LIB Leibniz Institute for the Analysis of Biodiversity Change

Patrick Kück

SeaLion version 1.0

March 2024

Contents

List of Figures

List of Tables

1 Features

1.1 Implementation & usage

SeaLion is a command-line-driven program with its main script written in `Perl`. The more computationally demanding supertree calculations are implemented in `C++` (icebreaker.o). Both the SeaLion `Perl` script and the icebreaker.o file have to share the same path directory. SeaLion is designed to operate on Linux operating systems and can be seamlessly integrated into automated pipelines for phylogenetic studies. The calculations performed by SeaLion are generally efficient and can be executed on a standard desktop computer, even when dealing with phylogenomic datasets. However, as the number of defined clades and possible species-quartets increases, the computation time also escalates. In such cases, it is advisable to run SeaLion on a more powerful computational server system.

1.2 OS compatibility

SeaLion is fully compatible with Linux (developed and tested on Ubuntu 18.04.6 LTS).

1.3 Source

SeaLion, Icebreaker, and a container file with all external script dependencies can be freely downloaded from: https://github.com/PatrickKueck/SeaLion.git

1.4 Script dependencies

SeaLion functions as a pipeline script, necessitating the prior installation of external software packages and libraries to ensure the smooth operation of its various processes (refer to Table 1 for an overview). The required script dependencies fall into two categories: basic software packages and output extension packages. Basic software packages include PERL, the Statistics-R package, and p4-phylogenetics (using Python 3), all of which are essential for SeaLion processing. On the other hand, output extension packages are required for additional R and/or LaTeX prints.

Table 1: Summary of SeaLion script dependencies: The installation of the two Perl and Python packages are mandatory. Additionally, for R and LaTeX respective pdf prints, all pdflatex and R packages listed here, as well as pdfunite, need to be installed.

Type	Package	Type	Installation-Command
Perl	Version 5 or higher	basic	`sudo apt-get install perl-base`
Perl	Statistics::R	basic	`sudo apt-get install -y libstatistics-r-perl`
R	Version $\geq$ 4	output extension	`sudo apt-get install r-base`
R	library(ggplot2) Version $\geq$ 3.4.0	output extension	`sudo apt-get -y install r-cran-ggplot2`
R	library(svglite)	output extension	`sudo apt-get -y install r-cran-svglite`
R	library(reshape)	output extension	`sudo apt-get -y install r-cran-reshape`
R	library(BiocManager)	output extension	`sudo apt-get -y install r-cran-BiocManager`
R	library(ggtree)	output extension	BiocManager::install("ggtree")
R	library(gridExtra)	output extension	`sudo apt-get -y install r-cran-gridExtra`
R	library(ggtern) Version $\geq$ 3.4.2	output extension	`conda install -c conda-forge r-ggtern`
Python	Python 3	basic	`sudo apt-get -y install python3.x`
Python	p4-phylogenetics	basic	see https://github.com/pgfoster/p4-phylogenetics/
pdfunite	poppler-utils	output extension	`sudo apt-get install poppler-utils`
pdflatex	texlive-latex-base	output extension	`sudo apt-get install texlive-latex-base`
pdflatex	texlive-fonts-recommended	output extension	`sudo apt-get install texlive-fonts-recommended`
pdflatex	texlive-fonts-extra	output extension	`sudo apt-get install texlive-fonts-extra`
pdflatex	texlive-latex-extra	output extension	`sudo apt-get install texlive-latex-extra`

Therefore, these packages need only be installed if the respective print format is selected. R prints require R version 4 or higher with aligned R packages, LaTeX prints use pdflatex based on texlive and

aligned packages, while pdfunite is mandatory for both types of print. Script dependencies can be installed individually by following the installation commands listed in Table 1. Alternatively, all dependencies can be installed in a single process run using the provided SeaLion singularity-container file (see section 1.4.1).
Note: Note that ggtern versions $<$ 3.4.2 may not be compatible with R version $\geq$ 4 or ggplot2 version $\geq$ 3.4.0. If you encounter issues with printing ternary plots, ensure that your installed version of `ggtern` is compatible with both R version $\geq$ 4 and ggplot2 version $\geq$ 3.4.0. For the installation of p4-phylogenetics follow the developers installation instructions at https://github.com/pgfoster/p4-phylogenetics/blob/master/INSTALL.rst

1.4.1 Using a SeaLion Singularity container file

To enhance the usability of the workflow, a Singularity container, SeaLion_container.dev was designed and built. This container facilitates a more user-friendly distribution of the software and improves the reproducibility of analyses performed by the workflow.

Build a SeaLion Singularity container Type...

- `user@linux:~$ cd ./code`
  `user@linux:~$ singularity build SeaLion_container.sif SeaLion_container.def`

Note: To incorporate the SeaLion perlscript into the container, the command above needs to be executed in the same directory where the main Perl-written SeaLion script and the additional icebreaker.o file are located. e.g. the provided subfolder `./code/`.

Run software Type...

- `user@linux:~$ cd testdata CONTAINER="../code/SeaLion_container.sif"`
- `user@linux:~$ singularity exec ${CONTAINER} sealion.pl`

The workflow can be initiated with the above command from within the directory containing the test data. Note: Make sure to provide the correct data path and the accurate SeaLion perlscript name. For further information about Singularity Containers, please have a look at the official documentation: https://docs.sylabs.io/guides/3.5/user-guide/quick_start.html

1.5 SeaLion input

SeaLion processes files containing multiple nucleotide and amino acid sequence alignments in FASTA and PHYLIP formats. Additionally, an input clade-definition file is required. This file should be in plain TEXT format and include at least four predefined clades, each consisting of one or more species-sequences. Additionally, one clade needs to be specified as the outgroup. The mandatory components for the analysis include the alignment, the cladefile, and the definition of the outgroup. Unless specified otherwise, SeaLion examines all possible outgroup including species-quartets, with a default limit of 20,000, between the four predefined clades of each relevant clade-quartet in the analysis. While there is no strict upper limit for the number of defined clades, it is advisable to limit the number to around ten due to the exponential increase in the number of different tree possibilities and subsequent computation time.
Note: SeaLion enforces several constraints...

- The input file(s) should not contain multiple species-name records.
- Species with dissimilar names between a predefined cladefile and a corresponding multiple sequence alignment are excluded from the analysis.
- One clade needs to be defined as the specific outgroup
- Forbidden site positions (positions with gaps, ambiguities, or missing characters) are excluded for each species-quartet in a given alignment independently.

1.5.1 Sequence-alignment file

To specify the alignment input file, use the command `-i name_of_alignment_infile`. The name of the alignment file should include the file format suffix (e.g., .fas). SeaLion supports two sequence input file formats: FASTA (.fas) and relaxed PHYLIP (.phy). It is important that all sequences in the input file have the same length. Sequence names may contain alphanumeric characters, underscores ("_"), and, in the case of FASTA files, spaces; other characters are not allowed. Sequences can consist of characters from the universal DNA/RNA or amino acid code, as well as ambiguity characters like "?", "X", and "-".

FASTA (.fas) format SeaLion can read sequences from FASTA files in two acceptable formats: either in a single line or with line interruptions (blocks). For both formats, sequence related species names must be in a single line and start with a ">" symbol. Each line should end with a line break. Table 2 provides an example of both acceptable FASTA formats.

Table 2: Known FASTA format in non-interleaved (format 1) and interleaved format (format 2).

FASTA format 1

```
>Name_sequence_1
AGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
>Name_sequence_2
AGCTCCGGCCCTTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
⋮
>Name_sequence_n
AGCTCCCGTCCTTTGGAGAGGTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
```

FASTA format 2

```
>Name_sequence_1
AGCTGTCCTTTCTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
AGCTGTCCTTTCTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
⋮
>Name_sequence_n
AGCTGTCCTTTCTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
AGCTGTCCTTTCTTG–AGA–GTGTCCTTTCCTAGCTCCCGTCCTTTG–AGA–GTGTCCTTTCCT
```

FASTA files can be generated by various programs, such as `MAFFT` (Katoh *et al.*, 2005; Katoh and Toh, 2008, 2010; Katoh and Standley, 2013), `MUSCLE` (Edgar, 2004), or `T-COFFEE` (Notredame *et al.*, 2000; Notredame, 2002). To convert aligned sequence files from other formats to FASTA, tools like `T-COFFEE` (Notredame *et al.*, 2000; Notredame, 2002), `FASconCAT` (Kück and Meusemann, 2010), or `FASconCAT-G` (Kück and Longo, 2014) can be employed. `FASconCAT` and `FASconCAT-G` are also useful for gene concatenation. Refer to the respective manuals and publications for further details.

Relaxed PHYLIP (.phy) format Each line in the PHYLIP file must end with a line break. Table 3 illustrates a typical PHYLIP file in interleaved format, although non-interleaved format is also permissible. Sequence names are allowed to contain more than ten characters at maximum and should be separated from the following sequence by a white space.

Table 3: Example of a relaxed interleaved PHYLIP formatted input file.

PHYLIP format (Interleaved)				
6 40				
Name_sequence_1	AGGGCCCTTG	CGCTTGGCCC	CGCTTGGCCC	AGGGCCCTTG
Name_sequence_2	AGGGCCCTTG	CGCCTCCCCC	CGCTTGGCCC	AGGGCCCTTG
Name_sequence_n	AGGCCCCTTG	CGCCGCCCGG	CGCTTGGCCC	AGGGCCCTTG
<line break>				
	ATTTCCCTTG	GGCTTCCCCC	CGCTTGGCCC	AGGGCCCTTG
	ATTTCCCTTG	GGGGGCCTCC	CGCTTGGCCC	AGGGCCCTTG
	ATCTCCCTTG	GGCCGGGGGC	CGCTTGGCCC	AGGGCCCTTG

Extant tools that can automatically generate aligned sequences in PHYLIP format include the `PHYLIP package` (Felsenstein, 1993) and `T-COFFEE` (Notredame *et al.*, 2000; Notredame, 2002). To convert aligned sequence files in FASTA format, software tools like `T-COFFEE` (Notredame *et al.*, 2000; Notredame, 2002), `FASconCAT` (Kück and Meusemann, 2010), or `FASconCAT-G` (Kück and Longo, 2014) can be employed. `FASconCAT` and `FASconCAT-G` can also be used for gene concatenation. For further details, refer to the respective manuals and publications.

1.5.2 Clade-definition file

SeaLion computes single clade-quartet and resulting rooted-clade tree support for user-defined clades. To specify clades, a clade-definition file must be provided using the `-p` option, and it should be in plain .txt format. Each clade is defined on a separate line in the file, starting with a user-specified clade name (alphanumeric signs and underscores are allowed). This is followed by a comma and then comma-separated sequence names. It is important to note that all sequence names must exactly match those in the sequence input file, and the matching is case-sensitive. There should be no blanks between comma-separated species names, and a species name can only belong to a single clade. Table 4 illustrates the correct format of a typical definition file. See also `-p` option (see section 2.6.12).
Note: Single clades are internally coded by the first character of the defined clade name. Therefore, clade names are not allowed to start with the same character to avoid conflicts in script internal coding. One clade needs to be additionally defined as the specific outgroup via the `-o` option (see section 2.6.10)

Table 4: Example of a typical clade-definition file.

Clade-Definition_File.txt
A_Clade,Sequence_name_1,Sequence_name_2,Sequence_name_3,Sequence_name_4
B_Clade,Sequence_name_a,Sequence_name_b,Sequence_name_c
C_Clade,Sequence_name_5,Sequence_name_6,Sequence_name_7,Sequence_name_8,Sequence_name_9
D_Clade,Sequence_name_V,Sequence_name_X,Sequence-name_Y,Sequence-name_Z

2 SeaLion processings

To use SeaLion, open the terminal of your operating system and navigate to the folder where SeaLion is located. SeaLion can be started directly from the command line, simplifying its integration into complex process pipelines. Type the name of the SeaLion version, followed by a space and the required options, each prefixed with a dash (-).
Note: Ensure accurate entry of input options, such as ”-i” and not ”- i”. Incorrect formatting will prompt SeaLion to display an error message and open the Help menu. Alternatively, you can access the configuration menu for step-by-step parameter specifications or enter the help menu. You can test SeaLion on your system by using the SeaLion provided example input files (see section 5).

2.1 Default processing

For default processing, type...

- `user@linux:~$ perl sealion.pl -i path/alignment.fas -p path/cladefile.txt -o outgroup -s <enter>`

2.2 Customized processing

For processing based on customized parameters, include the corresponding command followed by the desired parameter. E.g. to change the maximum number of single species-quartet analyses in each clade-quartet (`-l`) to 4000 and the minimum number of character-complete site positions in each species-quartet (`-M`) to 1206, type...

- `user@linux:~$ perl sealion.pl -i path/alignment.fas -p path/cladefile.txt -o outgroup -l 4000 -M 1206 -s <enter>`

2.3 Help menu

2.3.1 General help menu

For a concise overview of individual commands and their respective parameters (Figure 1), type...

- `user@linux:~$ perl sealion.pl -h <enter>`

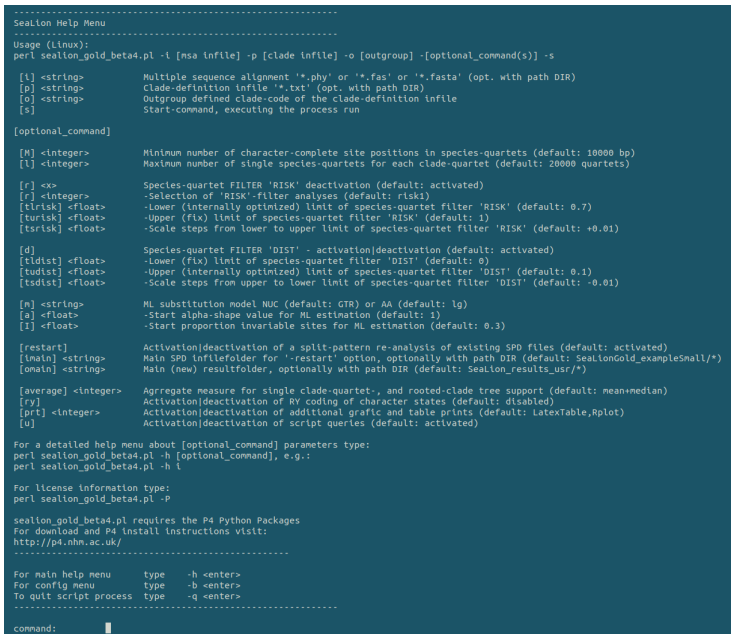

Figure 1: This figure provides an overview of the SeaLion general help menu, detailing individual parameter options. For more detailes about specific parameters type '-' followed by the corresponding command-code (no space sign inbetween) and press enter.

Note: Avoid initiating the SeaLion help menu on a server blade via a script job, such as `perl sealion.pl -h -s`. This may result in an endless loop caused by repeated calls to the help menu based on the `-h` command.

2.3.2 Command specific help menu

For a more in-depth understanding of individual commands and their corresponding parameters (Figure 2), include the relevant command code. For instance, to obtain more information on how the cladefile must be formatted, refer to the cladefile assigned command (`p`) and type...

- `user@linux:~$ perl sealion.pl -h p <enter>`

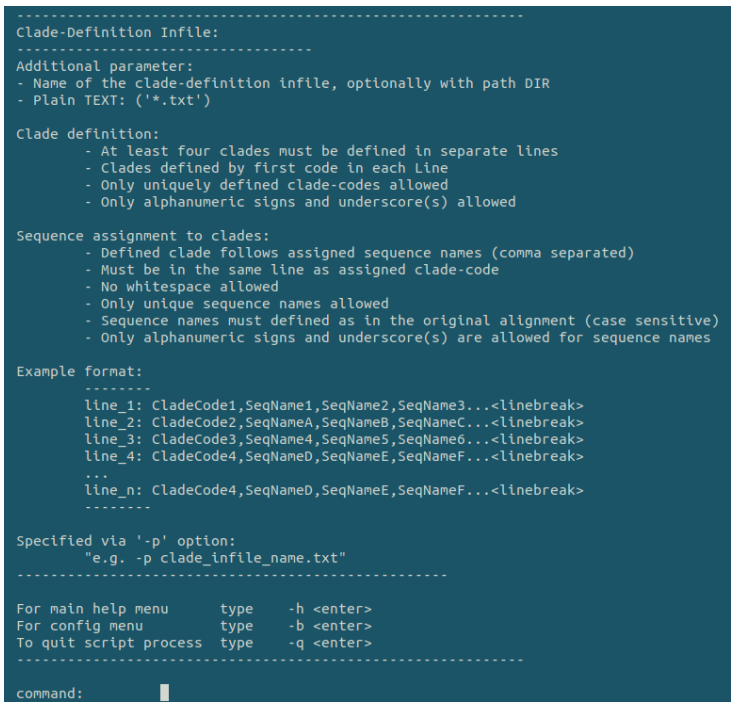

Figure 2: Example of an in-depth command explanation, providing additional details about the clade-definition file specified through the `-p` option.

2.4 Configuration menu

To access the configuration menu (Figure 3) and enable or disable individual commands or modify specific parameters step by step, type...

- `user@linux:~$ perl sealion.pl <enter>`

2.4.1 Paramter specification

Command code with additional parameter Within the config menu, single parameters can be changed step by step, by typing the parameter corresponding command code and (if applicable) the new parameter. For example, to change the maximum number of single species-quartets in clade-quartets from default to 5000, type command code followed by an integer ...

- `user@linux:~$ -M 2000 <enter>`

Command code without additional parameter For command codes that don't require additional parameters, simply enter the command code to toggle between different parameter setups. For instance, to alter the activation status of the `-restart` option, enter the following command ...

- `user@linux:~$ -restart <enter>`

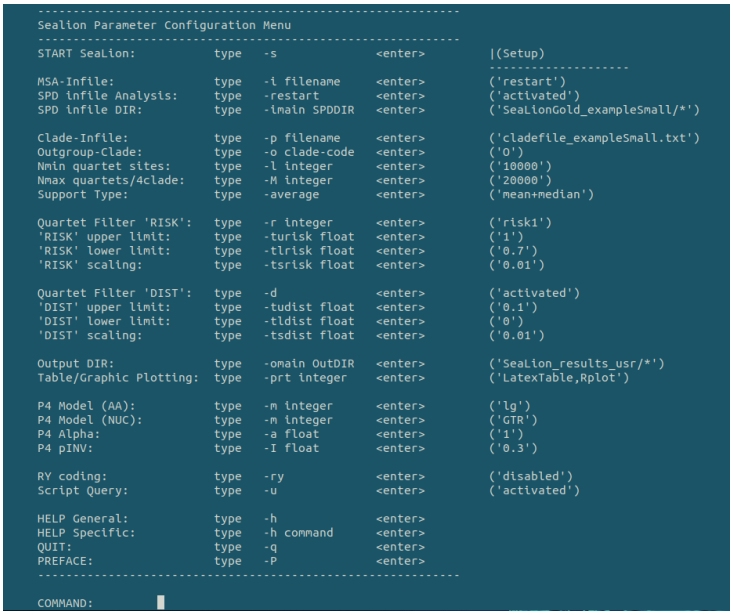

Figure 3: This figure provides an overview of the SeaLion configuration menu, detailing individual parameter options in the left column, aligned commands in the middle column, and their current settings in the right column. To modify specific parameters through the configuration menu, enter the corresponding command along with the required value (indicated next to the command code if applicable), and press enter. The terminal output will then reflect the updated parameter setting.

2.5 Menu switching

2.5.1 Switch to general help menu

To switch from the configarition menu to the help menu, type...

- `user@linux:~$ COMMAND: -h <enter>`

2.5.2 Switch to parameter-specific help menu

To switch from the configarition menu to a detailed help menu of a single command code, type `-h` and the desired command code without '-'. For example, to get more details about the clade-definition file (defined via `-p` option), type ...

- `user@linux:~$ COMMAND: -h p <enter>`

2.5.3 Switch to configuration menu

To switch from the help menu to the configarition menu, type...

- `user@linux:~$ COMMAND: -b <enter>`

2.5.4 Close menu

To exit the menu and close Sealion, type...

- `user@linux:~$ COMMAND: -q <enter>`

2.6 Command codes

SeaLion knows several input file options. It stops and opens the Help menu if an unknown option is encountered. SeaLion checks each input file according to correct format and forbidden sequence and structure characters. This subsection gives a short explanation for possible input file options and accepted file formats. Notice that not supported file formats cause SeaLion to abort.
Note: A classification of an outgroup clade via the `-o` option is mandatory.

2.6.1 ML α shape start parameter (-a)

- **Description:** ML alpha shape parameter for rate heterogeneity. Start parameter for ML Estimation of potentially convergently evolved split pattern frequencies.
- **Additional Parameter:** Float number 0.1 to 100.0
- **Default:** 1.0
- Note: To use P4 without estimation of among-site rate variation (ASRV), set α to 100 (`-a 100`). SeaLion will use just one rate category instead of four and the proportion of invariable sites will be set to zero, independent of given parameter value under `-I` option.
- **Specified via '-a' option:** `e.g., -a 0.5`

2.6.2 Aggregate measure of tree support (-average)

- **Description:** The `-average` option defines the average measure of single species-quartet support for individual clade-quartet trees. This measure is used as the basis for rooted-clade tree support calculation. If the combined option (`-average 3`) is selected, the mean and the median are separately used as a measure of support, with individual result prints for each of the two measures.
- **Additional Parameter:** Positive integer number: 1 to 3
- **Default:** median
- **Specified via '-average' option:** `-average`

Table 5: Selectable support measures.

OTU-Filter	Parameter	Process
Mean	1	Mean as aggregate support measure
Median	2 (Default)	Median as aggregate support measure
Mean+Median	3	Seperate analyses of both support aggregations

2.6.3 Species-quartet filter 'DIST' (-d)

- **Description:** The `-d` option activates or deactivates (if already activated) the species-quartet filter 'DIST' approach. This approach rejects species-quartets in single clade-quartet analyses if the quartet's corresponding tree support difference between the best and second-best (species-quartet-related) clade-quartet tree is below the upper (optimized) 'DIST' threshold. An activated `-d` option can be deactivated by typing `-d` again and vice versa.
- **Additional Parameter:** No extra parameter necessary
- **Default:** active
- **Specified via '-d' option:** `-d`

2.6.4 Multiple sequence-alignment infile (-i)

- **Description:** Name of the multiple sequence-alignment (MSA) file, optionally with the path DIR.
- **Additional Parameter:** String: Alphanumeric signs & underscores
- **Allowed Formats:**
  - FASTA ('*.fas' or '*.fasta')
  - PHYLIP ('*.phy') (strict or relaxed)
- **Allowed Site Conditions:**
  - Sequences of equal length
  - Nucleotide states
  - Amino-acid states
  - Ambiguity states
  - Indel/GAP states ('-')
  - Missing states ('?', 'X')
- **Allowed Sequence Names:**
  - Alphanumeric signs
  - Underscores ('_')
  - Only unique sequence names
- **Specified via '-i' option:** `-i MSA_filename.fas`

2.6.5 Main SPD infile pathDIR (-imain)

- **Description:** Path to the main folder of already analyzed split-pattern distribution (SPD) files ('SeaLion_detailed_split_calc_q*.txt'), located in SeaLion clade-quartet ('XYWZ') respective result subfolders ('SPD/XYWZ/*'), contributing to the actually specified clade-definition file (`-p` option) and input alignment (`-i` option).
- **Additional Parameter:** Pathstring
- **Default:** undefined
- Note: This parameter is used for starting a re-analysis (if -restart option is activated) without re-processing P4 of MSA observed and ML expected site pattern frequencies, which is one of the most time-consuming processes in the overall analysis.
- **Specified via '-imain' Option:** e.g. `-imain MySPDfile_resultFolder`

2.6.6 ML pINV start parameter (-I)

- **Description:** ML proportion of invariable site estimation. Start parameter for P4 performed ML estimation of potentially convergently evolved split pattern frequencies.
- **Additional Parameter:** Float number: 0.0 to 1.0
- **Default:** 0.3
- **Specified via '-I' Option:** e.g. `-I 0.15`

2.6.7 Maximum number of single clade-quartet analysed species-Quartets (-l)

- **Description:** Maximum number of single analyzed species-quartets for each clade-quartet. The total computation time can get very long if the number of quartets to be analyzed is large. Defining a maximum of single species-quartet analyses in each clade-quartet can help to reduce computation time (see Table 6).
- **Additional Parameter:** Integer number: ≥ 1
- **Default:** 20,000
- Note: If the number of possible unique species-quartets of a clade-quartet exceeds the maximum number of defined species-quartet analyses, the process run will stop with a command line request. The request can be suppressed by the `-u` option (see section 2.6.22). In that case, species-quartets are drawn randomly from the overall pool of available, clade-quartet corresponding species-quartets. To avoid terminal request in pipeline processes set the maximum number of allowed species-quartets sufficiently high, e.g. `-l 10000000000000`.
- **Specified via '-l' Option:**, e.g. `-l 10000`

Table 6: Examples of single species-quartet computation times for different sequence lengths using a normal desktop computer (2.8 GHz Intel Core i7).

Data Type	Quartet Sequence Length (bp)	Computation Time (sec)
Nucleotide		
	30,000	≈ 8
	5,000	≈ 4
	700	≈ 1
Amino Acid		
	2,000	≈ 100
	500	≈ 80

2.6.8 ML substitution model of P4 species-quartet analyses (-m)

- **Description:** Substitution model used for Maximum Likelihood (ML) estimation of potentially convergently evolved split pattern frequencies with P4. Under default the GTR model is used for nucleotide data and the WAG model for amino acids. Alternatively, PENGUIN provides optionally four other nucleotide and three other amino acid substitution models (Table 7).
- **Additional Parameter:** Integer number: 1 to 11
- **Default (NUC):** GTR
- **Default (AA):** lg
- **Specified via '-m' Option:**, e.g. `-m 1`

2.6.9 Minimum sequence length for species-quartets (-M)

- **Description:** Only character-complete site positions in species-quartets are analyzed. Character-incomplete sites, including e.g. gaps or ambiguities, are rejected from each species-quartet analysis. Species-quartets are rejected if the number of remaining sites is lower than specified by the `-M` parameter.
- **Additional Parameter:** Integer number: ≥ 1
- **Default:** 10,000
- **Specified via '-M' Option:**, e.g. `-M 2000`

Table 7: Implemented Maximum Likelihood substitution models for nucleotide and amino acid data.

Data Type	Model	Code	Parameter
Nucleotide			
	General Time-Reversible	GTR	1 (Default)
	Hasegawa, Kishino & Yano 1985	HKY	2
	Kimura 2-Parameter Model	K2P	3
	Felsenstein 1981	F81	4
	Jukes and Cantor 1969	JC	5
Amino Acid			
	Whelan & Goldman 2001	wag	6 (Default)
	Jones, Taylor, & Thornton 1992	jtt	7
	Dayhoff, Schwartz & Orcutt 1978	d78	8
	Adachi & Hasegawa 1996	mtrev24	9
	Le & Gascuel 2008	LG	10
	Henikoff & Henikoff 1992	BLOSSUM62	11

2.6.10 Outgroup definition (-o)

- **Description:** Clade code of the outgroup as defined in the analysis corresponding cladefile (`-p` option). The clade code of the outgroup must be identical to one of the defined clades of the cladefile (case sensitive).
- **Default:** undefined
- **Additional Parameter:** String: Alphanumeric signs & underscores
- **Specified via '-o' Option:**, e.g. `-o OUT`

2.6.11 Main outfile pathDIR (-omain)

- **Description:** Name of the new result output folder, with path DIR if the defined output folder is not located in the script DIR.
- **Additional Parameter:** String: Alphanumeric signs & underscores
- **Default:** SeaLion_results
- Note: With `-restart`, SPD files of `-imain` defined inputpath are copied to `-omain` defined output-path. Furthermore, already pre-existing SPD result files in `-omain` are always deleted in advance of a new analysis.
- **Specified via '-omain' Option:**, e.g. `-omain Output_Folder`

2.6.12 Clade-definition infile (-p)

- **Description:** Name of the cladefile (*.txt), optionally with path DIR.
- **Additional Parameter:** String: Alphanumeric signs & underscores
- **Default:** undefined
- **Allowed Formats:**
  - Plain TEXT ('*.txt')
- **Clade definition:** At least four clades must be defined in separate lines, and clades are defined by the first code in each line. Only uniquely defined clade codes are allowed, and only alphanumeric signs and underscores are allowed. The first character of each clade code is used in output tables and grafics. Thus, each clade code must start with a unique character.

- **Species assignment to clades:** The defined clade follows assigned species names (comma-separated) in the same line as the assigned clade code. No whitespace is allowed, and only unique species names are allowed. Species names must be defined as in the original alignment (case-sensitive), and only alphanumeric signs and underscores are allowed for species names.
- **Example format:**

```
line_1: AcladeCode1,SpeciesName1,SpeciesName2,SpeciesName3...
line_2: BcladeCode2,SpeciesNameA,SpeciesNameB,SpeciesNameC...
line_3: CcladeCode3,SpeciesName4,SpeciesName5,SpeciesName6...
line_4: OcladeCode4,SpeciesNameD,SpeciesNameE,SpeciesNameF...
...
line_n: NcladeCode5,SpeciesNameX,SpeciesNameY,SpeciesNameZ...
```

- **Specified via '-p' Option:** e.g. `-p cladefile.txt`

2.6.13 Species-quartet filter 'RISK' analysis (-r)

- **Description:** Activates or deactivates parameter corresponding (depending on the parameter) 'RISK' filter settings to filter single species-quartets based on their ratio of potentially convergent (Nc), and apomorphic (Na) evolved split signal among quartet related site positions (ratio Nc/Na). RISK1 and RISK2 are based on the original, uncorrected ratio of Nc/Na. The distinction between RISK1 and RISK2 lies in the fact that RISK2 rejects all species-quartets with a best tree support ratio of Nc to Na above the optimized lower RISK threshold (`tlrisk`). On the other hand, RISK1 retains species-quartets with best tree support above the lower threshold, as long as the two alternative trees have a Nc-to-Na ratio greater than 1 (upper, fix threshold `turisk`), provided that the best tree has a Nc-to-Na ratio below 1. In such instances, the convergent tree signal predominates for the two alternative trees, making the best tree, with a Nc-to-Na ratio below 1, the sole tree with a dominant apomorph signal and, consequently, potentially reliable (see example Figure 11). Nevertheless, trees with a Nc-to-Na ratio greater than 1 were only observed in data simulations yet. In the absence of such trees, the RISK1 and RISK2 filters behave identically. Parameter are assigned to different single or combined RISK filter (see Table 8). If the combined option is selected, such as RISK1+RISK2 (`-r 3`), the dataset is analyzed and the output is printed separately for each RISK scheme.
- **Additional Parameter:** Positive integer number: 1 to 15 or 'x' for deactivation.
- **Default:** RISK1
- **Specified via '-r' option:** e.g. `-r 3`

Table 8: Selectable RISK filter combinations.

Quartet-Filter	Parameter
RISK1	1 (Default)
RSIK2	2
RISK1+RISK2	3
Deactivate RISK Filter	x

2.6.14 Restart with existing SPD files (-restart)

- **Description:** The `-restart` option allows a re-analysis of a previously conducted SeaLion analysis by already using existing split-pattern distribution files (SPD) of a former analysis (defined via `-imain`). SPD files of `-imain` are copied to the new result output path (`-omain`) after deleting pre-existing `-omain` result files if SPD infile path (`-imain`) and output path (`-omain`) are not identical. An activated `-restart` option can be deactivated by typing `-restart` again.

- **Additional Parameter:** None
- **Default:** deactivated
- Note: To use the `-restart` option both, the path to the SPD files (`-imain`) and the SPD files corresponding cladefile (`-p`) have to be originated from the same analysis.
- **Specified via '-restart' Option:** `-restart`

2.6.15 RY coding of site characters (-ry)

- **Description:** With the `-ry` option activated, site characters are translated into R or Y states, depending on their chemical behaviour (see Table 9). This reduces the number of possible character states from four to two. An activated `-ry` option can be deactivated by typing `-ry` again.
- **Additional Parameter:** None
- **Default:** deactivated
- **Specified via '-ry' option:** `-ry`

Table 9: . RY code used for different data types

Data Type	Classification	Code
Nucleotide		
	Purine	R
	Pyrimidine	Y
Amino Acid		
	Hydrophobic	R
	Hydrophilic	Y

2.6.16 Species-quartet filter 'DIST' (upper) threshold (-tudist)

- **Description:** Start value for the upper 'DIST' threshold optimization. The upper 'DIST' threshold is stepwise decreased (following the `-tsdist` defined scaling) until either an optimum of rejected and remaining species-quartets is found, or the lower 'DIST' threshold (`-tldist`) is reached. The optimized upper threshold scale is subsequently used as the final lower limit of allowed clade-quartet respective support distance between the best and second-best tree.
- **Additional Parameter:** Float number: 0.0 to 1.0
- **Default:** 0.1
- Note: The upper (start) limit has to be greater/equal the lower threshold (`tudist` $\geq$ `tldist`).
- **Specified via '-tudist' option:** e.g. `-tudist 0.3`

2.6.17 Species-quartet filter 'DIST' (lower) threshold (-tldist)

- **Description:** Lower (fix) threshold limit for 'DIST' species-quartet filtering, used for upper 'DIST' threshold (`-tudist`) optimization. During the threshold optimization, the upper 'DIST' threshold is stepwise decreased (following the `-tsdist` defined scaling) until either an optimum of rejected and remaining species-quartets is found, or the lower 'DIST' threshold is reached. The adjustment of the lower threshold limit enables a more flexible range for threshold optimization. Instead of searching for an optimum between 0 and the upper threshold, the lower threshold limit can be shifted too, e.g., from 0 to 0.05, excluding an optimization analysis of clade-quartet tree distances lower than 0.05.

- **Additional Parameter:** Float number: 0.0 to 1.0
- **default:** 0.0
- Note: The lower threshold has to be less/equal to the upper threshold limit ('tldist' $\leq$ 'tudist').
- **Specified via '-tldist' option:** e.g. `-tldist 0.01`

2.6.18 Species-quartet filter 'DIST' threshold scaling (-tsdist)

- **Description:** Negative scale value for upper 'DIST' threshold optimization. Starting from the defined upper threshold value (`-tudist`), the number of 'DIST' rejected species-quartets is evaluated for each negative scale step until either an optimum of rejected and remaining species-quartets is found, or the lower threshold value (`-tldist`) is reached (whose value is then used as threshold criteria).
- **Additional Parameter:** Float number: 0.0 to 1.0
- **default:** 0.01
- **Specified via '-tsdist' option:** e.g. `-tsdist 0.05`

2.6.19 Species-quartet filter 'RISK' (lower) threshold (-tlrisk)

- **Description:** Start value of lower 'RISK' threshold optimization, regarding 'RISK' filtering of potentially Nc/Na biased species-quartets (see `-r` option). The lower 'RISK' threshold is stepwise increased (following the `'-tsrisk'` defined scaling) until either an optimum of rejected and remaining species-quartets or the upper 'RISK' threshold (`-turisk`) is reached.
- **Additional Parameter:** Float number: 0.0 to 1.0
- **default:** 0.7
- Note: The lower (start) threshold has to be less/equal the upper threshold limit (`tlrisk` $\leq$ `turisk`).
- **Specified via '-tlrisk' option:** e.g. `-tlrisk 0.92`

2.6.20 Species-quartet filter 'RISK' (upper) threshold (-turisk)

- **Description:** Upper (fix) threshold limit for 'RISK' species-quartet filtering.
- **Additional Parameter:** Float number: 0.0 to 1.0
- **default:** 1.0
- Note: The upper (fix) threshold has to be greater/equal the lower threshold limit (`turisk` $\geq$ `tlrisk`).
- **Specified via '-turisk' option:** e.g. `-turisk 1.01`

2.6.21 Species-quartet filter 'RISK' threshold scaling (-tsrisk)

- **Description:** Positive scale factor for lower RISK threshold optimization. Starting from the defined lower threshold value (`-tlrisk`), the number of 'RISK' rejected species-quartets is evaluated for each scale step until either an optimum of rejected and remaining species-quartets or the upper threshold value (`-turisk`) is reached (whose value is then used as threshold criteria).
- **Additional Parameter:** Float number: 0.0 to 1.0
- **default:** 0.1
- **Specified via '-tsrisk' option:** e.g. `-tsrisk 0.05`

2.6.22 Suppression of script queries (-u)

- **Description:** The `-u` option opposes possible script queries during the process run, which would stop automatic pipeline processes if the maximum number of allowed species-quartets is lower than the actual number of possible species-quartets (`-M` option). With the `-u`, script queries are suppressed, meaning that species-quartets are drawn randomly until the number of species-quartets exceeds the number of allowed quartets. An activated `-u` option can be deactivated by typing `-u` again.
- **Additional Parameter:** None
- **default:** deactivated
- **Specified via '-u' option:** `-u`

2.6.23 Additional print option (-prt)

- **Description:** The `-prt` option allows the output of additional result-graphics (generated with R) and LaTeX table prints. To use both print parameters successfully, additional R and LaTeX software packages are needed. With the 'x' parameter activated, extra prints are deactivated.
- **Additional Parameter:** Integer number: 1 to 3 or 'x' for deactivation
- **default:** deactivated
- **Specified via '-prt' option:** e.g. `-prt 3`

Table 10: Selectable combinations of additional result prints.

Additional Prints	Parameter
Print R graphics	1
Print LaTeX tables	2
Print LaTeX tables and R graphics	3
Deactivate additional prints	x (Default)

3 SeaLion output

SeaLion computes summarized output information, detailing information of the identified support for each potential species-quartet, clade-quartet, and final rooted-clade relationships. Any discrepancies in the topological support are presented through various text file details. Additionally, it provides information on species-quartet related split-pattern analyses concerning the analysis of potentially apormoprh (Na) and convergently (Nc) evolved tree signals. If specified, this information is also visualized using a diverse set of R plots and LaTeX tables, which are collectively summarized in individual PDF documents. The SeaLion results folder is organized into distinct subfolders (Table 11), each containing different types of output based on the specified parameter setting.

3.1 MCM subfolder output files

The 'MCM' subfolder contains concise information in .txt format summarizing the support of the final rooted-clade trees for both unfiltered and filtered analyses. In each analysis, four distinct files are generated. The '*nap.txt' file presents the score matrix, providing a comprehensive overview of analyzed tree scores for all clade-quartets used in determining the final rooted-clade tree support. Additionally, the '*all_topology_scores.txt' file compiles the ultimate support values for each potential rooted-clade tree, considering the top 100 best trees. The '*best_tree.txt' file contains the best rooted-clade tree identified

by each method in newick format. Meanwhile, the '*top_topology_scores.txt' file provides a summary of the support values for the top three rooted-clade trees.

Table 11: Overview of SeaLion subfolder content: The 'default' subfolder, which includes plain text result files, is consistently generated. Additionally, R plots in .pdf or .svg format, as well as LaTeX tables in .pdf format, are only created when explicitly specified.

Result Subfolder	Setup	Content
MCM	default	Final support matrix and rooted-tree support
SPD	default	Single species-quartet related split-pattern ditribution files, classified per clade-quartet
TRE	default	Best unfiltered and filtered rooted-clade trees provided in Newick format (.tre)
TSV	default	TSV tables of result specific meta-data used for R plots
TXT	default	TXT result information (tabstop delimited), similar to LaTeX table prints
PDF	`-prt 1` or `-prt 3`	PDF result plots generated by R
SVG	`-prt 1` or `-prt 3`	SVG result plots generated by R
TEX	`-prt 2` or `-prt 3`	LaTeX tables in .pdf and .tex format

3.2 SPD subfolder output files

The 'SPD' subfolder stores valuable information on the observed alignment and maximum likelihood (ML) estimated split-pattern frequencies for each analyzed species-quartet in a clade-quartet analysis. The information is compiled into separate .txt files, providing the flexibility for potential reanalysis of the identical cladefile under diverse parameter assumptions. This approach avoids the need to re-analyze the split-pattern distribution, which is the most time-consuming step in the process (see `-restart` option in section 2.6.14).

3.3 TRE subfolder output files

The final output includes the best unfiltered and filtered rooted-clade trees provided in Newick format (.tre) along with a summarized support report in .txt format.

3.4 TSV subfolder output files

The 'TSV' subfolder houses table files in .tsv format, each containing pertinent metadata essential for generating graphic plots in R. Table 12 provides an overview of the generated TSV files, including their content and the information presented in both text ('TXT') and LaTeX formats ('TEX'). Each TSV file is uniquely identified by a code at the beginning of its name, linking it to the corresponding R plots in subfolder 'PDF' and subfolder 'SVG' that share the same code. TSV output files are consistently generated, regardless of the specified print option (-prt), providing the flexibility for users to create customized R plots ('PDF', 'SVG') beyond the standard SeaLion graphic output.

3.5 TXT subfolder output files

The 'TXT' subfolder contains the main result information about single species-quartet, clade-quartet, and rooted-clade related tree support. If selected using the `-prt` option (see section 2.6.23), the information within these files can alternatively be formatted as LaTeX tables in the 'TEX' subfolder, identified by identical filename codes (see Table 12 for a short overview).

Table 12: Summary of SeaLion output file content and name conventions in individual output subfolders ('TXT', 'TEX', 'TSV') and associated R graphics (summarizing 'PDF' and 'SVG').

Content	TEX & TXT	TSV & R Grafic
SeaLion parameter overview	LP1	–
SeaLion assigned clades overview	LP2	–
SeaLion analysed clade-quartet(s) overview	LP3	–
Best unfiltered and filtered rooted-clade trees with final support	LRC1	MQ3
Best-three unfiltered and filtered rooted-clade trees with final support	LRC2	–
Summarised species-quartet support of single analysed clade-quartet trees	LQ1	MQ1
Average unfiltered support and number of analysed species-quartets for best clade-quartet trees	LQ2	MQ2
Average filtered support and number of analysed species-quartets for best clade-quartet trees	LQ2	MQ4
Species-related information about quartet particpations in filtered clade-quartet analyses	LQ3	Q5
Species-related information about quartet particpations in unfiltered clade-quartet analyses	LQ4	Q6
List of species in the cladefile without assignable alignment sequence	LQ5	–
Details on the optimization steps for species-quartet filters	LQ6	MQ6
Details on apomorph and convergent tree signal in single species-quartets	–	Q3
Species-quartet best-tree support relative to the number of analysed split-pattern	–	Q7
Number of filter rejected species-quartets in each clade-quartet	LQ7	Q6
Summary of triangle-corner coded clade-quartet trees as presented in the associated R plots	LQT1	T1
Tree support distances contributed by individual species	–	T2

3.6 PDF subfolder output files

The 'PDF' subfolder contains individual PDF summaries for single R plot graphics and LaTeX tables, separated for filtered and unfiltered analysis results (Table 13). Additionally, individual R grafic plots are stored as PDFs in the 'PDF/RPLOTS/' subfolder and as SVGs in the 'SVG' subfolder. For file-code corresponding TXT and LaTeX-formatted table files see Table 12.

Table 13: Summary of SeaLion output file content and name conventions in individual output subfolders ('TXT', 'TEX', 'TSV') and associated R graphics (summarizing 'PDF' and 'SVG').

Summary Output-File (PDF/*.pdf)	Description	File-Code
SeaLion_results_sumed_main_unfiltered	Rooted-clade trees	MQ3, R3
	unfiltered clade-quartets	MQ1, MQ2, Q3, Q6, T1, T2
SeaLion_results_sumed_mainTable_unfiltered	Rooted-clade trees	LRC1, LRC2
	unfiltered clade-quartets	LP1, LP2, LP3, LQ1, LQ2, LQ4, LQ5, LQ7
SeaLion_results_sumed_main_RISK	RISK-filtered clade-quartets	MQ1, MQ2, MQ4, MQ6, Q3, Q5, T1, T2
SeaLion_results_sumed_mainTable_RISK	RISK-filtered clade-quartets	LQ3, LQ6, LQT1
SeaLion_results_sumed_main_DIST	DIST-filtered clade-quartets	MQ1, MQ2, MQ4, Q5, T1, T2
SeaLion_results_sumed_mainTable_DIST	DIST-filtered clade-quartets	LQ3, LQ6, LQT1
SeaLion_results_sumed_main_RISK_DIST	RISK+DIST-filtered clade-quartets	MQ1, MQ2, MQ4, Q5, T1, T2
SeaLion_results_sumed_mainTable_RISK_DIST	RISK+DIST-filtered clade-quartets	LQ3, LQT1

3.7 SVG subfolder output files

The 'SVG' subfolder contains individual SVG files for single R plot graphics for filtered and unfiltered analysis results. Additionally, SVG corresponding R plots are stored as PDFs in the 'PDF/RPLOTS/' subfolder. Summarized PDFs of stored SVGs are printed to 'PDF/'. For plot corresponding TXT and LaTeX-formatted table files see Table 12.

3.8 TEX subfolder output files

The 'TEX' subfolder contains individual LaTeX tables in both PDF and LaTeX (.tex) formats. Each table information corresponds to TXT formatted table files with the same name code and to many R graphics

(see Table 12). Note: LaTeX Tables featuring clade-quartets in the header may become too large for PDFs in A4 format, particularly when the number of defined clades exceeds five. In such instances, not all table information is fully presented in the PDF. Nevertheless, all table details are retained in the original .tex formatted table files.

4 Short description of graphical R output files

With the `-prt` option set to 1 or 3, SeaLion prints additional R plots to the subfolder 'PDF/RPLOTS/'. Graphics underlying data are presented in TXT and LaTeX tables with usually related name code, distinguished by 'L' at the beginning (e.g., R plot Q1 corresponds to table LQ1). For a clearer interpretation of individual graphic outputs, concise descriptions of each graphic and assigned table formatting(s) are provided in the following subsections.

4.1 Overview R plot MQ1 – average species-quartet support for single clade-quartet trees

Lineplot showing average (mean and/or median) support for each clade-quartet tree. Support values in the range of zero to 1 (complete support) are subdivided by dashed horizontal lines, providing a visual guide to signal strength. Values below 0.4 suggest weak support, those below 0.6 indicate moderate signal strength, while values greater than or equal to 0.6 represent strong support. If analyses with both mean and median species-quartet related clade-quartet tree support are conducted, the difference of support between these two measures is shown by a black line between these two measures for each tree (Figure 4). Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information regarding mean and/or median species-quartet support and clade-quartet trees is presented in tables formatted as TXT and LaTeX with the name code LQ1, single species-quartet support is listed in TSV tables with the name code MQ1.

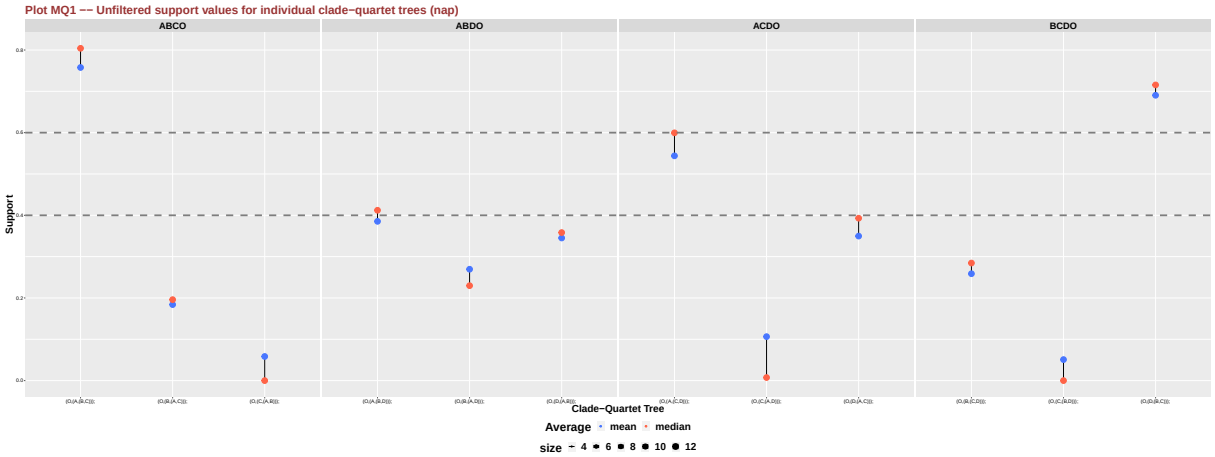

Figure 4: Example plot **MQ1:** Average (mean: blue dot; median: red dot) species-quartet support (y-axis) for each unfiltered analysed clade-quartet tree (x-axis) with support differences between both measures shown by black lines.

4.2 Overview R plot MQ2 – species-quartet support for single clade-quartet trees

Triangle plot depicting individual and average (mean and/or median) species-quartet support for each clade-quartet tree. Each corner of the triangle corresponds to one of the three clade-quartet relationships (QT1, QT2, QT3), with each species-quartet represented by a dot. The direction of stronger support towards one of the three triangle corners signifies a lower level of signal conflict between different topologies (Figure 5). Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information regarding mean and/or median species-quartet support and clade-quartet trees is presented

in tables formatted as TXT and LaTeX with the name code LQ2, single species-quartet support is listed in TSV tables with the name code MQ2.

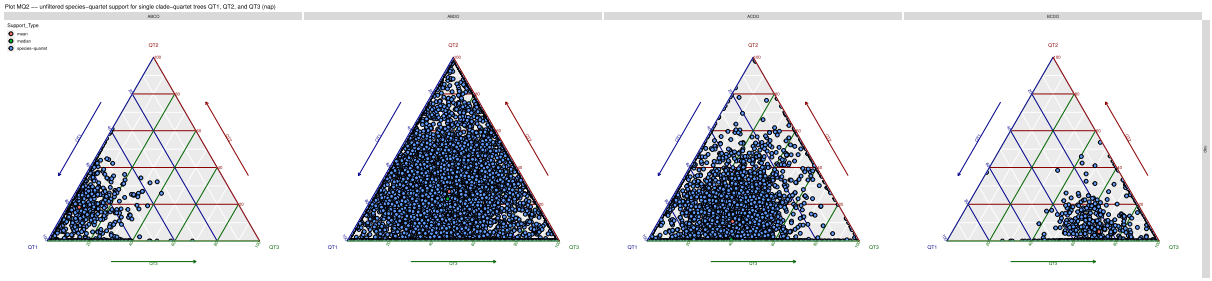

Figure 5: Example plot **MQ2:** Single (blue dot) and average (mean: red dot; median: green dot) species-quartet support (unfiltered) for single clade-quartet trees. Each clade-quartet is represented by a triangle. Each triangle corner represents one of the three clade-quartet trees. The stronger the signal strength for a tree, the stronger the direction of single species-quartet towards that triangle corner. The triangles on the left and right serve as illustrative examples of signal strength directed towards a specific corner, indicating robust support. Conversely, the second triangle on the left exhibits substantial conflict among different trees, while the third triangle on the left demonstrates moderate signal strength.

4.3 Overview R plot MQ3 – final rooted-clade tree support

Bar chart depicting the final scores for each best rooted-clade tree in both filtered and unfiltered analyses (Figure 6). Graphic-assigned information regarding best unfiltered and filtered rooted-clade trees with corresponding support is presented in tables formatted as TXT and LaTeX with the name code LRC1, and in TSV format coded as MQ3.

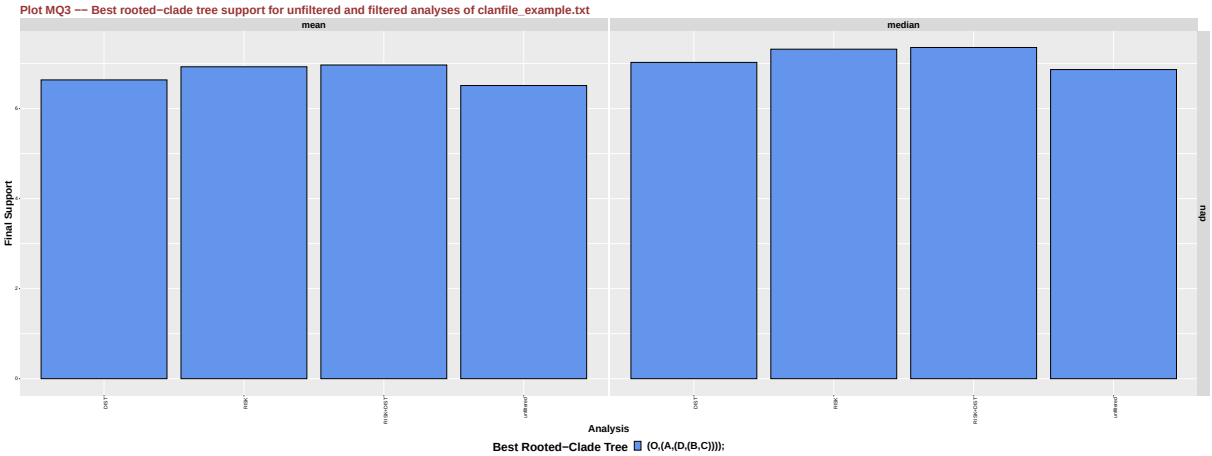

Figure 6: Example plot **MQ3:** Final support (y-axis) for each best rooted-clade tree in both filtered and unfiltered analyses (x-axis). Distinctive bar colors indicate diverse best trees. In this illustration, the analyses converge on the same best tree, represented by the blue bars. The header of each bar chart block emphasizes the specific support average utilized for clade-quartet tree assessment (left: mean; right: median).

4.4 Overview R plot MQ4 – filtered and unfiltered species-quartet support

Triangle plot depicting individual and average (mean and/or median) species-quartet support for each clade-quartet tree before (unfiltered; bottom) and after quartet filtering (top). Each corner of the triangle corresponds to one of the three clade-quartet relationships (QT1, QT2, QT3), with each species quartet represented by a dot (Figure 7). Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information regarding single species-quartet support and clade-quartet trees is presented in

tables formatted as TSV with the name code MQ4, information about best clade-tree support is given in TXT and LaTeX files with the name code LQ2.

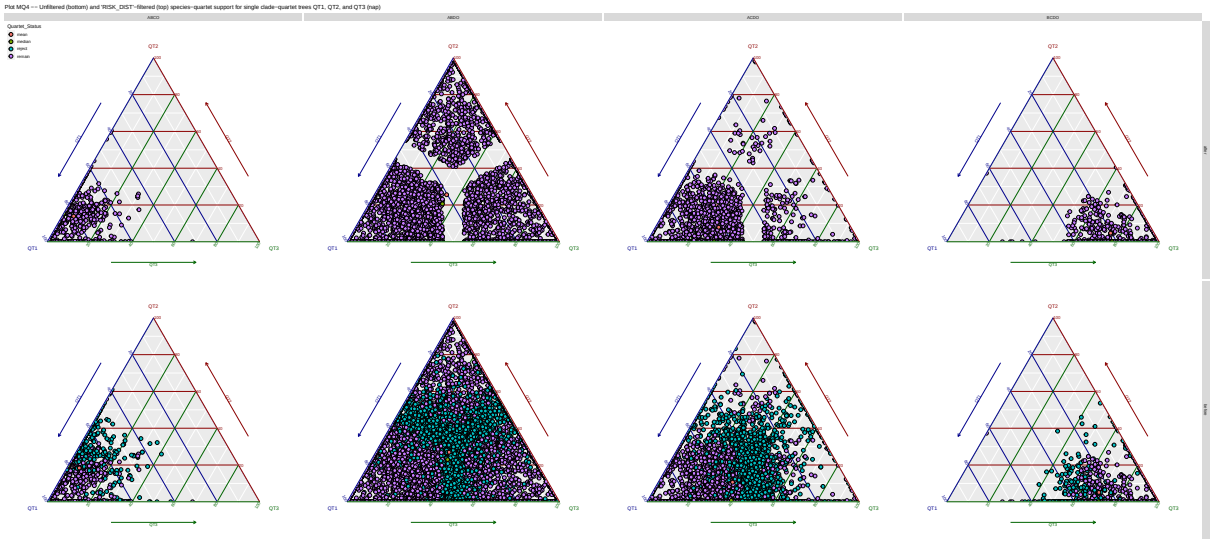

Figure 7: Example plot **MQ4:** Single and average (mean: red dot; median: green dot) species-quartet support (bottom: unfiltered top: filtered) for single clade-quartet trees. Filter remaining species-quartets are shown by purple dots, rejected species-quartets by green dots. Each clade-quartet is represented by a triangle. Each triangle corner represents one of the three clade-quartet trees. The stronger the signal strength for a tree, the stronger the direction of single species-quartet towards that triangle corner.

4.5 Overview R plot MQ6 – filter optimization

Line plot illustrating the count of retained species-quartets in each clade-quartet tree throughout the optimization process of species-quartet filters ('RISK', 'DIST') for various threshold values (Figure 8), culminating in the identification of the optimal threshold value (Figure 9). Single grafics are printed for each filter. Graphic-assigned information regarding single species-quartet support and clade-quartet trees is presented in tables formatted as TXT and LaTeX with the name code LQ6, and in TSV format coded as MQ6.

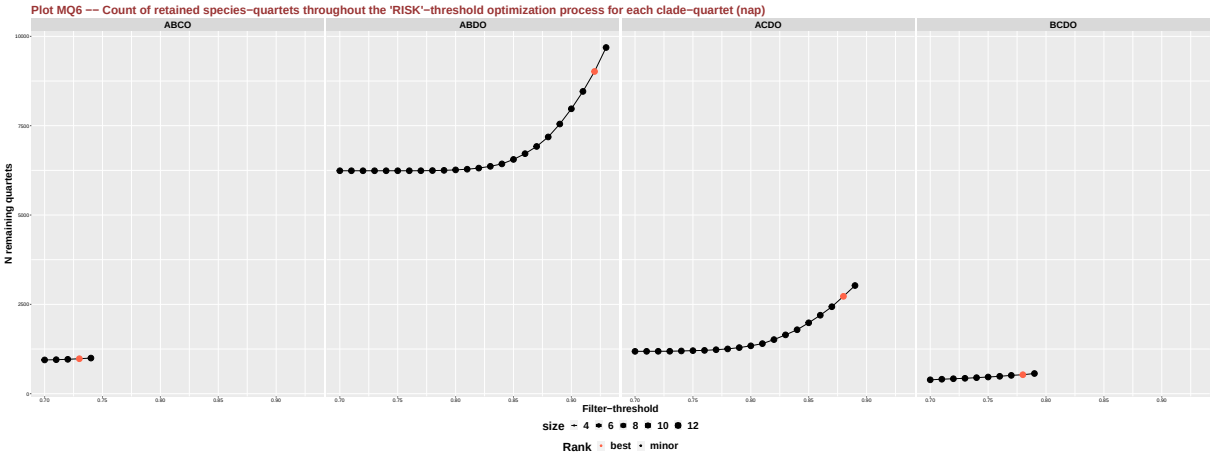

Figure 8: Example plot **MQ6 I:** Count of retained species-quartets (y-axis) across different threshold values (x-axis) during individual optimization steps of the 'RISK' and 'DIST' filters. The optimized threshold value, corresponding to the number of remaining species-quartets, is marked with a red dot.

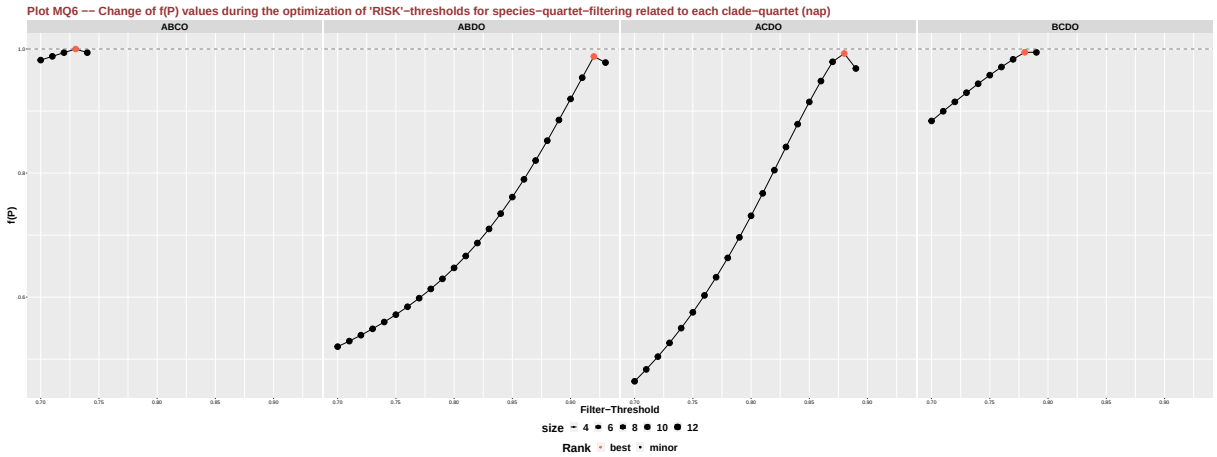

Figure 9: Example plot **MQ6 II:** Optimized function 'f(P)' (y-axis) across different threshold values (x-axis) during individual optimization steps of the 'RISK' and 'DIST' filters. The optimized threshold value, keeping a balance between rejected and remaining quartets in each clade-quartet, is marked with a red dot.

4.6 Overview R plot Q3 – tree signal in species-quartets

Q3 name coded R plots are focused on the distribution of potentially convergent (Nc) to apomorph split-pattern signal (Na) evaluated for each of the three species-quartet related trees with best supported trees highlighted different in comparison to the two alternative trees. The lower this ratio, the stronger the signal supporting the best tree, and the less surrounded is this signal in quartet cases where this tree was not the best supported. There are two distinct ways to illustrate the distribution of Nc-to-Na ratios in tree-related species-quartets. One approach involves using clouds, which enhances the visualization of potential overlaps between Nc-to-Na ratios for both the best and lower supported species-quartet trees (Figure 10). Individual graphics are generated for both filtered and unfiltered analyses.
In cases where the 'RISK' filter is chosen, an additional cloud graphic is provided to depict the distribution of Nc-to-Na ratios for best and alternative quartet trees, distinguishing between rejected and remaining species quartets (Figure 11) and, alternatively, the ratios can be depicted in a line format, where individual species-quartets are arranged side by side along the x-axis, showcasing 'RSIK' filtered best trees (Figure 12). Single grafics are printed for each clade-quartet. Graphically assigned information about single species-quartet support and clade-quartet trees is summarized for each clade-quartet in a TSV-formatted table labeled with the name code Q3.

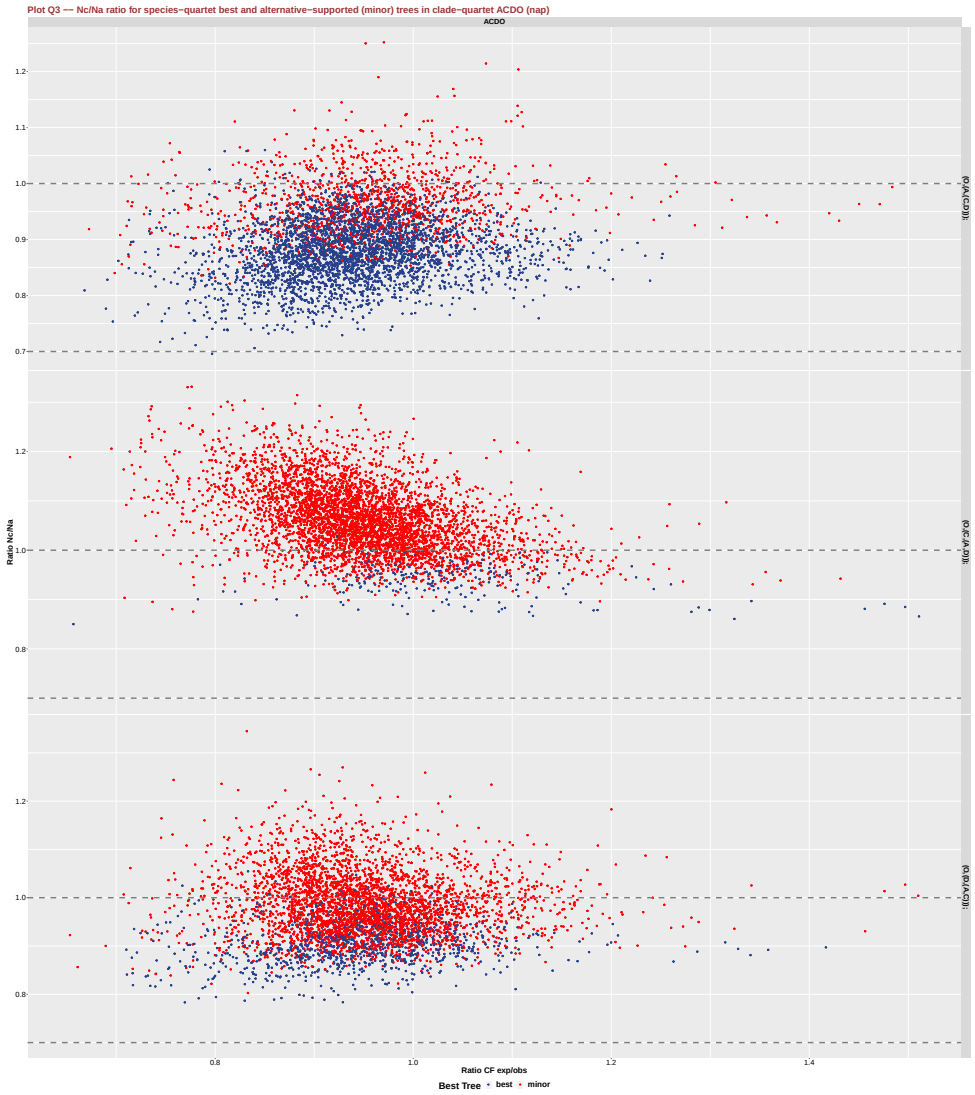

Figure 10: Example plot **Q3 I:** Graph showing support values for single species-quartet topologies for a single clade-quartet with related support expressed as ratios of potentially convergent ('Nc') to apomorphic ('Na') character states ('rNc/Na'; y-axis). The examined topologies are shown at the right of each graph segment. The visual representation comprises clouds of dots, with each dot symbolizing the Nc-to-Na proportion for a distinct species-quartet-related clade topology. Blue dots signify that this topology is the best-supported in a species-quartet, while red dots represent species-quartets where this topology is supported as alternative, second or third best tree. Within each clade combination, the lower horizontal line indicates the start threshold limit for Nc-to-Na ratios set before the 'RISK' filter optimization starts. If the 'RISK' filter is activated, the y-axis position of this line will be subsequently optimized, acting as a delineation between ratios of Nc-to-Na character states for each species-quartet's best tree that are considered to be more reliable (below the line) and values that are refused (above the line; see Figure 11). The top horizontal line in each graph segment represents the upper limit for the 'RISK' threshold during the optimization process.

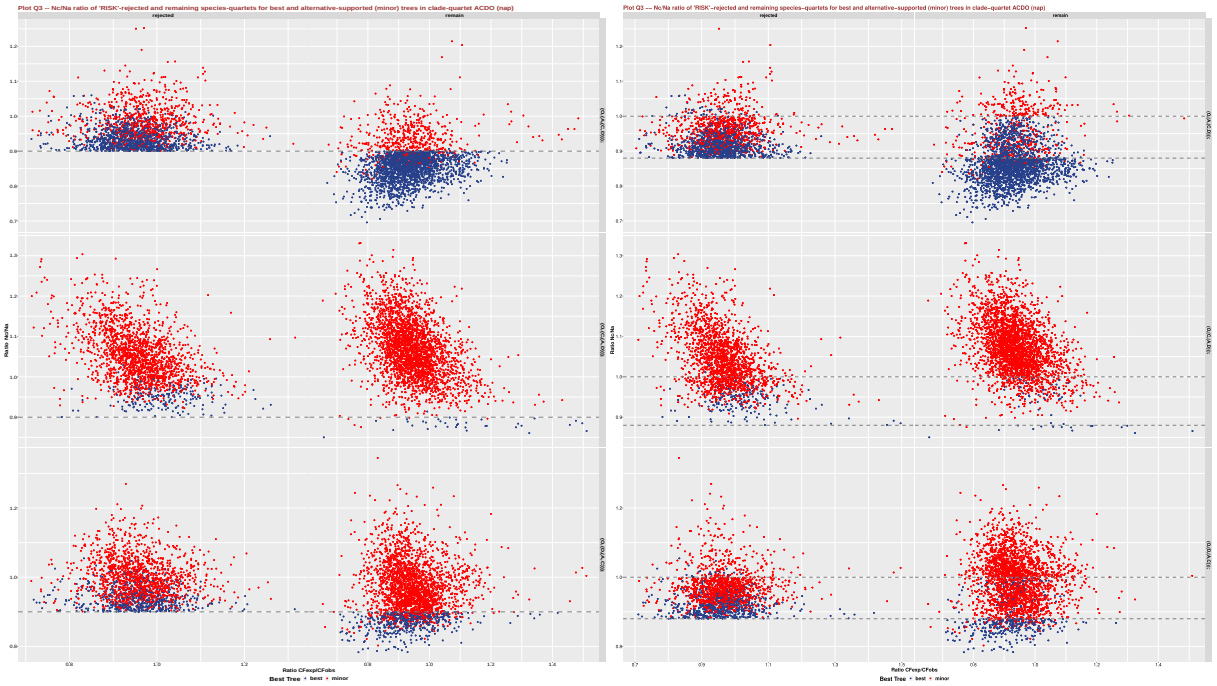

Figure 11: Example plot **Q3 II:** Graphs showing support values for single species-quartet topologies for a single clade-quartet with related support expressed as ratios of potentially convergent ('Nc') to apomorphic ('Na') character states ('rNc/Na'; y-axis), separated for 'RISK' filter remaining (right side of each plot) and rejected (left side of each plot) species-quartets. The examined topologies are shown at the right of each graph segment. The visual representation comprises clouds of dots, with each dot symbolizing the Nc-to-Na proportion for a distinct species-quartet-related clade topology. Blue dots signify that this topology is the best-supported in a species-quartet, while red dots represent species-quartets where this topology is supported as alternative, second or third best tree. Within each clade combination, the horizontal line indicates the optimized threshold limit for Nc-to-Na ratios, acting as a delineation between ratios of Nc-to-Na character states for each species-quartet's best tree that are considered to be more reliable (below the line) and best tree values that are refused (above the line). With 'RISK2' all species-quartets with best trees above that line are rejected (left plot). With 'RISK1' best trees above the line are not rejected if the two alternative trees of the species-quartet supporting the best tree, have a ratio Nc-to-Na greater than 1, as illustrated in this example for remaining best clade-quartet trees (above the line) at the top right. In such cases, convergences (Nc) dominate for the two alternative trees, rendering these trees unsupported and thus implausible. This makes the best tree, positioned above the line with a Nc-to-Na ratio below 1, the only tree with dominant apomorph tree signaland, and consequently, potentially reliable.

Figure 12: Example plot **Q3 III:** Graph showing the Nc-to-Na ratio (left y-axis) for each rejected and remaining species-quartet (x-axis) in relation to each clade-quartet tree (right y-axis). Best species-quartet trees are highlighted in orange, alternative trees in mint-green. Trees of remaining species-quartets are shown by triangles, rejected species-quartets by dots. Both horizontal (dashed) lines represent the lower and upper threshold limit respectively. In this example, the majority of remaining species-quartets support the tree (O, A, (C, D)) as the best tree. Additionally, these best trees exhibit, on average, the lowest Nc-to-Na ratio.

4.7 Overview R plot Q5 – count of single species participations in species-quartets

Line plots illustrating the count of species retained quartet participations after quartet filtering in for each clade-quartet (Figure 13) and summarized across all clade-quartets (Figure 14. Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information is presented in tables formatted as TXT and LaTeX with the name code LQ3, and in TSV format coded as Q5.

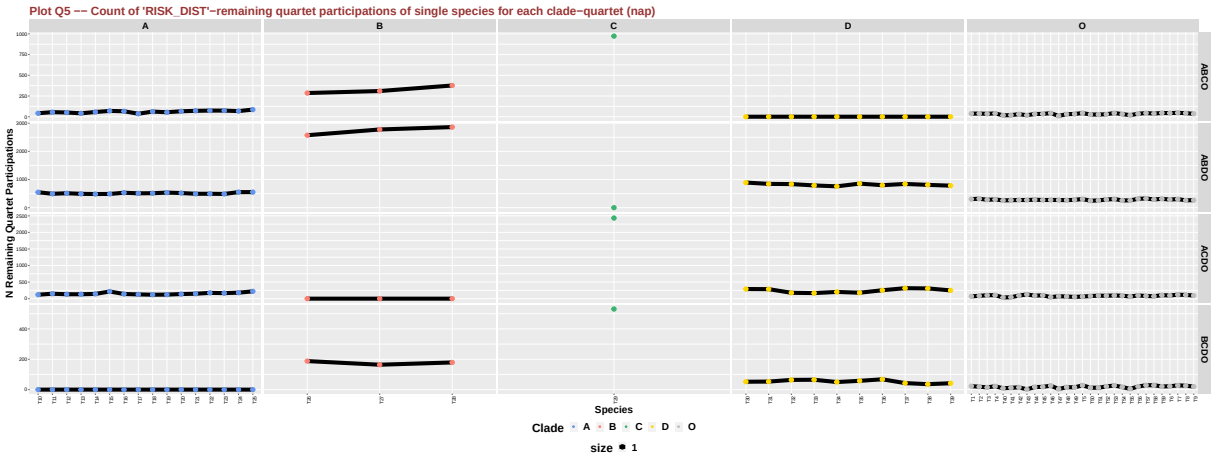

Figure 13: Example plot **Q5: I** For each species (x-axis), the count of filter remaining quartet participations (left y-axis) is presented for each clade-quartet (right y-axis), sorted based on species-assigned clades (as indicated at the top). Note that the count of ingroup species participations is zero if the species-assigned clade is not part of the clade-quartet.

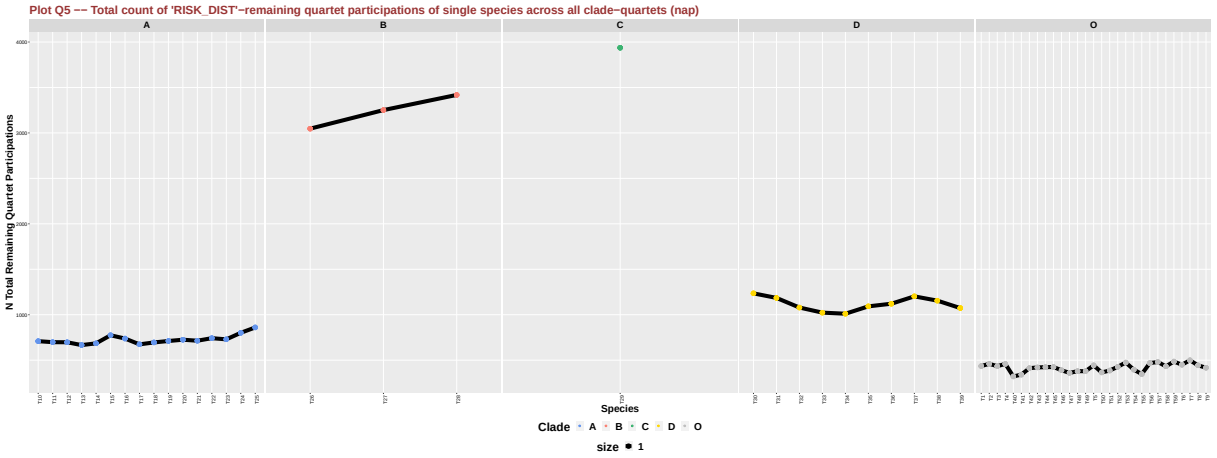

Figure 14: Example plot **Q5 II:** For each species (x-axis), the total count of filter remaining quartet participations (left y-axis), sorted based on species-assigned clades (as indicated at the top).

4.8 Overview R plot Q6 – count of analysed species-quartets

A bar chart showing the number of analyzed and rejected species-quartets for each clade-quartet in both unfiltered and filtered quartet analyses (Figure 15). Graphic-assigned information is presented in tables formatted as TXT and LaTeX with the name code LQ7, and in TSV format coded as Q6.

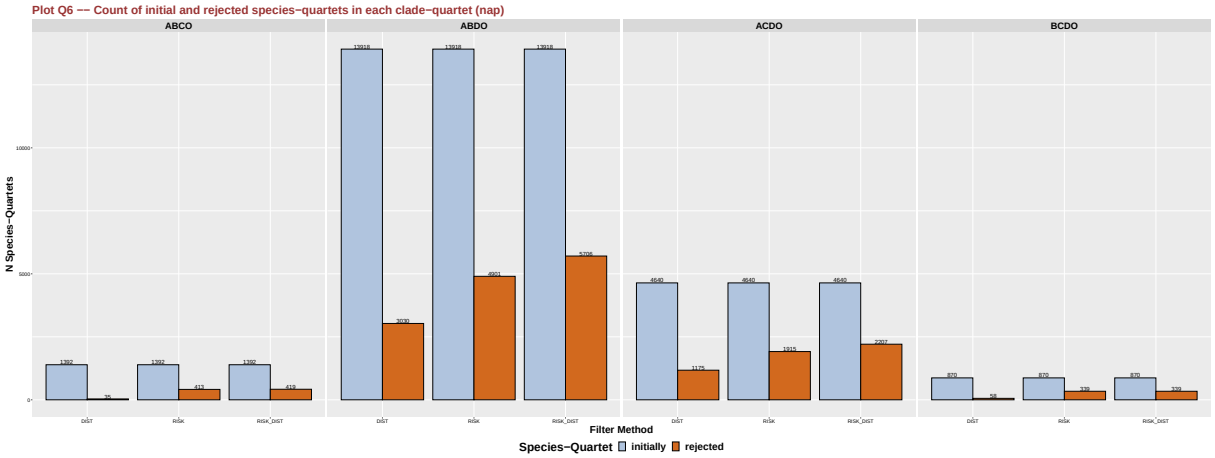

Figure 15: Example plot **Q6:** For each specific clade-quartet (as indicated at the top), the total count (N) of initially and rejected species-quartets is presented for both filtered and unfiltered support analyses (x-axis).

4.9 Overview R plot Q7 – species-quartet best-tree support to number of analysed split-pattern

A point plot illustrating the support of the best-supported clade-quartet trees for species-quartets in relation to the length of the corresponding species-quartet sequences. The analysis focuses on alignment sites devoid of missing or ambiguous character states for each species-quartet (Figure 16). Single grafics are printed for each unfiltered and filtered species-quartet analysed clade-quartet. Graphic-assigned information is presented in tables formatted in TSV format coded as Q7.

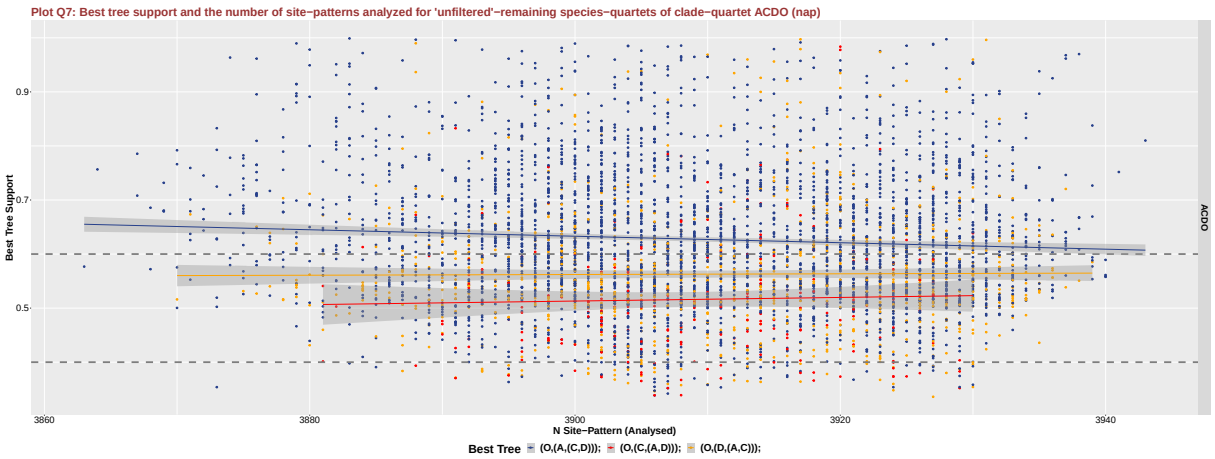

Figure 16: Example plot **Q7:** For a specified clade-quartet (as indicated on the right), the support of the best tree for individual unfiltered species-quartets (y-axis) is illustrated relative to the count of species-quartet corresponding sequence lengths after excluding sites with missing or ambiguous character states (x-axis). Best trees are represented as data points, with different trees displayed in distinct colors (see color legend at the bottom). Each regression line shows the linear fit of line representing best tree support to the number of species-quartet analysed number of site positions. The grey-shaded area surrounding each line represents the standard error.

4.10 Overview R plot R1 – best rooted-clade tree(s)

A tree plot illustrating best rooted-clade trees of each unfiltered and filtered analysis. Identical best trees of different analyses are shown by same branch colors (Figure **??**). Graphic-assigned information is presented in tables formatted as TXT and LaTeX with the name code LRC1 and LRC2, and in TSV format coded as MQ3.

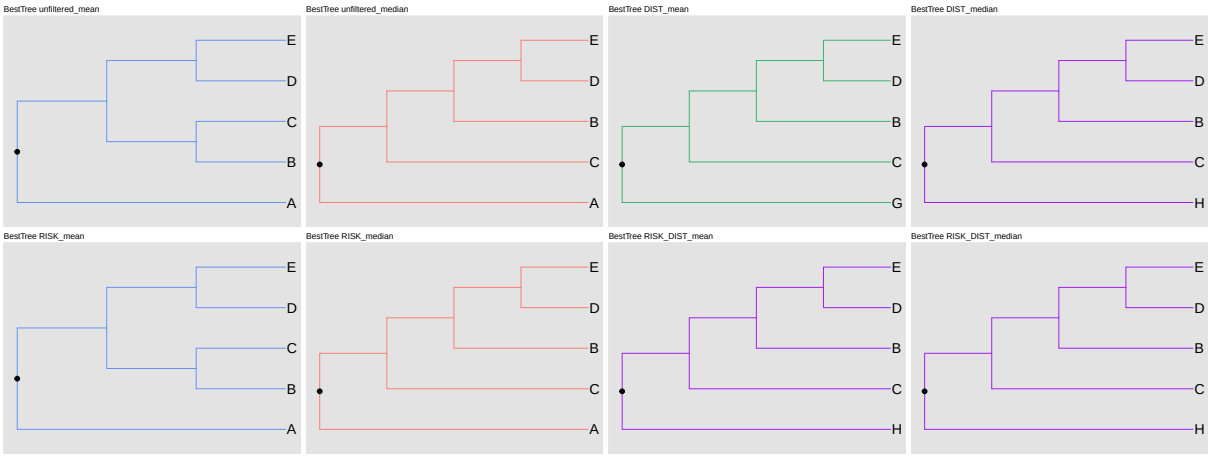

Figure 17: Example plot **R1:** Best filtered and unfiltered rooted-clade trees with similar trees highlighted by same branch colors.

4.11 Overview R plot T1 – single species support contribution to clade-quartet trees

Triangle plots illustrating the signal strength of single species-quartets in relation to the three possible clade-quartet topologies at the level of species. Each triangle corner represents the three possible relationships of a clade-quartet (top). The direction of stronger support towards one of the three triangle corners signifies a lower level of signal conflict between different topologies (Figure 18). Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information regarding mean and/or median species-quartet support and clade-quartet trees is presented in tables formatted as TSV with the name code T1. Additionally, clade-quartet trees of triangle corner labeled codes QT1, QT2, and QT3 are listed in tables formatted as TXT and LaTeX with the name code LQT1.

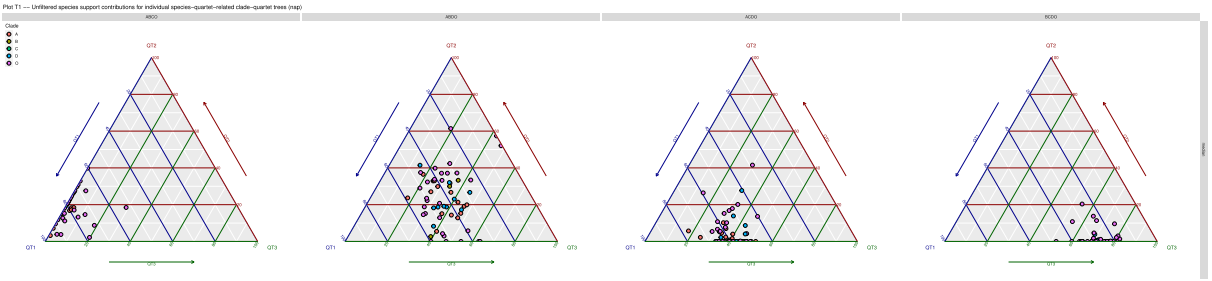

Figure 18: Example plot **T1:** Triangle graphs with averaged (mean and/or median) quartet support at the level of species. Each clade-quartet (top) is represented by a triangle. The proximity of points to corners indicates the averaged signal strength of species participating quartets in relation to the three possible clade-quartet topologies. Each species is depicted as a dot, with colors corresponding to the assigned clade (refer to the legend in the upper-left corner of the plot).

4.12 Overview R plot T2 – tree support distances contributed by individual species

Line plots displaying the average (mean and/or median) of individual species' contributions to species-quartet support and support distances for each clade-quartet related tree. The species are arranged on the x-axis based on their clade assignment (Figure 19). Single grafics are printed for filtered and unfiltered analyses. Graphic-assigned information regarding mean and/or median species-quartet support and clade-quartet trees is presented in tables formatted as TSV with the name code T2. Additionally, clade-quartet trees labeled codes QT1, QT2, and QT3 are listed in tables formatted as TXT and LaTeX with the name code LQT1.

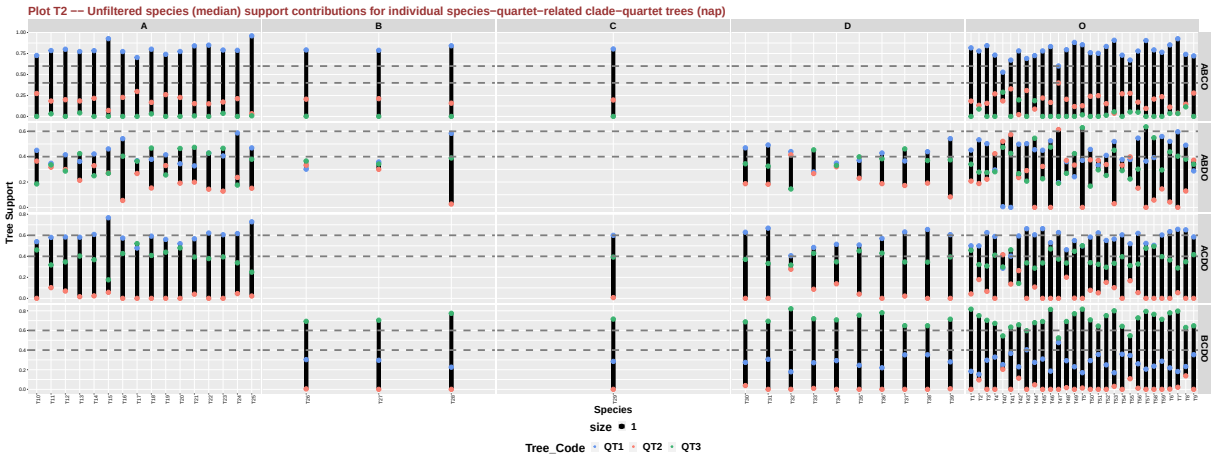

Figure 19: Example plot **T2:** Support contributions (y-axis) from individual species (x-axis) in unfiltered (see grafic header) species-quartets are shown for each of the three trees (QT1, QT2, QT3, highlighted by differently colored dots) in each clade-quartet (y-axis, right). Black vertical lines emphasize support distances between different trees. Dashed horizontal lines in each clade-quartet corresponding graph segment define ranges of low support (below upper line), moderate support (below upper line), and strong support (above upper line). The species are arranged on the x-axis based on their clade assignment (referred to at the top).

5 Example data

For a preliminary test run to ensure the smooth functioning of SeaLion on your system, particularly with respect to species-quartet analysis, P4 site-pattern inference, and extended output prints, sample input files are included with SeaLion on GitHub. The example directory, named "SeaLion_testdata_exampleManual", contains a sample alignment file ("sealion_example.fas"), a clade-definition file corresponding to the alignment ("sealion_example_cladefile.txt"), and clade-definition file pre-analysed SPD (split-pattern distribution) files summarized in a folder named "sealion_spd_imain/*".

5.1 Full processing

For a full analysis, including species-quartet split-pattern distribution analyses with P4, move to the path directory "SeaLion_testdata_exampleManual/" and type...

- `user@linux:~$ perl sealion.pl -i sealion_example.fas -p sealion_example_cladefile.txt -o 0 -M 1000 -s <enter>`

5.2 SPD file processing

For a re-analysis of already existing SPD files with the `-restart` option without including species-quartet split-pattern distribution analyses with P4, move to the path directory "SeaLion_testdata_exampleManual" and type...

- `user@linux:~$ perl sealion.pl -i sealion_example.fas -p sealion_example_cladefile.txt -o 0 -restart - imain sealion_spd_imain -M 1000 -s <enter>`

6 Trouble shooting

Usually, SeaLion should run without encountering unexpected process breaks. If there are issues with the input data or the specified parameter setup, SeaLion is designed to detect errors and provide relevant error messages. In such cases, SeaLion will display an error message and direct users to the corresponding section in the help menu for further guidance. However, there are exceptions, including error messages generated by external software processes such as R or LaTeX, or unexpected software bugs within SeaLion. In the first scenario, please verify if all script dependencies for the software package mentioned in the error message are correctly installed. If R graphic libraries are not installed properly, SeaLion may halt at the end of the process run, indicating that a specific graphic cannot be generated. By examining the filename code in the error message, it is possible to identify which R libraries may be incorrectly installed. Table 14 provides an overview of the R libraries associated with each filename code. Independent of each file assigned name-code, LaTeX tables are always based on the same five packages (see Table 15).
In case of unexpected errors or SeaLion error messages starting with "BUG-ERROR," please send an email to the respective system developer.

Table 14: Summary of individual R plot library dependencies. If any of these libraries is not correctly installed for the specific plot, the SeaLion process will be terminated at that output stage. Ensure that all required R plot libraries are installed to generate the desired visualizations.

File-Code	Plot Type	R Plot Libraries
MQ1	Lineplot	ggplot2, svglite, reshape
MQ2	Ternaryplot	ggplot2, svglite, reshape, ggtern
MQ3	Barplot	ggplot2, svglite, reshape
MQ4	Ternaryplot	ggplot2, svglite, reshape, ggtern
MQ6	Lineplot	ggplot2, svglite, reshape
R3	Treeplot	ggplot2, reshape, ggtree, gridExtra
Q3	Pointplot	ggplot2, svglite, reshape
Q5	Lineplot	ggplot2, svglite, reshape
Q6	Barplot	ggplot2, svglite, reshape
T1	Ternaryplot	ggplot2, svglite, reshape, ggtern
T2	Lineplot	ggplot2, svglite, reshape

Note: Note that older versions of the `ggtern` library may not be compatible with R version 4 or higher. If you encounter issues with printing ternary plots, ensure that your installed version of `ggtern` is compatible with R version 4 or higher.

Table 15: Summary of individual LaTeX package dependencies within texlive. If any of these packages is not correctly installed, the SeaLion process will be terminated at that output stage.

LaTeX Package	Part of Texlife-Basic
longtable	yes
table	yes
helvet	yes
pdflscape	yes
booktabs	yes

7 License/Help-Desk/Citation

SeaLion was developed by Patrick Kück in 2024 with the main script written in Perl. The additional Icebreaker file is written in C++ by Nathan I. Seidel. Both files are free software. It can be distributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation; either 2 of the license, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

If you have any problems, error-reports or other questions about SeaLion feel free and write an email to p.kueck@leibniz-lib.de which is the official help desk email account for the software. For other open source software for phylogenetic purposes visit also:

https://github.com/PatrickKueck.git

If you use SeaLion until the manuscript addressing SeaLion is published, please cite:
Kück P., Wilkinson M., Romahn J., Seidel, N.I. and Wägele J.W. (2024): SeaLion Manual, Version 1.0, Leibniz Institute for the Analysis of Biodiversity Change, Germany

References

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, **32**(5), 1792–1797.

Felsenstein, J. (1993). *PHYLIP: phylogenetic inference package*. Department of Genetics, University of Washington, Seattle, USA, version 3.5c edition.

Katoh, K. and Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol*, **30**(4), 772–780.

Katoh, K. and Toh, H. (2008). Recent developments in the MAFFT multiple sequence alignment program. *Brief Bioinform*, **9**(4), 286–298.

Katoh, K. and Toh, H. (2010). Parallelization of the MAFFT multiple sequence alignment program. *Bioinformatics*, **26**(15), 1899–1900.

Katoh, K., Kuma, K.-i., Hiroyuki, T., and Miyata, T. (2005). MAFFT version 5: Improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res*, **33**(2), 511–518.

Kück, P. and Longo, G. C. (2014). FASconCAT-G: extensive functions for multiple sequence alignment preparations concerning phylogenetic studies. *Front Zool*, **11**, 81.

Kück, P. and Meusemann, K. (2010). FASconCAT: Convenient handling of data matrices. *Mol Phylogenet Evol*, **56**, 1115–1118.

Notredame, C. (2002). Recent progresses in multiple sequence alignment: a survey. *Pharmacogenomics*, **3**(1), 1–14.

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-COFFEE: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, **302**(1), 205–217.

8 Copyright