ProDA User Manual

Version 1.0

Algorithm developed by

Tu Minh Phuong, Chuong B. Do, Robert C. Edgar, and Serafim Batzoglou.

Software written by

Tu Minh Phuong and Chuong B. Do.

Manual written by

Tu Minh Phuong.

Introduction

ProDA (Protein Domain Aligner) is public domain software for generating multiple alignments of protein sequences with repeats and rearrangements, e.g. proteins with multiple domains.

Given a set of protein sequences as input, ProDA first finds local pairwise alignments between all pairs of sequences, then forms blocks of alignable sequence fragments, and finally generates multiple alignments of the blocks. ProDA relies on many techniques used in Probcons (http://probcons.stanford.edu), a recent multiple aligner that shows high accuracy in a number of popular benchmarks.

Algorithm outline

1. Compute local pairwise alignments for each pair of sequences using either Viterbi or posterior decoding.
2. Infer repeats from pairwise alignments.
3. Generate a block of *alignable sequence fragments.*
4. Construct guide tree using *expected accuracies* and adjustment of the block.
5. *Progressively align* the block using the guide tree.
6. Extract final alignments from block alignment.
7. Remove used pairwise alignments.

Installation

The ProDA source code (proda_x_x.tar.gz) can be obtained from http://proda.stanford.edu

To install and use ProDA,

1. Decompress the files

```
gunzip proda_x_x.tar.gz
tar -xvf proda_x_x.tar
```

2. A subdirectory called `proda/` will be created in the current directory
3. Change to `proda/` directory and the the ProDA executable

```
cd proda
make
```

4. Align the sequences in file `input` and send the result to file `output`

```
./proda input > output
```

Input format

Proda accepts input files in the MFA format. The MFA format is specified below:

- The MFA format consists of multiple sequences.

- Each sequence in the MFA format begins with a single-line description, followed by lines of sequence data.

- The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column.

Output format

For a set of input sequences, Proda usually outputs several blocks in turn, each consists of alignable sequence fragments. Each block is followed by its multiple alignment.

A block is specified by listing its sequence fragments. Each fragment is output as *sequence_name(start-end)*, where *sequence_name* is the name of the original sequence and *start* and *end* are positions at which the fragment begins and ends respectively.

Proda produces block alignments in the ClustalW (ALN) format described below:

- The ClustalW format consists of a single header line followed by sequence data in blocks of 50 alignment positions.

- Each block consists of

    - one line of data for each of the sequences in the alignment; in particular, the name of the sequence

    - 50 characters of the alignment

    - one annotation line indicating fully conserved (*), strongly-conserved (:), or weakly-conserved columns (.)

FASTA format for output

If the `-fasta` option is specified, then, in addition to regular output, ProDA produces a file containing block alignments in the FASTA format. The output file has the same name as the first input file and extension "*.fasta*". Two consecutive block alignments are separated by a line containing character '#'.

The FASTA format is described below:

- The FASTA format consists of all the sequences given in the input files.

- Each sequence in the FASTA format begins with a single-line description, followed by lines of sequence data.

- The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column.

- Aligned residues are in upper case, unaligned residues are in lower case.

Since a final alignment contains each sequence only once, this option should be used only if input sequences do not contain repeats.

Example

Running ProDA on the following input file (containing protein sequences with SH3 and SH2 domains)

```
>GRB2_HUMAN
MEAIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAELNGKDGFIPKNYIEMKPHPW
FFGKIPRAKAEEMLSKQRHDGAFLIRESESAPGDFSLSVKFGNDVQHFKVLRDGAGKYFL
WVVKFNSLNELVDYHRSTSVSRNQQIFLRDIEQVPQQPTYVQALFDFDPQEDGELGFRRG
DFIHVMDNSDPNWWKGACHGQTGMFPRNYVTPVNRNV
>MATK_HUMAN
MAGRGSLVSWRAFHGCDSAEELPRVSPRFLRAWHPPPVSARMPTRRWAPGTQCITKCEHT
RPKPGELAFRKGDVVTILEACENKSWYRVKHHTSGQEGLLAAGALREREALSADPKLSLM
PWFHGKISGQEAVQQLQPPEDGLFLVRESARHPGDYVLCVSFGRDVIHYRVLHRDGHLTI
DEAVFFCNLMDMVEHYSKDKGAICTKLVRPKRKHGTKSAEEELARAGWLLNLQHLTLGAQ
IGEGEFGAVLQGEYLGQKVAVKNIKCDVTAQAFLDETAVMTKMQHENLVRLLGVILHQGL
YIVMEHVSKGNLVNFLRTRGRALVNTAQLLQFSLHVAEGMEYLESKKLVHRDLAARNILV
SEDLVAKVSDFGLAKAERKGLDSSRLPVKWTAPEALKHGKFTSKSDVWSFGVLLWEVFSY
GRAPYPKMSLKEVSEAVEKGYRMEPPEGCPGPVHVLMSSCWEAEPARRPPFRKLAEKLAR
ELRSAGAPASVSGQDADGSTSPRSQEP
>CRKL_HUMAN
MSSARFDSSDRSAWYMGPVSRQEAQTRLQGQRHGMFLVRDSSTCPGDYVLSVSENSRVSH
YIINSLPNRRFKIGDQEFDHLPALLEFYKIHYLDTTTLIEPAPRYPSPPMGSVSAPNLPT
AEDNLEYVRTLYDFPGNDAEDLPFKKGEILVIIEKPEEQWWSARNKDGRVGMIPVPYVEK
LVRSSPHGKHGNRNSNSYGIPEPAHAYAQPQTTTPLPAVSGSPGAAITPLPSTQNGPVFA
KAIQKRVPCAYDKTALALEVGDIVKVTRMNINGQWEGEVNGRKGLFPFTHVKIFDPQNPD
ENE
```

will generate the following output (with –posterior option, see instructions below):

```
MATK_HUMAN(51-100) GRB2_HUMAN(1-48) GRB2_HUMAN(159-205)
CRKL_HUMAN(126-173) CRKL_HUMAN(238-286)

MATK_HUMAN    TQ--CITKCEHTRPKPGELAFRKGDVVTILE-ACENKSWYRV-KHHTSGQEGLL
GRB2_HUMAN    ME--AIAKYDFKATADDELSFKRGDILKVLNEECDQN-WYKA-E--LNGKDGFI
GRB2_HUMAN    TY--VQALFDFDPQEDGELGFRRGDFIHVMD-NSDPN-WWKG-A--CHGQTGMF
CRKL_HUMAN    EY--VRTLYDFPGNDAEDLPFKKGEILVIIE-KPEEQ-WWSARN--KDGRVGMI
CRKL_HUMAN    VFAKAIQKRVPCAYDKTALALEVGDIVKVTR-MNINGQW-EG-E--VNGRKGLF
                                * :. *:.: :         *         *: *::

GRB2_HUMAN(50-146) MATK_HUMAN(112-207) CRKL_HUMAN(4-99)

GRB2_HUMAN    KNYIEMKPHPWFFGKIPRAKAEEMLSKQRHDGAFLIRESESAPGDFSLSVKFGNDVQHFK
MATK_HUMAN    SADPKLSLMPWFHGKISGQEAVQQLQPPED-GLFLVRESARHPGDYVLCVSFGRDVIHYR
CRKL_HUMAN    ARFDSSDRSAWYMGPVSRQEAQTRLQGQRH-GMFLVRDSSTCPGDYVLSVSENSRVSHYI
                  . .  .*: * :.  :*   *.  .. * **:*:*   ***: *.*. .  * *:

GRB2_HUMAN    VLRDGAGKYFLWVVKFNSLNELVDYHRSTSVSRNQQI
MATK_HUMAN    VLHRDGHLTIDEAVFFCNLMDMVEHYSKDKGAICTKL
CRKL_HUMAN    INSLPNRRFKIGDQEFDHLPALLEFYKIHYLDTTTLI
              :              *  *  :::.:          :
```

Command line options

ProDA command line options are detailed below.

General usage

```
./proda [OPTION] … MFAFILE [MFAFILE]…
```

-L min_length

Set minimal alignment length equal to min_length.

Description:

> ProDA finds alignments of length greater than or equal to a threshold L_{MIN}. By default, L_{MIN} = 30. This option sets the threshold to min_length.

Example usage:

```
./proda –L 20 input.mfa > output.aln
```

-posterior

Use posterior decoding when computing local pairwise alignments.

Description:

> ProDA computes local pairwise alignments between two sequences using a pair-HMM and either Viterbi decoding or posterior decoding. The default option is using Viterbi decoding which is faster than posterior decoding but may be less accurate. Turning on this option instructs the aligner to use posterior decoding instead. In the example above the output was generated with **–posterior** option turned on.

Example usage:

```
./proda –posterior input.mfa > output.aln
```

-silent

Do not report progress while aligning.

Description:

> Turning on this option instructs the aligner not to report the progress while aligning. By default, ProDA reports the progress on all pairwise alignments, block generation, and on block alignment. Since some stages of the algorithm, especially pairwise alignment, may take long time, reporting progress makes the program look alive while running.

Example usage:

```
./proda –silent input.mfa > output.aln
```

-tran

Use transitivity when forming blocks of alignable sequence fragments.

Description:

> Two sequence fragments are *directly alignable* if they are parts of a local pairwise alignment. By default, two fragments are considered *alignable* if and only if they are directly alignable. Turning on this option instructs the aligner to consider two fragments alignable when they are directly alignable or when both of them are directly alignable to a third fragment.

Example usage:

```
./proda –tran input.mfa > output.aln
```

-fasta

Use FASTA output format in addition to the ClustalW format.

Description:

> When this option is turned on, the aligner generates output in the FASTA format and stores in a file with the same name as the first input file and extension ".fasta", in addition to the normal output to stdout. This option should be used only if input sequences do not contain repeats.

Example usage:

```
./proda –fasta input.mfa > output.aln
```