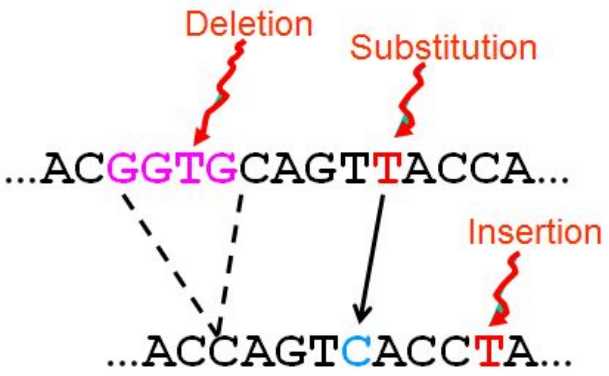
Multiple Sequence Alignment

Rosa Fernández, Lisa Pokorny & Marina Marcet-Houben

Multiple Sequence Alignment (MSA)



...ACGGTGCAGTTACC-A...

...AC----CAGTCACCTA...

The **true multiple alignment**

- Reflects historical substitution, insertion, and deletion events
- Defined using transitive closure of pairwise alignments computed on edges of the true tree

Fig. 9.1. Warnow. 2017. *Computational Phylogenomics. An introduction to designing methods for phylogeny estimation.* UIUC.

Multiple Seq Alignment (MSA)

Standard two-phase approach: 1st ALIGNMENT (positional homology)

```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```

→

```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```

2nd TREE BUILDING

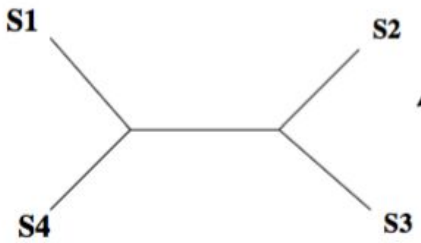


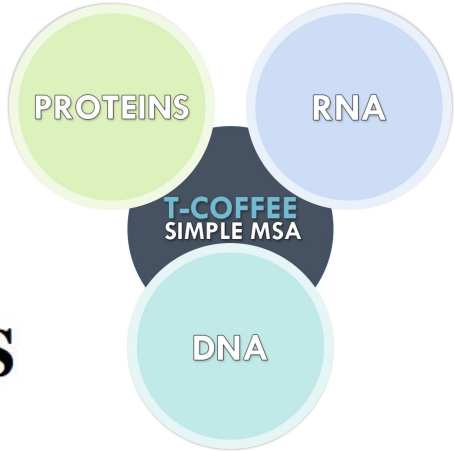
Fig. 9.1. Warnow. 2017. *Computational Phylogenomics. An introduction to designing methods for phylogeny estimation.* UIUC.

MSA Methods (MSAMs)

- Sum-of-Pairs Alignment (SOP)
- Tree Alignment and Generalized Tree Alignment
- Sequence Profiles
- Profile Hidden Markov Models (HMM)
- Reference-based Alignments
- Template-based Methods
- Seed Alignment Methods
- Weighted-Homology Pair Methods
- Progressive Methods
- Divide-and-Conquer Methods
- Co-estimation of Alignments and Trees
- Structure Informed Methods, etc.

Comparing MSAMs

Tool	Options	Algorithm	Alphabet
ClustalW	Defaults	Progressive	Amino Acid
Muscle	Defaults	Progressive (iterative)	Amino Acid
MAFFT	Defaults	Progressive (iterative)	Amino Acid
ProbCons	Defaults	Consistency	Amino Acid
ProbAlign	Defaults	Consistency	Amino Acid
Mummals	Defaults	Consistency/Structure	Amino Acid
Dialign-TX	Defaults	Greedy/Progressive	Amino Acid
Prank (AA)	+F (AA)	"Phylogenetically-aware"	Amino Acid
Prank	+F -codon	"Phylogenetically-aware"	Codon
BAli-Phy	Model M0	Statistical Alignment	Codon
BAli-Phy samples	Model M0	Statistical Alignment	Codon
BAli-Phy integrated	Model M0	Statistical Alignment	Codon



BAli---Phy

MUMMALS

MASS

Multiple Alignment by Secondary Structures

CLUSTAL

CLUSTAL

ESPript 3.0

EMBL-EBI

MUSCLE

PROBCONS

Probabilistic Consistency-based Multiple Alignment of Amino Acid Sequences

PRANK

MAFFT version 7

Multiple alignment program for amino acid or nucleotide sequences

Blackburne & Whelan. 2013. *Mol. Biol. Evol.* 30(3):642–653.

Mean Distance between MSAMs

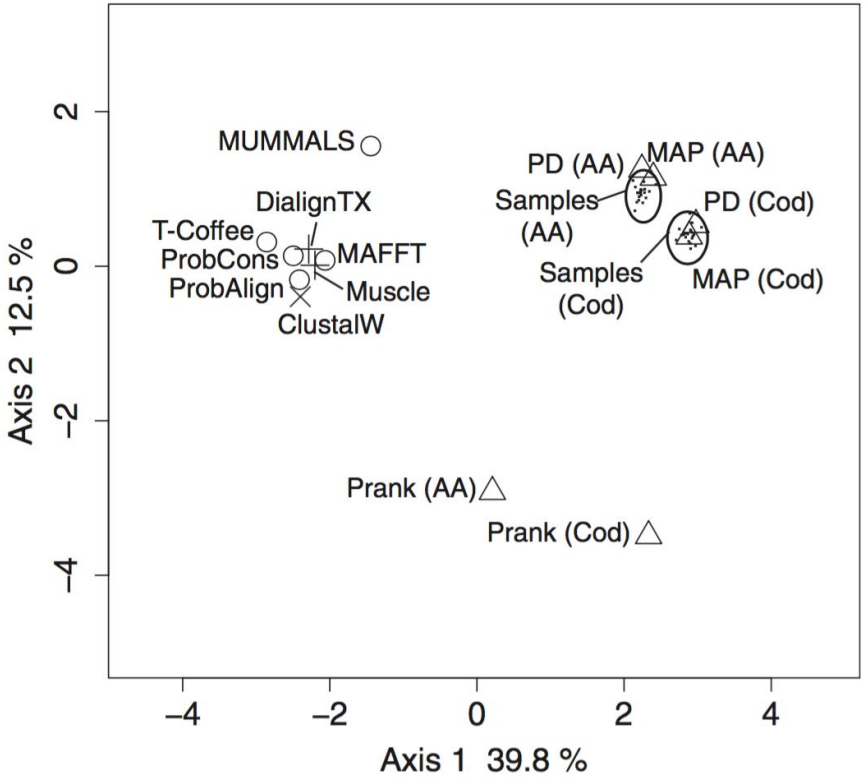


Fig. 1. Blackburne & Whelan. 2013. *Mol. Biol. Evol.* 30(3):642–653.

Distances btw. Tree Estimates from ≠ MSAMs

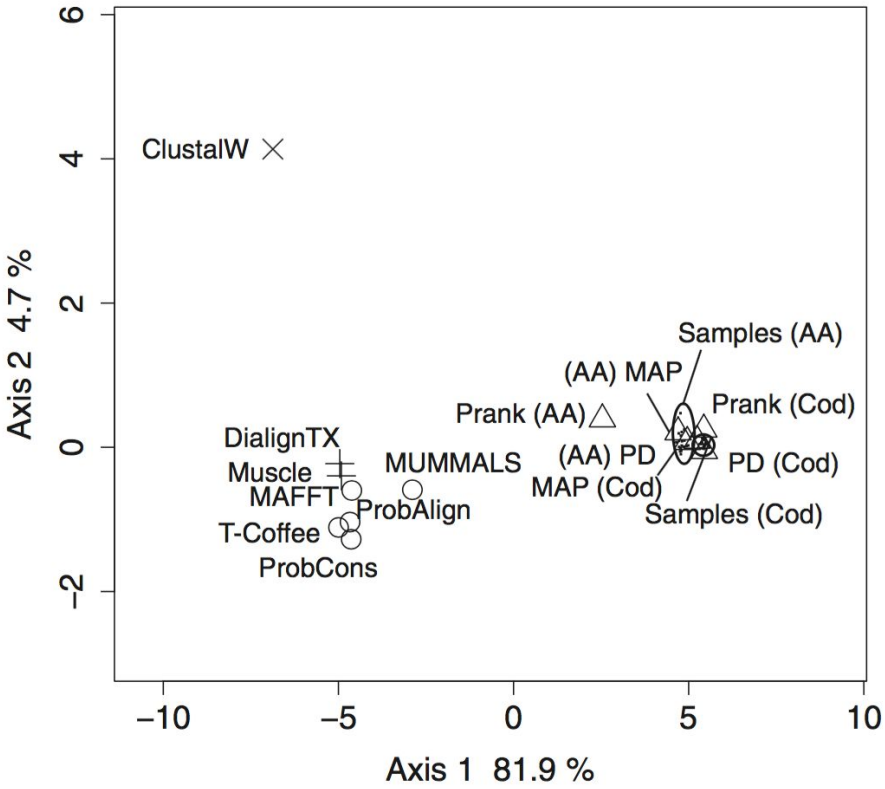


Fig. 2. Blackburne & Whelan. 2013. *Mol. Biol. Evol.* 30(3):642–653.

Even more MSAMs comparisons

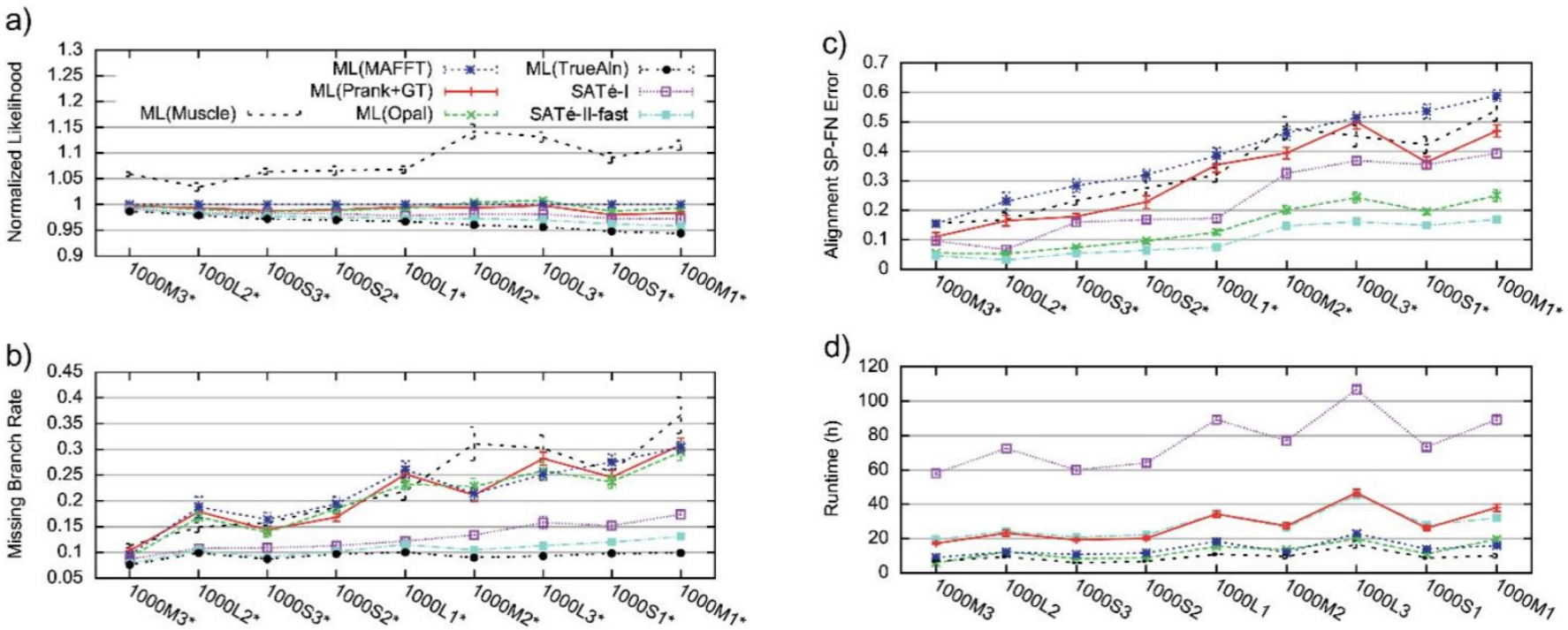


Fig. 4. Liu et al. 2013. *Syst. Biol.* 61(1):90–106.

PROTEINS

RNA

T-COFFEE
SIMPLE MSA

DNA

BAli---Phy

MUMMALS

MASS

Multiple Alignment by Secondary Structures

CLUSTAL

CLUSTAL

ESPript 3.0

EMBL-EBI

MUSCLE

PROBCONS

Probabilistic Consistency-based Multiple Alignment
of Amino Acid Sequences

PRANK

MAFFT version 7

Multiple alignment program for amino acid or nucleotide sequences

PASTA

PASTA estimates **alignments and ML trees** from unaligned sequences using an **iterative approach**. In each iteration, it first estimates a multiple sequence alignment using the current tree as a guide and then estimates a ML tree on (a masked version of) the alignment. By default, PASTA performs 3 iterations, but a host of options enable changing that behavior. In **each iteration**, a **divide-and-conquer** strategy is used for estimating the alignment. The **set of sequences** is **divided into smaller subsets**, **each** of which is **aligned using** an external alignment tool (default is **MAFFT**). These **subset alignments** are then **pairwise merged** (by default **using Opal**) and finally the **pairwise merged alignments** are **merged** into a **final alignment using** a **transitivity merge** technique. The division of the dataset into smaller subsets and selecting which alignments should be pairwise merged is guided by the tree from the previous iteration. The first step therefore needs an initial tree.

Acknowledgment: The current **PASTA** code is heavily based on the **SATé** code developed by Mark Holder's group at KU.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

PASTA

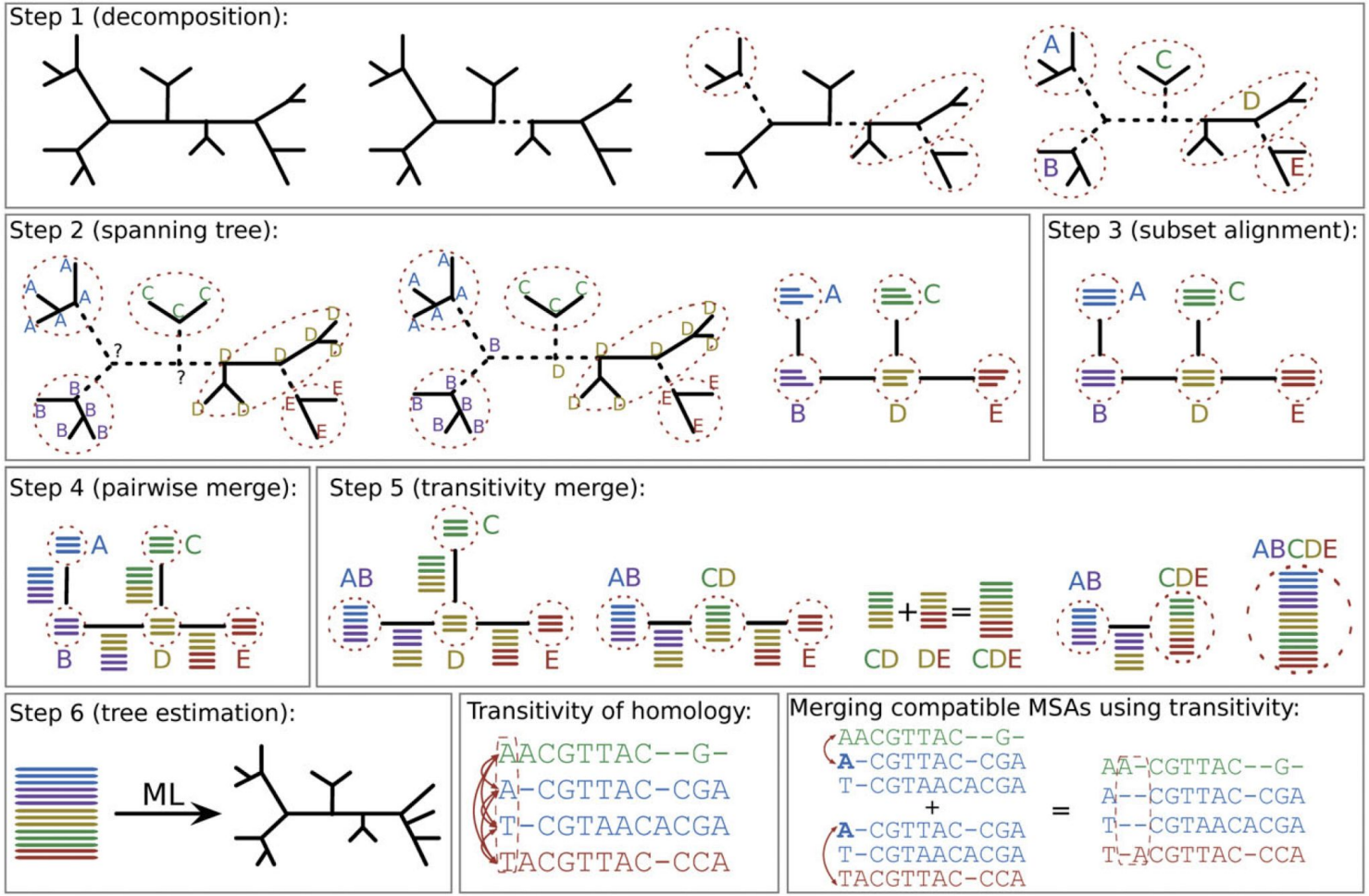


Fig. 1. Mirarab et al. 2015. *J. Comp. Biol.* 22(5):377–386.

PASTA

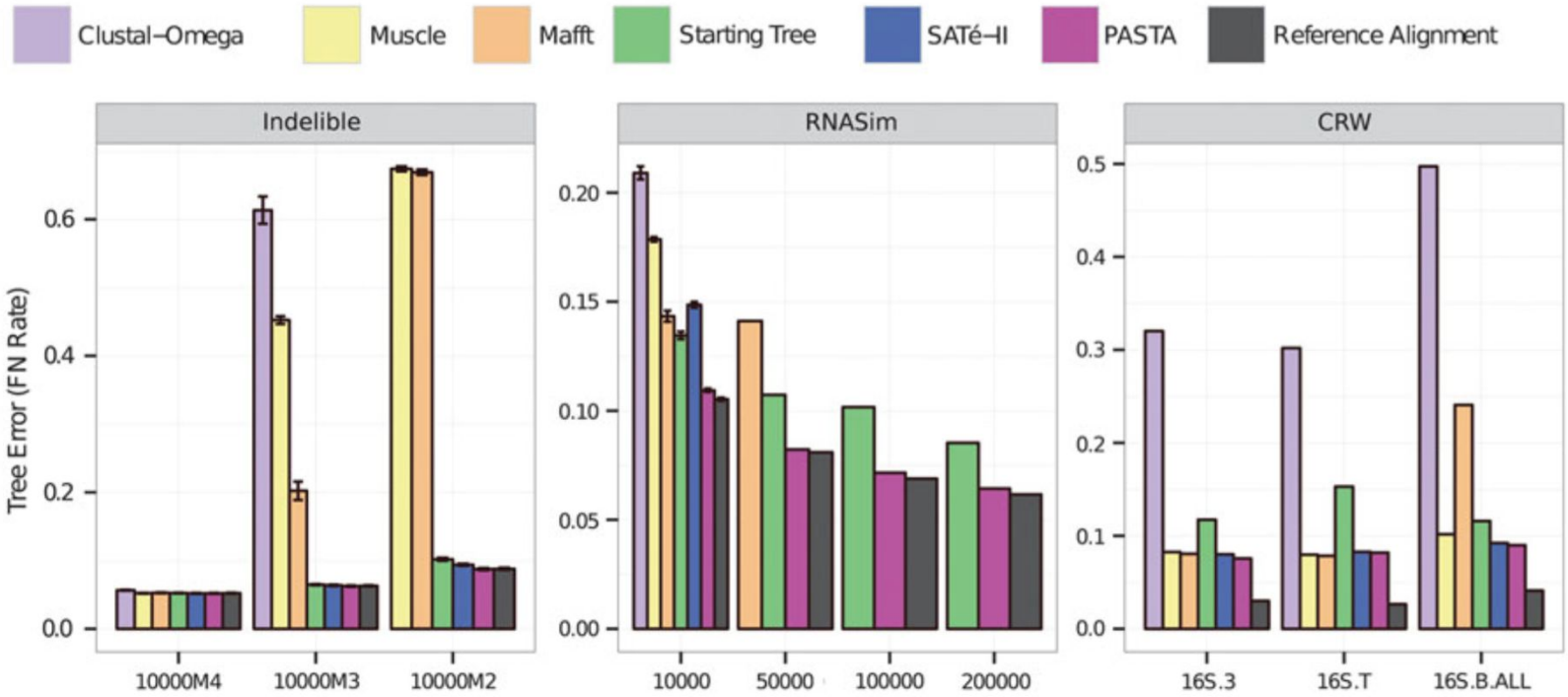


Fig. 2. Mirarab et al. 2015. *J. Comp. Biol.* 22(5):377–386.

Running PASTA (from Command-line)

If your installation is successful, you should be able to run **PASTA** by running the following command from any location. Open up a terminal window and type:

```
run_pasta.py --help
```

Running **PASTA** with the `--help` option produces the list of options available in **PASTA**. **PASTA** automatically picks its algorithmic settings based on your input, so you can ignore most of these options (but `-d` is essential if you have anything other than DNA sequences). The basic command-line usage you need to know is:

```
run_pasta.py -i input_fasta_file
```

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Running PASTA (from Command-line)

The `-i` option is used to specify the input sequence file. The input file needs to be in the relaxed FASTA format. This command will start **PASTA** and will run it on your input file.

For a test run, use the `cd` command to go to the `data` directory under your **PASTA** installation directory. From there, run

```
run_pasta.py -i small.fasta
```

This will start **PASTA** and will finish quickly (30 seconds to 5 minutes based on your machine). Read **PASTA** output and make sure it finishes without producing any errors. If **PASTA** runs successfully, it produces a multiple sequence alignment and a tree, which we will explore in the next step.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Inspecting the Output of PASTA

The two main outputs of **PASTA** are an alignment and a tree. The tree is saved in a file called `[jobname].tre` and the alignment file is named `[jobname].marker001.small.aln`. The `[jobname]` is a prefix which is by default set to `pastajob`, but can be changed by the user (see option `-j` below). When you start **PASTA**, if your output directory (which is by default where your input sequences are) already contains some files with the `pastajob` prefix, then the `pastajob1` prefix is used, and if that exists, `pastajob2` is used, and so forth. Thus the existing files are never overwritten. The name of your job and therefore the prefix used for output files can be controlled using the `- j` argument for command-line or the "Job Name" field on the GUI.

Tree viewing software → https://en.wikipedia.org/wiki/List_of_phylogenetic_tree_visualization_software

Alignment viewing software, e.g., http://doua.prabi.fr/software/seaview

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Running PASTA (from Command-line)

You can script a while loop in bash to run **PASTA** on multiple fasta files. First open a text editor

```
nano pasta_loop.sh
```

Write your bash script

```
#!/bin/bash
while read targetname;
   do
      python ABSOLUTE_PATH_HERE/run_pasta.py -i "$targetname".fasta -j $targetname
   done < targetlist.txt
```

Close `CTRL+x` and save your script. This script assumes all target files are in the same folder in fasta format. It also assumes that folder contains a text file listing all targets. From there, run

```
bash pasta_loop.sh
```

Understanding and Using PASTA Options

The command line allows you to alter the behavior of the algorithm using a variety of configuration options. Running **PASTA** with the `-h` option lists all the options that can be provided to the command-line (see below for the most important ones). In addition to the command-line itself, **PASTA** can read the options from one or more configuration files. The configuration files have the following format:

```
[commandline]
option-name = value

[sate]
option-name = value
```

Note that as mentioned before, with every run, **PASTA** saves the configuration file for that run as a temporary file called `[jobname]_temp_pasta_config.txt` in your output directory. You can view one of these files in a Text editor for better understanding the format of the configuration file.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

PASTA can read multiple configuration. Configuration files are read in the order they occur as arguments (with values in later files replacing previously read values). Options specified in the command line are read last. Thus these values "overwrite" any settings from the configuration files.

The following is a list of important options used by **PASTA**. Note that by default **PASTA** picks these parameters for you, and thus you might not need to ever change these (with the important exception of the `-d` option):

- **Initial tree**: As mentioned before, **PASTA** needs an initial tree for doing the first round of the alignment. Here is how the initial tree is picked.
  - If a starting tree is provided using the `-t` option, then that tree is used.

    ```
    run_pasta.py -i small.fasta -t small.tree
    ```

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

PASTA can read multiple configuration. Configuration files are read in the order they occur as arguments (with values in later files replacing previously read values). Options specified in the command line are read last. Thus these values "overwrite" any settings from the configuration files.

The following is a list of important options used by **PASTA**. Note that by default **PASTA** picks these parameters for you, and thus you might not need to ever change these (with the important exception of the `-d` option):

- **Initial tree**: As mentioned before, **PASTA** needs an initial tree for doing the first round of the alignment. Here is how the initial tree is picked.
  - If a starting tree is provided using the `-t` option, then that tree is used.
  - If the input sequence file is already aligned and `--aligned` option is provided, then **PASTA** computes a ML tree on the input alignment and uses that as the starting tree.
    - If the input sequences are not aligned (or if they are aligned and `--aligned` is not given), **PASTA** uses the following procedure for estimating the starting alignment and tree. It 1) randomly selects a subset of 100 sequences, 2) estimates an alignment on the subset using the subset alignment tool (default **MAFFT-l-insi**), 3) builds a **HMMER** model on this "backbone" alignment, 4) uses **hmmalign** to align the remaining sequences into the backbone alignment, 5) runs **FastTree** on the alignment obtained in the previous step.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

- **Data type**: **PASTA** does not automatically detect your data type. Unless your data is DNA, you need to set the data type using `-d` command. Your options are DNA, RNA, and PROTEIN.

```
run_pasta.py -i BBA0067-half.input.fasta -t BBA0067-half.startingtree.tre -d PROTEIN
```

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

- **Data type**: **PASTA** does not automatically detect your data type. Unless your data is DNA, you need to set the data type using `-d` command. Your options are DNA, RNA, and PROTEIN.
- **Tree estimation tool**: the default tool used for estimating the phylogenetic tree in **PASTA** is **FastTree**. The only other option currently available is **RAxML**. You can set the tree estimator to **RAxML** using the `--tree-estimator` option. However, Be aware that **RAxML** takes much longer than FastTree. If you really want to have a **RAxML** tree, we suggest obtaining one by running it on the final **PASTA** alignment. You can change the model used by FastTree (default: `-nt -gtr -gamma` for nt and `-wag -gamma` for aa) or **RAxML** (default `GTRGAMMA` for nt and `PROTWAGCAT` for AA) by updating the `[model]` parameter under `[FastTree]` or `[RAxML]` header in the input configuration file. The model cannot be currently updated in the command line directly as an option.
- **Subset alignment tool**: the default tool used for aligning subsets is **MAFFT**, but you can change it using the `--aligner` option. We strongly suggest alignment subset size should always be no more than 200 sequences, because for subsets that are larger than 200, the most accurate version of **MAFFT** (-linsi) is not used.
- **Pairwise merge tool**: the default merger too is **Opal**. You can change it using `--merger` option. If you have trouble with **Opal** (java version, memory, etc.) using **Muscle** should solve your problem and in our experience, it doesn't really affect the accuracy by a large margin.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

- **CPUs**: **PASTA** tries to use all the available cpus by default. You can use `--num_cpus` to adjust the number of threads used.

```
run_pasta.py -i small.fasta --num_cpus 1
```

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Understanding and Using PASTA Options

- **CPUs**: **PASTA** tries to use all the available cpus by default. You can use `--num_cpus` to adjust the number of threads used.
- **Number of iterations**: the simplest option that can be used to set the number of iterations is `--iter-limit`, which sets the number of iterations **PASTA** should run for. You can also set a time limit using `--time-limit`, in which case, **PASTA** runs until the time limit is reached, and then continues to run until the current iteration is finished, and then stops. If both options are set, **PASTA** stops after the first limit is reached. The remaining options for setting iteration limits are legacies of **SATé** and are not recommended.
- **Masking**: Since **PASTA** can produce very gappy alignments, it is a good idea to remove sites that are almost exclusively gaps before running the ML tree estimation. By default, **PASTA** removes sites that are more than 99.9% gaps. You can change that using the `--mask-gappy-sites` option. For example, using `--mask-gappy-sites 10` would remove sites that are gaps for all sequences except for (at most) 10 sequences. Increasing the masking can make **PASTA** a bit faster and can potentially reduce the memory usage. But it could also have a small effect on the final tree. If unsure, leave the option unchanged. Note that the final alignment outputted by **PASTA** is NOT masked, but masked versions of the output are also saved as temporary files (see below).

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Running PASTA Using Configuration Files

The configurations used for running **PASTA** are all saved to a configuration file, and also, **PASTA** can be run using a configuration file. These configuration files are useful for multiple purposes. For example, if you want to reproduce a **PASTA** run, or if you want to report the exact configurations used. Always make sure to keep the produced configuration files for future reference. Note however, that configuration files can be used as input only using command-line.

Let's open `myjob_temp_pasta_config.txt` under the data directory and take a look at it. Notice that the options we referred to are all mentioned here.

Now imagine that we wanted to instruct **PASTA** to use the JTT model instead of WAG for a protein run. Here is how we can accomplish that. Copy the `myjob_temp_pasta_config.txt` file as a new file (e.g. `cp myjob_temp_pasta_config.txt jtt_config.txt`). Then open `jtt_config.txt` using a text editor of your choice. Find `model = -wag -gamma -fastest` under the `[FastTree]` header. Remove the `-wag` option and save the config file. Note that the default model in FastTree is JTT, and therefore, when the `-wag` is removed, it automatically switches to using JTT. To run **PASTA** using this new configuration file, run:

```
run_pasta.py  jtt_config.txt
```

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Running PASTA Using Configuration Files

Adding custom parameters to aligners: It is also possible to add custom parameters to alignment and merge tools. To do so, you need to use the config file. Under each alignment tool in the config file, you can add an `args` attribute and list all the attributes you want to pass to that tool. For example, to run **Mafft** with your choice of gap penalty value, edit the config file under the `[mafft]` heading to something like:

```
[mafft]
path = [there will be a path here to your pasta directory]/bin/mafft
args = --op 0.2 --ep 0.2
```

and use this config file to run **PASTA**.

Note that **PASTA** does not try to understand these extra parameters you pass to external tools. It simply appends these parameters to the end of the command it executes.

https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Running PASTA with your own data

At this stage, if you have input files that you like to have analyzed, you know enough to start doing that.

Email: `pasta-users@googlegroups.com` for all issues.

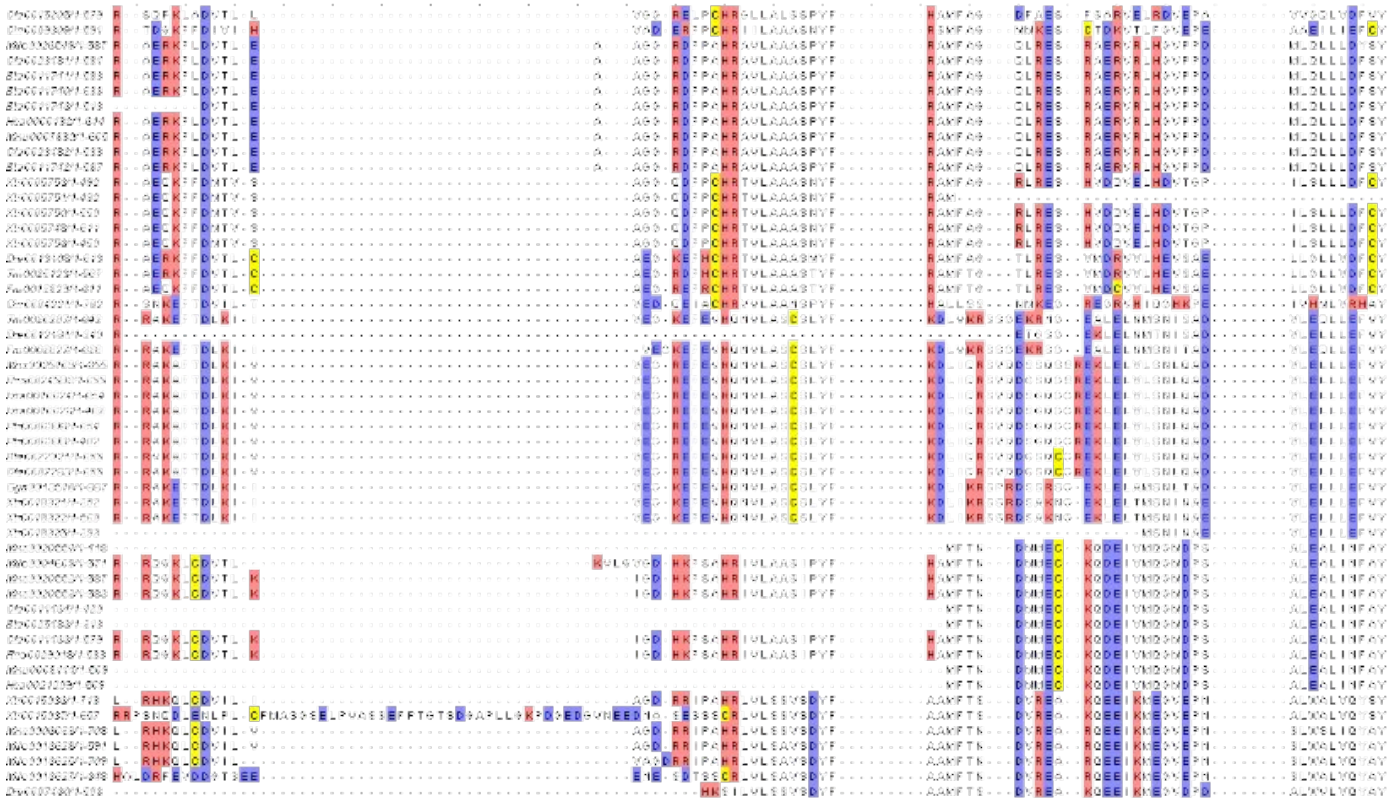
https://github.com/smirarab/pasta/blob/master/pasta-doc/pasta-tutorial.md

Trimming

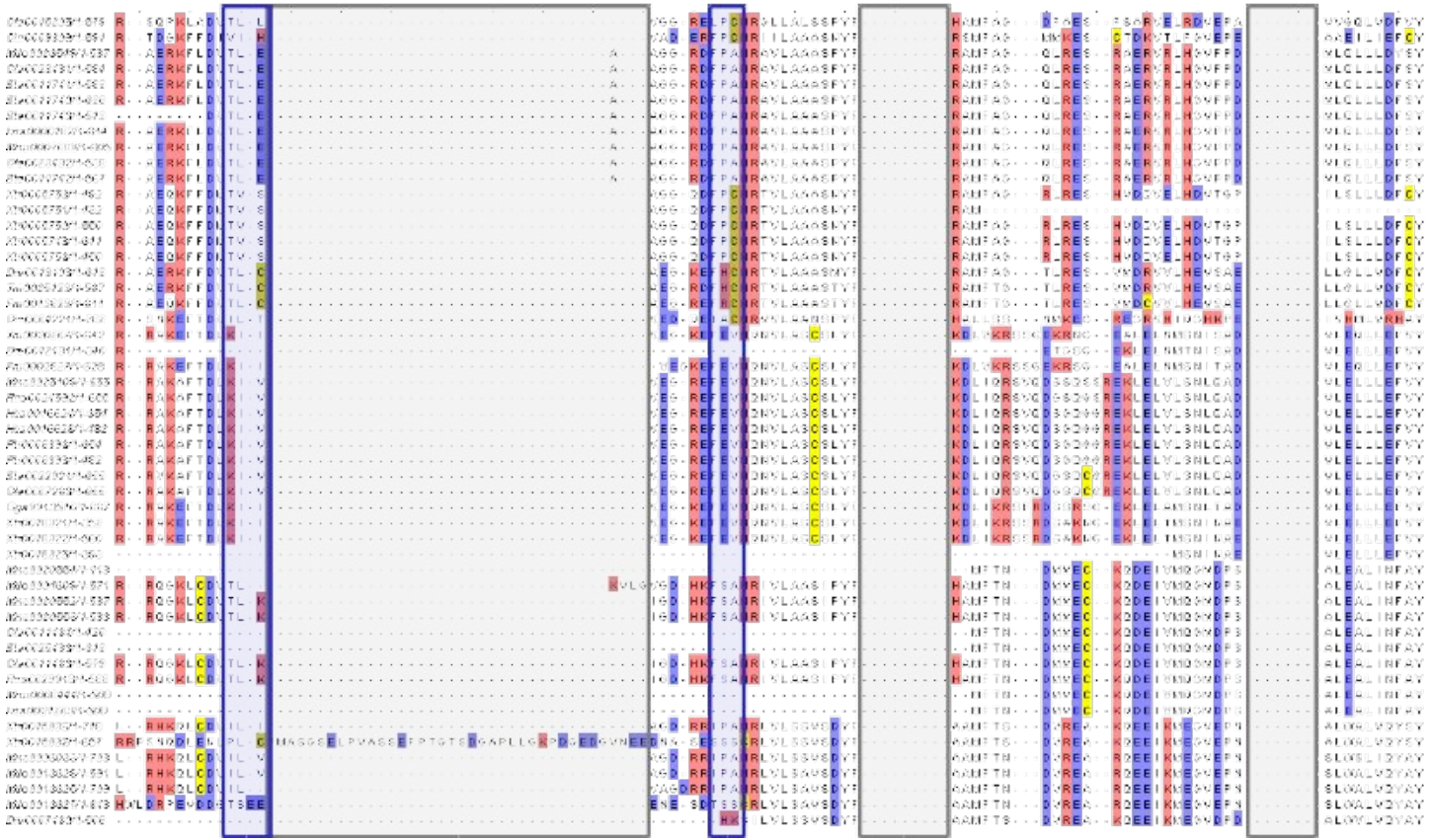
Rosa Fernández, Lisa Pokorny & Marina Marcet-Houben

Multiple Sequence Alignments

Trimming Multiple Sequence Alignments



Are these sites informative?



While gaps represent in theory insertions and deletions, phylogenetic methods are unable to use this information to reconstruct a tree

Alignment trimming

There are multiple programs to trim a MSA

Filtering methods	Type of "undesirable" sites filtered out by the method	Accounts for tree structure?	Uses a substitution matrix or model of evolution?	Adapts parameters for particular data sets?	References
Gblocks	Gap-rich and variable sites	No	No	No	Talavera and Castresana (2007)
TrimAl	Gap-rich and variable sites	No	Yes	Yes	Capella-Gutiérrez et al. (2009)
Noisy	Homoplastic sites	In part	No	No	Dress et al. (2008)
Aliscore	Random-like sites	No	Indirectly	No	Kück et al. (2010)
BMGE	High entropy sites	No	Yes	No	Criscuolo and Gribaldo (2010)
Zorro	Sites with low posterior	Yes	Yes	No	Wu et al. (2012)
Guidance	Sites sensitive to the alignment guide tree	Yes	Indirectly	No	Penn et al. (2010)

Current Methods for Automated Filtering of Multiple Sequence Alignments Frequently Worsen Single-Gene Phylogenetic Inference

Ge Tan; Matthieu Muffato; Christian Ledergerber; Javier Herrero; Nick Goldman; Manuel Gil; Christophe Dessimoz

http://trimal.cgenomics.org

Current stable version is 1.3 but version 1.4 contains a lot of new options so you may want to use that.

BIOINFORMATICS APPLICATIONS NOTE 2009, pages 1–2
doi:10.1093/bioinformatics/btp348

Phylogenetics

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

Salvador Capella-Gutiérrez, José M. Silla-Martínez and Toni Gabaldón*
Comparative Genomics group, Bioinformatics and Genomics Programme, Centre for Genomic Regulation (CRG), Dr. Aiguader, 88 08003 Barcelona, Spain

trimAl

A tool for automated alignment trimming

What can trimAl do?

1.- It allows the user to trim user-defined columns or sequences

2.- It allowed the user to define some thresholds and trim the alignment according to those thresholds:

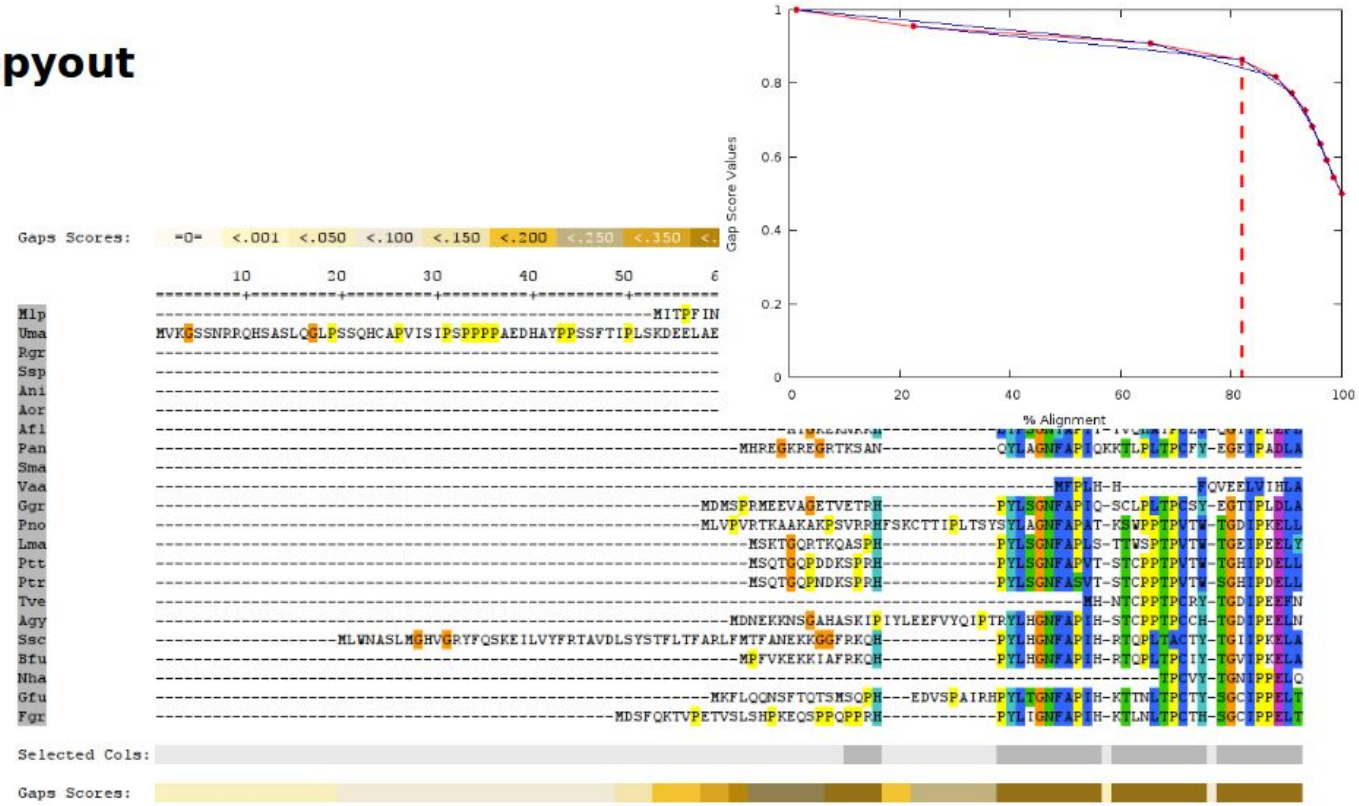
A.- Gap thresholds
B.- Similarity threshold
C.- Consistency threshold - Needs multiple alignments

3.- It allows the user to define a minimum percentage of alignment that has to be retained after trimming (Conservation score).

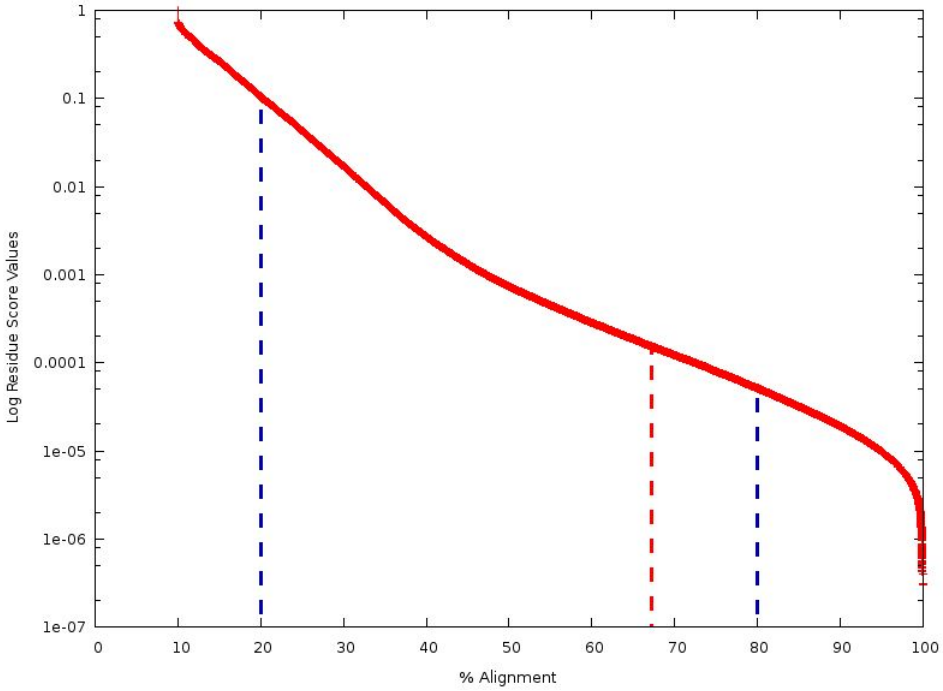
4.- It implements several automated methods that will chose the best trimming strategy based on the alignment.

Automated methods:

-gappyout



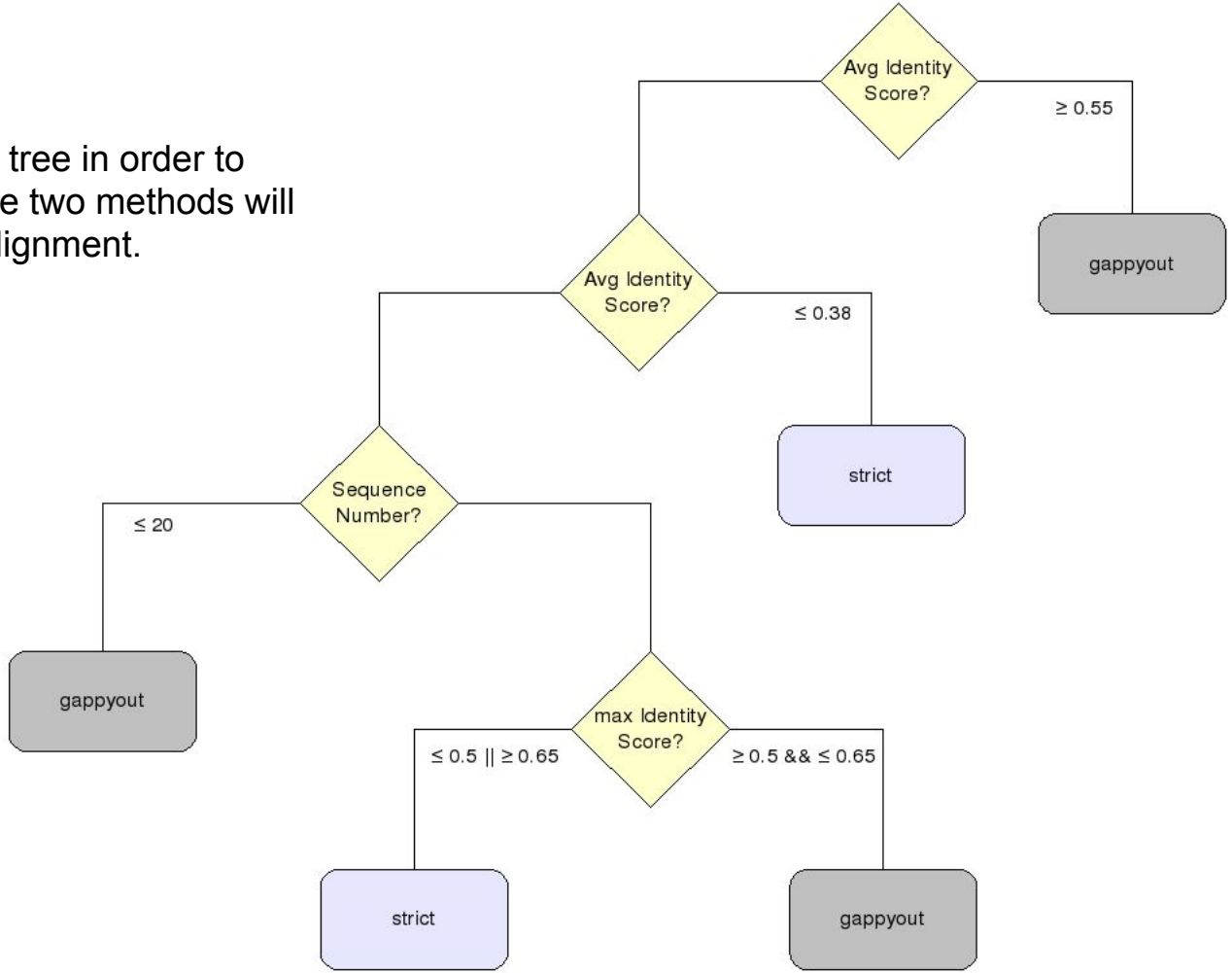
-strict -strictplus



Gappy-out + trimming by similarity scores → they will only delete blocks of data so if one column has been marked to be deleted but it is surrounded by non-marked columns it will be kept in the alignment. The two methods differ on how they define the block size.

-automated1

Will use a decision tree in order to choose which of the two methods will work best on the alignment.



Additional utilities implemented in trimAl

1.- Transform your amino acid alignment to a codon based alignment.

```
-backtrans <inputfile>   Use a Coding Sequences file to get a backtranslation for a given AA alignment
```

2.- You can delete a set of columns and obtain the alignment of the columns you have removed.
(i.e. Keep only those columns that have at least one mutation)

```
-complementary           Get the complementary alignment.
```

3.- Delete gaps

```
-nogaps                  Remove all positions with gaps in the alignment.
-noallgaps               Remove columns composed only by gaps.
```

http://trimal.cgenomics.org

trimAl

A tool for automated alignment trimming

OVERVIEW

- Introduction
- Publications
- News
- trimAl team
- FAQ
- Mailing list

DOCUMENTATION

- Getting started with trimAl v1.2
- Use of the trimAl v1.2 command line
- Use of the trimAl v1.2 webserver interface
- Getting started with readAl v1.2
- Changelog

DOWNLOADS

OLDER VERSIONS

This is **trimAl**'s information page. Also, you can find information related to the **readAl** program.

trimAl is a tool for the automated removal of spurious sequences or poorly aligned regions from a multiple sequence alignment Access trimAl publication

trimAl can consider several parameters, alone or in multiple combinations, in order to select the most-reliable positions in the alignment. These include the proportion of sequences with a gap, the level of residue similarity and, if several alignments for the same set of sequences are provided, the consistency level of columns among alignments. Moreover, **trimAl** is able to manually select a set of columns to be removed from the alignment.

Additionally, **trimAl** implements a series of automated algorithms that apply different thresholds, based on the characteristics of each alignment, to be used so that the signal-to-noise ratio after alignment trimming phase is optimized.

Moreover, the user can remove spurious sequences from the alignment before applying any method to improve the alignment's quality.

Among **trimAl**'s additional features, **trimAl** allows getting the complementary alignment (columns that were trimmed), to compute statistics from the alignment, to select the output file format, to get a summary of trimAl's trimming in HTML format, and many other options.

trimAl is being developed by the Comparative Genomics Group at the Centre for Genomic Regulation (CRG) at Barcelona, Spain.

In this site you can find the user manual, publications, news and also information related to the **trimAl** package. In this package, you can also find **readAl**, a tool for format alignment conversion.

You can also use an online version of **trimAl** and **readAl** through Phylemom2 webserver at the Centro de Investigacion Principe Felipe (CIPF) in Valencia, Spain.

Phylemon2

trimAl

A tool for automated alignment trimming

OVERVIEW

- Introduction
- Publications
- News
- trimAl team
- FAQ
- Mailing list

DOCUMENTATION

- Getting started with trimAl v1.2
- Use of the trimAl v1.2 command line
- Use of the trimAl v1.2 webserver interface
- Getting started with readAl v1.2
- Changelog

DOWNLOADS

OLDER VERSIONS

trimAl v1.4 (Beta version)

After the publication of a stable version of **trimAl v1.3**, **trimAl v1.4** will be our development version where we will add new functionality and continue improving trimAl's implementation. Official repository is here: http://github.com/scapella/trimal.

trimAl v1.3 (Release Candidate)

trimAl v1.3 has been our development version for almost 2 years. You can find the official repository here: http://github.com/scapella/trimal.

Development of **trimAl v1.3** has been frozen since we expect to release a stable version during May 2011. We'll only fix bugs before releasing this version.

If you have any suggestion or idea, don't hesitate to contact us. We'll add new functionality to **trimAl v1.4**, our new development branch. To get the latest news, you can subscribe to our **mailing list** .

trimAl v1.2 (Official release)

trimAl v1.2 can function across different platforms, here you can download the versions for three different Operative Systems, if you experience problems, you do not hesitate to contact us.

You can subscribe to our **mailing list** to get the latest news about these programs.

- trimAl v1.2 either for Linux or MacOS
- trimAl v1.2 for Windows

Remember that **trimAl v1.2** is a program that works with command line interface. Windows' users have a compiled version of **trimAl**/**readAl** in the 'bin' directory. Also, these users can compile the source code with a cross platform compiler, i.e. MinGW.

readAl: Reformatting MSAs

One of the main problems of alignments is the fact that different formats exist, and there may not be a match between the output format of an alignment program and the input format the next program needs.

```
#NEXUS

[!Imported PHYLIP file "030103phylip.phy" (Fri Jan 03 12:43:38 2003)]

Begin data;
    Dimensions ntax=29 nchar=949;
    Format datatype=nucleotide gap=- missing=? matchchar=. interleave;
options gapmode=missing;
    Matrix
487GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTCCATCGCCCCGGG
484GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTC-ATCGCCCCGGG
476GJS        AACGTTACCAAACTGTTGCCTCGGCGGGGAAA--TTCCATCGCCCCGGG
481GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTCCATCGCCCCGGG
497GJS        AACGTTACCAAACTGTTGCCTCGGCGGGGAAA--TCTCATCGCCCCGGG
501GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTTCAT-GCCCCGGG
477GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTCCATCGCCCCGGG
493GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTTCAT-GCCCCGGG
486GJS        AACGTTACCAAACTGTTGCCTCGGCGGGAA-----TTCAT-GCCCCGGG
TminuEX       AACGTTACCAAACTGTTGCCTCGGCGGGAAAA--TTTCAT-GCCCCGGG
Tharzia       GCCCATCTACGGAAGATCATTCCAGAACACCGCTGGTATTGGCCAGACT
Tvirens       GCCCATTTATGGACGATCATTCCAGAACACCGCTGGTATTGGCCAGACT
Thamatum      GCCCATCTATGGACGATCTTTCGAGAGCACCGGTGGAATTGGCCAGTCT
473gjs        GCCTATTTACGGACGATCCTTCGAGAGCACCACTGATATTGGCAAGTYT
Hpiluli       GCCTATTTACGGACGATCTTTTGAGGCCACNACNGATATTGGCCAGTTT
153DGJS       GCCTATTTACGGACGATCTTTTTAGKGAACCACTGATATTGGTCAGTCT
130DGJS       GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGGCT
135DGJS       GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGGCT
139DGJS       GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGTCT
147GJS        GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGTCT
138DGJS       GCCTATTTACGGGCGATCTTTCGAGGGCACCACTGATATTGGCAAGGCC
491GJS        GCCTATTTACGGACGATCTTTTGAGGGCACTACTGATATTGGCAAGGCT
460GJS        GCCTATTTACGGACGATCTTTTGAGGGCACTACTGATATTGGCAAGGCT
467GJS        GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGGCT
124DGJS       GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGGCT
150DGJS       GCCTATTTACGGACGATCTTTTGAGGGCACCACTGATATTGGCAAGGCT
croceum       GCCTATTTACGGACGATCTTTTGAGGGCACTACTGATATTGGCAAGGCT
polysp        GCCTATTTACGGACGATCTTTTGAGGGCACTACTGATATTGGCAAGGCT
toment        gcccatctacggacgatcattccagawcaccgmcggyattggccakact
    ;
End;
begin sets;
charpartition genes = ITS:1-383, EF:384-495, ECH:496-.;
;
end;
```

NEXUS format

```
>TRY2_RAT/24-239
-----------------------IVGGYTCQENSVPYQVSLNSGY------------HFC
GGSLI------NDQ-WV-VSAAHCYKS----------RIQVRLGE-HNINVLEGN-----
-----EQFVNAAKIIKHPNFDRKT-L-------------------------NNDIMLIKLS
SP--VKLNARVATVALPS---SCA---PAGTQCLISGWGN----------TLSSGV----
-------NEPDLLQ-CLDAP-LLPQADCEAS---YPGK--------ITDNMVCVGFL---
-EGG-KDSCQGDSGGPVVCNGE--------LQGIVSWG-YGCALPDN---PGVYTKVCNY
VDWI---------------

>Q16LB2_AEDAE/136-374
-----------------------ILNGIEADLEDFPYLGALALLDNYT-----STVSYRC
GANLI------SDR-FM-LTAAHCLFG--------KQAIHVRMGTLSLTDNPDED-----
----APVIIGVERVFFHRNYTRRPIT-------------------------RNDIALIKLN
RT---VVEDFLIPVCLYT---EQNDP-LPTVPLTIAGWGG----------NDSAS-----
-------LMSSSLM-KASVT-TYERDECNSL---LAKKI-----VRLSNDQLCALGRSEF
NDGLRNDTCVGDSGGPLELSIGR----RKYIVGLTSTG-IVCGNE-F---PSIYTRISQF
IDWI---------------
```

FASTA format

```
| 150 1075
Phy0007P00_GIBZE     -----------------------------------M---E-TQSSNG--N-APAT-K
Phy0007T05_GIBZE     ---------------MRG-A-FIP-WIDKYGELLRVCG---N---------VGTSEY-AR
Phy000D2RD_SCHPO     -----------------------------------------------------MG-K
Phy000FQYT_ASPFU     MFHT------------AL-N-PRA-A--------------------------TGEERR-G
Phy000FR4G_ASPFU     ----LR--RRL--HHSKH-R-ICL-QFALSLRKTSIGSVDSM---------GESCKAP-Q
Phy000FRUS_ASPFU     ----------MDP----NSSSST-QSLPKVSLPRITTAK---P-R-----------T---
Phy000FSLI_ASPFU     -----P--------S---PTAER-SVNSKVAIPRSANPS---S-W-----------T--S
Phy000FU66_ASPFU     MASH-------RTS-LQ-PG-SSK-YINL-NERYKYAWM---Q-HQLQPS--S-PQRP-K
Phy000FUZH_ASPFU     MFHT------------EG-P-GAS-APAK-----GRE--------------RQQSVG-R
Phy000FV0G_ASPFU     --------------------------MSGHPSEELSAD----YQLQSGGYFHYARSP-L
Phy000FW15_ASPFU     ------------------------------------MA-------DY-QN--VRPLRR-E
Phy000FWHP_ASPFU     -----------------------------------------------------MAS-T
Phy000FX8T_EMENI     ----G---SMSEQ-R---PSEPS-TPGSKIPIPRVSQLR---A-Y-------------G
Phy000FXVQ_EMENI     ---------------------------MDASVQPGHHLVRA---------PAPGAEP-S
Phy000FYYQ_EMENI     MFIA------------EA-A-NAS-KDLESGPRNPKSV---T----------GRTSTG-R
Phy000FZYA_EMENI     -------------PNDIR-A-LIT-GASGGIGAACAHQLHLA---------LTNS-AV--
Phy000G015_EMENI     ----------------------------------------------M---SSRGPT-R
Phy000G06G_EMENI     GQYEALQVSPILRQRSTL-A-VIG-GDLLQNGHSA---SEAY---------GSVFRSR-H
Phy000G0IY_EMENI     ----------------------------------------------M---SKNMGS-T
Phy000G1Z6_EMENI     MFHT------------EG-P-AAS-APARA---GRE--------------RQSSIG-R
Phy000G28K_EMENI     ------------------------------TDQPH------GAQPGILHASRLQGR-K
Phy000G57Q_ASPNG     ----------------------------------------------------DISQSR-R
Phy000G8Y0_ASPNG     MYRS-----------------------------------S--------------RHVE---K
Phy000G9MD_ASPNG     MASA--AD--EDDSDSFF-E-DHD-ASPGHDLMMKDDAL------GDGKP--DPLPMQ-K
Phy000GA9V_ASPNG     ---------------------------------------------MNQTHREGEPLA-K
Phy000GBJT_ASPNG     -----------------------------------------------------MAS-T
Phy000GBKP_ASPNG     -----SVPDGFPTRFNSG-I-I-----PSTPTKLCNNIW----LEMDS--SGRSRGKS-R
Phy000GEPP_ASPNG     -------------K---R-------------------------------------SEV--
```

PHYLIP format

readAl: Reformatting MSAs

readAl is a sister program to trimAl that allows us to convert alignment between each other.

readal -in [input file] -format -out [output file]

Input file → Alignment file

Output file → Resulting file

Format → Can be any of the formats that readAl has and that you wish to use as output: Fasta, phylip, mega, nexus, clustal,...

readAl: Colouring MSAs

```
                                 10        20        30        40        50        60        70        80        90       100       110       120
                        =========+=========+=========+=========+=========+=========+=========+=========+=========+=========+=========+=========+
Phy0007P00_GIBZE        ---------------------------------------M---E-TQSSNG--N-APAT-K-RRAVLACDRCHL-K-HYK-C-S-EAQP--CSHCKR--SGAE-CKYGR-GY-R-P---N-
Phy0007T05_GIBZE        ---------------MRG-A-FIP-WIDKYGELLRVCG---N---------VGTSEY-ARP---FSTVTPPF-IS-SLST--D-LMSP--HAHRMRKEVY------AH-A---K-IA-S-
Phy000D2RD_SCHPO        ------------------------------------------------------MG-KPKRIPRACDMCR-KR-HIR-CG---KQA--CSNCVS--HGIP-CVFTA-RPKR--R-GQ-
Phy000FQYT_ASPFU        MFHT------------AL-N-PRA-A-------------------------TGEERR-GRRPGLRACKECR-RR-HIRC--T-GTHP--CEPCLYYKKPEL-CRYSD-P-RVPH-----
Phy000FR4G_ASPFU        ----LR--RRL--HHSKH-R-ICL-QFALSLRKTSIGSVDSM---------GESCKAP-QIIRKSFACDECK-RR-HIRC--T-GGDN--CTNCVRD--AKP-CRYSS-P-SQK-LL-S-
Phy000FRUS_ASPFU        ----------MDP----NSSSST-QSLPKVSLPRITTAK---P-R-----------T----RRASRACISCR-EQ-HAK-C-T-GGSP--CERCRE--LKIE-CVFVAN-K----NEKI-
Phy000FSLI_ASPFU        -----P--------S---PTAER-SVNSKVAIPRSANPS---S-W-----------T--SSGRVSRACENCR-EQ-HAK-C-S-GHRPT-CQRCQE--SGIR-CSYGDR-K----REKM-
Phy000FU66_ASPFU        MASH-------RTS-LQ-PG-SSK-YINL-NERYKYAWM---Q-HQLQPS--S-PQRP-K-QRATQACDRCHL-K-HYK-C-D-ELYP--CTHCRK--NKLE-CIYQR-NY-R-QRE-D-
Phy000FUZH_ASPFU        MFHT------------EG-P-GAS-APAK-----GRE---------------RQQSVG-RRVTTLRACTSCR-RR-HIKC--D-----------------------------------
Phy000FV0G_ASPFU        ---------------------------MSGHPSEELSAD----YQLQSGGYFHYARSP-LPTRV-SACERCRR-H-HTR--CD-PFR-PC-SLCAR--AKVD-CQPLP-AS-Q-------
Phy000FW1S_ASPFU        ------------------------------------MA-------DY-QN--VRPLRR-ERLKVSTACQACR-LR-HAK-CG---GRA--CIRCLR--RGKA-CTYSI----------N-
Phy000FWHP_ASPFU        ------------------------------------------------------MAS-TRDSHSYACDECHL--RMSR--CSK-ERPTC-AQCKQ--LNKEC-NYSP-K-ITR-SP-L-
Phy000FX8T_EMENI        ----G---SMSEQ-R---PSEPS-TPGSKIPIPRVSQLR---A-Y--------------GNRRVKRACIECR-KQ-HTK-C-N-GQTP--CSRCIG--LGMD-CVYMDG-K----REVT-
Phy000FXVQ_EMENI        ---------------------------MDASVQPGHHLVRA---------PAPGAEP-SHRTAEQACRECR-RR-HSKC--D-RAPV--CRLCSKY--KRQ-CTYE--K-P-R-TP-L-
Phy000FYYQ_EMENI        MFIA------------EA-A-NAS-KDLESGPRNPKSV---T----------GRTSTG-RRGVVPRACSSCR-RR-HVKC--N-GEKP--CEACRWYKRPEE-CTYPE-Y-ER-------
Phy000FZYA_EMENI        -------------PNDIR-A-LIT-GASGGIGAACAHQLHLA---------LTNS-AV----NELVADLNSH-YA-DNKLAGG-GING--CHYAASKGSTRL-AEYNI-S-VNP-RN-G-
Phy000G015_EMENI        ------------------------------------------------M---SSRGPT-RPRRL-AAC---HS--QTTK--CSG-KR-PC-DGCLS--SGEEC-QFPAK-E-RK-IR-V-
Phy000G06G_EMENI        GQYEALQVSPILRQRSTL-A-VIG-GDLLQNGHSA---SEAY---------GSVFRSR-HILRKTFACDECK-RR-HIRC--S-GDEN--CLNCLRD--AKA-CRYSS-P-SHQ-LS-K-
Phy000G0IY_EMENI        ------------------------------------------------M---SKNMGS-TRDSHSYAC-----------------------------KQ--LDKEC-KYSP-K-ITR-SP-L-
Phy000G1Z6_EMENI        MFHT------------EG-P-AAS-APARA----GRE---------------RQSSIG-RRVTTLRACTSCR-RR-HIKC--D-GEKP--CEACRWYKKADL-CHYSD-P-RPS------
Phy000G28K_EMENI        ------------------------------TDQPH------GAQPGILHASRLQGR-KRKRVSVACRSCRA-R-HSR--CN-GTQP-C-SSCED--MDTE-CRYDQ-PS-TRPRT-SV
Phy000G57Q_ASPNG        ------------------------------------------------------DISQSR-RILRKTFACDECK-RR-HIRC--S-GNEN--CNNCRRD--ARS-CRYSS-P-SQR-LS-T-
Phy000G8Y0_ASPNG        MYRS-------------------------------S---------------RHVE---KLSTTL-------------------------------------------------------
Phy000G9MD_ASPNG        MASA--AD--EDDSDSFF-E-DHD-ASPGHDLMMKDDAL------GDGKP--DPLPMQ-KRRRVTRACDECH-RK-HIK-CG---KQP--CTHCTV--YSYE-CTYDQ-PSNR--RRPA-
Phy000GA9V_ASPNG        ---------------------------------------------MNQTHREGEPLA-KRRRIALACSACRA-R-HSR--CD-GQRPSC-STCLS--LTVE-CVYEP-SD-CSTNV-LV
Phy000GBJT_ASPNG        ------------------------------------------------------MAS-TRDSHSYACDECHL--RMSK--CSK-EKPTC-AQCKQ--LNKEC-NYSP-K-VTR-SP-L-
Phy000GBKP_ASPNG        -----SVPDGFPTRFNSG-I-I-----PSTPTKLCNNIW----LEMDS--SGRSRGKS-RGKYT-KACEECRR-R-HAK--CD-GVKPSC-SRCIQ--WNVS-CQYSG-TE-DGRRP-A-
Phy000GEPP_ASPNG        -------------K---R-------------------------------------SEV----QELRACVECR-RR-HSKC--S-GTAP--CSYCSRT--KKL-CRFEH-A-P-R-TP-L-
Phy000H59S_BOTFU        -------------------V-LST-E-------------------------------------------------AK-HIPC--H-IPYR--VDKAALAVINLQ-VIPDP-N-PIR-NL-SY
Phy000H9RX_BOTFU        ----------------------------MEDK------------DKDRDRDTVNPKR-RRLVT-NACTECHK-K-HAK--CD-GET-PC-ARCAS--QKVE-CVYEV-PV-RQSKE-N-
Phy000IALW_COCIM        MVPQ--CA--KSGANDQV-A-SRT-APPSQTPASADDNQ------AADDQ--LAIPVQ-KRRRVTRACDECH-RK-HIK-CG---KQP--CTHCTV--YSYA-CTYDQ-PSNR--RRPA-
Phy000IWRO_FUSOX        -------------------P-PQ-----------------------------------A-PTVEEPTTCRECR-RR-HVKC--D-GVPV--CSIC----------------------R---
Phy000IWT5_FUSOX        ------------------------------------MFCT-EPD---------SGTLKE-SKPTRGLHACGNCK-RR-HIRC--D-GRQP--CNQCESRDLQAR-CVYTQ-Q-IQR-VV-P-
Phy000J0D5_FUSOX        -------------G---R-------------------------------------GSS----SQLRACLNCQ-KR-HSRCPVN-GPGP--CSYCARA--GKD-CSFVN-P-P-R-TR-L-
Phy000J1KA_FUSOX        ------------------------------------------------ELPF-SET-GRQPRATSCAECR-RR-HVKC--E-GQFP--CRQCVYYLVPHL-CHYPA-R-KKR-HN-VS
```

statAl: Obtaining alignment statistics

While trimAl offers most of these options, this is a standalone program that will only give back some statistics of your alignment.

```
-sgc                    Print gap score per column from input alignment.
-sgt                    Print accumulated gap scores distribution from input alignment.

-ssc                    Print similarity score per column from input alignment.
-sst                    Print accumulated similarity scores distribution for input alignment.

-sfc                    Print sum-of-pairs score per column for the selected alignment
-sft                    Print accumulated sum-of-pairs scores distribution for the selected alignment

-sident                 Print identity scores for sequences in the alignemnt.
-scolidentt             Print general descriptive statistics for column identity scores from input alignemnt.
```

All the exercises can be found in the folder named: trimal_tutorial. At the beginning of each exercise a tag will let you know in which subfolder you should be.
1.- **[example_readal]** Open the alignment file and check in which format it has been generated. Now use readal to (make sure each result is in a different file):

- Change the format of the current alignment to fasta format
- Change the format of the current alignment to nexus format
- Change the format of the current alignment to clustal format
- Use the -onlyseqs option

Open the different files and notice the differences between the alignment formats.

Tip: readal is run like this:

readal -in alignment_file -out trimmed_alignment_file -format FORMAT_NAME

You can check out all the formats supported by readal by typing: readal -h

2.- **[example_trimal]** Use trimAl to trim the alignments according to a gap threshold using the following parameters:

- A gap threshold of 0.1 (-gt 0.1)
- A gap threshold of 0.5 (-gt 0.5)
- A gap threshold of 0.9 (-gt 0.9)

Make sure that the output of your alignment is in phylip format. Now you can visualize each alignment either using a text editor or using seaview. Which of the previous commands deletes the largest amount of columns?

3.- **[example_trimal]** Now use the -gt 0.5 command but add a conservation score of different values: 30, 50 and 80 (-cons option). Again make sure that your output alignment is in phylip format. Which effect does it have on the trimmed alignment?

4.- **[example_trimal]** Now instead of using the gap threshold, we'll be using the similarity threshold (-st). Repeat the trimming of the original alignment using different similarity thresholds (0.1, 0.5 and 0.9). Again, how does the alignment trimming vary? Which approach is more aggressive? How can you make sure you don't loose all the alignment?

6.- **[example_trimal]** Now we are going to use the automated trimming methods. Trim your alignment using:

- Use the different automated trimming methods: -gappyout, -strict, -strictplus, -automated1
- Use the more radical methods to delete all the columns with gaps in your alignment: -nogaps

Of all the trimming strategies you've tried, which is the best one? Can you know?

7.- **[example_consistency]** You will see seven different alignments in there and a file called paths. Each alignment has been generated in a different way and we want to trim one of them based on the consistency score. Run this command:

```
trimal  -compareset Phy007LWVO_COFCA.paths  -forceselect Phy007LWVO_COFCA.alg.metalig  -out
Phy007LWVO_COFCA.alg.clean -phylip -ct 0.1667
```

This command will trim an alignment based on a set of alternative alignments. The -ct score will trim out columns that are inconsistent in the dataset. Which advantage do you think this kind of trimming has over the others we've seen?

8.- **[example_backtranslate]** Backtranslate protein alignment into CDS. In order to do this you need to use the -backtrans option:

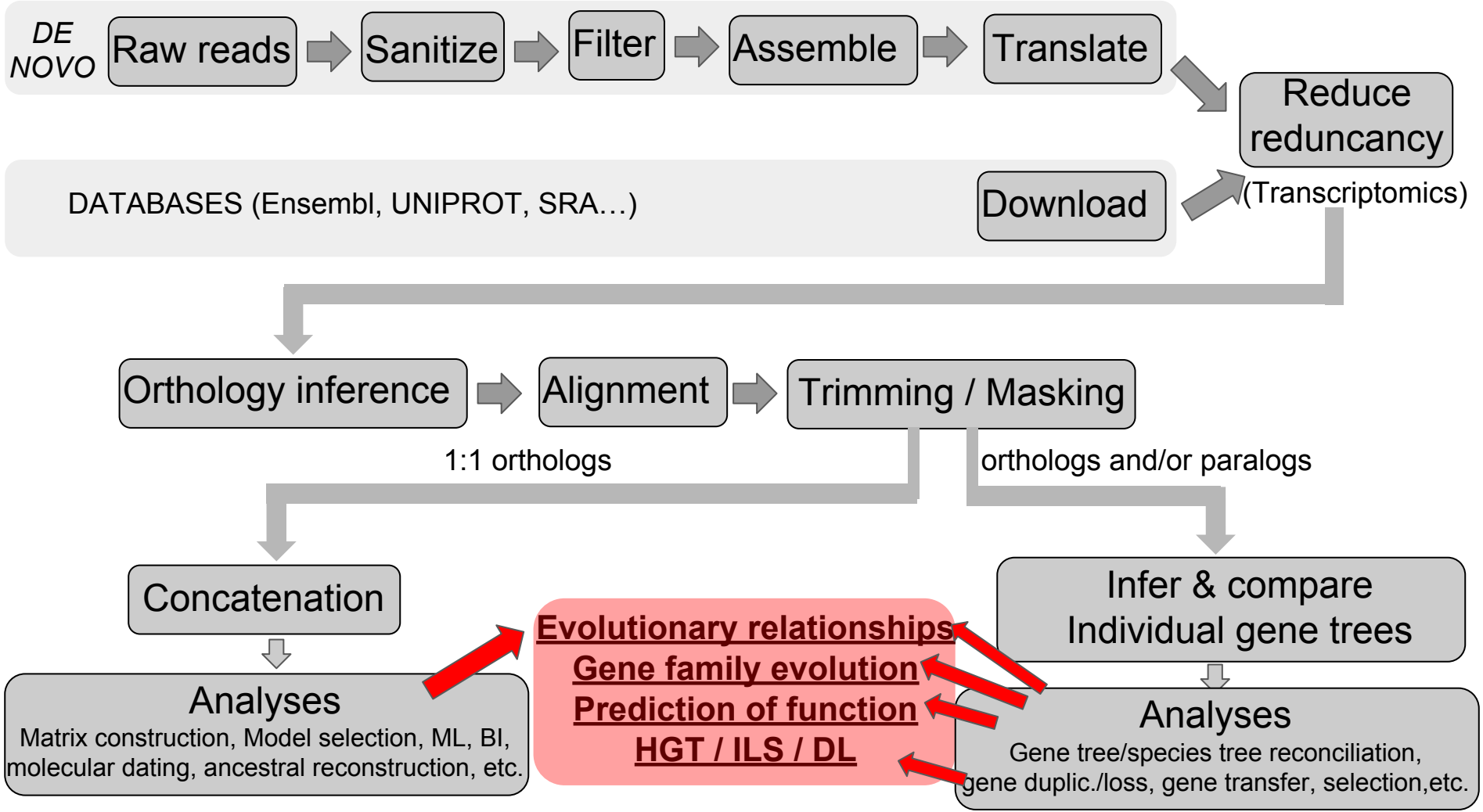
```
trimal -in Phy007LWVO_COFCA.alg -out example.cds -backtrans Phy007LWVO_COFCA.cds
```

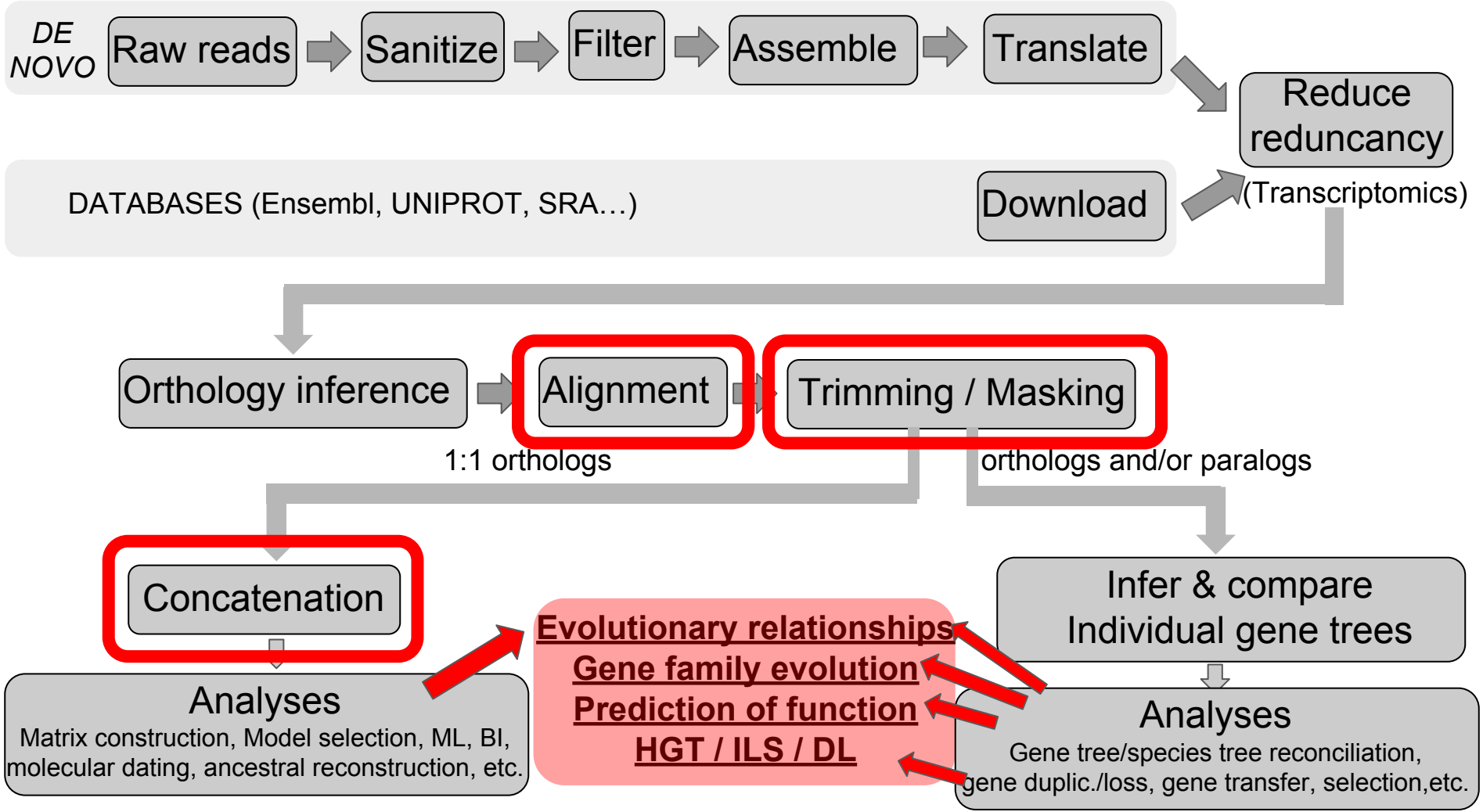
You can join this command with your preferred trimming methods.

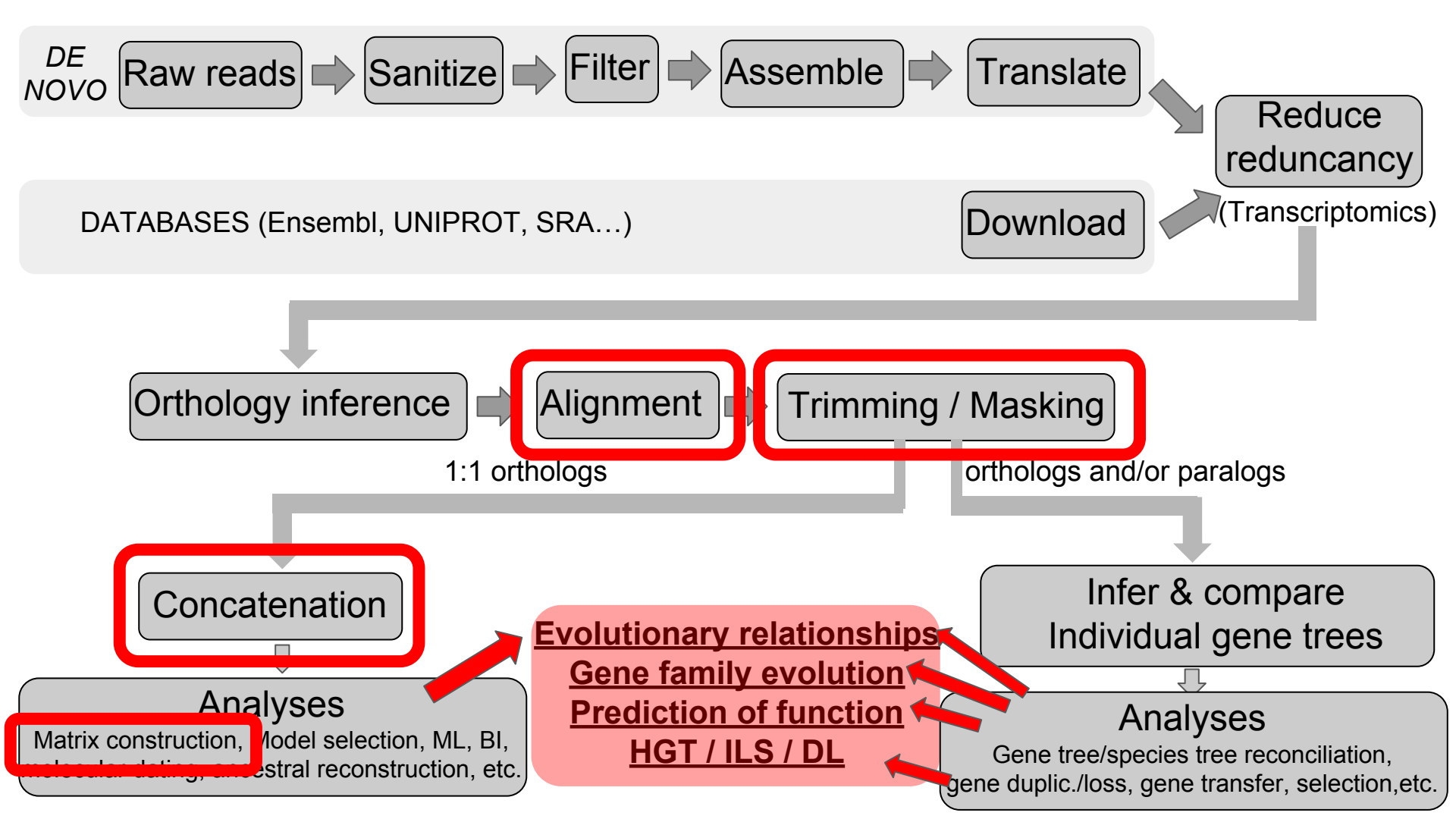
9.- Trim all the alignments found in a folder. In this case you'll have to use a bit of bash programming:

- Create a new folder called trimmed_alignments
- Now move to the folder where you have all your alignments
- for fileName in $(ls *); do trimal -in $fileName -out ../trimmed_alignments/$fileName -gt 0.1;done

(This will be needed in the main exercises after the explanations)


DE NOVO
Raw reads
Sanitize
Filter
Assemble
Translate
Reduce reduncancy
DATABASES (Ensembl, UNIPROT, SRA…)
Download
(Transcriptomics)
Orthology inference
Alignment
Trimming / Masking
1:1 orthologs
orthologs and/or paralogs
Concatenation
Infer & compare Individual gene trees
Evolutionary relationships
Gene family evolution
Prediction of function
HGT / ILS / DL
Analyses
Matrix construction, Model selection, ML, BI, molecular dating, ancestral reconstruction, etc.
Analyses
Gene tree/species tree reconciliation, gene duplic./loss, gene transfer, selection,etc.


DE NOVO
Raw reads
Sanitize
Filter
Assemble
Translate
Reduce reduncancy
(Transcriptomics)
DATABASES (Ensembl, UNIPROT, SRA…)
Download
Orthology inference
Alignment
Trimming / Masking
1:1 orthologs
orthologs and/or paralogs
Concatenation
Infer & compare Individual gene trees
Evolutionary relationships
Gene family evolution
Prediction of function
HGT / ILS / DL
Analyses
Matrix construction, Model selection, ML, BI, molecular dating, ancestral reconstruction, etc.
Analyses
Gene tree/species tree reconciliation, gene duplic./loss, gene transfer, selection,etc.


DE NOVO
Raw reads
Sanitize
Filter
Assemble
Translate
Reduce reduncancy
(Transcriptomics)
DATABASES (Ensembl, UNIPROT, SRA…)
Download
Orthology inference
Alignment
Trimming / Masking
1:1 orthologs
orthologs and/or paralogs
Concatenation
Infer & compare Individual gene trees
Evolutionary relationships
Gene family evolution
Prediction of function
HGT / ILS / DL
Analyses
Matrix construction, Model selection, ML, BI,
ancestral reconstruction, etc.
Analyses
Gene tree/species tree reconciliation,
gene duplic./loss, gene transfer, selection,etc.

Concatenation & Partition Files

Rosa Fernández, Lisa Pokorny & Marina Marcet-Houben

Concatenation vs. Coalescence in Phylogenomics

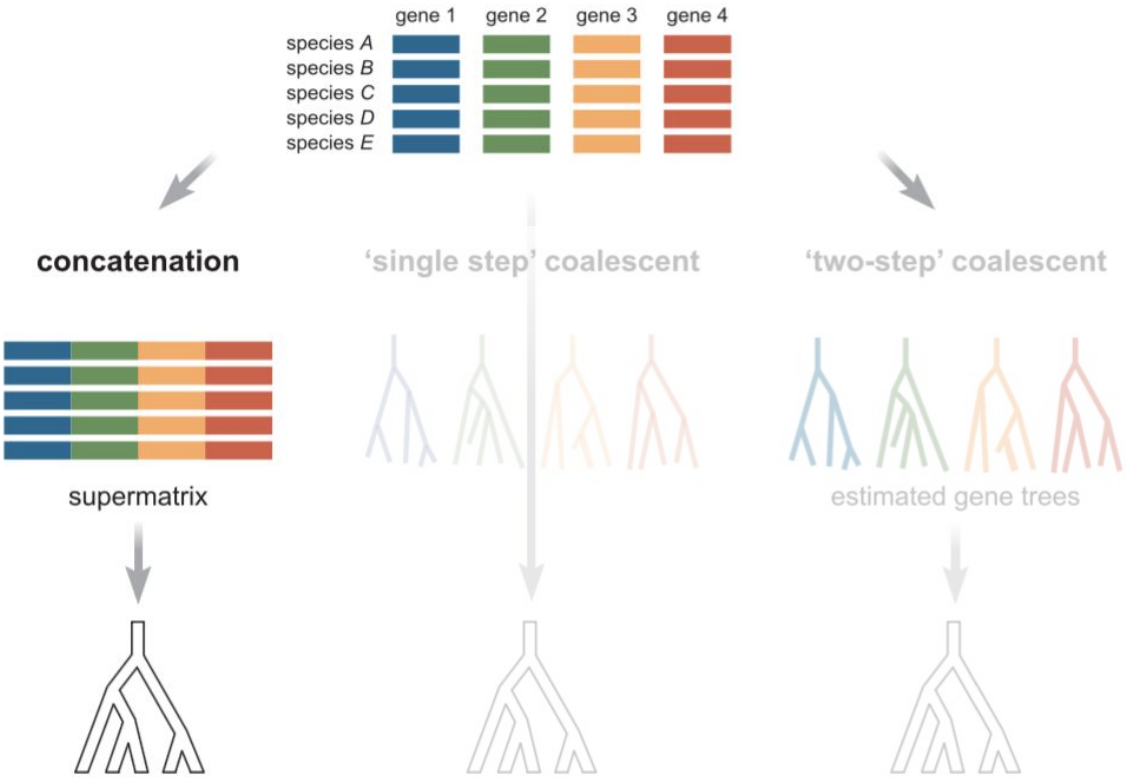


Fig. 1. Liu et al. 2015. *Ann. N.Y. Acad. Sci.* 0:1–18.

Concatenation vs. Coalescence in Phylogenomics

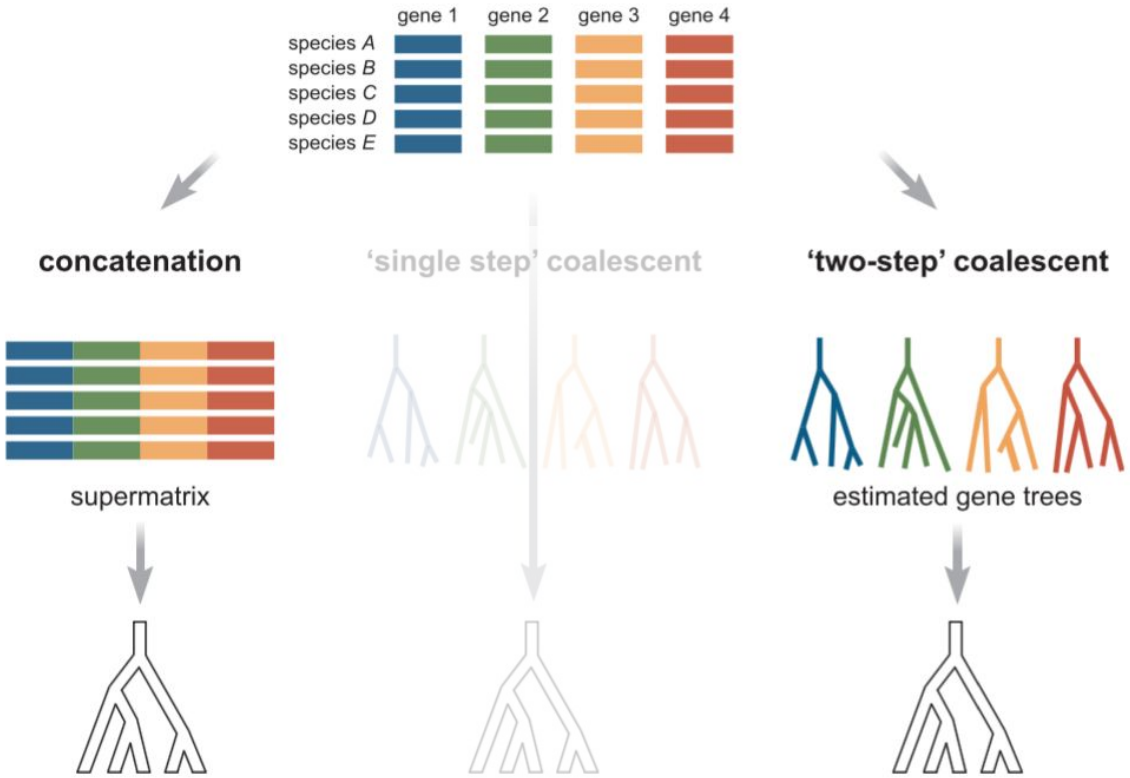


Fig. 1. Liu et al. 2015. *Ann. N.Y. Acad. Sci.* 0:1–18.

Concatenation vs. Coalescence in Phylogenomics

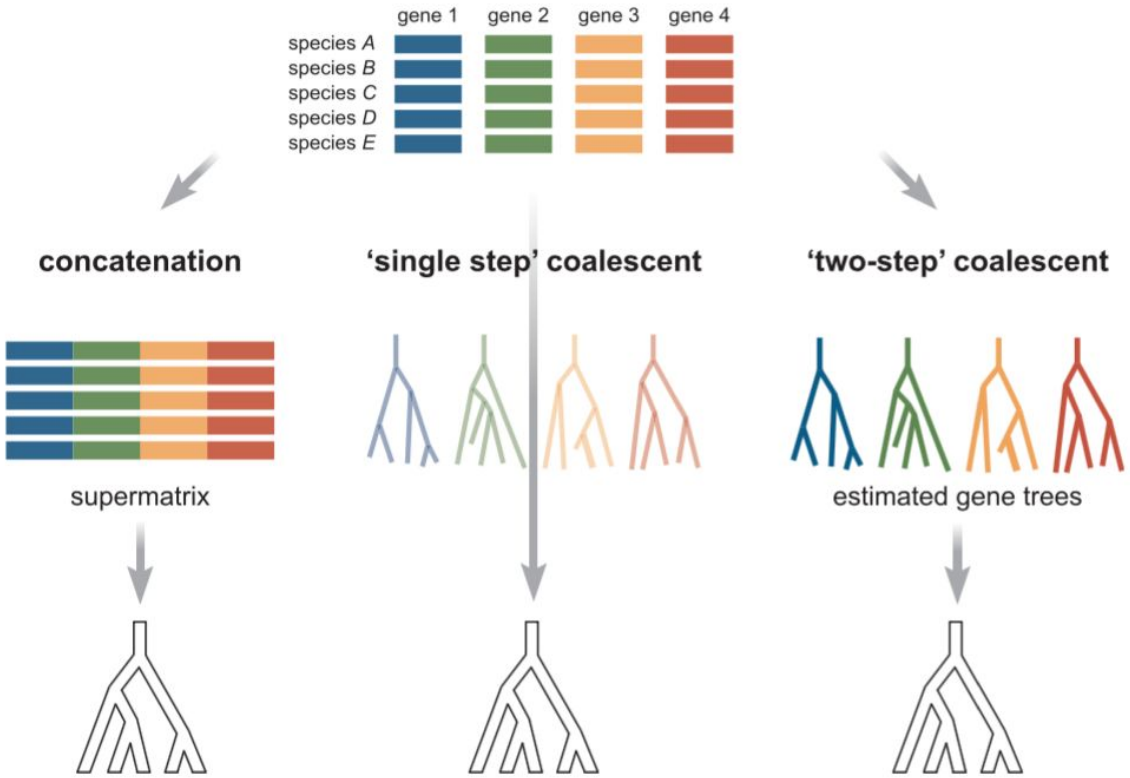


Fig. 1. Liu et al. 2015. *Ann. N.Y. Acad. Sci.* 0:1–18.

To concatenate or not to concatenate? That is the question...

To concatenate or not to concatenate? That is the question...

ELSEVIER

Molecular Phylogenetics and Evolution

Volume 91, October 2015, Pages 98–122

Coalescence vs. concatenation: Sophisticated analyses vs. first principles applied to rooting the angiosperms ☆

Mark P. Simmons[a], John Gatesy[b]

[a] Department of Biology, Colorado State University, Fort Collins, CO 80523, USA

[b] Department of Biology, University of California, Riverside, CA 92521, USA

Received 12 February 2015, Revised 1 May 2015, Accepted 14 May 2015, Available online 19 May 2015

Syst. Biol. 63(6):919–932, 2014

DOI:10.1093/sysbio/syu055
Advance Access publication July 30, 2014

Coalescent versus Concatenation Methods and the Placement of *Amborella* as Sister to Water Lilies

Zhenxiang Xi[1], Liang Liu[2], Joshua S. Rest[3], and Charles C. Davis[1,*]

[1]*Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA;* [2]*Department of Statistics and Institute of Bioinformatics, University of Georgia, Athens, GA 30602, USA; and* [3]*Department of Ecology and Evolution, Stony Brook University, Stony Brook, NY 11794, USA;*
**Correspondence to be sent to: Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA; E-mail: cdavis@oeb.harvard.edu.*

Received 7 January 2014; reviews returned 2 April 2014; accepted 24 July 2014
Associate Editor: Erika Edwards

Syst. Biol. 56(1):17–24, 2007

ISSN: 1063-5157 print / 1076-836X online
DOI: 10.1080/10635150601146041

Inconsistency of Phylogenetic Estimates from Concatenated Data under Coalescence

Laura Salter Kubatko[1] and James H. Degnan[2]

[1]*Departments of Statistics and Evolution, Ecology, and Organismal Biology, The Ohio State University, Columbus, Ohio 43210, USA; E-mail: lkubatko@stat.ohio-state.edu*
[2]*Department of Biostatistics, Harvard School of Public Health, Building 2, 4th Floor, 655 Huntington Avenue, Boston, Massachusetts 02115, USA*

Although multiple gene sequences are becoming increasingly available for molecular phylogenetic inference, the analysis of such data has largely relied on inference methods designed for single genes. One of the common approaches to analyzing data from multiple genes is concatenation of the individual gene data to form a single supergene to which traditional phylogenetic inference procedures—e.g., maximum parsimony (MP) or maximum likelihood (ML)—are applied. Recent empirical studies have demonstrated that concatenation of sequences from multiple genes prior to phylogenetic analysis often results

PLOS | CURRENTS TREE OF LIFE

Concatenation and Species Tree Methods Exhibit Statistically Indistinguishable Accuracy under a Range of Simulated Conditions

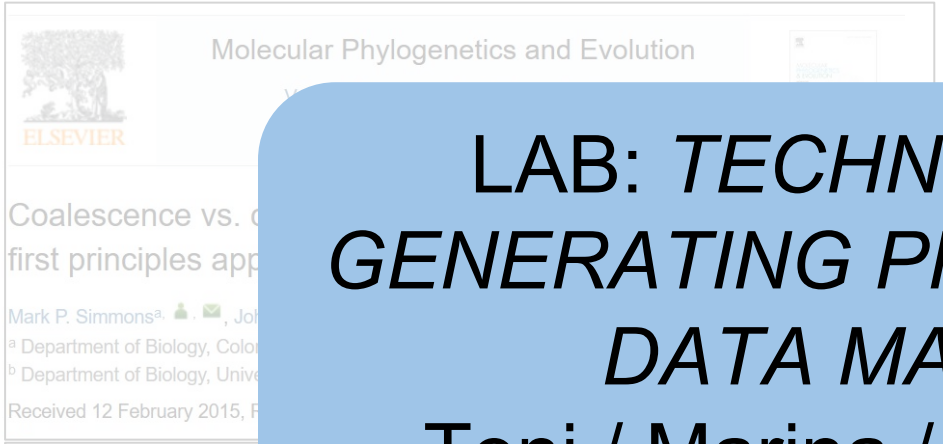
March 9, 2015 · Systematics

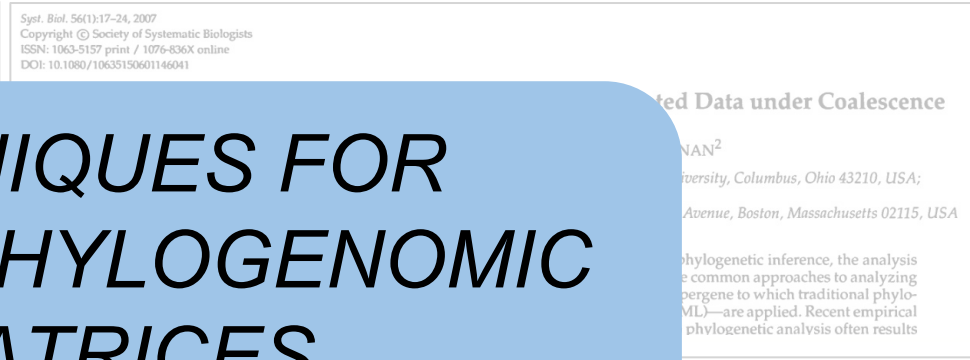
Citation

Tonini J, Moore A, Stern D, Shcheglovitova M, Ortí G. Concatenation and Species Tree Methods Exhibit Statistically Indistinguishable Accuracy under a Range of Simulated Conditions. PLOS Currents Tree of Life. 2015 Mar 9 . Edition 1. doi: 10.1371/currents.tol.34260cc27551a527b124ec5f6334b6be.

Tweet

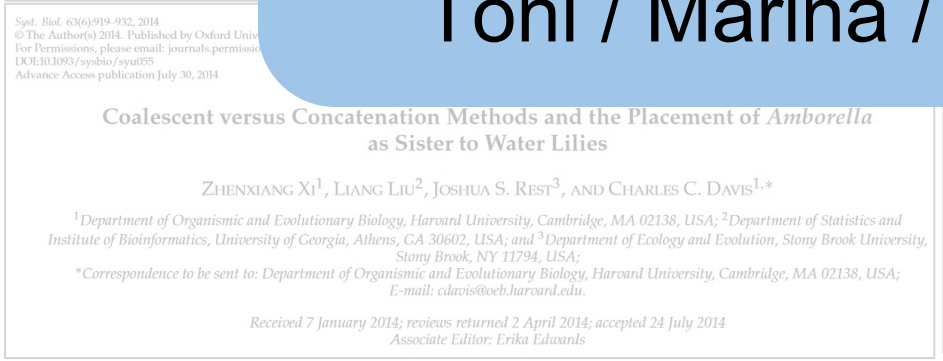
To concatenate or not to concatenate? That is the question...

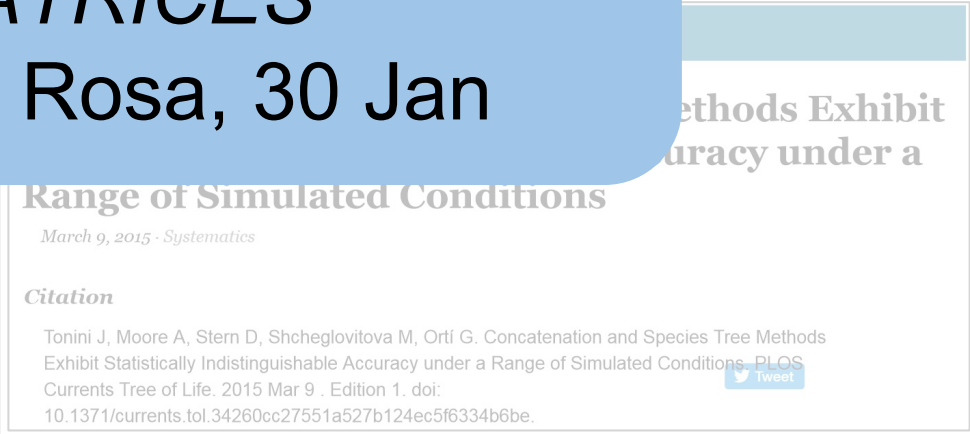




LAB: *TECHNIQUES FOR GENERATING PHYLOGENOMIC DATA MATRICES*

Toni / Marina / Rosa, 30 Jan





To concatenate or not to concatenate? That is the question...

LAB: *TECHNIQUES FOR GENERATING PHYLOGENOMIC DATA MATRICES*

Toni / Marina / Rosa, 30 Jan

Species-tree estimation

Laura Kubatko, 31 Jan

phyutility

Phyloinformatic utility

Phyutility (fyoo-til-i-te) is a command line program that performs simple analyses or modifications on both trees and data matrices. Makes use of JADE (PEBLS) and JEBL libraries. Please see the NEWS page for info concerning updates, etc.

See NEWS and download the new release for amino acid acceptance in the concatenate and cleaning functions (use the -aa argument in -clean and -concat functions).

Please use this citation when using Phyutility Smith, S. A. and Dunn, C. W. (2008) Phyutility: a phyloinformatics tool for trees, alignments, and molecular data. Bioinformatics. 24: 715-716

Currently it performs the following functions (to suggest another feature, submit an Issue and use the label Type-Enhancement) :

Trees

- rerooting
- pruning
- type conversion
- consensus
- leaf stability
- lineage movement
- tree support

Data Matrices

- concatenate alignments
- genbank parsing
- trimming alignments
- search NCBI
- fetch NCBI

phyutility

a) Concatenate the ortholog genes:

phyutility -concat -in INPUT_FILES -out OUTPUT_FILE

The output file will be in **nexus format**. It will include a block showing where each gene starts and ends.

phyutility

a) Concatenate the ortholog genes:

phyutility -concat -in INPUT_FILES -out OUTPUT_FILE

The output file will be in **nexus format**. It will include a block showing where each gene starts and ends.

b) Create a partition file (command line with VIM).

Tutorial: Concatenation with Phyutility

1) Phyutility is a command line program written in Java. We'll concatenate the same genes that were aligned with Pasta and trimmed with trimAl.

   If you had problems, you can use the pre-aligned genes in the folder **'~/workshop_materials/orthologs_concatenation/*.aligned.fa'.**

2) Open the terminal. You'll find the shortcut in the AMI Desktop.

3) Write the following commands:

phyutility -concat -in ~/workshop_materials/orthologs_concatenation/*.aligned.fa -out workshop_materials/orthologs_concatenation/concatenated_matrix.nexus

4) **concatenated_matrix.nexus** is your concatenated matrix in nexus format.

5) Write: **vim concatenated_matrix.nexus** and inspect the file (or **nano concatenated_matrix.nexus**).

How many taxa and amino acids does our matrix have?

Tutorial: Partitioning by gene

For most phylogenetics/phylogenomics programs, you'll want to prepare a partition file. There are different ways of partitioning. Let's first partition **by gene**.

Let's do a file with partitions by gene by using our concatenated matrix (nexus format), as it contains a block indicating the starting and ending position of each gene.

1) Open the concatenated matrix with vim:

vim concatenated_matrix.nexus

2) Select with your mouse the block of partitions. Copy it (right click -> copy).

3) Create a new, empty file called **partitions.txt**

touch partitions.txt

Tutorial: Partitioning by gene

4) Open the file:

vim partitions.txt (you can have a look at the Vim cheat sheet here)

5) Click ‘i’ to be able to edit the file. You’ll see that you’re now in INSERT mode.

6) Paste the block that you copied before (right click -> paste).

7) Remove the extension of the files:

:%s/.aligned.fa_//g

8) Now the names of all the genes are in the same line. Let’s put them in different lines:

:%s/ O/\r/g

Tutorial: Partitioning by gene

9) Delete everything but the gene number:

:%s/G.*g/g/g

10) Insert the '=' symbol after the name of each gene:

:%s/ / = /g

11) Add the evolutionary model that you want to apply to each partition (eg., LG4X, WAG, etc.). In the new versions of RAxML and in ExaML, there's the option of automatic selection (AUTO).

:%s!^!AUTO, !

12) Save the file. Just type:
:w

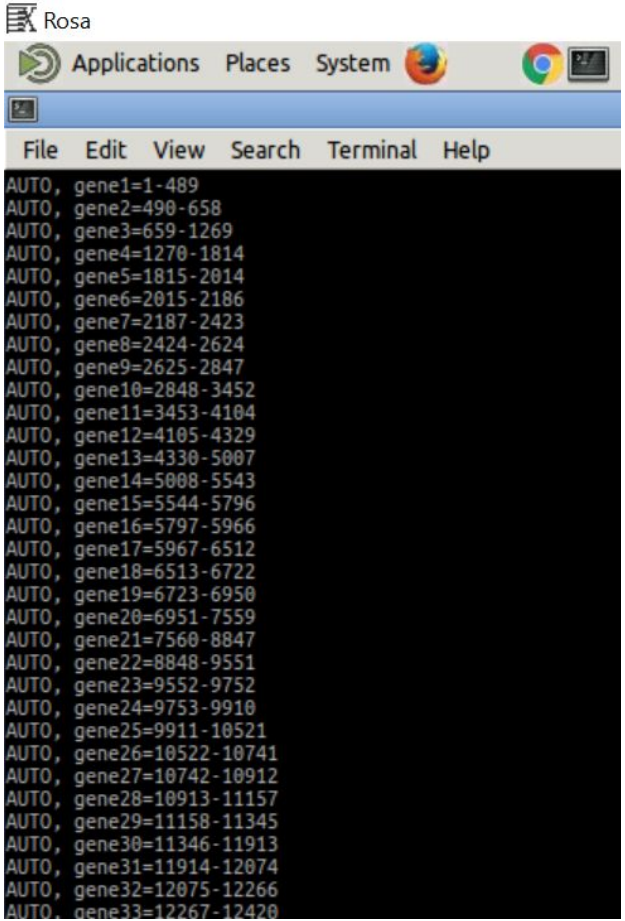
To quit vim, just type **:q!**

Tutorial: Partitioning by gene

12) **CONGRATS!! Your partition file is ready!!**

Check the partition file. It should look like this:

You can go to the beginning of the file by typing **‘gg’**, and to the end of the file with **‘shift + gg’.**



Tutorial: Partitioning with PartitionFinder2

Alternatively to partitioning by gene, we can also use some software to find best-fitting partition schemes. Let's try to find them with **PartitionFinder2**.

What PartitionFinder2 is for

PartitionFinder2 is a program for selecting best-fit partitioning schemes and models of evolution for nucleotide, amino acid, and morphology alignments. The user provides an alignment, and optionally some pre-defined data blocks (e.g. 9 data blocks defining the 1st, 2nd and 3rd codon positions of 3 protein-coding genes, see Figure 1). The program then finds the best partitioning scheme for this dataset, at the same time as selecting best-fit models for each subset of sites/columns. Here are a few things you can do with the program:

1. Find the best-fit partitioning scheme nucleotide, amino acid, or morphology datasets
2. Compare any number of user-defined partitioning schemes
3. Find best-fit models of evolution for each subset in any partitioned dataset (much like you might do with ModelTest or ProtTest).

The idea is that finding best-fit partitioning schemes and models of evolution will improve any downstream analyses of your data, like estimating phylogenetic trees or molecular dates. All of those kinds of analyses assume that your model of evolution is correct, and PartitionFinder2 helps make the model as good as it can be.

Tutorial: Partitioning with PartitionFinder2

Let's copy the folder from the workshop webpage to your instance in the AMI.

1) Download the program from this link. We're doing this because it's not in the instance, so you can delete it from your computer after the practice. Also, it's a good exercise to learn how to use the 'scp' option to copy files or folders between a local host and a remote host.

Tutorial: Partitioning with PartitionFinder2

Let's copy the folder from the workshop webpage to your instance in the AMI.

1) Download the program from this link. We're doing this because it's not in the instance, so you can delete it from your computer after the practice. Also, it's a good exercise to learn how to use the 'scp' option to copy files or folders between a local host and a remote host.

2) Open the terminal in your computer. Write the following:

scp -r path_to_folder phylogenomics@your_public_DNS:~/worskhop_materials

It will ask for the password: **evomics2017**

Tutorial: Partitioning with PartitionFinder2

Let's copy the folder from the workshop webpage to your instance in the AMI.

1) Download the program from this link. We're doing this because it's not in the instance, so you can delete it from your computer after the practice. Also, it's a good exercise to learn how to use the 'scp' option to copy files or folders between a local host and a remote host.

2) Open the terminal in your computer. Write the following:

scp -r path_to_folder phylogenomics@your_public_DNS:~/worskhop_materials

It will ask for the password: **evomics2017**

3) Go to your AMI. Open the 'workshop_material' folder. Double-click the zip file and extract all the folders (or use tar -xzvf partitionfinder-2.1.1.tar.gz from your terminal).

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

5) Go to the folder examples -> aminoacids. Open the **.cfg** file and have a look at it. There's information about each parameter in the documentation.

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

5) Go to the folder examples -> aminoacids. Open the .cfg file and have a look at it. There's information about each parameter in the documentation.

6) Let's run an example. In your AMI, open a terminal. Go to **'workshop_materials -> partitionfinder_2.1.1"**

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

5) Go to the folder examples -> aminoacids. Open the **.cfg** file and have a look at it. There's information about each parameter in the documentation.

6) Let's run an example. In your AMI, open a terminal. Go to **'workshop_materials -> partitionfinder_2.1.1"**

7) Type the following:

python PartitionFinderProtein.py examples/aminoacid/

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

5) Go to the folder examples -> aminoacids. Open the **.cfg** file and have a look at it. There's information about each parameter in the documentation.

6) Let's run an example. In your AMI, open a terminal. Go to **'workshop_materials -> partitionfinder_2.1.1"**

7) Type the following:

python PartitionFinderProtein.py examples/aminoacid/

8) Once its finished, go to 'examples -> aminoacid' and explore the output: **log.txt** and a folder called **analysis.**

Tutorial: Partitioning with PartitionFinder2

4) Navigate to the extracted folder. You'll see several python scripts. You'll need to chose the correct one depending on the type of data that you have (morphology, nucleotides or amino acids).

5) Go to the folder examples -> aminoacids. Open the **.cfg** file and have a look at it. There's information about each parameter in the documentation.

6) Let's run an example. In your AMI, open a terminal. Go to **'workshop_materials -> partitionfinder_2.1.1"**

7) Type the following:

python PartitionFinderProtein.py examples/aminoacid/

8) Once its finished, go to 'examples -> aminoacid' and explore the output: **log.txt** and a folder called **analysis.**

9) In the folder analyses, you have the file **best_scheme.txt.** Have a look at it.

Putting it all together...

Exercise: alignment, trimming, concatenating, creating partition files

In the folder ‘workshop_materials -> ortholog_concatenation’ there are ~ 100 1:1 orthologs. They’re called **OG[NUMBER].fa**

Following the same steps that you’ve learned in the labs today

- align them with PASTA
- trim them with trimAL
- create a concatenated matrix
- create a partition-by-gene file, and
- create a best-fit partition file with PartitionFinder2 (you’ll need to change to phylip format with SeaView)