LMAP_S: Lightweight Multigene Alignment and Phylogeny eStimation

MANUAL

Version: 1.0.0 Jun 18th, 2019

Authors:

Emanuel Maldonado[1] and Agostinho Antunes[1,2]

1 CIIMAR/CIMAR – Interdisciplinary Centre of Marine and Environmental Research, University of Porto, Porto, Portugal
2 Department of Biology, Faculty of Sciences, University of Porto, Porto, Portugal

Table of Contents

Figures and Tables

List of Notes

1. INSTALLATION

1.1. LMAP_S Archive

The Table 1, presents the applications (4) and modules (7) included in the LMAP_S archive (LMAP_Svx.x.x.zip).

Table 1. *Summary of LMAP_S applications.*

LMAP_S Application	Functionality	LMAP_S Module
lmap-s.pl	Alignment and phylogeny estimation.	MyPhylo.pm, MyUtil.pm, MyISWU.pm, MyNotify.pm, MyPhyloInfo.pm, MyMMAP.pm, MyPPMSF.pm
RYcode.pl	RY coding of multiple sequence alignment.	-
install.pl	Install LMAP_S requirements.	-
configure.pl	LMAP_S configuration.	MyNotify.pm, MyUtil.pm, MyISWU.pm

Beyond these central elements, the archive includes a folder containing a dataset ("EXAMPLEDATASET") for which results were calculated ("EXAMPLEDATASETRESULTS") with LMAP_S package. Additionally, the command which produced these results is indicated in the file named "*lmap-s.command.txt*" and explained in the accompanying "*README.txt*" file.

The dataset included demonstrates the usefulness and simplicity that LMAP_S package provides. Its sole purpose is to help users to understand how to prepare input files and to show how LMAP_S works by enabling immediate trial and testing.

1.1.1. Requirements

LMAP_S package was implemented in Perl (https://www.perl.org/) and it is assumed that it is already installed on the user workstation. CPAN (https://metacpan.org/) should also have been configured in your workstation. LMAP_S requires the following external modules and programs (Table 2). Additionally, requirements for each of the integrated software (Table 3) should be previously satisfied.

Table 2. *Required Perl CPAN Modules.*

Module	Observations
IO::All	Input/Output
Email::Sender	Email notifications
Email::MIME	Email notifications
Sys::Info	System information
Term::ReadKey	Terminal operations
Thread::Semaphore	Threads
Bio::TreeIO	Phylogeny editing operations (BioPerl [1])
File::Copy	File operations
File::Copy::Recursive	File operations

Table 3. *LMAP_S Required software (31).*

Software	Available from Ubuntu Repository	Information
Clustal Omega [2] (v.1.2.1)	Yes	http://www.clustal.org/omega/
ClustalW [3] (v.2.1)	Yes	http://www.clustal.org/clustal2/
CONSEL [4] (v.1.2.0/0.20)	No	http://stat.sys.i.kyoto-u.ac.jp/prog/consel/
Degen [5, 6] (v.1.4)	No	http://www.phylotools.com/ptdegenoverview.htm (Included)
Dialign-tx [7] (v.1.0.2)	Yes	http://dialign-tx.gobics.de/
EvalMSA [8] (v.1.0)	No	https://sourceforge.net/projects/evalmsa/
FSA [9] (requires MUMmer [10]) (v.1.15.9)	No (Yes)	http://fsa.sourceforge.net/ (http://mummer.sourceforge.net/)
Gblocks [11, 12] (v.0.91b)	No	http://molevol.cmima.csic.es/castresana/Gblocks.html
GramAlign [13] (v.3.0)	No	http://bioinfo.unl.edu/gramalign.php
IQ-TREE [14, 15] (v.1.3.11.1/v.1.6.2)	Yes	http://www.iqtree.org/
Kalign [16, 17] (v.2.0.4)	Yes	http://msa.cgb.ki.se/cgi-bin/msa.cgi
MACSE [18] (v.1.0.2)	No	http://bioweb.supagro.inra.fr/macse/index.php
MAFFT [19]	Yes	https://mafft.cbrc.jp/alignment/software/source.html

(v.7.271)		
MaxAlign [20] (v.1.1)	No	http://www.cbs.dtu.dk/services/MaxAlign/
MergeAlign [21] (not found)	No	http://mergealign.appspot.com/ http://www.stevekellylab.com/software/mergealign
MPBoot [22] (v.1.1.0)	No	http://www.cibiv.at/software/mpboot/
MUSCLE [23, 24] (v.3.8.31)	Yes	https://www.drive5.com/muscle/
Ninja [25] (v.1.2.2)	No	http://wheelerlab.org/software/ninja/download.html
Noisy [26] (v.1.5.12?)	No	https://www.bioinf.uni-leipzig.de/Software/noisy/
OD-Seq [27] (v.1.0)	No	http://www.bioinf.ucd.ie/download/od-seq.tar.gz
Opal [28] (v.2.1.3)	No	http://opal.cs.arizona.edu/index.html
Prank [29] (v.150803)	Yes	http://wasabiapp.org/download/prank/
Probalign [30] (v.1.4)	Yes	http://probalign.njit.edu/standalone.html
ProbCons [31] (v.1.12)	Yes	http://probcons.stanford.edu/download.html
PSAR-Align [32, 33] (v.1.0?)	No	http://bioen-compbio.bioen.illinois.edu/PSAR-Align/
RYcode (v.1.0.0)	No	https://lmap-s.sourceforge.io (part of LMAP_S)
SMS [34] (v.1.8.1?)	No	http://www.atgc-montpellier.fr/sms/binaries.php
T-COFFEE (TCS) [35, 36] (v.11.00.8cbe486)	Yes	http://www.tcoffee.org/Projects/tcoffee/
TreeCmp [37] (v.1.1)	No	http://kaims.eti.pg.gda.pl/~dambo/treecmp/
TrimAl [38] (v.1.4)	No	http://trimal.cgenomics.org/downloads
WeaveAlign [39] (v.1.2.1)	No	http://statalign.github.io/WeaveAlign/downloads.html
Screen(#)	Yes	https://www.gnu.org/software/screen/
Sendmail(*)	Yes	http://www.sendmail.com/sm/open_source/
Sed	Yes($)	https://www.unix.com/man-page/linux/1/sed/
Perl 5	Yes($)	https://www.perl.org/
Java	Yes($)	https://www.java.com/en/

Python	Yes($)	https://www.python.org/

(*) – Assuming the user will benefit from email notification, this is listed as required software and is installed automatically (see section **1.2.1.1**). If otherwise the email notification is not required, the installation of this utility can be discarded (see section **1.2.2** step 2).

(#) – Please see Note 13.

($) – Usually requires no installation (other required pre-installed software includes the Linux commands, such as "**cp**", "**mv**", "**rm**", "**ps**" and "**tput**").

NOTE: The versions indicated in gray text, were employed in the development and implementation of LMAP_S and hence are recommended, but not mandatory. Newer versions may replace these, especially for cases where software is mature and it is not expected to suffer drastic changes, e.g. clustalw/o, mafft or prank.

1.2. Instructions

These instructions make use of the Ubuntu package manager: APT (*apt* command).
If you are using a different Linux distribution, you will need to install programs manually or with an appropriate package manager available to your system.

LMAP_S contains the INTEGRATEDSOFTWARE directory, which separately contains the software from repository (in "AVAILABLEBYREPOSITORY", see section **1.2.1**) and the software to be installed manually (in "NOTAVAILABLEBYREPOSITORY", see section **1.2.2**). The former is provided simply to ensure that the LMAP_S original versions on top of which was developed are continuously available thus ensuring continued functionality. The latter lists other software that the user needs to manually place in binary folders like $HOME/BIN.

NOTES FOR MAC OS users:

For MAC OS users, the Xcode Developer tools (https://developer.apple.com/xcode/) are necessary, which will make Perl and other tools available (like *screen* and *sendmail*). See http://learn.perl.org/installing/osx.html.
Some of the required software (Table 3) may have to be installed manually, if the MacOS package manager is not providing their installation.

In Mac OS systems it may be possible that the install scripts and applications will not execute due to different Perl configurations or multiple Perl installations. In this sense, it may be either required to *(i)* change the first line occurring in each LMAP_S application and installation scripts or to *(ii)* just run each application with the *perl* command.

(i) The first line appearing at the top of each LMAP_S executable (files ending in ".pl") is

```
#!/usr/bin/perl -w
```

This line should be changed to:

```
#!/usr/bin/env perl -w
```

(ii) Alternatively, all LMAP_S applications can be executed with *perl* itself:

```
$ perl <appname.pl>
```

See also a list of available package managers:
https://en.wikipedia.org/wiki/List_of_software_package_management_systems

See also how to install BioPerl
http://www.bioperl.org/wiki/Installing_BioPerl_on_Unix

To install LMAP_S you may either (**1.2.1.**) run the included installation scripts; or (**1.2.2.**) install all the requirements manually.

1.2.1. Install Using the LMAP_S Programs

The easiest way to install LMAP_S package, is to 1.2.1.1) run the *install.pl* (version: 1.0.0 Nov 10th, 2018) and 1.2.1.2) *configure.pl* (version: 1.0.0 Nov 10th, 2018) programs located at the base of the LMAP_S unzipped directory.

```
$ cd LMAP_Sv1.0.0/LMAP_S
```

1.2.1.1 Install Requirements

```
$ sudo ./install.pl
```

It will automatically install all the LMAP_S requirements previously listed (Tables 2-3). If any modules or programs were previously installed, they will not be reinstalled.

NOTE 1 (LMAP_S install twice): You may need to run the install script twice, if CPAN was not previously configured in your computer/account.

NOTE 2 (LMAP_S SW replacement): Newer versions can/should replace the original ones, but in case there were profound changes, some of them might not work correctly. In this case, please contact the authors or use the provided versions.

This install procedure will do the following:

1) install CPAN Modules (Table 2)
This is done through specific CPAN oriented modules that allow the installation of other modules.

2) install Programs from repositories (Table 3)
This is done automatically for each program by using the *apt* command.

3) install Programs available from LMAP_S (NOTAVAILABLEBYREPOSITORY) (Table 3)
This is done automatically for each program located in this directory. This possibly includes installing *screen* utility 4.03.01 version.

1.2.1.2. Configure LMAP_S

After the installation of requirements is complete, the following step is to configure

LMAP_S. This step can be repeated any number of times, as per the user requirements, to change any previous configuration or reset configurations.

```
$ ./configure.pl
```

or;

```
$ sudo ./configure.pl
```

This application will do the following:

1) ask the user for configurations:
   i. Disable use of terminal colors;
   ii. Select location of Software executables (it provides auto-detection);
   iii. Email notification settings (optional step; see below, Table 4);
   iv. Select final location of LMAP_S applications

2) generate an integrated software configuration file in the user's $HOME directory (see section **2.3**)

3) generate a preferences file in the user's $HOME directory

4) move all the LMAP_S applications and modules into a user selected (binaries) location.

To have LMAP_S available throughout any directory location in your account/workstation, LMAP_S should be placed in a binaries location. Still, this is not a requirement.

In the first command case, the user will configure LMAP_S to the user's $HOME directory, usually in $HOME/BIN. In the second case with *sudo* command, the configuration of LMAP_S applications will be allowed in a system directory e.g.: /usr/local/bin/, thus available to any user account in the same workstation.

However, assuming there are multiple user accounts that will use LMAP_S, each account requires the LMAP_S configuration file. In this sense, since LMAP_S was already configured "system-wide" (second case), the application should now be executed in each account without *sudo* command (first case) and the last step (iv) requiring the selection of LMAP_S applications location, should be ignored. Here, to this end, the user must select the option "Do nothing, I will copy LMAP_S applications...". After this, the applications are ready and any user will have his own configuration settings in his/her own account. In this situation and in the first case, any user may configure LMAP_S in his account and not require administrative privileges. These are only required when installing/configuring LMAP_S in the system directories.

NOTE 3 (LMAP_S in system directory): to configure LMAP_S in a system directory (e.g.: /usr/local/bin/), run this configuration script with *sudo* command.

To enable email notification, the *sendmail* utility must have been installed (see Table 3 and sections **1.2.1.1** and/or **1.2.2**), and the settings in Table 4 are needed before proceeding to this configuration step. These settings are related to the required CPAN modules (Table 2). The configuration of email notification is optional, hence while interacting with configuration application; the user has the possibility to skip this step. If the user does not need email notifications, then this step is not necessary as also it is not necessary the installation of the *sendmail* utility.

Table 4. *Information required to enable email notifications.*

Information	Dummy Examples
SMTP server hostname and the required port number (or by default 25)	smtp.uni.fac.com:999999
SMTP server (HELO)	uni.fac.ehlo
SMTP server require a secure or encrypted connection	Yes / No
Username and Password for the SMTP account (performed in two steps; password entered in quiet mode)	-smtpaccount@uni.fac.com -smtppass
Default email address to which notifications can be sent	username@uni.fac.com

1.2.2. Install Manually

In order to install manually do the following essential three steps:

1) Install CPAN modules

In your terminal type:

```
$ sudo cpan
```

This will give the CPAN command-line, where you can type to install all modules at once or type install <module> for one module at a time.

```
cpan[1]> install IO::All Email::Sender Email::MIME Sys::Info Term::ReadKey Thread::Semaphore
Bio::TreeIO File::Copy File::Copy::Recursive
```

Alternatively, without root privileges, CPAN can be configured and modules installed in the user account.

2) Install required programs (with administrative privileges: *sudo*)

In your terminal type:

```
$ sudo apt install clustalw clustalo dialign-tx kalign mafft mummer muscle prank probalign probcons
t_coffee iqtree java screen sendmail-bin
```

or;

```
$ sudo apt install clustalw clustalo dialign-tx kalign mafft mummer muscle prank probalign probcons
t_coffee iqtree java screen
```

If email notification is not required, the installation of *sendmail* utility (Table 3) can be discarded (see section **1.2.1.2**). In both cases, it is assumed that all required programs will be installed for the first time, otherwise their installation can also be discarded.

Alternatively, without *sudo*,

download and install the software listed in Table 3 from their websites and follow instructions therein; and

download and install *screen* from https://www.gnu.org/software/screen/
and follow *screen*'s instructions; and

download and install *sendmail* from http://www.sendmail.com/sm/open_source/download/ , ftp://ftp.sendmail.org/pub/sendmail
and follow *sendmail*'s instructions.

Additionally to install remaining software from LMAP_S NOTAVAILABLEBYREPOSITORY folder, a few notes are presented (based on current experience):

-For **MACSE** the java archive is provided with the software version and thus should be renamed to just “macse.jar”.
-For **Degen** the perl program is provided with its version and thus should be renamed to just “Degen.pl”.
-For **PSAR-Align** the required **fsa** program should be located at a system directory (e.g. /usr/local/bin).
-For **TreeCmp** if located at $HOME/bin the “config” folder should be located at the previous directory, at the user $HOME.

3) Proceed to LMAP_S configuration.

See section **1.2.1.2.**

4) Configure $HOME/BIN.

To make all programs and scripts located at $HOME/BIN (i.e., ~/bin) available at any working directory, it is also necessary to change $HOME/.bashrc file.
Place the following line in $HOME/.bashrc and save.

```
export PATH=$PATH:~/bin/:.
```

(see also: http://askubuntu.com/questions/9848/what-are-path-and-bin-how-can-i-have-personal-scripts).

Finally, reopen your terminal so that changes may take effect.

2. GETTING STARTED

2.1. Preparing Input Files

To run LMAP_S, it is required the preparation of a set of files containing the nucleotide sequences (protein-coding genes) subject to study. These sequences can either be **2.1.1**) organized by gene within a multiple sequence file (MSF) named after each gene, or **2.1.2**) not organized, which means a single (or multiple) file(s) can contain several different genes. In this case, based on a specific sequence description format, LMAP_S is able to detect the GENENAME's within these files and organize them per gene file, as for in **2.1.1**.

2.1.1. Multiple Sequence Files - Ready

[GENENAME].[FILEEXT]

Where:
GENENAME – is any name or abbreviation given to the protein coding genes being analyzed. For instance, in case of mitochondrial encoded subunits cytochrome c oxidase III, this could be abbreviated as "COX3".
FILEEXT – file extensions accepted for FASTA format can be one of "fas", "fasta", "fst" or "txt".

In this case, the file name would be "COX3.fas".

Examples:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all
```

Use all GENENAME's found under Data/MSF location.

```
$ lmap-s.pl -A Data/MSF/ -g COX1,CYTB,ND1 -i 1 -d . -j MyDirectoryStruct -a all
```

In this case, assuming there are more GENENAME's in the MSF location, the option -g will limit the execution to the listed genes.

2.1.2. Multiple Sequence Files - Not Ready

In this case, the files can be identified by any means; however, it is recommended to avoid the use of spaces.

There are two ways to have the MSFs ready: i) while creating the LMAP_S command-line give the option -g to define the list of GENENAME's to create the sequence groups by searching the initial MSF file(s) sequence descriptions that will form the Ready MSF gene files (easiest case). Or ii) in the absence of -g (guess mode), LMAP_S needs to acquire this information directly from the initial MSF files sequences descriptions.
In either case, ensure that the sequences contained therein, follow a specific format, which enables LMAP_S to distinguish the GENENAME from other information. Two formats are

available for this case: 1) requires that the GENENAME or abbreviation is followed by the FASTA ">" sign and preceded by an underscore, and 2) allowing the GENENAME to be found anywhere in the middle, surrounded by two underscores (only two in the whole description!). This allows other information to be present surrounding these formats.

1) >**[GENENAME]_**[other]
2) >[other1]**_[GENENAME]_**[other2]

We recommend the user to employ the first case, since it is stricter and simpler.

Examples:

i)

```
$ lmap-s.pl -A Data/MSF/ -g COX3,CYTB,ND1 -i 1 -d . -j MyDirectoryStruct -a all
```

Or;

ii)

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all
```

Here, all the listed GENENAME's will be searched in the Not-ready MSF files, and taken to form the Ready files from section **2.1.1**.

2.2. LMAP_S Directory Structure

The directory structure created by LMAP_S, has the following profile from base to bottom:

LOCATION/PROJNAME/LMAP_SPOOL/GENENAME
LOCATION/PROJNAME/LMAP_SFINAL/MSAS/GENENAME
LOCATION/PROJNAME/LMAP_SFINAL/TREES/GENENAME
LOCATION/PROJNAME/LMAP_SFINAL/TREES/EDTREES/GENENAME
LOCATION/PROJNAME/LMAP_SREPORTS/

BASE → → → **BOTTOM**

Where:

LOCATION – as given in option -d (*mandatory*; see section **3** and **8**). Where the project will be created and located. May contain several projects.

PROJNAME – as given in option -j (*non-mandatory*; see section **3** and **8**). The project name and the base of the directory structure. If not specified, value defaults to "LMAP_SProjWXYZ", where W, X ,Y, Z, are random digits (from 0 to 9).

LMAP_SPOOL – the location where all provided MSF GENENAME's will be processed throughout the LMAP_S execution. It contains several subfolders for each gene, where all files and intermediate results will be contained.

LMAP_SFINAL – after LMAP_S finishes estimation of MSAs and/or PTs these will be copied into two specific folders located herein for easier access. One folder for MSAs, named 'MSAS', and one folder for PTs, named 'TREES'. The content for these subfolders is also organized in folders for each GENENAME. The PTs folder may additionally include a subfolder name 'EDTREES' to separate the edited trees (see

section **3.7**) from the original ones and has likewise organization per GENENAME.

LMAP_SREPORTS – during LMAP_S execution and depending on the programs/Stages selected to run, a some reports will appear in this directory.

GENENAME – the tip of the directory structure, which ends with the GENENAME identification. All files regarding MSA and PT estimation (and others) are located here (i.e., in each folder identified after the corresponding MSF GENENAME).

2.3. LMAP_S Integrated Software Configuration File

LMAP_S was implemented in a way that the user/researcher is not forced to deal with each integrated software settings, hence default behaviors were created for each case, which formed "versions" of each software that we call "algorithms". However, knowing that each researcher has its own preferences or requirements we have made it such that he/she is able to customize any of the algorithms settings whenever needed. Any algorithm is customizable, however to maintain the more specialized cases functional, we recommend that only the default ones are edited.

All these cases are hard-coded into LMAP_S modules, which provides their initial/starting configuration and further enables the user to restore them whenever necessary. To this end, the user is only required to re-run LMAP_S configuration application (*configure.pl* – see section **1.2**).

When this application is executed a hidden file is created/overwritten at the user $HOME directory, named "**.lmapsSwConfigs**". This file enables the user to make any modifications to the established algorithms command lines, for instance, add/modify existing arguments or change the location of the binaries/executables (e.g. to use different versions).

The exception to this rule is however, *TreeCmp*, which has very strict rules and requires careful editing. Hence, we recommend the user to read *TreeCmp* materials beforehand. Modifying or removing any of this software configuration will ultimately hamper LMAP_S from correct functioning regarding the LMAP_S Reports produced at PCC Stage (see section **3.6**).

NOTE 4 (Tags in .lmapsSwConfig file): In this file, there are tags of the form <tag> that should not be modified to avoid LMAP_S malfunctioning.

Beyond the LMAP_S integrated software command definitions, a few parameters need attention. Few software require additional parameters beyond the input files and output files, which is the case of *DIALIGN-TX*, *PSAR-Align*, *EVALMSA*, *MACSE* for a specific algorithm and *T-COFFEE TCS* methods (Table 5).

Table 5. *LMAP_S Software configuration file parameters.*

Software	PARAMETER	Description	Observations
Dialign-Tx	**DIALIGNTX_MTXDIR**	Dialign-tx matrices location.	/usr/share/dialign-tx/ (by default)

PSAR-Align	**PSARALIGN_PARAMFL**	PSAR-Align parameters file.	$HOME/bin/parameters.txt (by default)
EvalMSA	**EVALMSA_MTXFL**	EvalMSA matrix selection.	$HOME/bin/Matrix/blosum62 (by default)
MACSE (MACSEP - Table 9)	**MACSEPS_LRFL**	MACSE LR file containing the required sequences. If this file is missing, this algorithm will give error.	For the MACSE pseudogenes algorithm. To be set every time the filename changes.
T-COFFEE (TCS)	**TCS_NCORENV**	The number of cores for TCS execution.	4 (by default). This value can be adjusted, but was established to avoid/prevent CPU competition between TCS and LMAP_S, which may take to performance issues.

These parameters are configured with the *configure.pl* application and can also be changed or adapted to each user conditions by setting them as Linux system environment variables. The user may modify these values any time as he/she sees fit, before running LMAP_S. To set them as environment variables simply use the same **PARAMETER** name as follows, e.g.:

```
$ export MACSEPS_LRFL="myfile/path/location"
```

LMAP_S will first verify if these variables are defined and if yes, assume their values. Otherwise, it will resort to use the definitions from the configuration file. Hence, it means the environment variables provide more flexibility and have higher precedence than the values from the configuration file.

2.4. LMAP_S Programmer Configurations

This section is devoted to present a few useful aspects that can be modified in some modules and *lmap-s.pl* application. These changes take place in the beginning of the code/file. We do not recommend any changes, unless strictly necessary. If needed please, contact the authors for any help beforehand.

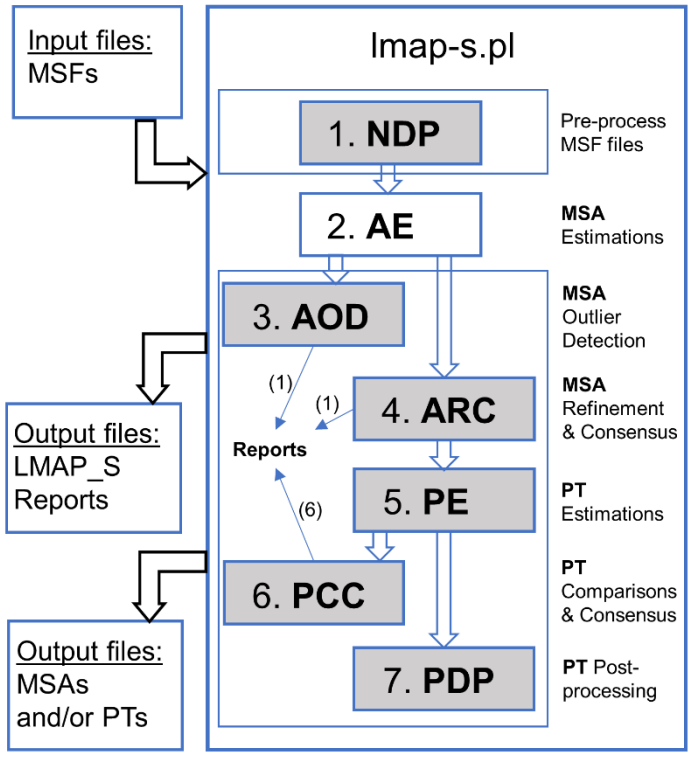
***LMAP_Slib/MyPhyloInfo.pm* – Modify Site-likelihoods values**
Here it is possible to modify LMAP_S behavior to enable site-likelihoods rounding and/or trimming of decimal places. By default, they are not used (off).
To modify this behavior, open this module with a text editor and set the values of the lines (first occurrences of) "ROUNDIT" to round and trim and of "LIM_DECPLACES" to only trim. The latter specifies the number of decimal places that will be used in either case; hence, the former will be sufficient to set to 1. Save and exit.

***LMAP_Slib/MyISWU.pm* – Modify/add other software/algorithm to any Stage**
This case is possible and somewhat uncomplicated, but much more delicate. If really needed the reader should have knowledge of the Perl language and programming skills. Please read **very carefully to understand** the instructions in MyISWU.pm (starting around line 90).

3. APPLICATION *lmap-s.pl* (version: 1.0.0 Apr 30th, 2019)

The *lmap-s.pl* application works in seven Stages (Figure 1), or at minimum one. AE Stage is the only one, which is mandatory and concerns the estimation of multiple sequence alignments (MSA).

These Stages will be covered each one per section.



Legend: **NDP** – Nucleotide Data Pre-processing ; **AE** – MSA Estimation ; **AOD** – MSA Outlier Detection ; **ARC** – MSA Refinement and Consensus ; **PE** – PT Estimation ; **PCC** – PT Comparison and Consensus ; **PDP** – PT Data Post-processing.

***Figure 1**. LMAP_S Workflow.*

Although these Stages are optional (boxes in gray color), they can be selected and included in the workflow as per the researcher's necessity. This gives the researcher the flexibility to perform different analysis or with different extensions.

In relation to each of these Stages, the following set of options are shown in Table 6.

Table 6. *LMAP_S Command-line options and corresponding Stages.*

LMAP_S CLOs	ARGUMENT	Description	STAGE
-A	LMAP_S input data directory	Select the location of nucleotide MSF files (see section **2.1**) (**MANDATORY**)	n.a.
-p	NDP operations	Select operations to perform on MSF files (see Table 8; section **3.1**)	**1**
-a	AE algorithms	Select MSA algorithms to apply to MSF gene files (see Table 9; section **3.2**) (**MANDATORY**)	**2**
-b	n.a.	Perform outlier detection and report (see section **3.3**)	**3**
-c	ARC algorithms	Select MSA refinement algorithms to apply to existing MSAs (see Table 10; section **3.4**)	**4**
-m	MSA group selector	Select which groups of MSAs will be used for posterior Stages (see Table 11; section **3.4**)	**Affects 5, 6 and 7**
-t	PE algorithms	Select phylogenetic tree estimation algorithms to apply to MSA files (see Table 12; section **3.5**)	**5**
-s	n.a.	Perform statistical and topological comparison and produce reports (see section **3.6**)	**6**
-q	PDP operations	Select operations to perform on PTs (see Table 13; section **3.7**)	**7**
-g	*GENENAME* comma-separated list	Select/Find genes given. To be used with -p RN or to define the (sub)group of genes to follow up (see section **2.1**)	n.a.
-i	Translation table code	The genetic code fit to the dataset (see Table 7) (**MANDATORY**)	n.a.
-n	Number of tasks/CPUs	Specify the number of tasks/CPUs to allocate. If not given, will try to maximize the CPU usage (see section **5**)	n.a.
-d	Project base location	The location to were the project will be created (see section **2.2**) (**MANDATORY**)	n.a.
-j	Project title	The project title (if not given, a default identification will be created) (see section **2.2**)	n.a.
-e	Email address	Specify a different email address for notification on termination (see section **3.9**)	n.a.
-l	Logging of final status	If specified, LMAP_S will produce a log file containing a compilation of all “Final Status” screens. They are also attached in email notifications (option -e) (see Note 15; section **3.10** and **5.1**)	n.a.
-h	Help menu	Display command-line help. Please see also variant help commands in section **3.11**	

n.a. – not applicable.
CLO – Command-line Option.
Blue text lines distinguishes the mandatory CLOs; the remainder are optional. The optional CLOs are also

marked in the *lmap-s.pl* help menu (see Figure 5; section 3.11).

Table 7. *LMAP_S Genetic codes as from GenBank and according to software.*

LMAP_S Code (option -i)	Genetic codes (TRANSLATION TABLE)	Description	Software availability(*)
0	1	Universal	MC/MT/IQ/DEG
1	2	Vertebrate mitochondrial	MC/MT/IQ/DEG
2	3	Yeast mitochondrial	MC/MT/IQ/DEG
3	4	Mold, Protozoan, and Coelenterate mitochondrial and the Mycoplasma/Spiroplasma	MC/MT/IQ/DEG
4	5	Invertebrate mitochondrial	MC/MT/IQ/DEG
5	6	Ciliate, Dasycladacean and Hexamita nuclear	MC/MT/IQ/DEG
6	9	Echinoderm and Flatworm mitochondrial	MC/MT/IQ/DEG
7	10	Euplotid mitochondrial	MC/MT/IQ/DEG
8	11	Bacterial, Archaeal and Plant Plastid	MC/MT/IQ/DEG
9	12	Alternative Yeast nuclear	MC/MT/IQ/DEG
10	13	Ascidian mitochondrial	MC/MT/IQ/DEG
11	14	Alternative Flatworm mitochondrial	MC/MT/IQ/DEG
12	15	Blepharisma nuclear	MC/MT
13	16	Chlorophycean mitochondrial	MC/MT/IQ
14	21	Trematode mitochondrial	MC/MT/IQ
15	22	Scenedesmus obliquus mitochondrial	MC/MT/IQ
16	23	Thraustochytrium mitochondrial	MC/MT/IQ
17	24	Pterobranchia mitochondrial	MT/IQ
18	25	Candidate Division SR1 and Gracilibacteria	MT/IQ
19	26	Pachysolen tannophilus nuclear	n.a.
20	27	Karyorelict nuclear	n.a.
21	28	Condylostoma nuclear	n.a.
22	29	Mesodinium nuclear	n.a.
23	30	Peritrich nuclear	n.a.
24	31	Blastocrithidia nuclear	n.a.

n.a. – not applicable.

(*) – This table lists all genetic codes available to LMAP_S. This column with the indicated software shows the range of codes that they enable or allow. The specification of the genetic code may be required under some conditions, not all. LMAP_S will still be able to use the remaining codes with other Stages or with other 'generic' software. MC = MACSE ; IQ = IQ-TREE ; DEG = DEGEN ; MT = MPBOOT

3.1. Stage 1 – MSF Pre-processing (NDP)

This Stage enables the researcher to enforce some data consistency and preparation of the initial sequences at hand to the following Stages. This is accomplished in two ways, *(i)* by default behavior and *(ii)* by researcher criteria.

In the first case, LMAP_S by default provides for the creation, distribution and preparation of MSF files across the directory structure, assuming files are Ready (see section **2.1.1**).

The second case, enables the researcher to select any MSF data pre-processing options listed in Table 8. This is an optional step that places NDP Stage as part of LMAP_S workflow when using option -p.

Table 8. *Stage 1. NDP – LMAP_S Pre-processing operations.*

LMAP_S code (option -p)	ARGUMENT	Description	Observations
RD	Filename.csv	Rename Sequence Descriptions	CSV data organized in columns. Two columns per gene, organized horizontally (A1, A2, B1, B2, etc). First row identifies the GENENAME (left column). Per gene and top-down, the left column containing the original description and right column the required description. Hyphens fill in the missing cell values within the matrix rectangle (Figure 2). Likewise, other can be added in lines below CYTB.
RI	Filename.csv	Remove Sequence Indexes	CSV data organized in rows. First column identifies each GENENAME per row. Following columns identify the sequence indexes to be removed from the MSF. Indexes are numbers (starting at 1) attributed to sequences top-down in the file. Hence, each row identifies the indexes per GENENAME (Figure 3). Likewise, other can be added in lines below CYTB.
RN	n.a.	MSF Not Ready	Input files are treated as “Not Ready”, meaning the sequences will be retrieved and organized and distributed in new files specific to each gene, which is indicated either in option -g or by the sequence description format existing in input files.
RC	n.a.	Remove Terminal Stop Codons	Removes stop codons found at the end of the sequences, according to the translation table code specified with option -i.
RG	n.a.	Remove Gaps	Removes any gaps (“-“) found in the sequences.

n.a. – not applicable.

	A	B	C	D	E	F	G	H	I	J	K
1	ATP6		COX1		CYTB		ND1				
2	>T7MT_MnEm	New61	>T8MT_MnIm	NewC1	>T59MT_TeHe	NewY1	>T9MT_PsPa	NewN1			
3	>T3MT_TeKl	New62	>T1MT_TeGr	NewC2	>T41MT_RaSw	NewY2	>T3MT_TeKl	NewN2			
4	>T59MT_TeHe	New63	>T5MT_InEl	NewC3	>T40MT_TrTr	NewY3	>T6MT_InFo	NewN3			
5	-	-	>T34MT_PeCa	NewC4	>T37MT_LiPu	NewY4	>T42MT_CaIn	NewN4			
6	-	-	-	-	>T35MT_PeSi	NewY5	>T38MT_ApFe	NewN5			
7	-	-	-	-	-	-	>T37MT_LiPu	NewN6			
8											
9											

***Figure 2**. General aspect of the CSV rename file (option code RD).*

	A	B	C	D	E	F	G	H	I
1	COX1	1	4	6	9	13			
2	ATP6	3	2	7	3	11			
3	ND1	6	9	8	13	14			
4	CYTB	4	5	9	12	15			
5									
6									
7									

***Figure 3**. General aspect of the CSV remove file (option code RI).*

These options must be separated by commas and without spaces, as follows:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -p rd[rename.csv],ri[remove.csv],rg,rc,rn -a all
```

This command will retrieve all the necessary files, MSFs (option -A), from the Data folder and create the directory structure named MyDirectoryStruct (option -j) in the current directory (option -d) prepared to pre-process these files with the indicated modifications (option -p). Option -a, specifies that all MSA algorithms will be executed (see section **3.2**).

This case will serve to present examples throughout the following sections and show the functioning of respective options.

Figures 2 and 3 show the general format of the LMAP_S input files for the corresponding option codes in Table 8. The CSV files are expected with a **semi-colon separator** instead. One example of each is given in the ExampleDataset folder. The reader can visualize their format with any text editor. In the case of the removal of sequences (code RI) the sequences indicated to be removed are kept in a separate file identified with the following format: *[GeneName]_REMOVED.fas.txt.* This file is not taken for any further analysis and is located at each *GeneName* directory.

NOTE 5 (option -p RI: removed sequences safe): the sequences indicated to be removed with code RI are automatically saved in a separate file identified with the following format: *[GeneName]_REMOVED.fas.txt.*

In this Stage, regardless of the data modifications made here, LMAP_S does not consider any specific file identification for the time being. After this Stage, file identification is maintained as the initial MSF *[GeneName].fas.*

E.g. COX1.fas or CYTB.fas.

3.1.1. Sequence Description Manipulation

At this stage by default and regardless of option -p (and its code) specification, LMAP_S manipulates sequence descriptions in order to create the homogeneity necessary throughout remaining stages. In fact, several integrated software make their own sequence descriptions manipulations, which sometimes takes to inconsistencies across different software and stages ultimately taking to errors that hinder LMAP_S workflow execution. Among the identified software, there is *T-COFFEE*, *Prank*, *GramAlign* and *ClustalW* from AE Stage (3.2) *PSAR-Align* and *NOISY* from ARC Stage (3.4) *Degen* and *SMS* from PE Stage (3.5). The characters targeted by these cases were essentially 'space', 'semicolon', 'comma' and 'hyphen'. Exceptionally, *NOISY* inserted an additional 'space' after the FASTA ">" sign, which LMAP_S corrects by removing it soon after *NOISY* terminates.
LMAP_S tries to alleviate the described cases, by replacing them by 'underscore'. Beyond the detected characters, LMAP_S also targets any existence of '@', '%', '&' and '|'. Additionally, if sequence description total length is found to be greater than 100 characters, any existing words like "genome", "transcriptome", "partial" and "complete" are automatically removed.

3.2. Stage 2 – MSA Estimation (AE)

In this Stage, LMAP_S provides several MSA algorithms to enable a wider choice of options and of comparability. Contrary to other Stage options/selectors, this selector must be present every time.

The integrated MSA algorithms amount to 33 options, respecting to 14 software (Table 9).

Table 9. *Stage 2. AE – LMAP_S MSA estimation algorithms.*

Software	ALGORITHM_S2	LMAP_S code (option -a)	Description
ALL	*ALL*	all	Select all algorithms at once (Single code usage)
Clustal Omega	CLUSTALO	co	ClustalO default
ClustalW	CLUSTALW	cw	ClustalW default
Dialign-TX	DIALIGN-TX	tx	Dialign-tx default
	DIALIGN-TXD	txd	Dialign-tx -D option
	DIALIGN-TXT	txt	Dialign-tx -T option (Translated)
FSA	FSA	fa	FSA default
	FSANP	fat	FSA -nucprot option (Translated)
GramAlign	GRAMALIGN	ga	GramAlign default

Kalign	KALIGN	ka	Kalign default
MACSE	MACSE	mc	MACSE default
	MACSEP (*)	mcp	MACSE PSEUDOGENES
MAFFT	MAFFT	ma	MAFFT default
	MAFFTA	maa	MAFFT AUTO
	MAFFTEI	mei	MAFFT E-INS-i
	MAFFTF1	mf1	MAFFT FFT-NS-1
	MAFFTF2	mf2	MAFFT FFT-NS-2
	MAFFTFI	mfi	MAFFT FFT-NS-i
	MAFFTGI	mgi	MAFFT G-INS-i
	MAFFTLI	mli	MAFFT L-INS-i
MUSCLE	MUSCLE	mu	MUSCLE default
Opal	OPAL	op	Opal default
ProbAlign	PROBALIGN	pa	ProbAlign default
ProbCons	PROBCONS	pc	ProbCons default
Prank	PRANK	pk	Prank default
	PRANKF	pkf	Prank +F option
	PRANKO	pko	Prank Once option
	PRANKCD	pcd	Prank Codon
	PRANKCDF	pcf	Prank Codon +F
	PRANKCDO	pco	Prank Codon Once
T-COFFEE	TCOFFEE	tc	T-COFFEE default (PROBA_PAIR)
	TCOFFEETC	ttc	T-COFFEE T_COFFEE_MSA
	TCOFFEEKT	tkt	T-COFFEE KTUP_MSA
	TCOFFEEPL	tpl	T-COFFEE PLIB_MSA

(*) – This option is only executed or included in 'all' set of options, if the necessary file has been defined (Table 5).

See the following examples below:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all
```

In this case, all the available options are selected to execute the input MSF files.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a ka
```

In this case, kalign (ka) was selected to estimate alignments for the input MSF files.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a mu,pk
```

In this case, muscle (mu) and prank (pk) were selected to estimate alignments for the input MSF files.

Hence, this Stage will generate estimations of alignments for each input MSF (gene) file, for instance, in the first case with one gene MSF, 30 alignments will be estimated, with two gene MSFs, 60 alignments, etc.

After the MSA algorithms are completed, LMAP_S ensures all resulting MSA files have the same taxa order. This becomes a useful characteristic for comparability and a necessity, since in the following stages some of the algorithms require MSA files with this setting (e.g. *trimalc* from ARC Stage; **3.4**).

The alignments resulting from this Stage will appear identified as *[GENENAME]_[ALGORITHM_S2].fas.*

Few examples:
COX1_**MUSCLE**.fas ; COX1_**PRANK**.fas ; CYTB_**MUSCLE**.fas ; CYTB_**PRANK**.fas

NOTE 6 (MACSE "!" symbol): It is important to note that *MACSE* as a software that considers frameshifts and stop codons, may take to alignments with unexpected characters like "!" (see [18] for more information). This presence in any number does not allow the correct functioning with software like *TRIMAL*, *WEAVEALIGN*, *MERGEALIGN* and *MAXALIGN*. Hence, the researcher may need to be careful when MACSE option(s) are selected. On the other hand, this conflict can be useful to detect such sequences or gene alignments. See also Note 14.

3.3. Stage 3 – MSA Outlier Detection (AOD)

This Stage has a different purpose. It is oriented to help users detect sequences that are unfit to be present in the MSA.

Two programs *OD-Seq* [27], and *EvalMSA* [8], are automatically executed by appending option -b.

Examples:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a cw,mu,pk -b
```

Here the outlier detection will be performed to clustalw (cw), muscle (mu) and prank (pk) MSAs.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b
```

Here the outlier detection will be performed to all MSAs.

The outcome for this Stage consists only in a report created in the LMAP_SREPORTS project directory identified by the name OUTLIER_REPORT.csv.

The report shows the results for each software, namely the outliers detected for each MSA case side-by-side, thus giving a corroborative view for each MSA estimated in the previous Stage.

These results do not interfere with following Stages, instead they are only meant to provide the information necessary to take better decisions regarding his/her dataset. In this sense and in an ideal scenario, the researcher will perform two executions of LMAP_S, one to get information about the possible outliers and a second execution to use this information towards his/her requirements. The first run to include this stage, and a second, to include any stages (except this) and account for such modifications. These modifications are the result of identifying MSF sequences to be removed with option -p ri[file.csv] (see Note 5; section **3.1**). This file (Figure 3) will have to be created by the user following the rules in Table 8.

Despite this ideal scenario, the researcher has the flexibility run any LMAP_S workflow construct, for instance, to include all stages. I.e., the researcher could perform an initial dataset study, to first understand how it behaves with all the required algorithms and reserve following runs to perfectioning/curating the dataset with information previously obtained from this stage/previous run.

3.4. Stage 4 – MSA Refinement and Consensus (ARC)

ARC Stage, directly follows from the AE Stage in terms of algorithms application. This Stage is focused in the refinement of MSAs and additionally in finding the consensus MSA among the many available.

Here the integrated MSA algorithms amount to 16 options (max. 19 output MSAs), respecting to 8 software (Table 10). Each TCS algorithm produces an extra MSA hence, the extra 3 MSAs forming a total of 19 (see below).

Table 10. *Stage 4. ARC – LMAP_S MSA refinement and consensus algorithms.*

Software	ALGORITHM_S4	LMAP_S code (option -c)	Description
ALL	*ALL*	all	Select all algorithms at once (Single code usage)
Gblocks	GBLOCKS	gb	Gblocks DNA option
	GBLOCKSC	gbc	Gblocks Codon option
MaxAlign	MAXALIGN	mx	MaxAlign default
Noisy	NOISY	ny	Noisy default

PSAR-Align	PSARALIGN	ps	PSAR-Align default
TCS (T-COFFEE)	TCS	tcs	TCS <default> (lib generation by probcons pair-HMM – proba_pair)
	TCSFM	tfm	TCSfm ENSEMBL COMPARA (lib generation by mafft+muscle+kalign)
	TCSOG	tog	TCSog ORIGINAL T-COFFEE (lib generation by clustalw+lalign)
TrimAl	TRIMAL	tl	TrimAl default
	TRIMALA	ta	TrimAl AUTOMATED1
	TRIMALG	tg	TrimAl GAPPYOUT
	TRIMALP	tp	TrimAl STRICTPLUS
	TRIMALS	ts	TrimAl STRICT
	TRIMALC	tt	TrimAl COMPARESET
MergeAlign	MERGEALIGN	mg	MergeAlign default
WeaveAlign	WEAVEALIGN	wa	WeaveAlign default

Rows painted in blue refer to the “consensus” algorithms and require more than one MSA algorithm to be selected for comparison. In fact *TRIMALC*, selects the best MSA among all the existing possibilities from AE Stage, hence it can be regarded as a different type of consensus. For this specific case of *TrimAl* a small report is produced in LMAP_SREPORTS identified by TRIMALCMPSET_REPORT.csv, allowing the user to quickly understand, which the selected cases were for each gene and the respective consistency scores. This comes from the fact that the output file created by LMAP_S to this algorithm is identified by [*GENENAME*]_ALL_TRIMALC.fas.

The alignments resulting from this Stage, appear identified with one additional portion relative to the MSA from AE Stage: *[GENENAME]_[ALGORITHM_S2]_[ALGORITHM_S4].fas*. In case of the consensus algorithms it is of the form *[GENENAME]_ALL_[ALGORITHM_S4].fas*

All *TCS* versions here indicated produce two MSA files instead of one; hence, six files are expected from all three cases. The reason lies in the output indicated in *TCS* command-lines: *sp_ascii,score_ascii,score_pdf,tcs_column_filter2,tcs_weighted*. This indicates that two modified MSAs (*tcs_column_filter2,tcs_weighted*) are produced. An extra character ‘w’ for the weighted version and ‘f’ for the filtered version specifically identifies these cases. Note that the value ‘2’ can be modified (see section **2.3**) to better reflect the sites that are maintained.

Option -c enables the selection of any of the algorithms, or with just ‘all’ select all at once.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a mu,pk -b -c mg,ny,ps,tt
```

Here, the mergealign (mg), trimalc (tt), noisy (ny) and pasaralign (ps) will be applied to all

MSAs from muscle (mu) and prank (pk).

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b -c all
```

Here, all the ARC Stage algorithms will be applied to all AE Stage algorithms data.

Few examples:
COX1_MUSCLE_**NOISY**.fas ; COX1_MUSCLE_**PSARALIGN**.fas ; COX1_**ALL**_**TRIMALC**.fas ; COX1_**ALL**_**MERGEALIGN**.fas ; CYTB_PRANK_**NOISY**.fas ; CYTB_PRANK_**NOISY**.fas ; CYTB_**ALL**_**TRIMALC**.fas ; CYTB_**ALL**_**MERGEALIGN**.fas
(hidden COX1_PRANK.fas and CYTB_MUSCLE.fas to avoid redundancy)

Before the user proceeds, LMAP_S provides another important selector through option -m (Table 11). It enables the user to select which MSAs will be employed for the following Stages.

Table 11. *LMAP_S MSA groups selector.*

LMAP_S code (option -m)	Observations
a	Use all MSAs available (Stages 2 and 4)
i	Use only MSAs from ARC Stage [**default, with -c used**]
m	Use only MSAs from AE Stage [**default, with -c not used**]

Assuming the user requires results from AE and ARC Stages, but only phylogeny estimations from AE Stage, this option is the solution.

3.5. Stage 5 – Phylogeny Estimation (PE)

In this Stage, the previously selected MSAs (Table 11) will be subject to input to the following selected algorithms and estimate the corresponding phylogenies. LMAP_S currently provides two Maximum Likelihood (ML) software, *IQ-TREE* and *SMS*, a Neighbor-Joining (NJ) software, *Ninja*, and a Maximum Parsimony (MP) software, *MPBoot*.

Here the integrated PT algorithms amount to 22 options, respecting to 6 software (Table 12).

Table 12. *Stage 5. PE – LMAP_S Phylogeny estimation algorithms.*

Software	ALGORITHM_S5	LMAP_S code (option -t)	Description
ALL	*ALL*	all[NBOOTS]	Select all algorithms at once. Single code usage.
IQ-TREE	NIQTREE	nit	IQ-TREE **DNA** TEST

	NSBIQTREE	nsit[NBOOTS]	IQ-TREE **DNA** TEST **STDBOOT**
	NUBIQTREE	nuit[NBOOTS]	IQ-TREE **DNA** TEST **UFBOOT**
	DIQTREE	dit	IQ-TREE DNA(**DEG**) TEST
	DSBIQTREE	dsit[NBOOTS]	IQ-TREE DNA(**DEG**) TEST **STDBOOT**
	DUBIQTREE	duit[NBOOTS]	IQ-TREE DNA(**DEG**) TEST **UFBOOT**
	RIQTREE	rit	IQ-TREE DNA(**RY**) TEST
	RSBIQTREE	rsit[NBOOTS]	IQ-TREE DNA(**RY**) TEST **STDBOOT**
	RUBIQTREE	ruit[NBOOTS]	IQ-TREE DNA(**RY**) TEST **UFBOOT**
	CIQTREE	cit	IQ-TREE **CODON** TEST
	CSBIQTREE	csit[NBOOTS]	IQ-TREE **CODON** TEST **STDBOOT**
	CUBIQTREE	cuit[NBOOTS]	IQ-TREE **CODON** TEST **UFBOOT**
	TIQTREE	tit	IQ-TREE **NT2AA** TEST
	TSBIQTREE	tsit[NBOOTS]	IQ-TREE **NT2AA** TEST **STDBOOT**
	TUBIQTREE	tuit[NBOOTS]	IQ-TREE **NT2AA** TEST **UFBOOT**
SMS (PhyML [40])	SMSAN	san[NBOOTS]	SMS AIC + NNI
	SMSAS	sas[NBOOTS]	SMS AIC + SPR
	SMSBN	sbn[NBOOTS]	SMS BIC + NNI
	SMSBS	sbs[NBOOTS]	SMS BIC + SPR
Ninja	NINJA	nj	Ninja default
MPBoot	NMPBOOT	nmp	MPBoot **DNA** default
	NUBMPBOOT	nump[NBOOTS]	MPBoot **DNA** "**UFBOOT**"
Degen	**DEG**	n.a.	Degen MSA coding
RYcode	**RY**	n.a.	RY MSA coding

n.a. – not applicable.

Where available [NBOOTS] indicates the possibility to provide the required number of bootstraps, e.g. -t all500 or -t tsit400. **UFBOOT** required minimum is 1000, **STDBOOT** recommended minimum is 100, **SMS** may run with 0 bootstraps (LMAP_S ensures 100 minimum). Following these rules, the 'all' code applies the given number where fit and preventing the number of bootstraps to be inferior to those above indicated.

From this table it is visible that LMAP_S *IQ-TREE* algorithms allow several character-coding (CC) options, and these are a requirement for the next Stage. These 5 CC options include **DNA**, **DEG** (Degeneracy coding), **RY** (puRine/pYrimidine coding), **CDN** (codon coding), and **2AA** (Amino acid coding - translated).

NOTE 7 (MSA format conversion): *SMS* as a software that integrates *PhyML* [40] software, requires the MSAs in PHYLIP format. LMAP_S automatically converts MSA formats from FASTA to PHYLIP and vice-versa. This is also the case of *TCS* (from ARC

Stage) that generates PHYLIP MSAs, which are converted to FASTA to maintain proper functioning and coherence.

When selected, the *Degen* and *RYcode* applications presented in blue, are executed by LMAP_S beforehand to prepare the coding of the MSA files to be served to *IQ-TREE* executions, respectively for D(SB/UB)IQTREE and R(SB/UB)IQTREE algorithms.

The resulting phylogenetic trees are identified by: *[GENENAME]_[ALGORITHM_S2]_[ALGORITHM_S4]_[ALGORITHM_S5].nwk* or; *[GENENAME]_[ALGORITHM_S2]_[ALGORITHM_S5].nwk*

Option -t enables the selection of any of the algorithms, or with just 'all' select all at once.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a mu,pk -b -c mg,ny,ps,tt -t nit,rit,san
```

Here the riqtree (rit), niqtree (nit) and smsan (san) will be applied to all MSAs from mergealign (mg), noisy (ny), psaralign (ps) and trimalc (tt).

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b -c all -t all
```

Here all the S5 algorithms will be applied to all S4 algorithms MSAs. With the -m m option it would apply to S2 algorithms instead (Table 9).

Few examples:
COX1_MUSCLE_NOISY_**RYt**_**RIQTREE**.nwk ;
COX1_MUSCLE_PSARALIGN_**RYt**_**RIQTREE**.nwk ; COX1_ALL_TRIMALC_**RYt**_ **RIQTREE**.nwk ; COX1_ALL_MERGEALIGN_**RYt**_**RIQTREE**.nwk ;
CYTB_PRANK_NOISY_**RYt**_ **RIQTREE**.nwk ; CYTB_PRANK_NOISY_**RYt**_ **RIQTREE**.nwk ; CYTB_ALL_TRIMALC_**RYt**_ **RIQTREE**.nwk ;
CYTB_ALL_MERGEALIGN_**RYt**_ **RIQTREE**.nwk
COX1_MUSCLE_NOISY_**NIQTREE**.nwk ; COX1_MUSCLE_PSARALIGN_**NIQTREE**.nwk ; COX1_ALL_TRIMALC_**NIQTREE**.nwk ; COX1_ALL_MERGEALIGN_**NIQTREE**.nwk ;
CYTB_PRANK_NOISY_**NIQTREE**.nwk ; CYTB_PRANK_NOISY_**NIQTREE**.nwk ;
CYTB_ALL_TRIMALC_**NIQTREE**.nwk ; CYTB_ALL_MERGEALIGN_**NIQTREE**.nwk
COX1_MUSCLE_NOISY_**SMSAN**.nwk ; COX1_MUSCLE_PSARALIGN_**SMSAN**.nwk ;
COX1_ALL_TRIMALC_**SMSAN**.nwk ; COX1_ALL_MERGEALIGN_**SMSAN**.nwk ;
CYTB_PRANK_NOISY_**SMSAN**.nwk ; CYTB_PRANK_NOISY_**SMSAN**.nwk ;
CYTB_ALL_TRIMALC_**SMSAN**.nwk ; CYTB_ALL_MERGEALIGN_**SMSAN**.nwk
(hidden COX1_PRANK.fas and CYTB_MUSCLE.fas to avoid redundancy)

As seen from examples above, it is possible to note that RY (and DEG likewise) cases add another component to the file identification, _**RYt**_ (and _**DEG**_).

A special attention is required to the '**t**' following "RY", which allows the user to understand that the RY coding procedure was done for the third codon positions of the MSA. Other types of RY-coding will have their respective indication letter to appear in replacement of the '**t**'. In fact, *RYcode* program is able to perform RY-coding for other positions or combinations (see section **4.**). This behavior can be easily modified in LMAP_S configuration file (see section **2.3**).

3.6. Stage 6 – Phylogeny Comparison and Consensus (PCC)

In this Stage, the phylogenetic trees are compared both statistically and topologically, respectively using the software, *CONSEL* [4] and *TreeCmp* [37], with option -s. Its purpose is to establish a common ground from where the researcher can take an informed decision about which may be the best phylogenetic tree as well as the best chain of algorithms applied since AE Stage (to which we refer hereafter as strategy). The method hereafter described is illustrated in Figure 4.

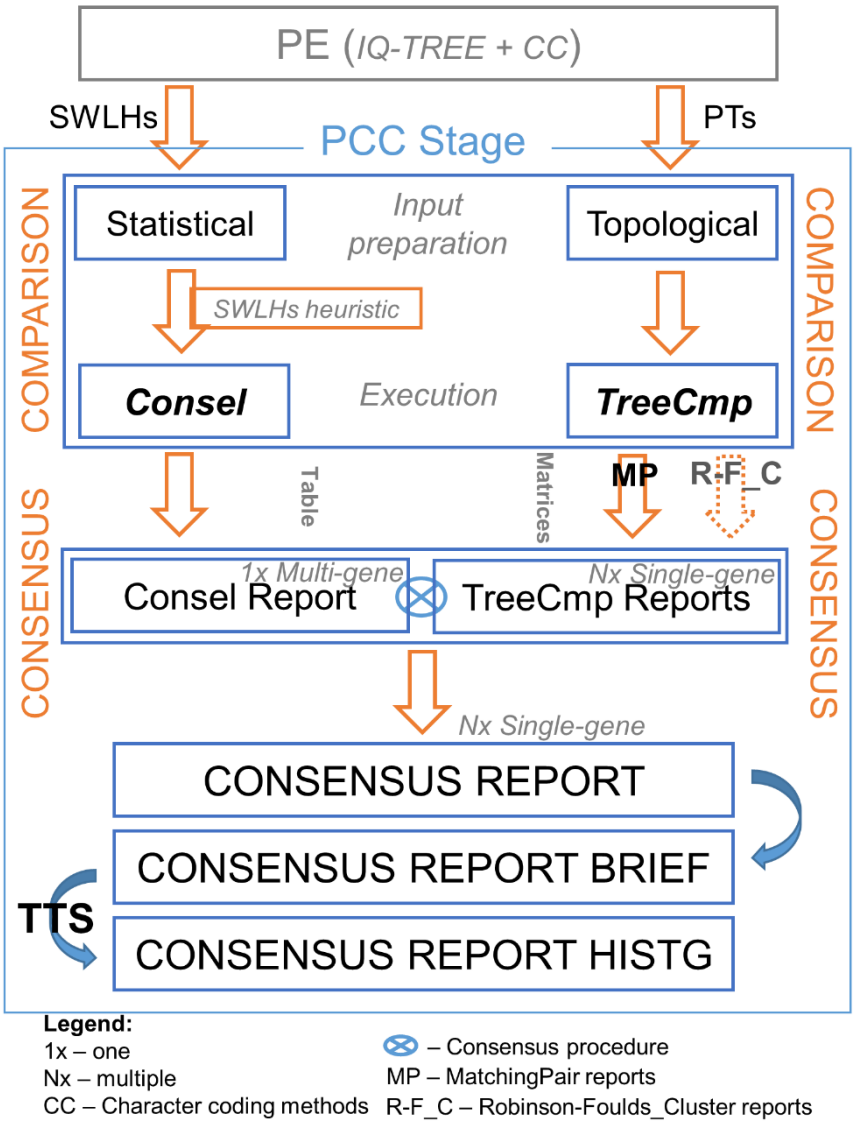


***Figure 4**. Flowchart describing the PCC method.*

Hence, statistical comparison is provided from the site-wise log-likelihoods (SWLH) that result from the execution of the selected *IQ-TREE* algorithms. To this end and for each gene, all SWLH are joined together in the same *CONSEL* input file by following the grouping of the same CC methods. For instance, if ARC Stage is included, all alignments (AE Stage) processed by the ARC Stage algorithm will have their SWLHs joined. Hence, *CONSEL* input files will be of the form [*GENENAME*]_[*ALGORITHM_S4*]_i_All[*CC*].sitelh. In case

ARC Stage, is left out of the workflow or option -m was set to 'm' (Table 11), then the input files identification becomes of the form [*GENENAME*]_i_All[*CC*].sitelh and all SWLHs (related to AE Stage MSAs) with the same CC are compared. This procedure ensures that the same gene SWLH are compared among the same CC options and possibly within the same S4 algorithms.

Since some SWLH files may have different site-wise lengths, LMAP_S ensures that *CONSEL* functions correctly (i.e., all concatenated strategies have same length) by selecting the SWLH files according to *(i)* same gene *(ii)* the same CC method and if ARC Stage was executed, *(iii)* to the same refinement/consensus algorithm. Even though, there may be SWLH files that result with different lengths due to e.g., different gap insertions. Hence, to maximize the usefulness of our procedure we have implemented an intermediate heuristic. It consists in *(i)* distributing the SWLH files by their site-wise lengths in bins created for every novel site-wise length; and by the end, *(ii)* select the bin with the maximum number of items to be concatenated, which respect a single specific length. For instance, while analyzing the SWLH strategies, one may have an item with length of 348 sites, thus a new bin for this length will be created and its identity inserted. If a different length appears with the next item, then a novel bin is created with its identity inserted, otherwise it will be inserted to the existing bin. This is repeated for every case. By the time all are processed, a bin with the maximum amount of strategies is selected. Finally, all will be concatenated and posteriorly compared with *CONSEL* package.

NOTE 8 (CONSEL SWLH items lengths): The alignments estimated may have different lengths due to gap insertions and/or other factors. This scenario poses a problem in *CONSEL*, which requires that all the SWLH data arrays for the input file should have the same length. LMAP_S tries to solve these cases by collecting the maximum SWLH items possible for each length, and discard the cases with different lengths in inferior number(s).

A similar strategy is followed for *TreeCmp* input files, where all the resulting ML topologies are joined together in input files identified by [*GENENAME*]_All.nwk. The result from *TreeCmp* is identified by [*GENENAME*]_All_TREECMP.nwk.txt

NOTE 9 (Phylogeny comparison data requirements): It is important to note that *CONSEL* and *TreeCmp* requires 2 MSAs at minimum from previous stages and 2 *IQ-TREE* algorithms, one from each CC.

From these programs, six reports are produced in LMAP_SREPORTS, one from *CONSEL* and two from *TreeCmp*. The remaining three reports were devised to combine both methodologies results and to present a condensed view of its results.

The *CONSEL* report identified by ICONSEL_REPORT.csv, summarizes the best selected cases for every gene (the ones in the first rank and other ranks, if their values equal the top rank). This ranking, is achieved by the selected column sort (option -s) given to *catpv* program (it can be modified to another column in LMAP_S software configuration file (see section **2.3**) – please see *CONSEL* guide http://stat.sys.i.kyoto-u.ac.jp/prog/consel/quick.html). Additionally, the report shows the original strategies clear identification that were selected.
The *TreeCmp* reports identified by [*GENENAME*]_TREECMP_RC_REPORT.csv and [*GENENAME*]_TREECMP_MP_REPORT.csv show the topology comparisons for the several topologies obtained from *IQ-TREE* algorithms. The difference lies in the method, which are

by default used: in first case, comparisons are made following Robinson-Foulds_Cluster (RF_C) method and the second following MatchingPair (MP), both under a matrix comparison scheme where all topologies are compared to each other without repetitions. Although two reports are produced, only the MP is used by default.

The last three reports, identified by *(i)* [*GENENAME*]_CONSENSUS_PHYLOGENY_REPORT.csv, *(ii)* [*GENENAME*]_CONSENSUS_PHYLOGENY_**BRIEF**_REPORT.csv and *(iii)* [*GENENAME*]_CONSENSUS_PHYLOGENY_**HISTG**_REPORT.csv have the purpose to combine both the statistical and topological methodologies. The first contains the base information from the *TreeCmp* MP report, on top of which the best candidates from *CONSEL* report are searched and marked. To this end two searches are performed: first, a search is performed to place *CONSEL* marks and a second search to place the *TreeCmp* marks.
In the first, it searches the column heading, if it finds a match with a *CONSEL* best candidate; it becomes marked with surrounding '*#*' (*CONSEL* mark) and the same strategy is marked in the transposed position. In the second search, the column heading is explored and if finds the *CONSEL* mark, it searches this corresponding column top-down to find all zero cells (which means, best topological score), if found, then the column header becomes marked with surrounding "***" (*TreeCmp* mark). For all the zero cells in this column, the respective row headers are also marked in the same manner. After searching the matrix column heading, it will apply the same procedure to the row heading after transposition of the matrix. During this process, it is ensured that only one mark of each type is applied to one strategy.
In conclusion, the best result would be the phylogenetic case that has agreement from both phylogeny comparison methodologies, i.e., zero topological score and rank 1 from *CONSEL*, which is found marked as "*** # **strategy** # ***".
The second report (BRIEF) follows from the first, by simply resuming the information to only rows and columns marked with either or both mark types.
The third report (HISTG) functions like an histogram and follows from the BRIEF report by presenting a sorted count of topology agreements, i.e., the number of zero cells that were found for each strategy in its column, thus showing further support to the given strategy. This report shows strategies sorted in two ways: first by their zero cell counts – i.e. Total Topological Support (TTS) – and secondly, by alphabetical order for groups of strategies that have the same TTS (e.g.: A, B, C, D all with count 24), hence the strategy line position should not be considered as criteria for supporting the strategy itself. This last sorting is only implemented to enable easier comparisons (inspected by the user) among identical LMAP_S executions and reports. In fact, where the same TTS is found, all strategies have equal support, thus any selection from such group is equally good.
When available, the consensus MSAs from ARC Stage, are also incorporated in the topologies comparison, but not in the statistical comparisons of CONSEL since they are unique cases for each gene and CC method. Hence, the consensus reports may show these cases and their TTS (HISTG) if any.

NOTE 10 (TreeCmp missing reports): It may be possible that *TreeCmp* terminates with an error if it finds the trees contain "different leaf sets". This scenario might be the reason for the eventually missing *TreeCmp* reports. If this happens, *TreeCmp* recommends adding its -P (prune) option to its command-line, hence the user might want to edit the LMAP_S software configuration file (see section **2.3**).

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a cw,mu,pk -b -c mg,ny,ps,tt -t nit,rit,san -s
```

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b -c all -t all -s
```

Here, all the phylogenetic trees previously generated and their SWLHs scores will be joined in single files per gene and CC for their comparison and report generation.

NOTE 11 (SWLH and topology differences): It is known that different runs from *IQ-TREE* may result in different topologies and mainly in different SWLH data with slight differences at a few decimal places. This behavior may lead to different results in the LMAP_S reports, namely in the consensus reports. Hence, we recommend the user to repeat specific *IQ-TREE* cases or whole LMAP_S command-line to ensure that the consensus reports results are stable and thus confer to them even more confidence.

These topological and statistical differences may cause some strategies to disappear from one identical execution to another, which only demonstrates that such strategy might suffer from data problems and requires user attention.

If necessary, *IQ-TREE* provides the "-seed [number]" option that enables a deterministic algorithm operation and thus always ensure the results are same. However, it is advised that it may cause such different results to pass undetected and thus the user may have no chance to verify if the reason could stem from the data or other sources. The user may add this option in LMAP_S configuration file (see section **2.3**).

By our experience, the best supported strategy(ies) at the top are not expected to vary from one LMAP_S execution to another (eventually only the overall TTS itself), and the user may select the optimal strategies without proceeding like explained in NOTE 11, unless more confidence is still required for any reason.

While selecting the best strategy from several LMAP_S replicates, what is important is to verify that the strategies persist in both/all reports and their relative positions are maintained (maintained through their TTS; regardless if they are different from one report to another). This means, there may be a noticeable difference from report A to report B in TTSs which is visible in all strategies TTSs, e.g. a difference of one may be found from A to B and still the strategies ranking is maintained.

NOTE 12 (Choosing the best strategy): For more confidence, we recommend that the criteria for selecting a best strategy is to *(i)* perform at least two identical executions of LMAP_S and *(ii)* to compare both HISTG reports. Then select the strategy that is having higher TTS at the top. If there are several strategies with the same TTS, select one that is **common to both reports**. If they have the same order but different TTS in both (meaning that due to reasons above the counts may vary from one report to the other, but with the same overall ranking), select the common and with common maximum TTS.

3.7. Stage 7 – Phylogeny Post-processing (PDP)

The last Stage in LMAP_S has the purpose to provide small, but useful modifications to all the estimated phylogenies and prepare them for further downstream analyses.

Table 13. *Stage 7. PDP – LMAP_S Post-processing operations.*

LMAP_S code (option -q)	Description	Observations
all	Select all arguments at once	Single code usage
rbs	Remove Branch Lengths	n.a.
rbl	Remove Bootstraps	n.a.
wur	Unroot tree	Only for rooted

Examples:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a cw,mu,pk -b -c mg,ny,ps,tt -t nit,rit,san -s -q rbs,rbl
```

Here all the phylogenies will be modified to have their bootstraps and branch lengths removed.

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b -c all -t all -s -q all
```

Here all the phylogenies will be modified to have their bootstraps, branch lengths removed and tree unrooted (LMAP_S only makes this modification if it detects the tree is initially rooted).

NOTE 13 (File identification): This additional file identification allows the researcher to understand which modifications or algorithms are covered in each file. At the end of LMAP_S execution, the MSAs and PTs can be easily found in LMAP_SFINAL directory (see secction **2.2**). The files resulting from PDP Stage are placed in a sub-directory named 'EDTREES'. Thus, this identification enables the user to quickly find and select the ones that might be necessary for his/her downstream analyses.

The resulting phylogenetic trees are identified by: *[GENENAME]_[ALGORITHM_S2]_[ALGORITHM_S4]_[ALGORITHM_S5]_[ARGUMENTS_S7].nwk* or; *[GENENAME]_[ALGORITHM_S2]_[ALGORITHM_S5]_[ARGUMENTS_S7].nwk*

Few examples:
(case 1)
COX1_MUSCLE_NOISY_RYt_RIQTREE_**RBSRBL**.nwk ;
COX1_MUSCLE_PSARALIGN_RYt_RIQTREE_**RBSRBL**.nwk ;
COX1_ALL_TRIMALC_RYt_ RIQTREE_**RBSRBL**.nwk ;
COX1_ALL_MERGEALIGN_RYt_RIQTREE_**RBSRBL**.nwk ;
CYTB_PRANK_NOISY_RYt_ RIQTREE_**RBSRBL**.nwk ; CYTB_PRANK_NOISY_RYt_ RIQTREE_**RBSRBL**.nwk ; CYTB_ALL_TRIMALC_RYt_ RIQTREE_**RBSRBL**.nwk ;
CYTB_ALL_MERGEALIGN_RYt_ RIQTREE_**RBSRBL**.nwk ;

(case 2)
COX1_MUSCLE_NOISY_NIQTREE_**RBSRBLWUR**.nwk ;
COX1_MUSCLE_PSARALIGN_NIQTREE_**RBSRBLWUR**.nwk ;
COX1_ALL_TRIMALC_NIQTREE_**RBSRBLWUR**.nwk ;
COX1_ALL_MERGEALIGN_NIQTREE_**RBSRBLWUR**.nwk ;
CYTB_PRANK_NOISY_NIQTREE_**RBSRBLWUR**.nwk ;
CYTB_PRANK_NOISY_NIQTREE_**RBSRBLWUR**.nwk ;
CYTB_ALL_TRIMALC_NIQTREE_**RBSRBLWUR**.nwk ;
CYTB_ALL_MERGEALIGN_NIQTREE_**RBSRBLWUR**.nwk
(hidden COX1_PRANK.fas and CYTB_MUSCLE.fas to avoid redundancy)

3.8. File Count Metrics

When LMAP_S terminates its execution, a few metrics are given to help the user understand how many files/results were generated and mainly if there are some executions missing (due to possible errors).

LMAP_S essentially presents the number of existing and valid MSA and PT files that were copied to the LMAP_SFINAL directory respective locations. Additional messages can be present if the number of expected MSAs and/or PTs is different from what was produced and if there were empty files (which are not copied).

NOTE 14 (Valid vs. Expected MSAs/PTs): At the end of LMAP_S execution the researcher may frequently encounter a different number of valid MSAs compared to that of expected MSAs (and/or same for PTs). We have found this to be a normal outcome whereby the successive application of certain algorithms may take to a non-valid chain of algorithms (e.g., loss of data – entire/partial sequences or nucleotide sites, arising intermediate stop codons, changes in the number of codons – "multiple of 3" errors, or changes to reading frames). We have performed significant testing and debugging to ensure that all data flow and algorithms are applied in correct order and with correct settings. Nevertheless, errors may still be found that may justify LMAP_S additional bug fixes. In this case, if the researcher/user detects anything relevant, please send us all the information possible, so that we can quickly clear any doubt or difficulty. See also **3.10** and Note 6.

The expected number of MSAs is calculated with the following expressions depending on the selected cases. This includes the option -m (Table 11; section **3.4**) which controls the groups of MSAs that are taken to the next stages.

NG = Number of gene MSFs;

EM2 = Expected MSAs from Stage 2 (-m 'm');
EM4 = Expected MSAs from Stage 4 (-m 'i');
EM24 = Expected MSAs from Stages 2 + 4 (-m 'a');
N2 = Number of selected MSA algorithms Stage 2;
NR4 = Number of selected MSA refinement algorithms Stage 4;
NC4 = Number of selected MSA consensus algorithms Stage 4;
NTCS = Number of TCS algorithms selected

ET2 = Expected PTs (-m 'm');

ET4 = Expected PTs (-m ‘i’);
ET24 = Expected PTs (-m ‘a’);
NR5 = Number of selected PT algorithms Stage 5;

ETT2 = Expected total PTs (-m ‘m’);
ETT4 = Expected total PTs (-m ‘i’);
ETT24 = Expected total PTs (-m ‘a’);

Stage **2**:

$$EM2 = NG \times N2$$

Stage **4**:

$$EM4 = (NG \times N2 \times (NR4 + NTCS)) + (NG \times NC4)$$

The NTCS here, reflects the additional MSA (‘w‘ and/or ‘f’ versions) produced by each TCS algorithm. Each one produces 2 MSAs each, hence if all three are selected there will be 6 in total (similarly, as if there were 3 more algorithms being selected).

Stage 2+**4**:

$$EM24 = EM2 + EM4 = (NG \times N2) + (NG \times N2 \times (NR4 + NTCS)) + (NG \times NC4)$$

Stage 2+**5**:

$$ET2 = NR5 \times EM2$$

Stage 4+**5**:

$$ET4 = NR5 \times EM4$$

Stage 2+4+**5**:

$$ET24 = NR5 \times EM24$$

Stage 2+5+**7**:

$$ETT2 = 2 \times ET2$$

Stage 4+5+**7**:

$$ETT4 = 2 \times ET4$$

Stage 2+4+5+**7**:

$$ETT24 = 2 \times ET24$$

For instance, in an all-to-all case scenario, with 12 genes, options by default, applying all possible algorithms (33 MSAs (S2), 16+3 MSAs (S4), 24 PTs (S5)) it is expected to get a total of EM4 = 6372 (6336+36) MSAs and ET4 = 152928 PTs or double with additional PDP Stage editing.

3.9. Email Notifications

After LMAP_S has been configured with the correct SMTP information (see Table 4 and section **1.2.1.2.**), the user is ready to receive notifications. Notifications are sent, as soon as, the *lmap-s.pl* application terminates the various batches of executions.
Email notification is enabled via option -e, as in:

```
$ lmap-s.pl -A Data/MSF/ -i 1 -d . -j MyDirectoryStruct -a all -b -c all -t all -s -q all -e
username@uni.fac.com
```

A default email address can be defined during LMAP_S configuration (Table 4), which will

be used whenever the option -e is employed without arguments. If this default address was not initially defined, then an address is required at all times as shown above.
This option can thus function in two ways: *(i)* send notification to the default address (-e); or *(ii)* to the address specified in front of this option (-e <address>). Thus the user has the versatility of two modes of functioning. In case no address was supplied in either case (in the command-line and during initial configuration), no notification will be sent and setting this option will have no effect.

3.10. Output Logging and Support

To help the user understand what possibly went wrong, LMAP_S provides three log cases: *(i)* through option -l, *(ii) screen* utility logging and *(iii)* $HOME logging.

The first case, enables the creation of a file that contains all algorithms executions (compilation of “Final Status” screen), with their ranking (first terminated), input file executed, and time used. Hence, the CSV log file header consists of “Algorithm,Rank,File,Time Used”. The “Time Used” column information should be taken carefully, since few integrated software depend on other programs and thus for these cases the time used may not correspond to the exact total time used. This file is located in same directory as the LMAP_S project folder. See also section **5**.

The second case is by default enabled with *screen* utility execution. This utility by default creates the “screenlog.0” file (in every *GENENAME* folder) in which the user may find the output produced by all software executed (for each *GENENAME*). These files may become very long.

The third case, is by default enabled by LMAP_S and is located at the user’s $HOME under the format “.lmaps[STARTDATE]Log.txt” (e.g.: .lmapsSat_Jun__2_21:53:20_2018Log.txt) . STARTDATE is the time and date that LMAP_S started executing, it becomes useful to help identify the corresponding LMAP_S execution. These files register the warnings, errors and other information produced by LMAP_S and are very helpful to detect the reasons for any malfunctioning. From time to time, these files might need to be manually removed to free up space.

LMAP_S does not use or send this information anywhere. The user is recommended to send this information to us (or part of it) in case any help or support is required with LMAP_S.

3.11. Help From Command-line

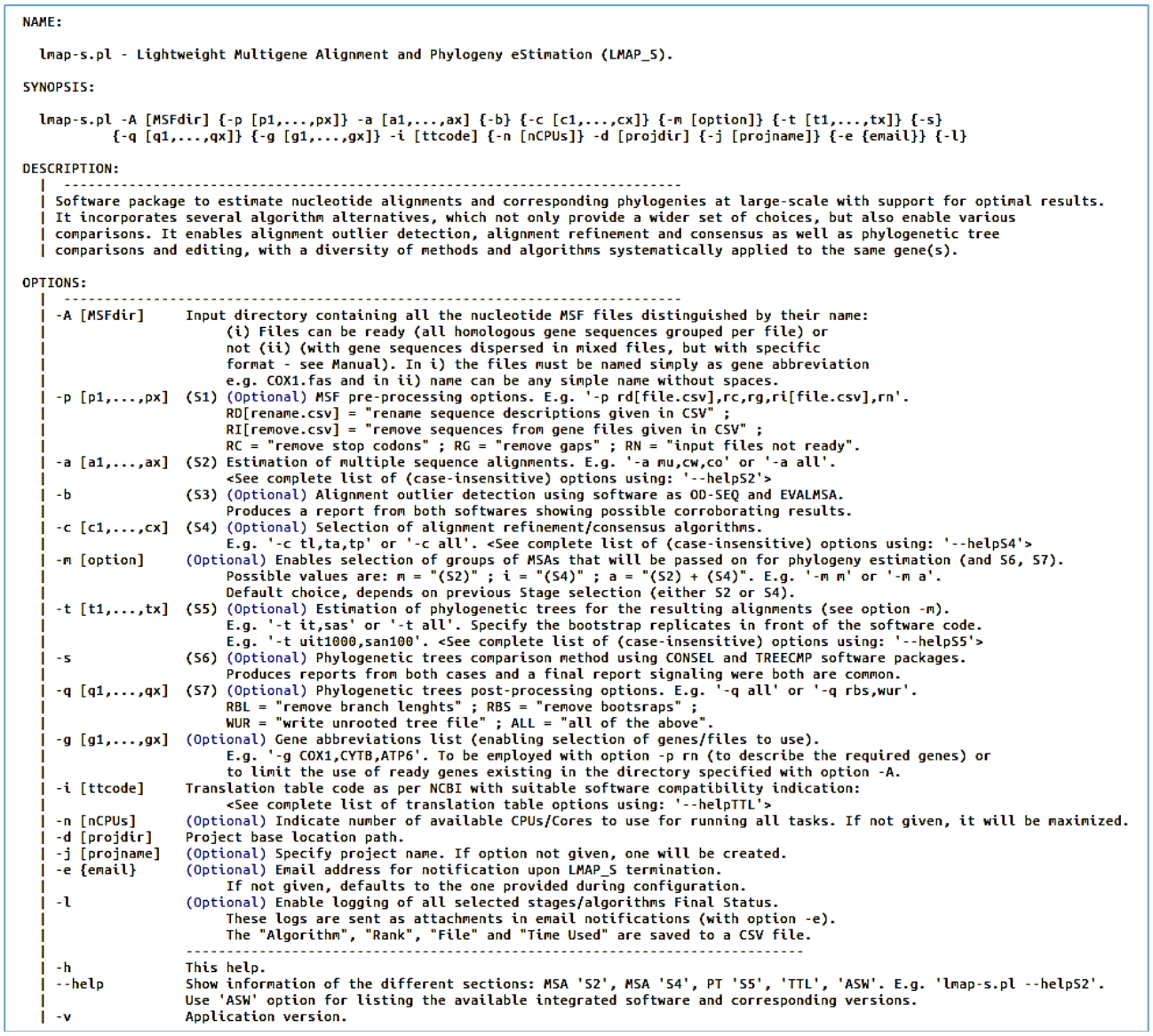
LMAP_S can present five specific quick help messages that can be invoked from the command-line to, e.g. help the user visualize and select the software to run. These messages show the software options for AE Stage (Table 9), ARC Stage (Table 10), PE Stage (Table 12), Genetic codes (Table 7) and available integrated software (Table 3).

To view the listing for each case, type:

```
$ lmap-s.pl -h
```

This will show all the available LMAP_S options and which are non-mandatory/optional

(Figure 5).



```
NAME:

  lmap-s.pl - Lightweight Multigene Alignment and Phylogeny eStimation (LMAP_S).

SYNOPSIS:

  lmap-s.pl -A [MSFdir] {-p [p1,...,px]} -a [a1,...,ax] {-b} {-c [c1,...,cx]} {-m [option]} {-t [t1,...,tx]} {-s}
            {-q [q1,...,qx]} {-g [g1,...,gx]} -i [ttcode] {-n [nCPUs]} -d [projdir] {-j [projname]} {-e {email}} {-l}

DESCRIPTION:
  | ----------------------------------------------------------------------------
  | Software package to estimate nucleotide alignments and corresponding phylogenies at large-scale with support for optimal results.
  | It incorporates several algorithm alternatives, which not only provide a wider set of choices, but also enable various
  | comparisons. It enables alignment outlier detection, alignment refinement and consensus as well as phylogenetic tree
  | comparisons and editing, with a diversity of methods and algorithms systematically applied to the same gene(s).

OPTIONS:
  | ----------------------------------------------------------------------------
  | -A [MSFdir]     Input directory containing all the nucleotide MSF files distinguished by their name:
  |                      (i) Files can be ready (all homologous gene sequences grouped per file) or
  |                      not (ii) (with gene sequences dispersed in mixed files, but with specific
  |                      format - see Manual). In i) the files must be named simply as gene abbreviation
  |                      e.g. COX1.fas and in ii) name can be any simple name without spaces.
  | -p [p1,...,px]  (S1) (Optional) MSF pre-processing options. E.g. '-p rd[file.csv],rc,rg,ri[file.csv],rn'.
  |                      RD[rename.csv] = "rename sequence descriptions given in CSV" ;
  |                      RI[remove.csv] = "remove sequences from gene files given in CSV" ;
  |                      RC = "remove stop codons" ; RG = "remove gaps" ; RN = "input files not ready".
  | -a [a1,...,ax]  (S2) Estimation of multiple sequence alignments. E.g. '-a mu,cw,co' or '-a all'.
  |                      <See complete list of (case-insensitive) options using: '--helpS2'>
  | -b              (S3) (Optional) Alignment outlier detection using software as OD-SEQ and EVALMSA.
  |                      Produces a report from both softwares showing possible corroborating results.
  | -c [c1,...,cx]  (S4) (Optional) Selection of alignment refinement/consensus algorithms.
  |                      E.g. '-c tl,ta,tp' or '-c all'. <See complete list of (case-insensitive) options using: '--helpS4'>
  | -m [option]     (Optional) Enables selection of groups of MSAs that will be passed on for phylogeny estimation (and S6, S7).
  |                      Possible values are: m = "(S2)" ; i = "(S4)" ; a = "(S2) + (S4)". E.g. '-m m' or '-m a'.
  |                      Default choice, depends on previous Stage selection (either S2 or S4).
  | -t [t1,...,tx]  (S5) (Optional) Estimation of phylogenetic trees for the resulting alignments (see option -m).
  |                      E.g. '-t it,sas' or '-t all'. Specify the bootstrap replicates in front of the software code.
  |                      E.g. '-t uit1000,san100'. <See complete list of (case-insensitive) options using: '--helpS5'>
  | -s              (S6) (Optional) Phylogenetic trees comparison method using CONSEL and TREECMP software packages.
  |                      Produces reports from both cases and a final report signaling were both are common.
  | -q [q1,...,qx]  (S7) (Optional) Phylogenetic trees post-processing options. E.g. '-q all' or '-q rbs,wur'.
  |                      RBL = "remove branch lenghts" ; RBS = "remove bootsraps" ;
  |                      WUR = "write unrooted tree file" ; ALL = "all of the above".
  | -g [g1,...,gx]  (Optional) Gene abbreviations list (enabling selection of genes/files to use).
  |                      E.g. '-g COX1,CYTB,ATP6'. To be employed with option -p rn (to describe the required genes) or
  |                      to limit the use of ready genes existing in the directory specified with option -A.
  | -i [ttcode]     Translation table code as per NCBI with suitable software compatibility indication:
  |                      <See complete list of translation table options using: '--helpTTL'>
  | -n [nCPUs]      (Optional) Indicate number of available CPUs/Cores to use for running all tasks. If not given, it will be maximized.
  | -d [projdir]    Project base location path.
  | -j [projname]   (Optional) Specify project name. If option not given, one will be created.
  | -e {email}      (Optional) Email address for notification upon LMAP_S termination.
  |                      If not given, defaults to the one provided during configuration.
  | -l              (Optional) Enable logging of all selected stages/algorithms Final Status.
  |                      These logs are sent as attachments in email notifications (with option -e).
  |                      The "Algorithm", "Rank", "File" and "Time Used" are saved to a CSV file.
  |                 ----------------------------------------------------------------------------
  | -h              This help.
  | --help          Show information of the different sections: MSA 'S2', MSA 'S4', PT 'S5', 'TTL', 'ASW'. E.g. 'lmap-s.pl --helpS2'.
  |                 Use 'ASW' option for listing the available integrated software and corresponding versions.
  | -v              Application version.
```

***Figure 5**. LMAP_S Help Menu.*

```
$ lmap-s.pl --helpS2
```

This will show the software codes for AE Stage (Table 9; 3.2).

```
$ lmap-s.pl --helpS4
```

This will show the software codes for ARC Stage (Table 10; 3.4).

```
$ lmap-s.pl --helpS5
```

This will show the software codes for PE Stage (Table 12; 3.5).

```
$ lmap-s.pl --helpTTL
```

This will show the list of genetic codes from where to select for option -i (Table 7; 3).

```
$ lmap-s.pl --helpASW
```

This will show the available software, their locations and respective versions. This last option will also show if any software is missing with a red tag “[N/F]” (not found) showing instead of the version number. Alternatively, a “[N/A]” (not available) may appear indicating that the software was found, but LMAP_S could not find the respective software version. This tag can also show during MMAP interface (Figure 6; **5.1**) execution with the same meaning. Integrated software versions may not be found, when the authors have not provided versioning to their applications directly from their help/version messages on screen.

3.11.1. Synopsis Section

The SYNOPSYS section in *lmap-s.pl* application help (displayed by using the command-line option -h), has the purpose to quickly elucidate the user on how to write the command for the application at hand by giving the correct format for each command-line option and argument (if any).

```
SYNOPSIS:

  lmap-s.pl -A [MSFdir] {-p [p1,...,px]} -a [a1,...,ax] {-b} {-c [c1,...,cx]} {-m [option]} {-t [t1,...,tx]} {-s}
          {-q [q1,...,qx]} {-g [g1,...,gx]} -i [ttcode] {-n [nCPUs]} -d [projdir] {-j [projname]} {-e {email}} {-l}
```

As shown above, the mandatory options are presented with the argument description enclosed in brackets (e.g.: “-a [a1,…,ax]”). Otherwise non-mandatory options are found enclosed in curly braces (e.g.: “{-b}”). Overall, curly braces have the meaning of “optional”, while the brackets, the meaning of mandatory. Generally, whenever an option is specified, an argument (value) is expected to follow, with the exception being the option -e.

4. APPLICATION *RYcode.pl* (version: 1.0.0 Mar 30th, 2018)

The *RYcode.pl* application was implemented to fill the gap of a missing solution for the problem of RY-coding. It can be used independently from LMAP_S and its modules.

RY-coding allows purines (A, G) to be coded (replaced) with 'R' and pyrimidines (C, T, U [RNA]) to be coded (replaced) with 'Y'. Five ways of RY-coding (Table 14) were implemented regarding the codon: to code the *(i)* first, *(ii)* second or *(iii)* third position, *(iv)* the first and second, and *(v)* all three.

Table 14. *RYcode.pl codon coding alternatives.*

Option	ARGUMENT	Description
	a	All three codon positions
	w	First and second positions
-p	t	Third position [**default**]
	s	Second position
	f	First position

RYcode.pl currently accepts both NEXUS and FASTA input formats. The output file if not given by user, is by default formed by appending _RY[ARGUMENT].[inputfileext] to the input file name.

Examples:

```
$ RYcode.pl -i InputMSAFile.fas -p t
```

This is the command-line specification integrated in LMAP_S. Although it would not be necessary to include -p t since it is default, it was included to help the user make modifications to this parameter in LMAP_S software configuration file (see section 2.3) and thus allow the *IQ-TREE* RY-coding algorithms to have different behaviours. The output file will be InputMSAFile_RYt.fas.

```
$ RYcode.pl -i InputMSAFile -p a -o MyRYcodedFile.fas
```

RY-code all codon positions and save to MyRYcodedFile.fas.

```
$ RYcode.pl -i InputMSAFile -p w
```

Here, the output file will be InputMSAFile_RYw.fas.

5. LMAP_S User Interaction - MODULE *MyMMAP.pm*

(**version: 1.0.0 Apr 24th, 2019**)

The *MyMMAP.pm* module was adapted and improved from the *LMAP* (*mmap.pl*) [41] application and has the purpose to execute and monitor all the available files in every Stage, by every software integrated in LMAP_S and existing in the directory structure.

To perform the execution of the several Stages (and respective software), the main procedure from *mmap.pl* application was transformed into a main method and invoked from *lmap-s.pl* application.

All the main logical *mmap.pl* functions were maintained, with slight improvements in threading and in graphical presentation. The code was adapted to accommodate the several integrated software differences, which is also accomplished with the help of *MyISWU.pm* module.

LMAP_S also provides the -n option (non-mandatory) to specify the maximum number of tasks/CPUs to use. In case this option is not provided, MyMMAP.pm will automatically estimate the maximum CPUs available and thus parallelize that number of tasks.

```
$ lmap-s.pl -A Data/MSF/ -d . -j MyDirectoryStruct -a all -b -c all -t all -s -q all -n 30
```

This command will have *lmap-s.pl* application to start executing all files, found in the whole directory structure recursively, given in the option -d. Furthermore, assuming that the workstation where LMAP_S is installed has at least a total of 30 CPU cores, this command will schedule a maximum number of 30 tasks (option -n) executing at same time (this also largely depends on the number of tasks available at each Stage).

The option -n is useful to control how many tasks will be running at a given time and to make possible the management of workstation capacity occupancy. The user can give a maximum number of 10 tasks and thus be able to execute two more LMAP_S instances with 10 tasks each, in the same workstation without any interference from the other instances. In fact, LMAP_S provides an ID (MMAPID or MPID) randomly generated. This MPID is composed of 2 consecutive capital letters and four digits as in “AZ0887”. This ID identifies all the *screen* instances (and hence any project instances) executing with respect to the *lmap-s.pl* application instance which created them.

In a case, where multiple LMAP_S instances might be executing in the same workstation, each instance will show the tasks of other instances. Hence, to understand which is the instance responsible for some of the tasks, it is enough to observe the top left corner MPID (Figures 6-7) and compare to the MPIDs of the listed tasks. Additionally, and assuming the use of colors in LMAP_S is configured from the start (see section **1.2.1.2**), the tasks belonging to the specific instance will appear colored, contrarily to others.

The option -l (lower case L) (see section **3.10**) enables the creation of a log file for reporting all tasks executions, which have successfully finished. Here, the information is placed in four related columns, the first indicates the software algorithm “*Algorithm*”; the second the task precedence numbering, “*Rank*”; the third indicates the data “*File*” absolute path used in execution; and the fourth, the “*Time used*”.

5.1. Monitoring of Executions and Available Screens

While LMAP_S is running, the user can monitor the executions through the MMAP interface, which enables the user to understand the progress of LMAP_S and of the dataset. This monitoring of tasks is performed in cases that appear as scheduled, as running and as finished, in two different screens. The default screen 1 (Figure 6), shows pairs of "SCREEN" and "PRGM" lines. The "SCREEN" lines identify the *screen* utility, which is executing the following associated "PRGM" line. For instance, *screen* 12567 is executing *T-COFFEE* 12769 (Figure 6 – first two lines/numbers). Thus, each pair corresponds to one task. The task can be identified by the session name shown in the "SCREEN" line. The session name contains the *screen* process number, followed by the MPID, which launched this task, the current software executing under SCREEN (PRGM), and followed by the path to the data file being handled. This location is represented with slashes converted to hyphens.

NOTE 15 (*Screen* utility versions and installation): *Screen* utility software was found to behave differently from older version 4.03.01 (16.04LTS) to current 4.06.02 (18.04LTS). It was detected that the newer version does not allow session names longer than 81 characters and it affected LMAP_S integrated software executions and results. Hence, to minimize this effect we provide the version 4.03.01 (binary) with LMAP_S package. During installation (see section **1.2.1**), it will ask the user his preference either to install this older one or to keep using the one currently installed. When installing the one provided, the current will be backed up to a different name in same folder ("screen.bck"). LMAP_S, namely MMAP interface, will present different behavior depending on the *screen* version being used. For the older one, the session names will be longer (as described above) and for the newer, it will be limited to the file name of the dataset being handled. This issue has been reported: http://savannah.gnu.org/bugs/index.php?54458.

```
RUN STATUS screen
====================
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        ::: Status of the 12 instances running:
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

SCREEN:         12567.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ATP6-ATP6.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12769  0.0  0.0  57336 43312 pts/2  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12569/404712624_208_TCtmp.pl
SCREEN:         12576.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ATP8-ATP8.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12773  0.0  0.0  58600 44940 pts/3  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12578/404712629_208_TCtmp.pl
SCREEN:         12584.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-COX1-COX1.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12815  0.0  0.0  67240 53312 pts/4  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12588/404712668_208_TCtmp.pl
SCREEN:         12595.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-COX2-COX2.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12818  0.0  0.0  57352 43224 pts/5  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12599/404712683_208_TCtmp.pl
SCREEN:         12605.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-COX3-COX3.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12820  0.0  0.0  58208 44532 pts/6  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12608/404712695_208_TCtmp.pl
SCREEN:         12613.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-CYTB-CYTB.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12840  0.0  0.0  58292 44352 pts/7  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12617/404712718_208_TCtmp.pl
SCREEN:         12627.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND1-ND1.fas   (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12852  0.0  0.0  59992 46132 pts/8  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12634/404712760_208_TCtmp.pl
SCREEN:         12646.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND2-ND2.fas   (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12854  0.0  0.0  60680 46604 pts/9  [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12652/404712758_208_TCtmp.pl
SCREEN:         12670.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND3-ND3.fas   (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12855  0.0  0.0  56504 42688 pts/10 [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12677/404712779_208_TCtmp.pl
SCREEN:         12687.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND4-ND4.fas   (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12866  0.0  0.0  64824 50744 pts/11 [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12697/404712802_208_TCtmp.pl
SCREEN:         12717.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND4L-ND4L.fas (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12861  0.0  0.0  54808 40896 pts/12 [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12727/404712813_208_TCtmp.pl
SCREEN:         12743.MPID:HT5070:t_coffeetc:NRresults-LMAP_SPool-ND5-ND5.fas   (25/10/2018 16:12:49)   (Detached)
   \_   PRGM:   12868  0.0  0.0  71164 57296 pts/13 [R: RUNNING]  16:12   0:00 /var/tmp/t_coffee.tmp/tmp.labpc10c.12750/404712841_208_TCtmp.pl

        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

                            Press s to [show]/hide other MPIDs.
                Press +/- to increase/decrease maximum number of instances running.
```

***Figure 6**. MMAP Screen 1 – Run Status screen.*

By pressing the keyboard key '2', the MMAP interface displays two lists in screen 2 (Figure 7). At the top, the next 10 (maximum) scheduled tasks are displayed ("Files running next"). These are tasks ready to enter execution, as soon as any task(s) from the screen 1 terminate(s). Below, the most recent 10 (maximum) finished tasks are displayed ("Files finished") together with their "Time Used" at the left. The terminated tasks from screen 1, appear in this list, with the most recent always showing at the top. Press keyboard key '1' to go back to screen 1 (Figure 6).

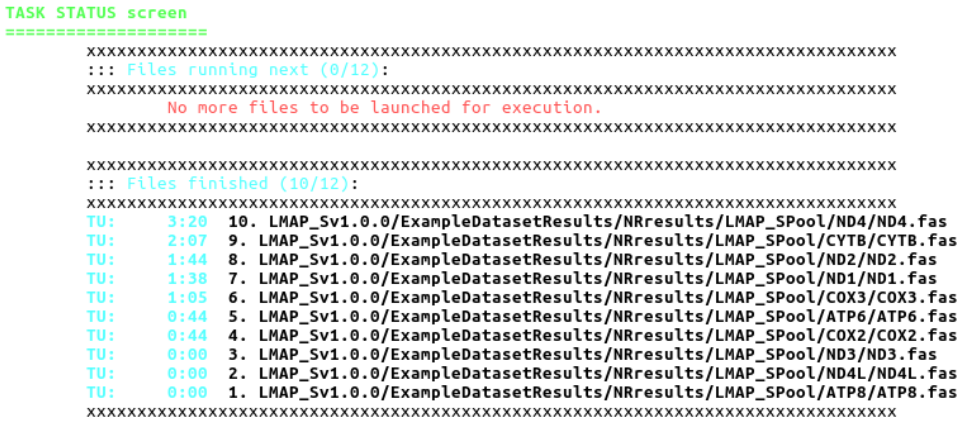


***Figure 7**. MMAP Screen 2 – Task Status screen.*

Beyond exhibiting all tasks, the screen 1 allows the user to understand if there is a not running task. This is displayed in the same column where the **"[R: RUNNING]**" tag is normally seen (Figure 6). Any other tag (Table 15) being exhibited in its place indicates that something could be wrong with the execution. The *screen* utility program generates additional log files (see section **3.10**), in each folder of the LMAP_SPOOL directory. Such logs contain the complete and joined output of all selected software executed individually for each gene target. They aid in inspecting the source of the problems causing the different behaviors, thus showing a different tag. In this case, the tag likely to occur more often would be the tag in the second row of Table 15. In this situation, the user may opt to terminate the task that is preventing another task to take its place. To this, the user may access the built-in Process Manager (PM) screen (Figure 9), by pressing "Ctrl-\" or "Ctrl-c" to exit. It will then be possible to enter this screen by replying "m" to the presented query. Please note that despite this continues to be valid, it has been verified that the several integrated software in LMAP_S have different requirements and behaviors, and thus the same or other tags may be shown alternatively for brief periods without harm.

NOTE 16 (Final Status screen and option -l): At the end of each algorithm execution, additional resume information is presented in a "Final Status" screen that shows time, rank and respective file location (Figure 8). This is only visible for the last algorithm when *lmap-s.pl* terminates and briefly (few seconds) for the remainder. To visualize and save the complete information for all algorithms the user must specify option -l (Table 6) in the command-line. This will create a log file (option -l) with all information compiled (see section **3.10**).

```
:z: All executions are finished successfully!

:z: A total of 12 / 12 files were started in specified program:

FINAL STATUS screen
====================
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        ::: Tasks completed:
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        TU:     2:54    12. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND5-ND5.fas
        TU:     2:08    11. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-COX1-COX1.fas
        TU:     1:41    10. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND4-ND4.fas
        TU:     1:06    9. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-CYTB-CYTB.fas
        TU:     0:58    8. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND2-ND2.fas
        TU:     0:50    7. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND1-ND1.fas
        TU:     0:32    6. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-COX3-COX3.fas
        TU:     0:24    5. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-COX2-COX2.fas
        TU:     0:24    4. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ATP6-ATP6.fas
        TU:     0:04    3. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND3-ND3.fas
        TU:     0:03    2. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ND4L-ND4L.fas
        TU:     0:00    1. MPID:DX1029:probalign:IQtest_1-LMAP_SPool-ATP8-ATP8.fas
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

***Figure 8**. MMAP – Final Status screen.*

***Table 15.** MMAP possible task status tags.*

Troubleshoot Tag	Observations
[R: RUNNING]	Execution OK.
[S: WAITING FOR USER INPUT]	Execution waiting for user input.
[T: STOPPED (BY SIGNAL)]	Execution stopped by external signal.
[D: UNINTERRUPTIBLE SLEEP (IO)]	Execution OK, but waiting for input/output access.
[Z: WRONGLY TERMINATED]	Execution terminated incorrectly.
[X: !DEAD!]	Should never be seen.

http://www.petefreitag.com/tools/man-pages/ps.html

NOTE 17 (Quitting MMAP/LMAP_S): After typing “Ctrl-\” or “Ctrl-c” to exit, this action may not be instantly triggered, and so it may require a moment to be presented the possibility to correctly terminate *lmap-s.pl* execution together with all *screen* instances. Three queries are presented to the user: the first, allows one to select the PM screen (Figure 9) or to quit; the second, confirms the quit decision and the third, requires the user to decide what action to take towards the running tasks. In this last case, three actions are possible, *(i)* to terminate only spawned/created instances (current MPID), *(ii)* to terminate every *screen* instance running (includes other *lmap-s.pl* instances **(!!!)** – all available MPIDs); or *(iii)* leave all running and just exit *lmap-s.pl*.

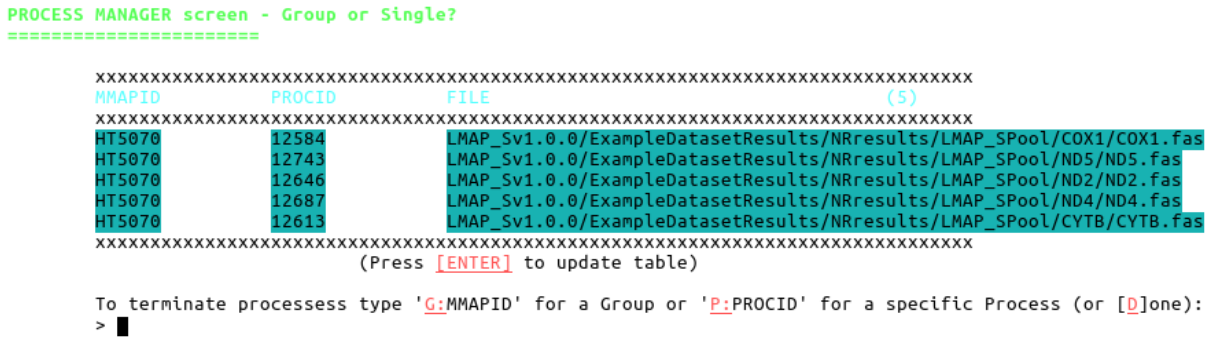


***Figure 9**. MMAP – Process Manager screen.*

The PM screen presents a table with three related columns (Figure 9), the “FILE” column shows all running tasks, and two columns identify two IDs: *(i)* the “MMAPID” column, which allows the user to terminate a group of tasks that have the same ID and (ii) the “PROCID” column, which allows one to terminate a single task. By identifying the problematic execution, the user may here issue the command “P:PROCID” or “G:MMAPID” with the corresponding ID to terminate it (e.g.: “P:12584” or “G:HT5070”). After all is done, the user may return to the main screen 1 or 2, following the *lmap-s.pl* queries.

10. REFERENCES

1. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C et al. The Bioperl toolkit: Perl modules for the life sciences. Genome research. 2002;12(10):1611-8. doi:10.1101/gr.361602.

2. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011;7:539. doi:10.1038/msb.2011.75.

3. Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research. 1994;22(22):4673-80. doi:10.1093/nar/22.22.4673.

4. Shimodaira H, Hasegawa M. CONSEL: for assessing the confidence of phylogenetic tree selection. Bioinformatics. 2001;17(12):1246-7. doi:10.1093/bioinformatics/17.12.1246.

5. Regier JC, Shultz JW, Zwick A, Hussey A, Ball B, Wetzer R et al. Arthropod relationships revealed by phylogenomic analysis of nuclear protein-coding sequences. Nature. 2010;463(7284):1079-83. doi:10.1038/nature08742.

6. Zwick A, Regier JC, Zwickl DJ. Resolving discrepancy between nucleotides and amino acids in deep-level arthropod phylogenomics: differentiating serine codons in 21-amino-acid models. PLoS One. 2012;7(11):e47450. doi:10.1371/journal.pone.0047450.

7. Subramanian AR, Kaufmann M, Morgenstern B. DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment. Algorithms Mol Biol. 2008;3:6. doi:10.1186/1748-7188-3-6.

8. Chiner-Oms A, Gonzalez-Candelas F. EvalMSA: A Program to Evaluate Multiple Sequence Alignments and Detect Outliers. Evol Bioinform Online. 2016;12:277-84. doi:10.4137/EBO.S40583.

9. Bradley RK, Roberts A, Smoot M, Juvekar S, Do J, Dewey C et al. Fast statistical alignment. PLoS Comput Biol. 2009;5(5):e1000392. doi:10.1371/journal.pcbi.1000392.

10. Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C et al. Versatile and open software for comparing large genomes. Genome Biol. 2004;5(2):R12. doi:10.1186/gb-2004-5-2-r12.

11. Talavera G, Castresana J. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. Syst Biol. 2007;56(4):564-77. doi:10.1080/10635150701472164.

12. Castresana J. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Molecular biology and evolution. 2000;17(4):540-52. doi:10.1093/oxfordjournals.molbev.a026334.

13. Russell DJ, Otu HH, Sayood K. Grammar-based distance in progressive multiple sequence alignment. BMC bioinformatics. 2008;9:306. doi:10.1186/1471-2105-9-306.

14. Nguyen LT, Schmidt HA, von Haeseler A, Minh BQ. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Molecular biology and evolution. 2015;32(1):268-74. doi:10.1093/molbev/msu300.

15. Hoang DT, Chernomor O, von Haeseler A, Minh BQ, Vinh LS. UFBoot2: Improving the Ultrafast Bootstrap Approximation. Molecular biology and evolution. 2018;35(2):518-22. doi:10.1093/molbev/msx281.

16. Lassmann T, Frings O, Sonnhammer EL. Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. Nucleic acids research. 2009;37(3):858-65. doi:10.1093/nar/gkn1006.

17. Lassmann T, Sonnhammer EL. Kalign--an accurate and fast multiple sequence alignment algorithm. BMC bioinformatics. 2005;6:298. doi:10.1186/1471-2105-6-298.

18. Ranwez V, Harispe S, Delsuc F, Douzery EJ. MACSE: Multiple Alignment of Coding SEquences accounting for frameshifts and stop codons. PLoS One. 2011;6(9):e22594. doi:10.1371/journal.pone.0022594.

19. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular biology and evolution. 2013;30(4):772-80. doi:10.1093/molbev/mst010.

20. Gouveia-Oliveira R, Sackett PW, Pedersen AG. MaxAlign: maximizing usable data in an alignment. BMC bioinformatics. 2007;8:312. doi:10.1186/1471-2105-8-312.

21. Collingridge PW, Kelly S. MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments. BMC bioinformatics. 2012;13:117. doi:10.1186/1471-2105-13-117.

22. Hoang DT, Vinh LS, Flouri T, Stamatakis A, von Haeseler A, Minh BQ. MPBoot: fast phylogenetic maximum parsimony tree inference and bootstrap approximation. BMC Evol Biol. 2018;18(1):11. doi:10.1186/s12862-018-1131-3.

23. Edgar RC. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC bioinformatics. 2004;5:113. doi:10.1186/1471-2105-5-113.

24. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research. 2004;32(5):1792-7. doi:10.1093/nar/gkh340.

25. Wheeler TJ, editor. Large-Scale Neighbor-Joining with NINJA2009; Berlin,

Heidelberg: Springer Berlin Heidelberg.

26. Dress AW, Flamm C, Fritzsch G, Grunewald S, Kruspe M, Prohaska SJ et al. Noisy: identification of problematic columns in multiple sequence alignments. Algorithms Mol Biol. 2008;3:7. doi:10.1186/1748-7188-3-7.

27. Jehl P, Sievers F, Higgins DG. OD-seq: outlier detection in multiple sequence alignments. BMC bioinformatics. 2015;16:269. doi:10.1186/s12859-015-0702-1.

28. Wheeler TJ, Kececioglu JD. Multiple alignment by aligning alignments. Bioinformatics. 2007;23(13):i559-68. doi:10.1093/bioinformatics/btm226.

29. Loytynoja A, Goldman N. An algorithm for progressive multiple alignment of sequences with insertions. Proc Natl Acad Sci U S A. 2005;102(30):10557-62. doi:10.1073/pnas.0409137102.

30. Roshan U, Livesay DR. Probalign: multiple sequence alignment using partition function posterior probabilities. Bioinformatics. 2006;22(22):2715-21. doi:10.1093/bioinformatics/btl472.

31. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S. ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome research. 2005;15(2):330-40. doi:10.1101/gr.2821705.

32. Kim J, Ma J. PSAR: measuring multiple sequence alignment reliability by probabilistic sampling. Nucleic acids research. 2011;39(15):6359-68. doi:10.1093/nar/gkr334.

33. Kim J, Ma J. PSAR-align: improving multiple sequence alignment using probabilistic sampling. Bioinformatics. 2014;30(7):1010-2. doi:10.1093/bioinformatics/btt636.

34. Lefort V, Longueville JE, Gascuel O. SMS: Smart Model Selection in PhyML. Molecular biology and evolution. 2017;34(9):2422-4. doi:10.1093/molbev/msx149.

35. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000;302(1):205-17. doi:10.1006/jmbi.2000.4042.

36. Chang JM, Di Tommaso P, Notredame C. TCS: a new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction. Molecular biology and evolution. 2014;31(6):1625-37. doi:10.1093/molbev/msu117.

37. Bogdanowicz D, Giaro K. Comparing Phylogenetic Trees by Matching Nodes Using the Transfer Distance Between Partitions. J Comput Biol. 2017;24(5):422-35. doi:10.1089/cmb.2016.0204.

38. Capella-Gutierrez S, Silla-Martinez JM, Gabaldon T. trimAl: a tool for automated

alignment trimming in large-scale phylogenetic analyses. Bioinformatics. 2009;25(15):1972-3. doi:10.1093/bioinformatics/btp348.

39. Herman JL, Novak A, Lyngso R, Szabo A, Miklos I, Hein J. Efficient representation of uncertainty in multiple sequence alignments using directed acyclic graphs. BMC bioinformatics. 2015;16:108. doi:10.1186/s12859-015-0516-1.

40. Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W, Gascuel O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol. 2010;59(3):307-21. doi:10.1093/sysbio/syq010.

41. Maldonado E, Almeida D, Escalona T, Khan I, Vasconcelos V, Antunes A. LMAP: Lightweight Multigene Analyses in PAML. BMC bioinformatics. 2016;17(1):354. doi:10.1186/s12859-016-1204-5.