trimAl: a tool for automated alignment trimming in large-scale phylogenetics analyses

Salvador Capella-Gutiérrez, Jose M. Silla-Martínez and Toni Gabaldón

Tutorial
Version 1.2

trimAl tutorial

trimAl is a tool for the automated trimming of Multiple Sequence Alignments. A format inter-conversion tool, called readAl, is included in the package. You can use the program either in the command line or webserver versions. The command line version is faster and has more possibilities, so it is recommended if you are going to use trimAl extensively. The trimAl webserver included in Phylemon 2.0 provides a friendly user interface and the opportunity to perform many different downstream phylogenetic analyses on your trimmed alignment.

This document is a short tutorial that will guide you through the different possibilities of the program. Additional information can be obtained from http://trimal.cgenomics.org where a more comprehensive documentation is available.

If you use trimAl or readAl please cite our paper:

> **trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses.** Salvador Capella-Gutierrez; Jose M. Silla-Martinez; Toni Gabaldon. Bioinformatics 2009 25: 1972-1973.

If you use the online webserver *phylemon* or *phylemon2*, please cite also this reference:

> **Phylemon: a suite of web tools for molecular evolution, phylogenetics and phylogenomics.** Tárraga J, Medina I, Arbiza L, Huerta-Cepas J, Gabaldón T, Dopazo J, Dopazo H. Nucleic Acids Res. 2007 Jul;35 (Web Server issue):W38-42.

1. Program Installation.

If you have chosen the trimAl command line version you can download the source code from the Download Section in trimAl's wikipage.

For Windows OS users, we have prepared a pre-compiled trimAl version to use in this OS. Once the user has uncompressed the package, the user can find a directory, called **trimAl/bin**, where trimAl and readAl pre-compiled version can be found.

Meanwhile for the OS based on Unix platform, e.g. GNU/Linux or MAC OS X, the user should compile the source code before to use these programs. To compile the source code, you have to change your current directory to **trimAl/source** and just execute "**make**".

Once you have the trimAl and readAl binaries program, you should check if trimAl is running in appropriate way executing **trimal** program before starting this tutorial.

2. trimAl. Multiple Sequence Alignment dataset.

In order to follow this tutorial, we have prepared some examples. These examples have been taken from phylomeDB.org and you can use the codes from these files to get more information about it in this database. You can find three different directories called **Api0000038**, **Api0000040** and **Api0000080** with different files. The directory contains these files:

- A file .seqs with all the unaligned sequences.
- A file .tce with the Multiple Sequence Alignment produced by T-Coffee[1].
- A file .msl with the Multiple Sequence Alignment produced by Muscle[2].
- A file .mft with the Multiple Sequence Alignment produced by Mafft[3].
- A file .clw with the Multiple Sequence Alignment produced by Clustalw[4].

A file .cmp with the different names of the MSAs in the directory. This file would be used by trimAl to get the most consistent MSA among the different alignments.

You can use any directory to follow the present tutorial.

3. Useful trimAl's features.

Among the different trimAl parameters, there are some features that can be useful to interpret your alignment results:

- **-htmlout filename**. Use this parameter to have the trimAl output in an html file. In this way you can see the columns/sequences that trimAl maintains in the new alignment in grey color while the columns/sequences that have been deleted from the original alignment are in white color.
- **-colnumbering**. This parameter will provide you the relationship between the column numbers in the trimmed and the original alignment.
- **-complementary**. This parameter lets the user get the complementary alignment, in other words, when the user uses this parameter trimAl will render the columns/sequences that would be deleted from the original alignment.
- **-w number**. The user can change the windows size, by default 1, to take into account the surrounding columns in the trimAl's manual methods. When this parameter is fixed, trimAl take into account *number* columns to the right and to the left from the current position to compute any value, e.g. gap score, similarity score, etc. If the user wants to change a specific windows size value should use the correspond parameter **-gw** to change window size applied only a gap score assessments, **-sc** to change window size applied only to similiraty score calculations or **-cw** to change window size applied only to consistency part.

4. Useful trimAl's/readAl's features.

Both programs, trimAl and readAl, share common features related to the MSA conversion. It is possible to change the output format for a given alignment, by default the output format is the same than the input one, you can produce an output in different format with these options:

- **-clustal**. Output in CLUSTAL format.
- **-fasta**. Output in FASTA format.
- **-nbrf**. Output in PIR/NBRF format.
- **-nexus**. Output in NEXUS format.
- **-mega**. Output in MEGA format.

- **-phylip3.2**. Output in Phylip NonInterleaved format.
- **-phylip**. Output in Phylip Interleaved format.

5. Getting Information from Multiple Sequence Alignment.

trimAl computes different scores, such as gap score or similarity score distribution, from a given MSA. In order to obtain this information, we can use different parameters through the command line version.

To do this part, we are going to use the MSA called **Api0000038.msl**. This file is in the Api0000038 directory.

```
$ cd Api0000038

$ trimal -in Api0000038.msl -sgt
$ trimal -in Api0000038.msl -sgc

$ trimal -in Api0000038.msl -sct
$ trimal -in Api0000038.msl -scc

$ trimal -in Api0000038.msl -sident
```

You can redirect the trimAl output to a file. This file can be used in subsequent steps as input of other programs, e.g. gnuplot, openoffice.org, microsoft excel, etc, to do plots of this information.

```
$ trimal -in Api0000038.msl -scc > SimilarityColumns
```

For instance, in the lines below you can see how to plot the information generated by trimAl using the GNUPLOT program.

```
$ gnuplot
        plot 'SimilarityColumns' u 1:2 w lp notitle
        set yrange [-0.05:1.05]
        set xrange [-1:1210]
        set xlabel 'Columns'
        set ylabel 'Residue Similarity Score'
        plot 'SimilarityColumns' u 1:2 w lp notitle
        exit
```

In this other example you can see the gaps distribution from the alignment. This plot also was generated using GNUPLOT

```
$ trimal -in Api0000038.msl -sgt > gapsDistribution


$ gnuplot
        set xlabel '% Alignment'
        set ylabel 'Gaps Score'
        plot 'gapsDistribution' u 7:4 w lp notitle
        exit
```

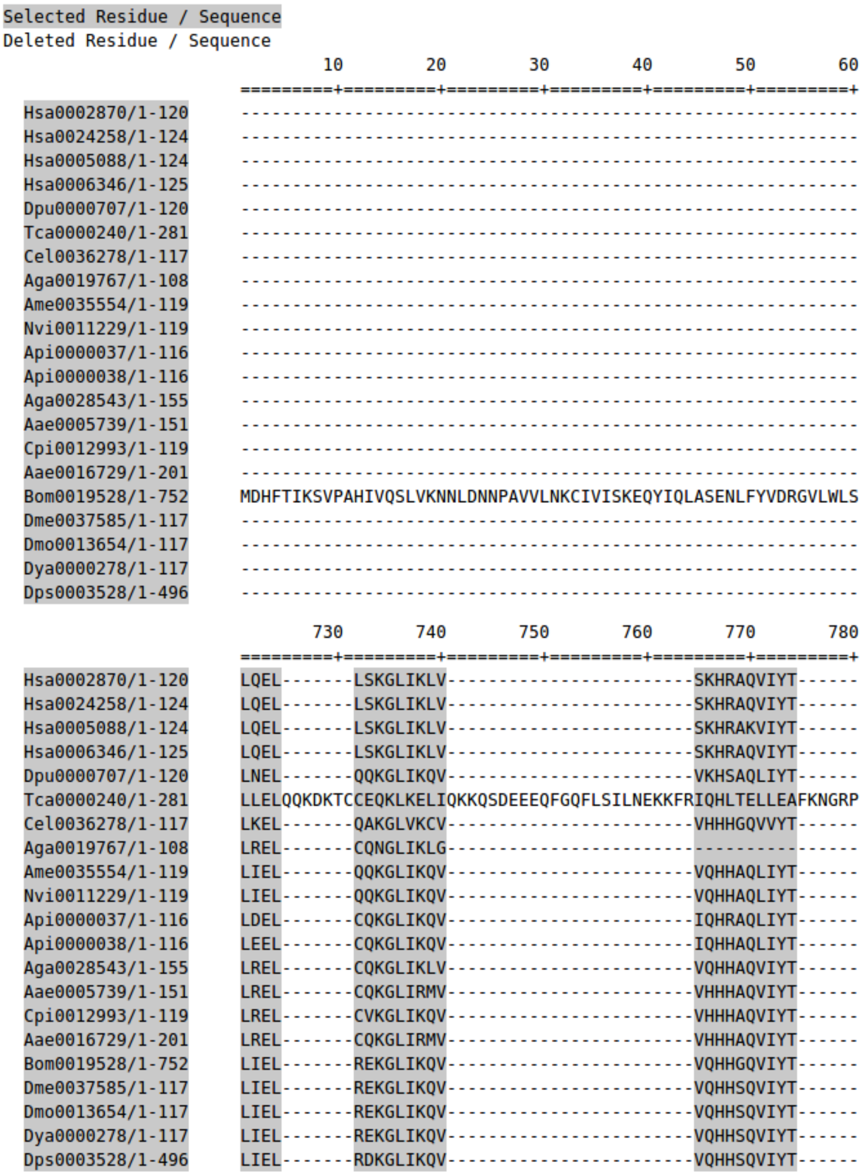
6. Using user-defined thresholds.

If you do not want to use any of the automated procedures included in trimAl (see sections 7 and 8) you can set your own thresholds to trim your alignment. We will use the parameter **-htmlout filename** for each example so differences can be visualized. In this example, we will use the **Api0000038.msl** file from the Api0000038 directory.

Firstly, we are going to trim the alignment only using the **-gt value** which is defined in the [0 - 1] range. In this specific example, those columns that do not achieve a gap score, at least, equal to 0.190, meaning that the fraction of gaps on these columns are smaller than this value, will be deleted from the input alignment.

```
$ trimal -in Api0000038.msl -gt 0.190 -htmlout ex01.html
```

You can see different parts of the alignment in the image below. This figure has been generated from the trimAl's HTML file for the previous example.



```
Selected Residue / Sequence
Deleted Residue / Sequence
                                 10        20        30        40        50        60
                       =========+=========+=========+=========+=========+=========+
  Hsa0002870/1-120     ------------------------------------------------------------
  Hsa0024258/1-124     ------------------------------------------------------------
  Hsa0005088/1-124     ------------------------------------------------------------
  Hsa0006346/1-125     ------------------------------------------------------------
  Dpu0000707/1-120     ------------------------------------------------------------
  Tca0000240/1-281     ------------------------------------------------------------
  Cel0036278/1-117     ------------------------------------------------------------
  Aga0019767/1-108     ------------------------------------------------------------
  Ame0035554/1-119     ------------------------------------------------------------
  Nvi0011229/1-119     ------------------------------------------------------------
  Api0000037/1-116     ------------------------------------------------------------
  Api0000038/1-116     ------------------------------------------------------------
  Aga0028543/1-155     ------------------------------------------------------------
  Aae0005739/1-151     ------------------------------------------------------------
  Cpi0012993/1-119     ------------------------------------------------------------
  Aae0016729/1-201     ------------------------------------------------------------
  Bom0019528/1-752     MDHFTIKSVPAHIVQSLVKNNLDNNPAVVLNKCIVISKEQYIQLASENLFYVDRGVLWLS
  Dme0037585/1-117     ------------------------------------------------------------
  Dmo0013654/1-117     ------------------------------------------------------------
  Dya0000278/1-117     ------------------------------------------------------------
  Dps0003528/1-496     ------------------------------------------------------------

                                730       740       750       760       770       780
                       =========+=========+=========+=========+=========+=========+
  Hsa0002870/1-120     LQEL-------LSKGLIKLV------------------------SKHRAQVIYT------
  Hsa0024258/1-124     LQEL-------LSKGLIKLV------------------------SKHRAQVIYT------
  Hsa0005088/1-124     LQEL-------LSKGLIKLV------------------------SKHRAKVIYT------
  Hsa0006346/1-125     LQEL-------LSKGLIKLV------------------------SKHRAQVIYT------
  Dpu0000707/1-120     LNEL-------QQKGLIKQV------------------------VKHSAQLIYT------
  Tca0000240/1-281     LLELQQKDKTCCEQKLKELIQKKQSDEEEQFGQFLSILNEKKFRIQHLTELLEAFKNGRP
  Cel0036278/1-117     LKEL-------QAKGLVKCV------------------------VHHHGQVVYT------
  Aga0019767/1-108     LREL-------CQNGLIKLG----------------------------------------
  Ame0035554/1-119     LIEL-------QQKGLIKQV------------------------VQHHAQLIYT------
  Nvi0011229/1-119     LIEL-------QQKGLIKQV------------------------VQHHAQLIYT------
  Api0000037/1-116     LDEL-------CQKGLIKQV------------------------IQHRAQLIYT------
  Api0000038/1-116     LEEL-------CQKGLIKQV------------------------IQHHAQLIYT------
  Aga0028543/1-155     LREL-------CQKGLIKLV------------------------VQHHAQVIYT------
  Aae0005739/1-151     LREL-------CQKGLIRMV------------------------VHHHAQVIYT------
  Cpi0012993/1-119     LREL-------CVKGLIKQV------------------------VHHHAQVIYT------
  Aae0016729/1-201     LREL-------CQKGLIRMV------------------------VHHHAQVIYT------
  Bom0019528/1-752     LIEL-------REKGLIKQV------------------------VQHHGQVIYT------
  Dme0037585/1-117     LIEL-------REKGLIKQV------------------------VQHHSQVIYT------
  Dmo0013654/1-117     LIEL-------REKGLIKQV------------------------VQHHSQVIYT------
  Dya0000278/1-117     LIEL-------REKGLIKQV------------------------VQHHSQVIYT------
  Dps0003528/1-496     LIEL-------RDKGLIKQV------------------------VQHHSQVIYT------
```

In this other example, we can see the effect to be more strict with our threshold. An usual consequence of higher stringency is that the trimmed MSA has fewer columns. Be careful so you do not remove too much signal

```
$ trimal -in Api0000038.msl -gt 0.8 -htmlout ex02.html
```

To be on the safe side, you can set a minimal fraction of your alignment to be conserved. In this example, we have reproduced the previous example with the difference that here we required to the program that, at least, conserve the 80% of the columns from the original alignment. This will remove the most gappy 20% of the columns or stop at the gap threshold set.

```
$ trimal -in Api0000038.msl -gt 0.8 -cons 80 -htmlout ex03.html
```

Secondly, we are going to introduce other manual threshold **-st value**. In this case, this threshold, also defined in the [0 - 1] range, is related to the similarity score. This score measures the similarity value for each column from the alignment using the Mean Distance method, by default we use Blosum62 similarity matrix but you can introduce any other matrix (see the manual). In the example below, we have used a smaller threshold to know its effect over the example.

```
$ trimal -in Api0000038.msl -st 0.003 -htmlout ex04.html
```

In this example, similar to the previous example, we have required to conserve a minimum percentage of the original alignment in a independent way to fixed by the *similarity threshold*. A given threshold maintains a larger number of columns than the *cons* threshold, trimAl selects this first one.

```
$ trimal -in Api0000038.msl -st 0.003 -cons 30 -htmlout ex05.html
```

Thirdly, we are going to see the effect of combining two different thresholds. In this case, trimAl only maintains those columns that achieve or pass both thresholds.

```
$ trimal -in Api0000038.msl -st 0.003 -gt 0.19 -htmlout ex06.html
```

Finally, we are going to see the effect of combining two different thresholds with the *cons* parameter. In this case, if the number of columns that achieve or pass both thresholds is equal or greater than the percentage fixed by *cons* parameter, trimAl chose these columns. However, if the number of columns that achieve or pass both thresholds is less than the number of columns fixed by *cons* parameter, trimAl relaxes both to thresholds in order to retrieve those columns that lets to achieve this minimum percentage.

```
$ trimal -in Api0000038.msl -st 0.003 -gt 0.19 -cons 60 -htmlout ex07.html
```

7. Selection of the most consistent alignment.

trimAl can select the most consistent alignment when more than one alignment is provided for the same sequences (and in the same order) using the **-compareset filename** parameter. To do this part, we are going to move to Api0000040 directory, we can find there a file called

Api0000040.cmp listing the alignment paths. Using this file, we execute the instruction below to select the most consistent alignment among the alignment provided

```
$ trimal -compareset Api0000040.cmp
```

As in previous section, once trimAl has selected the most consistent alignment, we can get information about the alignment selected using the appropriate parameters. For example, we can use the follow instructions to know the consistency value for each column in the alignment or its consistency values distribution

```
$ trimal -compareset Api0000040.cmp -sct
$ trimal -compareset Api0000040.cmp -scc
```

Also, we can trim the selected alignment using a specific threshold related to the consistency value. To do that, we should use the **-ct value** where the value is a number defined in the [0 - 1] range. This number refers to the average conservation of residue pars in that column with respect to the other alignments.

```
$ trimal -compareset Api0000040.cmp -ct 0.6 -htmlout ex08.html
```

On the same way than the previous section, we can define a minimum percentage of columns that should be conserve in the new alignment. For this purpose, we have to use the *cons* parameter as we explained before.

```
$ trimal -compareset Api0000040.cmp -ct 0.6 -cons 50 -htmlout ex09.html
```

Finally, we can combine different thresholds, in fact, we can use all of them as well as we can define a minimum percentage of columns that should be conserve in the output alignment. In the line below, you can see an example of this situation.

```
$ trimal -compareset Api0000040.cmp -ct 0.6 -cons 50 -gt 0.8 -st 0.01
    -htmlout ex10.html
```

8. Applying automated methods.

One of the most powerful aspects of trimAl is that it provides you with several automated options. This option will automatically select the most appropriate thresholds for your alignment after examining the distribution of various parameters along your alignment. Among the alignment features that trimAl takes into account to compute these optimal cut-off are the gap distribution, the similarity distribution, the identity score, etc.

You can find a complete explanation about all of these methods in the trimAl's Publications Section. Here, we provide some examples on how to use these methods. The automated methods, *gappyout*, *strict* and *strictpus*, can be used independently if you are working with one or more than one alignment, in the last case, for the same sequences.

In the lines below, you can see how to use the *gappyout* method in both ways. This method will eliminate the most gappy fraction of the columns from your alignment. For this, we are going to continue using the same directory than the previous section.

```
$ trimal -compareset Api0000040.cmp -gappyout -htmlout ex11.html
$ trimal -in Api0000040.mft -gappyout -htmlout ex12.html
```

In this case, we are going to use the same files than in the example before but we have changed the method to trim the alignmnet. Now, we are using *strict* and *strictplus* methods. These two methods combine the information on the fraction of gaps in a column and their similarity scores, being strictplus for more stringent than strict method.

```
$ trimal -compareset Api0000040.cmp -strict -htmlout ex13.html
$ trimal -in Api0000040.clw -strictplus -htmlout ex14.html
```

9. Using an heuristic method to decide which is the best automated method for a given MSA.

Finally, we implemented an heuristic method to decide which is the best automated method to trim a given alignment. The heuristic method takes into account alignment features such as the number of sequences in the alignment as well as some measures about the identity score among the sequences in the alignment or among the best pairwise sequences in that MSA. According to these characteristics trimAl will decide upon one of the two automated methods (gappyout or strictplus).

To illustrate how to use this method, we provide a couple of example using the same directory than the section before. First, we used trimAl to selecte the most consistent alignment and then we trimmed that alignmnet using our heuristic method.

```
$ trimal -compareset Api0000040.cmp -automated1 -htmlout ex15.html
```

Then, we trim a single MSA using the previously mentioned method.

```
$ trimal -in Api0000040.msl -automated1 -htmlout ex16.html
```

10. Getting more information.

We hope that this short introduction to trimAl's features has been useful to you.

We advise you to visit periodically the trimAl's wikipage (trimal.cgenomics.org) where you could get the latest news about the program as well as more information, examples, etc, about trimAl's package. You can also subscribe to the mailing list if you want to be updated in new trimAl developing.

11. References.

1. **T-Coffee: A novel method for fast and accurate multiple sequence alignment**. Notredame C, Higgins DG, Heringa J. J Mol Biol. 2000 Sep 8;302(1):205-17.

2. **MUSCLE: multiple sequence alignment with high accuracy and high throughput**. Edgar RC.Nucleic Acids Res. 2004 Mar 19;32(5):1792-7.

3. **MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform**. Katoh K, Misawa K, Kuma K, Miyata T. Nucleic Acids Res. 2002 Jul 15;30(14):3059-66.

4. **CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice**. Thompson JD, Higgins DG, Gibson TJ. Nucleic Acids Res. 1994 Nov 11;22(22):4673-80.