ProAlign, a probabilistic multiple alignment program

(version 0.5a0)

Ari Löytynoja & Michel C. Milinkovitch

Unit of Evolutionary Genetics, Free University of Brussels (ULB), Belgium.
http://www.ulb.ac.be/sciences/ueg/

SUMMARY.

Optimal multiple alignment of molecular sequences is computationally impractical for more than a few sequences, and heuristics have been developed in the form of progressive alignment methods (e.g. ClustalW from Thompson, Higgins & Gibson 1994). These algorithms (i) construct a distance matrix of all N(N-1)/2 pairs followed by approximate conversion of similarity scores to evolutionary distances, (ii) build a guide tree with NJ clustering, and (iii) progressively perform pairwise alignments of sequences and profiles at nodes in order of decreasing similarity.

Generally multiple alignment algorithms provide a single solution without any measure of its local or global reliability. We have earlier developed program SOAP that uses a parameter perturbation approach to define the unstable blocks, and computes the consensus among several ClustalW-alignments. Although the method is intuitive, it lacks statistical background and may be biased on its ability to recognize the low quality regions.

Durbin et al. (1998) defined algorithms for a probabilistic sequence alignment using hidden Markov models (HMM). They described models for the most probable state path (Viterbi path) and the forward and backward full probability recursion of an affine-gap-score alignment of two sequences. Unlike traditional methods, their HMM algorithms provide tools for probabilistic comparison of different solutions.

We have further developed the HMMs of Durbin et al. and generalized them for probabilistic multiple alignment of nucleotide and amino-acid sequences. We use a progressive method, such that multiple alignment is created stepwise by performing pairwise alignments in the nodes of a guide tree. Sequences are described with vectors of character probabilities, and each pairwise alignment reconstructs the ancestral (parent) sequence by computing the probabilities of different characters according to an evolutionary model. As the process is completely probabilistic, it allows sampling alignments either by tie-breaking or from posterior probabilities.

The novel algorithms are implemented in the software "ProAlign" that includes a graphical interface allowing to (i) perform alignments of nucleotide or amino-acid sequences, (ii) view the quality of solutions, (iii) filter the unreliable alignment regions and (iv) export alignments to other softwares. As far as simulated nucleotide datasets are concerned, ProAlign performs better than the traditional alignment methods and, in many cases, the result can be further improved by tie-breaking and/or sampling of sub-optimal alignments. The probability of the total alignment correlates only weakly with the number of correctly aligned columns, and selecting the absolute best alignment can be difficult. However, the minimum posterior probability of each aligned column is an efficient criterion for detecting and removing the most unreliably-aligned sites.

PROGRAM AVAILABILITY.

An up-to-date version of this program manual can be found under http://www.ulb.ac.be/sciences/ueg/. The main parts of this document are included in the Tutorial (pdf-format).

ProAlign is written in the platform-independent programming language Java, and the same jar-file should run on any Linux, Windows or Mac OSX computer equipped with Java2 v.1.2 or higher (on Linux IBM Java2 v.1.3, and on Windows Java2 v.1.4 recommended). The program should usable on a standard personal computer, as the multiple alignment of 20 sequences of ~500 nucleotides takes only 21 seconds on a Intel 1GMHz Linux-computer.

We view ProAlign, which is distributed free of charge, as a service to our community. Hence, (i) support cannot be guaranteed (but will be provided as fast/best as possible) and (ii) we do not take any responsibility on erroneous interpretation of data and analyses resulting from the use of our program. Obviously, debugging and further development of ProAlign will require alpha-testing and feedback from you. We wish any user to report crashes, apparent errors, or other anomalous results by an e-mail to Ari Löytynoja (aloytyno@ulb.ac.be).

The paper describing the method is accepted for publication in Bioinformatics. You can download the manuscipt in pdf-format from the program home page. To publish results that have been obtained with ProAlign, we request the program be cited as "Ari Löytynoja & Michel C. Milinkovitch, 2003. A hidden Markov method for progressive multiple alignment. Bioinformatics, in press".

DOWNLOADING THE PROGRAM.

The program jar-file can be found in the ProAlign download site. To keep count on real users we insist that you register yourself using the form at the ProAlign homepage. If you have filled the form, and do not receive anything in a couple of hours, please, repeat the registration or write an email to aloytyno@ulb.ac.be.

INSTALLATION AND RUNNING.

To install ProAlign on your computer, download the ProAlign*jar-file and..

A) Linux Click the "ProAlign*.jar" icon and give the application "java -jar", or start the program manually by typing a command 'java -jar ProAlign*.jar'.

B) MacOSX Download the "clustalw"-file. Click the "ProAlign*.jar" icon, or start the program manually by typing a command 'java -jar ProAlign*.jar'.

C) Windows (95/98/NT/2k) Download the "clustalw.dll"-file. Click the "ProAlign*.jar" icon, or start the program manually by typing a command 'java -jar ProAlign*.jar'.

(* is replaced by the current version number).

ProAlign either creates the alignment guide tree by itself, reads a tree from a file, or calls a native ClustalW program do that. ClustalW for different platforms can be downloaded e.g. from EBI. Some binaries are also provided at the ProAlign download site. The most current releases of Java can be downloaded from the Sun home page, http://java.sun.com/products.

The native ClustalW-file (Linux & MacOSX) should be located so that the path to it does not contain blank spaces or special characters. The path to the ClustalW-program (Linux & MacOSX), or the shared object file (Windows) is specified during the run of the program.

ProAlign is also integrated in the program SOAP, such that it is convenient e.g. to sample alignments and compute their consensus. Find more information at the program home page at http://evol-linux1.ulb.ac.be/ueg/SOAP/.

MEMORY ISSUES.

ProAlign is reasonably fast as it only aligns a band around the diagonal of the pairwise alignment matrix. It requires that the alignment of the two neighboring sequences does not contain long gaps, and the alignment path stays inside the band. If the cumulative length of the gaps (excluding the trailing gaps caused by the

missing data) in one sequence gets high (i.e. more than 50 characters), the default band width may need be to increased. However, it should be noticed that the computation time and memory requirements grow linearly according to the band width.

The alignment of amino-acid data sets or sequences of different length (with a large band) requires a plenty of memory. Irrespective of the actual amount of available memory, the Java Virtual Machine, by default, sets an upper limit of 64MB for the allocation space. That is hardly enough for alignment of 20 sequences of 750 aa, and may need to be increased.

The maximum size of allocation memory can be overrun with flag '-Xmx' when the program is launched from command line, e.g. command

```
java -Xmx128m -jar ProAlign_0.3a3.jar
```

sets the maximum to 128MB. Currently there seems not to be any way to set that value permanently, or without the usage of command line. More information about the JVM memory allocation can be found at Java documentation.

USING THE PROGRAM.

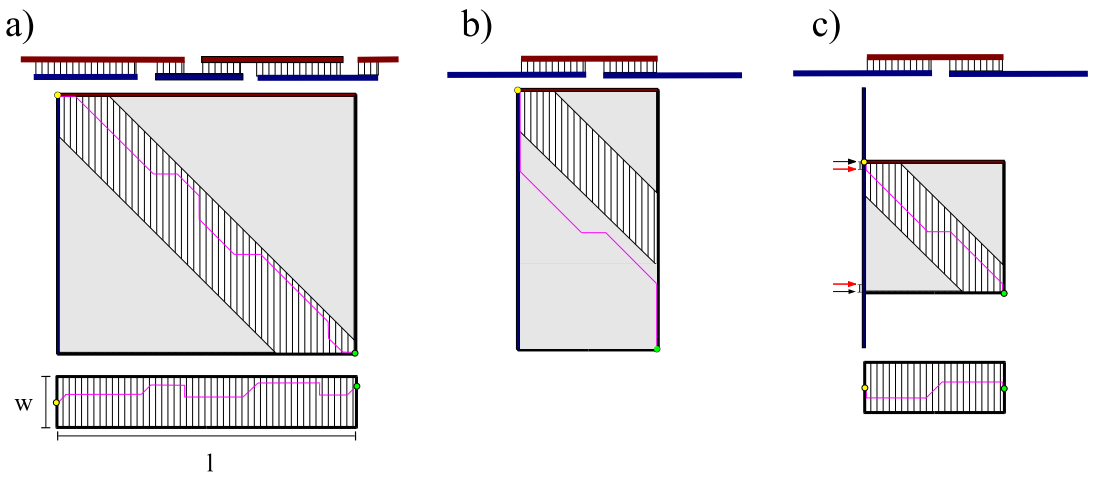
A normal analysis starts with the import of nucleotide or amino-acid sequences in Fasta-, PIR-, Phylip-(interleaved), MSF-, or Nexus-format. If the sequences are aligned, the gap signs are automatically removed. In the case that a guide tree in ClustalW- or Phylip-format exists, it can be imported, otherwise, ProAlign creates a guide tree, or a native ClustalW program is called to do that. The algorithm assumes bifurcating tree, and if necessary, the program roots the guide tree at the midpoint. A specific outgroup can be defined by rooting the guide with another program (such as Treeview, or Njplot), and importing the tree to ProAlign.

A ProAlign guide tree is created with the NJ-clustering after performing all the pairwise alignments and creating a rough distance table. The pairwise alignments are computed using Gotoh's algorithm without any memory or computation time shortcuts. By default ProAlign does not penalize terminal gaps (often caused by missing data), but the option can be changed in the program settings. For the alignment of nucleotide sequences the standard substitution score table (from ClustalW) is provided, whereas the amino-acid alignments (all comparisons with the same substitution scores) can be performed using a range of different PAM-tables.

The pairwise distances are computed from the proportion of identities after excluding the terminal gaps. By default ProAlign corrects the nucleotide and amino-acid distances with the JC and Kimura methods, respectively, and estimates the branch lengths longer than ClustalW does. The distance correction ("Multiple hits") can be unchecked in the program settings, which should give branch length estimates more similar (although not identical) to those of ClustalW. As negative branch lengths are not allowed in the guide tree, the algorithm sets the branch lengths to their absolute values. If the sequences are already aligned, the pairwise distances are naturally computed from the multiple alignment. Thus, the multiple alignment step can easily be iterated with an improved guide tree.

The multiple alignment is created by performing a pairwise alignment at each node of the guide tree in the order of decreasing similarity. The pairwise alignment of two sequences (either terminal or internal) reconstructs their common ancestor, and that sequence is further used to proceed with the alignment towards the root of the guide tree. The algorithm uses the branch length information from the guide tree to compute a substitution table of its own for each branch, and thus automatically adjusts the substitution probabilities according to the relatedness of the sequences.

When performing a pairwise alignment, ProAlign does no compute the complete matrices but only a band on the diagonal. This is schematically shown in figure *a* where the red and blue bars above represent an optimal alignment of two sequences. The square shows the complete pairwise alignment matrix where the optimal path (in magenta) proceeds from the yellow dot (start) to the green dot (end). The diagonal moves are matches whereas the horizontal and vertical moves create gaps. ProAlign ignores the gray area, and only aligns the white hatched band in the middle, as shown below. This drops the memory and computation time from $(O)l^2$ to $(O)wl$ but requires that the optimal alignment between the sequences does not contain long internal gaps.



If one of the sequences lacks a terminal region (e.g. due to incomplete sequencing), the optimal path does not stay in the middle band, and the alignment fails (figure *b*). However, if the shared region is continuous, ProAlign can search its start and end points with the traditional pairwise algorithm (the red arrows in figure *c*), and perform the probabilistic alignment only for the mid-part. The maximum allowed length of the trailing sequence before its exclusion from the alignment can be defined in the program settings. As the traditional pairwise alignment algorithm may not be exact when defining the end point of the pairwise algorithm, these adjacent regions are also included in the probabilistic pairwise alignment (the distance bars between the black and red arrows in the figure). The cut-out trailing sequences are handled as gaps, and the character probabilities for the ancestral sequence are defined accordingly.

It should be noticed that penalizing the terminal gaps may have an effect when defining the start and end points of the shared region. Also, high values for the allowed length of the trailing sequence (before performing a cut) will decrease the effective width of the band by causing long gaps right in the start (figure *c* below). If the sequences are complete and the band width is reasonably high, the correction of trailing sequences can turned off (done in the program settings).

After finishing the multiple alignment, ProAlign displays the quality of each pairwise alignment with a color code that roughly reflects the posterior probability of the alignment at that site: bright yellow means high probability, red low probability. The color code is drawn between the two aligned subsets of sequences at the level of the corresponding internal node on the left panel. A click in the color box will give information on the character probabilities at the ancestral sequence.

For fast browsing a quality plot can be opened by a click at the tree nodes or by choosing the option in the “Results”-menu. Any sequences or alignment sites can be selected/unselected manually from the set of exported data, or the sites having posterior probability lower than the threshold value can be filtered out automatically. The complete alignment can be saved and re-opened in ProAlign-format, or it can be exported in PIR/Phylip/MSF/Nexus-formats to be used elsewhere.

Many of the alignment model parameters can be set manually, and some of them can be estimated from the data. As the method allows sampling of alignments from their posterior probabilities, a window of its own is provided for that. A user can there set the alignment path search strategy (“best” or “sample”), and define where to save the log information and/or alignments. The options at the three menus and two additional windows are explained more detailed below.

File-menu:

About ProAlign: some information.

Open alignment: open a binary format ProAlign-alignment. The alignment is recovered exactly as it was.

Save alignment: save a ProAlign-alignment in the binary format. The format is not readable for any other programs but ProAlign itself.

Import data: import sequence data in Fasta-, PIR-, Phylip- (interleaved), MSF-, or Nexus-format. Ambiguity characters are the same that are accepted by program PAUP*. Gap signs are removed.

Import guide tree: import a ClustalW- or Phylip-guide tree. A guide tree can not be imported before the data has been selected. If the tree is unrooted, midpoint-rooting is performed.

Export Fasta: export selected data in PIR-format. Unselected taxa/columns are excluded from the exported alignment. Posterior probability of sites is saved in a separate file.

Export Phylip: export selected data in Phylip interleaved-format. Unselected taxa/columns are excluded from the exported alignment.

Export Nexus: export selected data in Nexus-format. A complete alignment is exported but unselected taxa/columns are removed by commands in the Paup-block. Posterior probability of sites is saved as a 'weight set' in Assumptions-block.

Save guide tree: save the guide tree.

Exit: exit .

Font-menu:

Size: font size.

Horizontal space: space between characters.

Vertical space: space between characters.

Align-menu:

Do ClustalW-guide tree: call a native ClustalW-program to create a guide for the data set. The correct path to ClustalW-program has to be set, cf. 'Set Parameters' below. This option may take some time.

Do ProAlign-guide tree: if sequences are not aligned, do all pairwise alignments using the current settings for pairwise alignment, compute a distance matrix, and create a guide tree using the NJ-clustering. If sequences are aligned, compute pairwise distances from the multiple alignment, create a guide tree, and remove gaps from the alignment.

Do multiple alignment: align the data according to the guide tree.

Sample alignments: cf. 'Sample alignments-window' below.

Remove all gaps: remove all gaps but keep the data and the guide tree. Removing gaps is necessary before a new alignment.

Set parameters: cf. 'Set Parameters-window' below.

Result-menu:

Plot min. probability: plot minimum posterior probability for each site. A click on the plot will focus the main window around the region of interest and will display some information on the text field.

Filter sites: unselect columns that minimum posterior probability lower than the selected value. The set of unselected columns can be modified manually.

Set parameters-window:
Character frequencies for nucleotide data:

Char. 'gap': lock/unlock the frequency of '-'.

Char. Set: set the background frequencies for characters. If 'gap' is locked, take the current values and set '-' to sum up to 1. If gap is unlocked, take the current values and scale them to sum up to 1.

Char. Estimate: estimate the background frequencies for characters from the data.

Character frequencies for amino-acid data:

Gap frequency: the background frequency for "gap". It is considered 21st amino-acid, and the frequency of other amino-acids is scaled accordingly.

Gap probability: the probability of substitution between a "gap" and a non-gap character. The probabilities between other amino-acids is scaled accordingly.

Pairwise alignment (the guide tree computation/trailing sequences):

pam-table: the substitution score table for amino-acid data.

Gap opening: the gap opening penalty.

Gap extension: the gap extension penalty.

Terminal gaps: penalize terminal gaps.

Trailing sequences: allow and correct the trailing sequences.

Max allow: maximum length of the trailing sequence before exclusion. The value should be less than a half of the band width.

Multiple hits: correct the pairwise distances for multiple hits.

HMM parameters:

Delta: the probability of moving to an insert state ("gap opening penalty"). A low "delta" corresponds to a high gap opening penalty.

Epsilon: the probability of staying at an insert state ("gap extension penalty"). A low "epsilon" corresponds to a high gap extension penalty.

Estimate: estimate "delta" and "epsilon" from the data. The guide tree is required.

Traceback route: select the best or sample a path from posterior probabilities. (also for "best", in a case of two equally good paths, one of the paths is chosen randomly).

Write log: a log of the analysis. Currently meant for debugging. The default file is "proalign.log".

Band width: the width of the diagonal alignment band. If sequences are of very different length, the band width may need to be increased. High values are warned but accepted.

ClustalW: the path to ClustalW-program. The default is 'clustalw', the first program named like that on the execution path.

Temp folder: the path to a folder where the temporary files for ClustalW are stored. The default is '/tmp' on unix, 'C:\TEMP' on Windows. On Unix there should not be any reason to change this, on Windows the location may vary.

OK: confirm chosen values.

Sample alignments-window:

Permutations: set the number of multiple alignments created.

Write results: write the results as comma separated values in a file. The default file is 'sample_result.csv'.

Save .paa: save the alignments in ProAlign-format.

Save .nex: save the alignments in Nexus-format.

Save .fas: save the alignments in Fasta-format.

Traceback route: select the best or sample a path from posterior probabilities. Sampling alignments with "select best" checked can sometimes found a globally better alignment.

Start: start sampling alignments. The best scoring alignment is updated in the alignment window.

Stop: stop after the current alignment.

For a multiple alignment, there are also some additional actions available with a mouse click:

guide tree

- a click at a guide tree node will open a new window plotting the quality of the alignment on that node.
- a click at a guide tree node with the 'Shift'-button down will open a new window instead of updating the old plot.

quality plot

- a click on the quality plot will focus the main window around the region of interest.

sequence names

- a click at a sequence name will select/unselect that sequence from the exported alignment.
- two clicks with the -button pressed will select a region.

sequence data

- a click at the column ruler will select/unselect that column from the exported alignment.
- two clicks with the -button pressed will select a region.
- a click at a color box between any two characters will display some information of that site on the text field.

COMMAND LINE.

Limited usage of the program is possible from command line. Recognized options are:

-nogui (force command line)
-seqfile=<sequence file>
-treefile=<tree file>
-newtree (compute a new guide tree)
-sample (sample traceback path; if not given, Viterbi is chosen)
-delta=<HMM delta> or -delta=estimate (if not given, default is used)
-epsilon=<HMM epsilon> or -epsilon=estimate (if not given, default is used)
-gapfreq=<gap frequency> (if not given, default is used)
-gapprob=<gap substitution probability> (if not given, default is used)
-bwidth=<search band width> (if not given, default is used)
-distscale=<distance scale factor> (for branch lengths; default is 1)
-nocorrection (no correction for pairwise distances on guide tree computation)
-notrailing (no trailing sequence corrcection)
-trailing=<trailing sequence correction length> (for missing ends)
-penalize=true, or =false (penalize end gaps on pairwise alignments for guide tree; default is true)
-writemean (write mean posterior probability of sites)
-writeall (write posterior probability of each node)
-writeroot (write root node character probabilities)
-wag (use WAG probability table; default value)
-dayhoff (use Dayhoff probability table)
-jtt (use JTT probability table)
-outfile=<alignment file>
-outformat=pir, -outformat=msf, -outformat=phylip, or -outformat=nexus (output format)
-quiet (no log)

For example, following is a valid command:

```
java -jar ProAlign_0.5a0.jar -nogui -seqfile=prim.fas -treefile=prim.tre
-outfile=prim.pir -outformat=pir
```

METHODS.

We owe a lot to Durbin et al. (1998) who defined the basic algorithms for a probabilistic sequence alignment using hidden Markov models.

The paper describing our improved methods is accepted for publication in Bioinformatics. You can download the manuscipt in pdf-format from the program home page. The program source code is available from aloytyno@ulb.ac.be.

FEEDBACK.

We are happy to get positive feedback on our work, and we try to cope with the negative one, too. Please, write an e-mail to aloytyno@ulb.ac.be if you have detected bugs in the program or you have some comments how it could be made better.

When you report a bug, please, provide us with sufficient information: (1) a description of the problem, (2) the error messages given by JVM (may require starting the program from a terminal), (3) the file 'proalign.log', (4) the data set, and (5) the guide tree that cause the problem.

REFERENCES.

J. D. Thompson and D. G. Higgins and T. J. Gibson, 1994. 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: 4673-80.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, UK.

Ari Löytynoja and Michel C. Milinkovitch, 2001. SOAP, cleaning multiple alignments from unstable blocks. Bioinformatics, 17:573-574.