices start from 1 up to any number, since the sequence length is not fixed by the format. An index smaller than or equal to 0 is also allowed and considered as a gap. Also, the same index may occur more than once.

(c) The last data line contains the score of each aligned column, normalised by the number of residue pairs in it. The scores may be fractional values.

(d) An exception is a record for only 1 sequence, or many sequences with no residue pairs, which has no data lines.

Example:

```
ALIGNM
# A list of all sequences (not needed):
>seq1
>seq2
>seq3
# Alignment of the start of seq1 and seq2:
>seq1 seq2
1   2   3   5
1   2   3   4
1   0.9 1   0.5
# Alignment of seq1, seq2 and seq3
# Notice the first column spans only the first 2 sequences
# but since the score is between residue pairs, it is comparable
# in magnitude to the other columns
>seq1 seq2 seq3
6   7   8   9
5   6   7   8
0   3   4   6
0.5 1   1   0.7
```

4.3 Substitution matrix

Several amino acid substitution matrices are built in and can be chosen from the command line (table 2): `PAM20`, `PAM40`, `PAM80`, `PAM200`, `PAM250`, `PAM320`, `BLOSUM35`, `BLOSUM45`, `BLOSUM62`, `BLOSUM80`, `BLOSUM100`, `DNA`, `DNA2`, `RNA`, `Identity_protein`, `Identity_DNA` and `Identity_RNA`. Align-m chooses a default matrix if the alphabet is protein (`BLOSUM62`), DNA (`DNA2`) or RNA (`RNA`). If there are problems with recognition of the alphabet, force the use of a specific substitution matrix with the `-m` option. Through this option it is also possible to provide a custom substitution matrix that describes symbol similarities for any alphabet that does not contain whitespace or lowercase characters. The file format is as follows:

1. First line: substitution matrix name.
2. Second line: alphabet of n unique characters. Whitespace is ignored.

3. Lines 3 through $n + 2$, i.e. for each letter of the alphabet, start with the alphabet character followed by n substitution scores all separated by whitespace. Substitution scores may be fractional values.

Example:

```
Custom
   A  B
B  0  1
A  1  0.5
```

4.4 Command line arguments file

A file with command line arguments simply contains the same text that would be used as a command line, but it may be spread across multiple lines. Also, empty lines and lines starting with '#' are ignored.

Example:

```
-i infile.fasta
-o outfile.fasta
#-o outfile2.fasta
-p2p on
#-p2p off
```

References

[1] Johan Desmet, Jan Spriet, and Ignace Lasters. Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization. *Proteins*, 48(1):31–43, Jul 2002.

[2] C Notredame, DG Higgins, and J Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1):205–17, Sep 2000.

[3] JD Thompson, DG Higgins, and TJ Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80, Nov 1994.

[4] Ivo Van Walle, Ignace Lasters, and Lode Wyns. Consistency matrices: quantified structure alignments for sets of related proteins. *Proteins*, 51(1):1–9, Apr 2003.

[5] Ivo Van Walle, Ignace Lasters, and Lode Wyns. Align-m–a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics*, 20(9):1428–35, Jun 2004.

[6] Ivo Van Walle, Ignace Lasters, and Lode Wyns. An assessment of the limits of sequence and structure comparison with Align-m 2. *Bioinformatics*, 2004. Submitted.

[7] Ivo Van Walle, Ignace Lasters, and Lode Wyns. SABmark – a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 2004. In publication.

Table 1: General command line arguments for Align-m

Parameter	Default	Description	Value
`-i`*		Input sequence file	File name
`-optfile`*		File with command line argument, inserted before the other arguments	File name
`-o`		Output file or directory (for pairwise output files)	File name
`-otype`		Output file(s) type. Not all types are allowed for all settings. If not given, the type is guessed from the `-o` file's extension.	`FASTA`, `CLUSTAL` or `ALIGNM`
`-v`		Verbose comments during execution	No value

*More than one may be given.

Table 2: Command line arguments for S2P

Parameter	Default	Description	Value
`-m`		Alias for `-s2p_m`	
`-s2p`	`on`	Switch S2P on or off	`on` or `off`
`-s2p_i`		Input pairwise comparison file. Residue pairs from alignments are given the maximum score of the substitution matrix	File name
`-s2p_guided_aln`	`on`	Switch guided alignments on or off	`on` or `off`
`-s2p_m`		Substitution matrix name. For protein sequences, `BLOSUM62` is used by default.	Standard name (§4.3) or file name of custom matrix
`-s2p_w`	`15`	Width of ungapped segment to average over	Positive uneven integer
`-s2p_nsigma_max`	`3`	Maximum number of sigma that a new set of columns may score on average below the first one	Positive real
`-s2p_minscore`	`0`	Minimum score that a new set of columns may score on average (supersedes `-s2p_nsigma_max`)	Real
`-s2p_go`	`12`	Gap opening penalty for guided alignments	Positive real
`-s2p_ge`	`2`	Gap extension penalty for guided alignments	Positive real
`-s2p_naln`	`3`	Number of guided alignments per sequence pair	Strictly positive integer
`-s2p_pn`	`1`	Penalty for aligned residue pairs before calculating the next guided alignment	Strictly positive real

*More than one may be given.

Table 3: Command line arguments for P2P

Parameter	Default	Description	Value
`-p2p`	`off`	Switch P2P on or off	`on` or `off`
`-p2p_i`*		Input sequence alignment file (FASTA or CLUSTAL)	File name
`-p2p_otype`	`MCAs`	Output type	`MCAs` or `CMs`

*More than one may be given.

Table 4: Command line arguments for P2M

Parameter	Default	Description	Value
`-p2m`	`on`	Switch P2M on or off	`on`
`-p2m_i`*		Input pairwise comparison file	File name
`-p2m_Sbad_max`	`0`	Maximum residue pair score that is still insignificant	Real value
`-p2m_Fmin`	`0.7`	Minimum fraction of insignificant scores of a residue with the other aligned residues in the same column. A value of 0 is interpreted as 1 signficant score being sufficient	Real value [0,1]
`-p2m_Nmin`	`5`	Minimum number of aligned residues per column. If larger than the actual number of sequences N, it is lowered to N	Integer value [2,N]
`-p2m_nseq_min`		Alias for `-p2m_Nmin`. Deprecated.	
`-p2m_allow_topdiff`		Do not force the output alignment to be colinear. ALIGNM output format only.	No value
`-p2m_pre_cluster_columns`		Separate totally disjunct residue clusters in columns before pruning them, improving handling of sequences sets with clear subsets.	No value

*More than one may be given.