Aln (Ver. 4.0.4)

Alignment of a pair of nucleotide and amino acid sequences (for UNIX)

Osamu Gotoh

Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto, Kyoto 606-8501, Japan
Tel: +81-75-753-9109, Fax: +81-75-753-9110
E-mail: o.gotoh@i.kyoto-u.ac.jp

Original version: August 1992
Major updated version: March 2000
Major updated version: December 2009
Distributed version: July 2-14

Overview

Aln is a program for aligning a pair of nucleotide or amino acid sequences or groups of sequences. **Aln** accepts various combinations of distinct types of inputs: a single nucleotide sequence (Sn), a single genomic DNA fragment containing potential introns (Sg), a single amino acid sequence (Sa), a pre-aligned group of nucleotide sequences (Mn), and a pre-aligned group of amino acid sequences (Ma). **Aln** attempts to obtain the exact optimal alignment between input sequences or alignments, but the current version of **aln** also supports a heuristic yet faster algorithm for alignment of a pair of single sequences. **Aln** also implements spliced alignment algorithm [6-8], which enables us to predict eukaryotic gene structures (exon-intron organizations) based on sequence homology with known protein or mRNA (cDNA) sequence(s) together with intrinsic statistical features such as coding potential and exon-intron boundary signals. The details of these procedures will be described in detail in a later section entitled 'Gene-structure prediction'.

In sequence alignment, the hardest problem is to develop an efficient optimization algorithm under realistic constraints on deletions and insertions (indels or gaps). Currently the most popular methods use affine functions (AF) of the form, $g(k) = uk + v$, to penalize an indel [1]. A slightly more general penalty function is a piecewise linear function [3], a special and the simplest form of which is a double affine (DA), $g(k) = uk + v$ for $k <= K$ and $g(k) = g(K) + u'(k-K)$ for $(k > K)$, where u, u' $(< u)$, v, and K are non-negative constants. With the default settings, **aln** automatically selects seemingly the best algorithm, depending on the types of input sequences, as follows. Choice of AF (default) or DA depends on the command line option: -yl2 (AF) or –yl3 (DA).

(1A)	Sa x Sa:	DP-AF or DA
(1N)	Sn x Sn:	DP-AF or DA
(2A)	Sa x Ma or Ma x Ma:	DP-AF or DA
(2N)	Sn x Mn or Mn x Mn:	DP-AF or DA
(3A)	Sg x Sa or Sg x Ma:	SA-AF or DA
(3N)	Sg x Sn:	SA-AF or DA

where DP, CL, and SA stand for dynamic programming, candidate-list, and spliced alignment algorithms, respectively. CL is a generalized version of DP, and was introduced to rigorously optimize a group-to-group alignment score with an affine gap-penalty function [4]. **Aln** converts the input Ma or Mn into a generalized profile [5], if the performance is expected to be improved compared to the ordinary character-based method. In (3A) and (3N), **aln** uses the same internal codes as **spaln** [7, 8], but the interface and default settings are different, e.g. **aln** outputs alignment by defaults, whereas **spaln** reports information of the gene structure. Moreover, **spaln** does but

aln does not support genome mapping.

References

[1] Gotoh, O. (1982) “An improved algorithm for matching biological sequences.” *J. Mol. Biol.* **162**, 705-708.
[2] Gotoh, O. (1987) “Pattern matching of biological sequences with limited storage.” *Comput. Applic. Biosci.* **3**, 17-20.
[3] Gotoh, O. (1990) “Optimal sequence alignment allowing for long gaps.” *Bull. Math. Biol.* **52**, 359-373.
[4] Gotoh, O. (1993) “Optimal alignment between groups of sequences and its application to multiple sequence alignment.” *Comput. Applic. Biosci.* **9**, 361-370.
[5] Gotoh, O. (1994) “Further improvement in group-to-group sequence alignment with generalized profile operations.” *Comput. Applic. Biosci.* **10**, 379-387.
[6] Gotoh, O. (2000) “Homology-based gene structure prediction: simplified matching algorithm by the use of translated codon (tron) and improved accuracy by allowing for long gaps.” *Bioinformatics*, **16**, 190-202.
[7] Gotoh, O. (2008) “A space-efficient and accurate method for mapping and aligning cDNA sequences onto genomic sequence.” *Nucleic Acid Res.*, **36**, 2630-2638.
[8] Gotoh, O. (2008) “Direct mapping and alignment of protein sequences onto genomic sequence.” *Bioinformatics*, **24**, 2438–2444.
[9] Gotoh, O., Morita, M. Nelson, D.R. (2014) "Assessment and refinement of eukaryotic gene structure prediction with gene-structure-aware multiple protein sequence alignment", submitted

Changes from the previous version

1. The output format specified by -O option is unified with that of spaln.
2. Gene-structure-Aware alignment is evoked without additional option if the information about exon-intron organizations of the corresponding genes is provided in the input sequence files.
3. Algorithm C in Ref 3, instead of Algorithm D is now used by default.
4. The -yl3 option can handle double affine gap penalty for group-to-sequence of group-to-group alignment as well as sequence-to-sequence alignment.
5. The 'quick' option given by -Q and -LS option mentioned below are temporarily disabled.
6. The conversational mode described below is also now disabled.

Changes made in Version 3.5

1. For alignment of single sequences (Sa x Sa, Sn x Sn), the “quick” mode is added in (1A) and (1N). To use this mode, set `-QN` (*N* = 1, 2, or 3) option. The quick mode has already been implemented in spliced alignment, (3A) and (3N).
2. If information about the exon-intron organization of the corresponding gene is provided in the sequence files **and** `-yJN` (*N* > 0) option is set, a bonus is given to a match of intron positions along the resultant alignment.
3. The `-LS` (local similarity) option invokes Smith-Waterman type local alignment. Thus **aln** can now virtually replace **swg** that is an implementation of the Smith-Waterman algorithm modified by Gotoh [2] so that suboptimal local alignments can also be reported. However, **aln** sets the “cutting corners approximation”, and hence some similarities far from the main diagonals might be missed. In combination with `-M2` option, **aln** reports only "unique" alignments suppressing repetitious outputs, which may be reported with `-M3` option.

System

The programs are written in C++, and distributed as source code. Users must compile the programs on their own system. Although the programs have been tested only a few operating systems, they are

likely to be portable to most UNIX systems with little or no modifications. The distribution package also contains source codes of a few related programs, including **`prrn`**, which is used for multiple sequence alignment by randomized iterative refinement strategies. By following the instructions below, you should obtain executable forms of all of the programs in the package.

Install

After decompressing and extracting `prralnxxx.tar.gz`, you will find four directories under the `./prraln` directory (`xxx` is a version code):

`bin`: binary files will be stored
`doc`: contains this document
`perl`: perl scripts
`sample`: sample sequences in various legal formats
`seqdb`: indexed sequences are stored
`src`: source codes
`table`: score matrix of amino-acid substitutions and other tables

To compile

(1) Move to the directory `'./prralnxxx/src'`.
(2) % `./configure`.
(3) Edit the first line in `Makefile` so that "exec_prefix" defines the directory to which the executables are copied.
(4) Edit the second line in `Makefile` so that "table_dir" defines the directory where the contents of "table" are copied.
(5) If CC indicates a C compiler, edit `Makefile` so that CC indicates a C++ compiler (ex. g++).
(6) `% make all`.
(7) `% make install`. You must have the right to write into the ”`exec_prefix`” directory.
(8) `% make clearall`, which deletes all .o, .a and executables.

The "`make install`" command uses **`makmdm`** to create two files, "`mdm_cmp`" and "`mdm_mtx`", in the "`table`" directory. The present version of **`makmdm`** produces PET91 matrixes (Jones et al. (1992) CABIOS 8, 275-282) by default. The `-y78` option forces it to create the original MDM78 matrixes of Dayhoff et al.

Sequence Files

A sequence file may contain either a single sequence or a set of pre-aligned sequences. Avoid file names starting with a numeral.

<Single sequence>

Almost all popular formats for nucleotide and amino acid sequences, FASTA, GenBank, EMBL, SwissProt, PIR, and ProDB, are acceptable. A simple text file without any additional information is also fine. In any format, a line beginning with a semicolon (;) is regarded as a comment line. The specific format is recognized automatically by the first word in the first non-comment line of each file. Only single-letter codes are accepted for an amino acid. The nomenclature recommended by NC-IUB (Eur. J. Biochem. (1985) 150, 1-5) is followed to represent ambiguous nucleotides. The programs are not case-sensitive.

<Sequential format of multiple sequences>

The concatenation of multiple sequences in one of the above single-sequence formats may be used as a program input. Deletion characters are automatically padded at the end of a shorter sequence. If the sequences are pre-aligned, internal deletion characters are preserved. Two examples of sequential formats are shown below.

```
File 1:                    |     File 2:
                           |
>Seq1                      |     LOCUS  Seq1
;C (1..10, 101..200)       |     ORIGIN
aaatttcccggg               |           1 AAATTTCCCGGG
>Seq2                      |     //
atcgatcgatcgat             |     LOCUS  NCODE
; this is comment          |     ORIGIN
>NCODE                     |           1 ACMGR-SVTWYHKDBN
ACMGRSVTWYHKDBN            |     //
```

<Interleaved (native) format of multiple sequences>

This native format is designed for multiple-sequence alignment to be naturally recognized by human eyes. The alignment produced by **`aln`** can be used as an input to **`aln`** or **`prrn`**, and this is the most common way to have access to sets of pre-aligned sequences. Thus, the format of an aligned sequence file is the same as the default output format of **`aln`**. The first non-blank line in a file must indicate the number of sequences, *N*, involved in the alignment. This number is obtained as the sum of numbers in square brackets, *e.g.*, when the first line is

```
Seq1[3] - Seq2[4]
```

N is calculated to be 7. Subsequent lines up to the first blank line are ignored. The rest of the file is composed of one or more blocks of a fixed column width of less than 254 characters. Each block is composed of *N* 'sequence lines' and other (optional) 'non-sequence lines'. The general format of a sequence line is:

```
<Position>  <Sequence|>  <Name>,
```

where `<Position>` is a numeral that indicates the sequence position of the first letter in `<Sequence|>` (Usually all `<Position>`s in the first block are 1, but it is not a prerequisite. Negative values are also appropriate). A line lacking the `<Position>` field is regarded as a 'non-sequence line' and ignored upon reading. The *i*-th sequence line in the second block is concatenated to the *i*-th sequence line in the first block, and so on. There is no particular limit on *N* or the length, but the total number of characters to be stored is limited by `MAXAREA` defined in `src/sqio.h`. Several examples of native format are provided in the sample directory.

<Special Characters>

Three characters, dash `'-'`, tilde `'~'` and caret `'^'` have special meanings in a multiple-sequence file of the native format. A dash indicates a 'deletion' introduced by some alignment procedure. Be careful not to use a space or dot instead of a dash. Spaces and dots are simply ignored, so that the file may be interpreted in a totally unexpected way. A tilde means the same residue as that in the first sequence line on that column in the block. On the other hand, a caret means the same residue as that in the previous sequence line on that column. These ditto characters are convenient in representing an alignment of closely related sequences. Neither `'~'` nor `'^'` is allowed in the first sequence line in each block.

<Information on gene structures>

To represent the exon-intron structure of the parental gene, the FASTA file should be extended. A line starting with ';C' shows the exon boundaries (inclusive). More than one line may be used if necessary. The format after ";C " is essentially the same as that of Feature field of a GenBank file. Start and end positions of each exon are separated by two dots. Individual exons are delimited by a comma. The term 'complement' indicates that the corresponding gene lies in the complementary strand of the genomic sequence. An example is the second line in Seq1 shown above.

When the parental gene contains one or more frame shifts, an output from `aln` or `spaln` contains one or more lines starting with ';M' such as:

;M Deleted *n* chars at *p*
;M Insert *n* chars at *p*

The first line indicates that *n* nucleotides (*n* = 1 or 2) have been deleted from the parental genomic sequence beginning at site *p*. Likewise, the second line indicates that *n* blank characters are inserted after site *p* to maintain the open reading frame. Such kinds of information are uaed to properly juxtapose intron positions along the alignment.

For native format of multiple sequences, lines starting with ";B ", ";b ", and ";m " represent the information about organizations of corresponding genes. The first number that follows ";B ", *NP*, indicates the number of alignment positions where an intron intervenes at least one of the genes corresponding to the aligned protein or cDNA sequences. For a protein sequence, the position means that of coding nucleotide sequence so that phase as well alignment column is also significant. The second number in this line, *NI*, indicates the total number of introns. The numbers that follow ";b " indicate the number of sequences that contain the intron at the 1st, 2nd, ..., *NP*-th position. The numbers that follow ";m " present the list of sequences (1..*N*) that contain the intron at the 1st, 2nd, ..., *NP*-th position in this order. See the example ce13a.mfa in sample/pas directory.

User Operation

There are three methods to invoke a program (currently Method [1] is disabled).

```
[1]    aln [option2]
[2]    aln [option2] seq1 seq2
[3]    aln option1 [option2] [seq2]
```

Option2 is common to other programs (e.g. **`prrn`**), and default values will be used if omitted. Method [1] is a 'conversational mode', which will be discussed in detail below. Method [2] is an "instant mode"; the calculation starts immediately using the sequence data in files seq1 and seq2. Method [3] is a "batch mode", which will be discussed after explanation of conversational mode.

Notice

At the beginning of each process, several files in the `'table'` directory are read. Follow the instructions under Install to properly locate these files.

Usually, **`aln`** automatically judges the type of a sequence (amino acids or nucleotides) according to the composition of characters read in. You may explicitly specify the sequence type by adding an "attribute" as described below.

[1] Conversational mode (presently disabled)

This menu-driven mode provides an easy way to use the programs. You may read a sequence file,

change parameter values, or send the results to a desired device or file. However, this mode is somewhat obsolete, and does not entirely support some new features.

Menu

In response to the prompt, you can select one or more single-letter commands. Case is significant except for a few commands. Numerals must be separated from other commands by one or more spaces, whereas other characters should be typed contiguously. The commands are processed from left to right. For example

```
Menu Prompt: 1 2 pgr
```

is equivalent to

```
Menu Prompt: 1
Menu Prompt: 2
Menu Prompt: p
Menu Prompt: g
Menu Prompt: r
```

Sequence

When the programs successfully start, or when you select a numeral (1 or 2) as the menu command, input of a sequence is prompted. (If you give a plus sign in addition to a numeral, i.e., 1+ or 2+, the new sequence is added (concatenated) to the existing one, otherwise the new sequence replaces the older one.) You should respond with

```
[[path][filename]] [start] [end] [attribute]
```

Any terms may be omitted, except `start` when you wish to specify `end`. When `path` is omitted, the default path is used (this can be specified with option2, see [2] instant mode below). `Start` and `end` inclusively specify the range of the sequence to be analyzed. The default values are 1 and the last position of the sequence, respectively. Some attribute terms are applicable only to a nucleotide sequence. Currently meaningful attributes are defined below.

`^`	: the sequence is complemented. (nucleotide sequence only)
`-`	: the sequence is reversed.
`<`	: the sequence is reverse-complemented.
`%`	: forces the sequences into profile.
`A`	: amino acid sequence.
`P`	: amino acid sequence.
`D`	: nucleotide sequence.
`R`	: nucleotide sequence.
`G`	: genomic sequence with potential introns.
`N`	: nucleotide sequence, delete ambiguous codes ('N').
`T`	: tron sequence. See ref. [6].
`@`*N*	: read coding parts only, according to the *N*-th CDS (GenBank only).
`#`*N*	: retain only the columns composed of >*N*% of non-deletion characters.

If there is no attribute character (default), the sequence remains as read. If filename is specified, the remaining terms are reset to the default values, otherwise non-specified terms are unchanged. Of course, filename should not be omitted in the beginning session.

Parameters

When you select the 'p' or 'P' command in the menu, you can change the current parameter values, which are shown in parentheses. Lowercase 'p' and uppercase 'P' refer to different sets of parameters. Carriage/Return terminates the input, leaving the remaining terms unchanged. Hence, only hit Carriage/Return if you do not want to change the current values shown in the line. The following table shows the default values and meanings of the parameters used. Note that the default values vary with the sequence types to be aligned, and may be different from those shown here.

Parameters

	Default	Meaning
pam	250	PAM level of Dayhoff's mutation data matrix MDM [0 - 300] step by 10.
bias	0	Value to be added to each element of MDM.
scale	1	Precision of the MDM matrix (>= 1).
v	9	Constant term of the gap-weighting function.
u	2	Proportional coefficient of the gap-weighting function for a gap shorter than or equal to k1.
s[=]	2	Similarity measure between identical nucleotides.
s[#]	-2	Similarity measure between different nucleotides.
u1	1	Proportional coefficient of the gap-weighting function for a gap longer than k1.
k1	10	The flection point.
shldr	100	Window size (>= 0), see note 1.
trial	0	Number of pairs of randomized sequences used for a Monte Carlo test for the significance of sequence similarity.
lpw	60	Number of residues per line in an output alignment.
lseg	2	Number of pieces of linear functions that represent a gap-weighting function [1 - 3], see Ref. [2]. For DNA vs. protein sequence alignment, lseg = 2 (affine gap penalty) and lseg = 3 (restricted affine penalty) indicate the potential existence of introns in the DNA sequence.
pmode	1	Control the format of the output alignment.
consl	0	do. How strictly to regard a site as a consensus site.
lorn	0	do. Sequence label is name (0) or numbering (1).
cmode	1	Calculation mode [0 - 1], see note 2.
algor	0	Select the algorithm for group-to-group alignment. If 0, the program chooses conceivably the best one.
av/sum	1	Sum of pairs measure (1) or averaged score (0).

(note 1)
The program sets a window of size (|*M* - *N*| + 2 * *shldr*) along the center of the main diagonal, where *M* and *N* are the lengths of the sequences. For DNA vs. protein sequence alignment, the length of the protein sequence is multiplied by 3 to calculate the window size.

(note 2)

Mode = 0:	Get only the distance value and other statistics (no alignment).
Mode = 1:	Get a single optimal alignment.

Go, and Reprint

The 'g' command starts actual calculation. After calculation is complete, **aln** prompts for where the result is to be output. At the prompt, select a number (-1, 0, 1, 2, 3, or 4, the default is 0). When you select 3 or 4, the destination file name is prompted (if you select 4, the file is opened in an

append mode). The alignment is saved in a file which may be further edited or used as an input file for multiple-sequence alignment. The `'r'` command re-displays the latest result, and is effective only after a `'g'` command.

Quit

To exit from the program, type `'q'` in response to the menu prompt.

[2] Instant mode

This mode is the simplest to use. Two sequence files to be compared are given in the command line arguments, and the result is reported to `stdout` (standard output). Command line options, option2, are common to all modes, but they are described here because they are most likely to be used in non-conversational modes.

Option2

Each option is specified by a dash followed by a single or double character. Some options take a second argument as shown below, where *N* is a number (real or integer), and *S* is a word (file name in most cases). The options with an asterisk are meaningful only for spliced alignment.

option	default	range	description
`-A`*N*	0	0-4	Select algorithm. *N*=0: `aln`'s choice; *N*=1: DP-AF; *N*=2: DP-DA; *N*>=3: CL-AF. A value of other than 0 is not recommended.
`-F`*N*	1	1-9	Output format. N=1-5: native; *N*=6: Phylip; *N*=7: GCG; *N*=8: GDE; *N*=9: Concatenated FASTA.
`-l`*N*	60	>8	Set line width = *N*.
`-L[`*N*`]`	0	0-15	Control penalty values given to terminal gaps.
`-LS`			Smith-Waterman-Gotoh local alignment. (currently disabled)
`-M`*N*	1	>0	Multiple outputs when *N* > 1. With `–LS` option, *N*=2 does but *N*=3 does not suppresses repetitious outputs.
`-m`*S*			Amino acid or nucleotide substitution matrix.
`-n`			Report only statistics rather than an alignment.
`-o`*S*			In the interactive mode, *S* indicates the default directory where the result is written. In other modes, *S* is the file name itself. If this option is not set, the output goes to `stdout`.
`-O`*N*			Output mode. See the 'Gene structure prediction'.
`-pq`			suppress output of additional messages.
`-r`			Don't remove intermediate files (with `-a` or `-b`).
`-R`*N*	0	>=0	The number of random sequence pairs used for a shuffle test.
`-s`*S*			Set the default path where sequence files reside.
`-T`*S*	ALN_TAB		The `'table'` directory.
`-u`*N*	2	>=0	Set the gap-extension penalty u = *N*.
`-v`*N*	9	>=0	Set the gap-opening penalty v = *N*.
`-w`*N*	100	>=0	Set the width around the main diagonals shldr = *N*.
`-yb`*N*	0		Add *N* to each element of an amino acid exchange matrix.
`-yc`*N**	10	>=0	Define the "neighborhood" exon range adjacent to EI boundary.
`-ye`*N*	0	>=0	Additional penalty assigned to each deletion.
`-yi`*N**	68	>0	Intron penalty.
`-yj`*N*	0	>=0	The u' value in a DA gap penalty. Must be 0 <= u' < u.
`-yk`*N*	10	>0	The K value in a DA gap penalty.
`-yl`*N**	1	1-3	*N*=1: assume the absence of introns; *N*=2: assume the presence of

			introns, and use the AF penalty: *N*=3: assume the presence of introns, and use the DA penalty.
-ym*N*	2	>=0	Similarity measure between identical nucleotides.
-yn*N*	-2	<=0	Similarity measure between different nucleotides.
-yo*N**	100	>=0	Penalty assigned to a premature termination codon.
-yp*N*	250	0-300	The 'PAM' value of the mutation data matrix.
-yu*N*	2	>=0	Same as -u*N*.
-yv*N*	9	>=0	Same as -v*N*.
-yw*N*	100	>=0	Same as -w*N*.
-yx*N**	20	>=0	Penalty for a frameshift.
-yy*N**	16	>=0	Relative contribution rate of translational and splicing boundary signals to the total alignment scores.
-yz*N**	2	>=0	Relative contribution rate of coding potential to the total alignment scores.
-yB*N*	0	>=0	A bonus given a matched intron position
-yE*N**	18	>=0	Shortest exon length.
-yI*S**			Set the length-dependent intron penalty function
-yJ*N**	0.001	>=0	Bonus given to a matched intron positions.
-yL*N**	30	>=0	Lower limit of intron length
-yT*N**	0	>=0	Assumes as a gene end if the terminal gap is shorter than this value.
-yY*N**	1	>=0	Relative contribution of length-dependent term of intron penalty.

The following options are used for fast heuristic alignment. The options with upper-case letter after x are used in the first stage of seed search, while those with lower-case letter are used in the second stage when –Q2 or –Q3 is set.

-Q*N*	0	0-3	*N*=0: without fast mode, *N*>=1: *N* levels of seed search
-xB*S*			Discontinuous bit pattern. Default pattern if *S* is empty
-xD*N*	10		Maximum length of direct repeat encompassing EI boundaries
-xE*N*	120		A bonus given HSP matching a region near gene end
-xG*N*	4		Score gain per a single hit
-xH*N*	35,45		Lower limit of the score of HSP to be considered
-xK*N*	5,8		Tuple size
-xN*N*	128		Maximum number of HSPs to be stored
-xR*N*	16,4		Alphabet size*
-xX*N*	0		Value for X-drop off; no extension of HSP when 0
-xb*S*	“”	"1..1"	Discontinuous bit pattern. Default pattern if *S* is empty
-xd*N*	0		Maximum length of direct repeat encompassing EI boundaries
-xe*N*	120		A bonus given HSP matching a region near gene end
-xg*N*	4		Score gain per a single hit
-xh*N*	35,35		Lower limit of the score of HSP to be considered
-xk*N*	4,5		Tuple size
-xn*N*	128		Maximum number of HSPs to be stored
-xr*N*	12,4		Alphabet size*
-xx*N*	0		Value for X-drop off; no extension of HSP when 0

* Alphabet sizes of $6 \leqq N \leqq 16$ (even number) indicate reduced amino acid codes. For example, the 20 amino acids are grouped and reduced into six elements such that {C}, {SJTPAG}, {NDEQBZ}, {HRK}, {MILV}, and {FYW} with *N*=6.

With option -p, you are asked to input new parameter values just as with 'p' + 'P' commands in

the Interactive mode. Prompt and echo are simply for your convenience.

[3] Batch mode

This mode enables several calculations with a single command. Given a list of sequence names in a `catalog` file, you can compare various combinations of sequences according to option1. Option1's are mutually exclusive; you should choose one of them. In a special case, you get a multiple-sequence alignment by recurrently applying pairwise alignment to groups of sequences. In this case, you must prepare a special format of tree file instead of a `catalog` file. If `:catalog` in the following list is omitted, individual sequences given in the arguments are compared with one another.

Option1

`-acatalog`	Construct multiple-sequence alignment simply by adding on in the order of the file names in the catalog
`-btree**`	Construct multiple-sequence alignment by the progressive method
`-ia:catalog`	(2i-1)-th sequence vs. 2i-th sequence (i > 0).
`-ie:catalog`	All pairs of sequences in the catalog.
`-if:catalog`	First sequence vs. all other sequences in the catalog.
`-ig:catalog`	Every sequence in group1 vs. every sequence in group2. The two groups are separated by a blank line.
`-ii:catalog`	Self comparisons. Not useful with aln.
`-ij:catalog`	(2i-1)-th sequence vs. 2i-th sequence (i > 0) (same as -ia).
`-il:catalog`	Last sequence vs. all other sequences in the catalog.
`-ip File1 File2`	parallel read from File1 (e.g. genomic) and File2 (e.g. protein)

**Notes

A tree file appropriate for an input to aln can be made by upg/nj included in the distribution version. You can also perform the equivalent calculations with the prrn -b option without generating intermediate files. The tree topology may be specified in the 'New Hampshire Standard' (Newick) format, in which the OTU names correspond to the sequence file names to be aligned. Alternatively, you may directly indicate the order of calculation as shown in the following example.

```
AA   ---------
1             |
BB            |
0             |     Content of a tree file.
CC            |     AA,...EE  show sequence file names.
3             |     Numbers indicate the order of calculations.
DD            |
2             |
EE   --------
```

In this example, `BB` and `CC` are aligned first, and the result is stored in a file named `_s0` in the current directory. `_s0` is then aligned with `AA` to give `_s1`. Alignment between `DD` and `EE` is stored in `_s2`, which is further aligned with `_s1`, to give `_s3`, the final result. The number between sequence names indicates the distance to the internal node from the leaf furthest from the root. Since the present implementation uses only topological information, you need not care about precise distance values. These intermediate files `_s..` are deleted automatically unless you give the `'-r'` command line option, and the final result is written in either the file specified by `-o` option or

`'./ALN'` if not specified. A second run of `'aln -b tree'` may replace the previous files, so that the final result (`./ALN` or `_s3` with `-r` option in this example) should be copied or renamed before another run of **`aln`**.

The strategy adopted here is the so-called 'progressive method' similar to those proposed by Hogeweg and Hesper (1984) *J. Mol. Evol.* **20**, 175-186, Feng and Doolittle (1987) J. Mol. Evol. **25**, 351-360, and others. This strategy rapidly generates a reasonably good alignment, which may be further refined by **`prrn`**. Please refer to the accompanying document `'prrn.doc'` for further details.

`Catalog` file

Each line in a `catalog` file has the same format as that used for input of a sequence name under Conversational mode:

```
[[path][filename]] [start] [end] [attribute]
```

With the `-g` option, a blank line separates two groups, otherwise a blank line (optional) indicates the end of the list.

Gene-structure prediction

`Aln` also supports spliced alignment between a genomic sequence with a nucleotide sequence, a protein sequence, or a protein profile. To run the program in this mode, you must confirm that the following five files with asterisk are present in the 'table' directory. Species specific (ss) files may be present in each subdirectory.

```
AlnParam (ss)
Branch (ss)
CodePotTab*
Intron53 (optional)
IntronPotTab (ss)
Splice3* (ss)
Splice5* (ss)
TransInit*
TransTerm*
```

These files are included in the distribution package. Sixteen one subdirectories in the `table` directory contain species-specific parameters. You can select the species closest to your genome by the -Txxxxxx option. Otherwise, the default parameter set is used. The default files are adjusted for ‘generic species’ (average of several representative species).

The option `-yl2` or `-yl3`, together with `G` attribute, tells the program that the nucleotide sequence may contain introns which are skipped in alignment. As a consequence, the exon/intron structure of a genomic sequence can be predicted [6-8]. The `G` attribute may be omitted in genome vs. protein alignment but indispensable for genome vs. cDNA sequence alignment. An affine (`-yl2`) or a double affine (`-yl3`) function may be chosen to penalize a gap (indel) at the translated amino-acid sequence or `'mature'` nucleotide sequence level. The `-L` option should also usually be set to perform semi-global alignment. The standard command would look like:

```
% aln -yl2 (or -yl3) -L -ON 'DNA N1 N2 G' reference
```
(sense strand)

```
% aln -yl2 (or -yl3) -L -ON 'DNA N1 N2 G <' reference
```
(antisense strand)

where *`N1`* and *`N2`* specify the region in the genomic DNA sequence within which the objective gene may reside, and 'reference' indicates a mRNA sequence or protein sequence or profile used as the

reference. The `-ON` option controls the output:

`-O0:` GFF gene format
`-O1:` Alignment between the DNA sequence and the mRNA or protein
`-O2:` GFF match format
`-O3:` Bed format
`-O4:` Exon-oriented information such as lengths, %matches, and boundary coordinates
`-O5:` Intron-oriented information such as lengths, boundary coordinates and signal strengths
`-O6:` Concatenated exonic sequence i.e. predicted 'cDNA' sequence
`-O7:` Translated amino acid sequence

With -O6 and -O7, boundary signal strengths are also reported after the end of the sequence. In addition, translation is conducted after correction for potential frameshift errors with -O7.

The accompanying file 'Exam' contains some examples for the application of aln to test data with several different option settings.

`Aln` and `spaln`

The following commands are nearly equivalent:

```
% spaln -Q0 'DNA N1 N2' reference
% aln -yl2 -L -O4 –yp150 'DNA N1 N2 G' reference
```

However, there are several differences between `spaln` and spliced alignment mode of `aln`.

1. `Aln` requires `-yl2 -L` options to perform spliced alignment while `spaln` does not.
2. The default output mode is `–O1` with `aln` whereas `–O4` with `spaln`.
3. `Spaln` cannot use multiple alignment as the reference while `aln` can.
4. `Spaln` has the genome mapping phase but `aln` does not.
5. `Spaln` has quick modes whereas the quick modes of `aln` are currently disabled.
6. Score values are floating point in `aln` whereas integer in `spaln` and 10 times larger than those of `aln`.
7. The default PAM level is 150 in `spaln` whereas 250 in `aln`.

Osamu Gotoh <o.gotoh@aist.go.jp>
Last modified on 2014-06-02