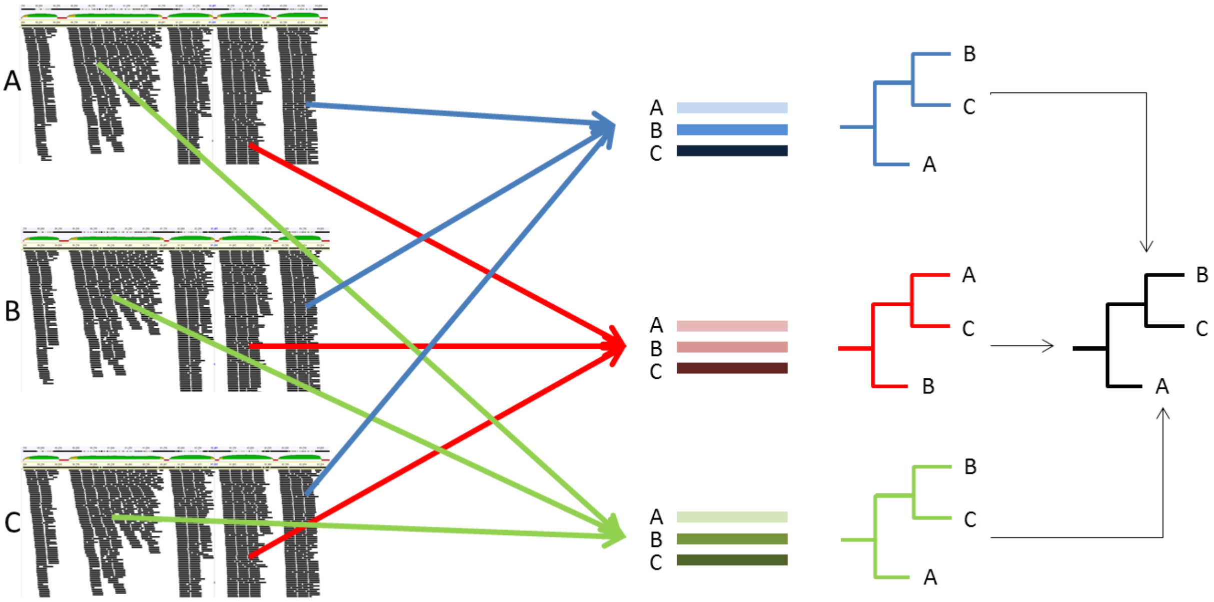
HybPhyloMaker

Pipeline for generating phylogenies based on target enriched genomic libraries (Hyb-Seq data)

https://github.com/tomas-fer/HybPhyloMaker



Version 1.8.4.

11th June 2025

Tomáš Fér

tomas.fer@natur.cuni.cz

Department of Botany
Charles University, Prague
Czech Republic

Content

Appendices

1. Introduction

HybPhyloMaker is a set of BASH scripts for UNIX-like environment that is designed for compact and easy-to-use processing of raw Illumina paired-end reads that originate from target enriched genomic libraries, selecting suitable loci and constructing gene and species trees using different methods. Most of the scripts are wrappers around high-throughput sequencing and phylogenomics software (see Appendix 2) that must be installed prior to analysis. The HybPhyloMaker pipeline is generally based on the pipeline proposed by Weitemier et al. (2014). At the beginning, reads are formatted to conform to HybPhyloMaker requirements, and all results are later saved in a synoptic folder structure. See Appendix 1 for the HybPhyloMaker workflow.

HybPhyloMaker is intended for local use on UNIX-based computer systems (it was tested on several Linux distributions, MacOS X and under WSL and Cygwin for Windows) or it can be run on a computer cluster with job scheduling. All scripts are currently optimized for the Czech National Grid Infrastructure (MetaCentrum; http://www.metacentrum.cz/en) and Smithsonian Institution High Performance Cluster Hydra (SI/HPC; https://confluence.si.edu/display/HPC), but they could easily be modified for usage in other cluster environments.

CITATION: Fér T & Schmickl RE (2018): HybPhyloMaker: target enrichment data analysis from raw reads to species trees. *Evolutionary Bioinformatics*, 14: 1–9. doi: 10.1177/1176934317742613.

DISCLAIMER: This software is provided "as is" without any warranties and support. You use it at your own risk, and we accept no responsibility for anything that may occur as a result of your use of it (or inability to use it). The user is especially responsible for reading up on program assumptions, parameters, and all the software on which HybPhyloMaker depends, and their possible influences on species tree reconstruction before starting the pipeline.

2. Preparing data and software for the analysis

Before running your analysis the following steps are usually necessary: (a) install all necessary software (see Appendix 2) and ensure that it is in PATH, (b) install all appropriate R packages (see Appendix 3), (c) create a directory, which is hereafter called `'homedir'`, (d) download all HybPhyloMaker files (including all folders) from GitHub (https://github.com/tomas-fer/HybPhyloMaker) to `'homedir'` and make scripts executable, (e) put your project-specific files to the `'HybSeqSource'` directory that is within your `'homedir'`, (f) prepare your FASTQ.gz files (two per sample) with Illumina reads in `'homedir'`, (g) edit analysis settings in the file `'settings.cfg'` in `'homedir'`.

2.1. Installation of all necessary software

Install all the software that is necessary for successfully running HybPhyloMaker (see Appendix 2) by following the instructions on the webpages of their developers. Ensure that all the software is in PATH and can be called from anywhere.

Running on GNU/Linux

If you are running HybPhyloMaker on Linux, you should consider running `'install_software.sh'` which will install, download and/or compile all the necessary software and checks whether appropriate binaries are in PATH. Open the script in a text editor and set your 'Linux distribution' and 'package management tool' (lines 15 and 17 of the script). Look through the script to see what it is going to do to prevent unwanted changes to your computer. Automatic software installation and HybPhyloMaker performance was tested on clean installations of Debian, Ubuntu, OpenSUSE, Fedora, CentOS, and Scientific Linux. The installation script will also install appropriate R packages, clone the HybPhyloMaker GitHub repository and make the scripts executable.

Running on MacOS

MacOS users are advised to use homebrew (https://brew.sh/) to install all of the necessary software. Moreover, the `coreutils` package should also be installed using `brew install --with-default-names` command. Some basic commands (cp, sed, etc.) require the GNU version instead of the version provided by MacOS. Another possibility is to use MacPorts (https://www.macports.org/). The pipeline was tested on MacOS 10.7.5 and higher, however, it is sometimes tricky (but generally possible) to install some of the required software on older system versions.

There are also numerous smaller supportive scripts and utilities written in Perl, Python, Java or R that are provided together with HybPhyloMaker within the folder `'HybSeqSource'` (see Appendix 4). These scripts are ready-to-use and need not to be installed.

IMPORTANT: You should appropriately cite these scripts/software when using HybPhyloMaker in your publications.

COMMENT: On older distribution versions some software is not properly installed. This is, however, indicated by the installation script. See comments at the end of the script `'install_software.sh'` and installation logs in the folder `'install'` how to solve known issues.

2.2. Installation of R packages

HybPhyloMaker requires R (R Core Team, 2016) in several scripts for calculating summary statistics of alignment and gene tree properties and for plotting boxplots, histograms and correlation plots, and for generating summary tables. After installing R you also need to install eight R packages: ape (Paradis et al., 2004), seqinr (Charif et al., 2007), data.table (Dowle & Srinivasan, 2017), openxlsx (Walker, 2017), phangorn (Schliep, 2011), treeio (Guangchuang, 2018), gplots (Warnes et al., 2024), and PERMANOVA (Vicente-Gonzalez & Vicente-Villardon, 2021. Refer to Appendix 3 how to do that locally or in the cluster environment. The script `'install_software.sh'` will do it for you, however, it sometimes installs older versions of R packages.

IMPORTANT: Without installation of the appropriate R packages some scripts might not work properly and some plots and tables will not be generated.

IMPORTANT: On some older Linux distributions (incl. Ubuntu 12.04 LTS and Debian 7) old version of R is automatically installed from the default repository. Sometimes this version is incompatible with some functions used in the HybPhyloMaker R scripts. Consider installation of the most recent version of R using instructions from CRAN (e.g., https://cran.r-

project.org/bin/linux/ubuntu/README.html) to ensure smooth processing of your data with HybPhyloMaker. Short advice is also given at the end of `'install_software.sh'` script.

2.3. Directory structure

HybPhyloMaker is working with a dedicated folder structure. Prepare the directory structure as depicted in Fig. 1 (i.e., copy all data from GitHub to `'homedir'` or clone the whole repository – the name of your `'homedir'` will be `'HybPhyloMaker'` in the latter case). This structure is ready if you run '`install_software.sh`'.

IMPORTANT: You also have to make all the scripts executable, e.g., by running `'chmod +x *.sh'` from your `'homedir'`. You also have to make executable ASTRID binary in `'HybSeqSource'` folder (e.g., `'chmod +x HybSeqSource/ASTRID'`). This is not necessary if you run '`install_software.sh`'.

During run each script creates its own `'work'` directory within `'homedir'`, copies all input and other necessary files to it and, after finishing calculation, copies the results back to the newly created subfolder(s) in `'datadir'`. By default, the `'work'` directory is deleted after the script finishes.

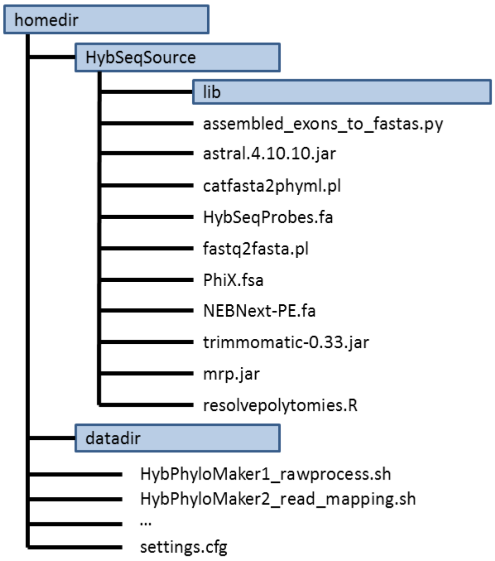


Fig.1: Folder structure before running HybPhyloMaker. Create `'homedir'` and put all BASH scripts (`'HybPhyloMakerX*.sh'`) and `'settings.cfg'` to it. In `'HybSeqSource'` folder there are all other supporting scripts, sequences of target enrichment probes, plastid coding genes, adapters and PhiX. In 'datadir' there will be all input and output files. `'homedir'` folder is called 'testdata' after running '`install_software.sh`'. **IMPORTANT:** This is just an example figure; there are more 'HybPhyloMaker*.sh' files in 'homedir' and more files in 'HybSeqSource'.

2.4. Project-specific files (to be put to `'HybSeqSource'`)

You must provide reference sequences and adaptor sequences that are specific to your project and put them to the `'HybSeqSource'` folder. The names of these files must be specified in `'settings.cfg'` (see 2.7.).

2.4.1. Sequences of target enrichment probes

A FASTA file in sequential sequence format (i.e., not interleaved, only one line per sequence) with sequences of target enrichment probes (exons) is required. This is the file with the sequences that were used for bait design in order to enrich your genomic libraries. The naming within this file must follow the scheme: `'>Assembly_geneNumber_exonNumber_whatever'` (e.g., `'>Assembly_1_Contig_1_413'`). **The word 'Assembly' is mandatory.** Files with the same 'geneNumber' will later be merged to a single file (exon concatenation). Specify the name of this file under 'probes=' in `'settings.cfg'` (see 2.7.).

2.4.2. 'Pseudoreference' for read mapping

A 'pseudoreference' is a long sequence consisting of sequences of target enrichment probes (see 2.4.1.) separated by 'nrns' Ns (number of Ns is specified in `'settings.cfg'`). The same number of Ns is also added to the beginning and to the end of this 'pseudoreference'. Sequences are separated in order to avoid read mapping to possibly unrelated but in 'pseudoreference' adjacent exons. We recommend using 400 Ns for 2×150 PE reads and 800 Ns for 2×250 PE reads. Run `'HybPhyloMaker0b_preparereference'` and a pseudoreference called `'{name_of_the_probe_file}_with{nrns}Ns_beginend.fas'` is saved to `'HybSeqSource'` folder.

2.4.3. Sequences of organellar coding genes

A FASTA file with the coding sequences of the organellar (e.g., plastid) reference needs to be provided if you want to use chloroplast reads. The naming within this file must follow the scheme: `'Number_number_geneName'` (e.g., `'>008854573_1_rps12'`). Such a file can be obtained from GenBank, e.g., by extracting coding sequences from properly annotated whole plastome record. Specify the name of this file under 'cpDNACDS=' in `'settings.cfg'`. A chloroplast 'pseudoreference' also has to be created with `'HybPhyloMaker0b_preparereference'` when you set 'cp=yes' in `'settings.cfg'` (see 4.10. for detailed information about cpDNA analysis).

2.5. File with adapters to be trimmed with Trimmomatic

A text file with adapter sequences used during library preparation needs to be provided. This file is later used to trim adapter sequences from reads using Trimmomatic (see 4.1). You might use one of the fasta-formatted adapter files distributed with Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic; TruSeq and Nextera adapters), or you can supply a file with sequences that are specific for your library preparation method. Place the file in the `'HybSeqSource'` folder and specify its name by setting 'adapterfile=' in `'settings.cfg'` (see 2.7.). `'NEBNext-PE.fa'` with adapters for NEBNext Ultra libraries is provided with HybPhyloMaker.

2.6. Input FASTQ files

Input Illumina FASTQ files need to be paired-end and gzipped. Put these FASTQ.gz files (two files per sample) to `'homedir'`. Prepare a file for automated renaming of the FASTQ.gz files. It must have two entries per line separated by TAB. The first entry is the name of the sample, which you want to give it for usage throughout the pipeline and which must follow the naming scheme: 'Genus-species_Code' (e.g., 'Curcuma-longa_S01'). The second entry is the first unique (!) part of the name of the FASTQ.gz file of this sample (e.g., 'Z1065_' if the name of the FASTQ.gz file is 'Z1065_S1_L001_R2_001.fastq.gz'). Avoid usage of '-' in the original FASTQ.gz files. Name the file 'renamelist.txt' and save it to 'homedir'.

IMPORTANT: Be sure to use unique part of the FASTQ.gz file name in order to avoid unexpected behaviour during subsequent data preparation & processing (see 4.0.).

2.7. Edit analysis settings

Open the file 'settings.cfg' that is in your 'homedir' and edit the general settings and parameter options. Here you can set whether you are going to run HybPhyloMaker locally or in a cluster environment, set 'datadir', type of data (exons or organellar DNA), tree building method and much more. See Appendix 5 for a thorough explanation of general settings, parameters and parameter options.

IMPORTANT: Carefully set/review all options before running any HybPhyloMaker script. These settings influence what will be calculated (and how) and where the results will be saved.

3. Test dataset and files

There is a small test dataset available, if you want to try HybPhyloMaker. Download the folder 'testdata' from GitHub and put it to your 'homedir'. Alternatively, you can clone the whole GitHub project, and the resulting 'HybPhyloMaker' folder will be your 'homedir' (this will the 'install_software.sh' do for you as well). The folder 'testdata' includes the subfolder '10rawreads' with six sample subfolders with two FASTQ files each. Each FASTQ file includes a random selection of 100,000 reads from the original file (phylogeny of the family Zingiberaceae; Fér et al., in prep.). Inside '10rawreads' there is also a list of samples in 'SamplesFileNames.txt'. This test dataset serves as an example how the initial data structure should look like before running HybPhyloMaker (see also Fig. 2). In the case of this test dataset there is no need to run the script 'HybPhyloMaker0a_ preparedata.sh' (see 4.0.), as the data structure is already prepared. You can now run 'HybPhyloMaker1_rawprocess.sh' (see 4.1.).

In the 'HybSeqSource' folder there is a FASTA file with sequences of the target enrichment probes called 'curcuma_HybSeqProbes_coursetest.fa'. This includes a subset of the total number of exons (i.e., the first 100 exons originating from 30 genes) that was utilized for target enrichment. The name of this file should be specified in 'probes=' in 'settings.cfg'. Use this file to generate a 'pseudoreference' for read mapping. The 'pseudoreference' has to be created by running the script 'HybPhyloMaker0b_preparereference.sh'

In the 'HybSeqSource' folder there is also a file 'CDS_Curcuma-roscoeana_plastome.txt' which was created by exporting CDS from *Curcuma roscoeana* plastome (GenBank accession NC_022928; Barrett et al., 2014). This file can be used as a reference for read mapping ('HybPhyloMaker2_readmapping.sh') and is also used for generating PSLX files when working with organellar (plastome) data using

'HybPhyloMaker3_generatepslx.sh' script. Before running you should specify 'cpDNACDS=CDS_Curcuma-roscoeana_plastome.txt' in 'settings.cfg' (see 2.6.).

4. Running the pipeline

Now you are ready to run HybPhyloMaker. The whole pipeline consists of several consecutively numbered BASH scripts that must be run in this order. The initial scripts are numbered '0' and serve for data preparation and renaming (and optional downloading from Illumina BaseSpace; script 'HybPhyloMaker0a_preparedata.sh'), 'pseudoreference' building (script 'HybPhyloMaker0b_preparereference.sh' and setting up R environment (only if you run HybPhyloMaker on MetaCentrum; 'HybPhyloMaker0c_Rsetup_MetaCentrum.sh'.

4.0. Prepare input files for analysis

HybPhyloMaker takes two gzipped FASTQ (FASTQ.gz) files per sample as input. Before running the analysis, these files must be arranged in a specific folder structure (to the folder '10rawreads' within 'datadir'), renamed to conform to pipeline standards, and a list of files must be specified (Fig. 2).

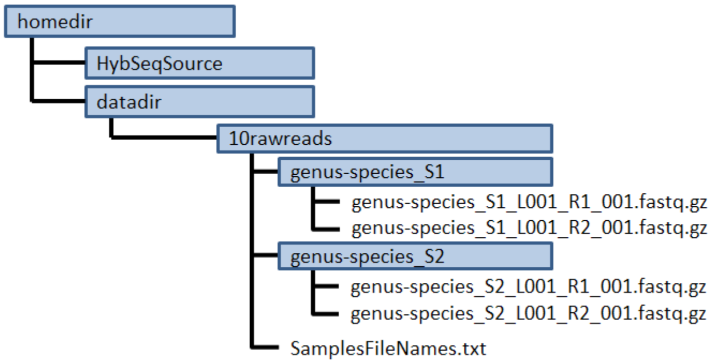

homedir
HybSeqSource
datadir
10rawreads
genus-species_S1
genus-species_S1_L001_R1_001.fastq.gz
genus-species_S1_L001_R2_001.fastq.gz
genus-species_S2
genus-species_S2_L001_R1_001.fastq.gz
genus-species_S2_L001_R2_001.fastq.gz
SamplesFileNames.txt

Fig. 2: Input data structure before running 'HybPhyloMaker1_rawprocess.sh'. All input data (two *.FASTQ.gz files per sample) must be in a specific folder structure within 'datadir/10rawreads'. Files from each sample must be in a separate folder named 'Genus-species_Code'. The names of the *.FASTQ.gz files must also follow this convention, and a list of all samples ('SamplesFileNames.txt') must be provided, which follows the same 'Genus-species_Code' naming scheme.

This could be done manually but it is recommended to put all FASTQ.gz files to 'homedir' together with 'renamelist.txt' (see 2.5.) and run the script 'HybPhyloMaker0a_preparedata.sh'. The script will accordingly rename the FASTQ.gz data, create the required folder structure, move the files to the appropriate folders and create a list of files ('SamplesFileNames.txt'). The whole input data structure is then copied to '10rawreads'. In case your data are stored in Illumina BaseSpace (https://basespace.illumina.com) you can use this script for data download. In order to access Illumina BaseSpace you need to have a personal access token, which should be saved in 'token_header.txt' in 'homedir'. Provide IDs of the Illumina BaseSpace project and run in 'settings.cfg' and do not forget to set the option 'download=yes'. Consult Appendix 6 how to obtain these IDs and how to get a personal token. The script uses BaseSpace API to downloads all the fastq.gz files from the BaseSpace project and also basic information about the run. This information is saved in the folder '00downloadinfo'. Moreover, you can also use this script to download/modify the fastq.gz files from NCBI Sequence Read Archive (SRA) or EMBL-EBI European Nucleotide Archive (ENA). In this case you need to prepare a list of SRR and/or ERR numbers (one number per line), save it as `'SRRandERRlist.txt'` in `'HybSeqSource'`, and set the option 'download=sra'. The script then downloads all pairs of fastq.gz files [either using SRA Toolkit (https://github.com/ncbi/sra-tools) and Entrez Direct (https://www.ncbi.nlm.nih.gov/books/NBK179288/) or ffq (Gálvez-Merchán et al. 2023)], rename them according organism name metadata and save them to `'data/10rawreads'`. Check script output as well as naming of the data whether everything went as expected.

4.1. Raw read filtering

Once all the data are in `'10rawreads'` you can start with the first step of data processing by running `'HybPhyloMaker1_rawprocess.sh'`. First, the scripts checks whether the structure of input data within `'10rawreads'` is correct. Second, it conducts the following operations and creates a subfolder `'20filtered'` in `'datadir'` with a subfolder for each sample:

- removal of PhiX reads: a PhiX index is created using bowtie2-build command, reads are mapped to this index utilizing Bowtie 2 (Langmead & Salzberg, 2012) and removed using SAMtools (Li et al., 2009) and bam2fastq (https://gsl.hudsonalpha.org/information/software/ bam2fastq),
- adapter trimming and quality filtering using Trimmomatic (Bolger et al., 2014),
- duplicate read removal utilizing FastUniq (Xu et al. 2012),
- creation of a summary table (`'reads_summary.txt'`) with the original number of all reads and the number of reads after each filtering step (also reporting the percentage of reads that were filtered out),
- gzipping all files with processed reads.

Each sample-specific folder in `'20filtered'` now contains six files with filtered reads:

- {name}-1P.gz – paired forward reads with duplicates,
- {name}-1P_no_dups.fastq.gz – paired forward reads without duplicates,
- {name}-2P.gz – paired reverse reads with duplicates,
- {name}-2P_no_dups.fastq.gz – paired reverse reads without duplicates,
- {name}-1U.gz – unpaired forward reads,
- {name}-2U.gz – unpaired reverse reads,

and four log files from the filtering process (Fig. 3). The important information from these files is used to make a summary table (`'reads_summary.txt'`), which is also located in `'20filtered'`. In the subfolder `'for_Geneious'` there is a tar gzipped file (`'*-all-no-dups.tar.gz'`) containing all sample files to be imported to Geneious for read mapping (optional, see 4.2.2.).

If you are on MetaCentrum, you can parallelize this filtering step by running the script 'HybPhyloMaker1a_rawprocessParallel.sh' instead. This script creates as many submission job file as you have samples and store them in `'20filtered'`. Depending on sizes of your fastq.gz files (i.e., number of read pairs) you might need to change the values

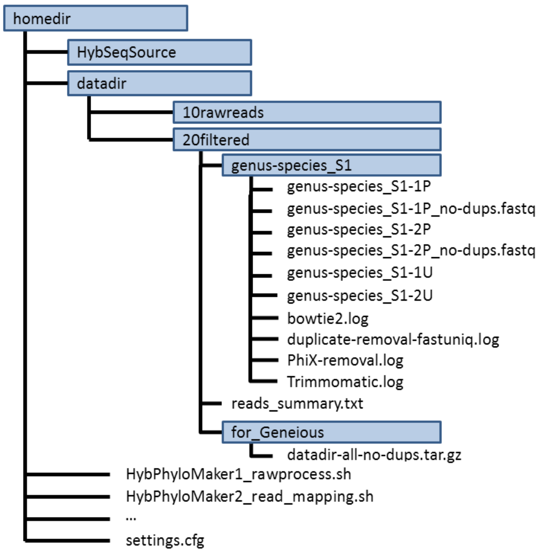


Fig. 3: Folder structure after running `'HybPhyloMaker1_rawprocess.sh'`. `'reads_summary.txt'` is the read filtering summary. A single *.tar.gz file is found in `'20filtered/for_Geneious'` ready for decompression and for import of included files to Geneious for read mapping. Or you can directly continue with `'HybPhyloMaker2_read_mapping.sh'`.

4.2. Read mapping

HybPhyloMaker uses mapping to 'pseudoreference' (i.e., reference-guided assembly) to obtain sample-specific sequence for each exon. After mapping the majority-rule consensus sequences is exported and used in subsequent steps. The user can choose between in-built mapping using script `'HybPhyloMaker2_read_mapping.sh'` [implements Bowtie 2 (Langmead & Salzberg, 2012) or BWA (Li & Durbin, 2009)] and mapping in the software Geneious (Kearse et al., 2012). We recommend considering Geneious for read mapping as Geneious seems to be more efficient than all other alternative mappers including Bowtie 2 or BWA (see http://assets.geneious.com/documentation/geneious/GeneiousReadMapper.pdf). However, the mapping within HybPhyloMaker using Bowtie2/BWA allow users to continue with the analysis from command line and without the necessity to buy a licence for Geneious. Moreover, our comparisons showed that all approaches are nearly compatible. Mapping in these three software packages were tested on several datasets and almost identical results (species trees) were obtained. However, Geneious performs best; BWA and Bowtie 2 map around 90% and 80% of reads mapped in Geneious, respectively (Supplement Table 2 in Fér & Schmickl, 2018).

4.2.1. Read mapping within HybPhyloMaker

The script `'HybPhyloMaker2_read_mapping.sh'` takes paired reads without duplicates (`{name}-1P_no_dups` and `{name}-2P_no_dups`) and upaired forward ({`name}-1U`) and reverse ({`name}-2U`) reads and maps them to the 'pseudoreference' using Bowtie 2 or BWA. The mapping option is set under 'mappingmethod=' in `'settings.cfg'`. Bowtie 2 runs with settings that are similar to the '--very-sensitive-local' preset option. The user might change these setting directly in the script. BWA (using the command 'bwa mem') runs twice with default settings: first with paired reads and second with unpaired reads. The resulting SAM file is converted to BAM file using samtools and saved to `'exons/21mapped_{mappingmethod}'` folder. The two BAM files resulting from two BWA runs are merged using samtools. Log from Bowtie2/BWA mapping is saved as `'genus-species_Code_{mappingmethod}_out.txt'`. Mapping results are summarized in the table `'mapping_summary.txt'`. Finally, a **consensus sequence is generated** from the BAM file just from mapped reads, i.e., without considering 'pseudoreference' sequence, using one of the two approaches: (1) variable **majority rule consensus** sequence using kindel (https://github.com/bede/kindel; Constantinides & Robertson, 2017) or OCOCO (Břinda et al., 2016) – in the case of two (or multiple) bases per site the base which is present in more than x% reads is called (specified as 'majrule=x'); otherwise 'N' is called; (2) **ambiguity DNA codes** (e.g., Y, R, S, M…) are called in case of multiples bases per site using ConsensusFixer (https://github.com/cbg-ethz/ConsensusFixer); the minimum relative abundance of alternate base is defined by 'pluraliy='. In case of lower than minimum coverage (specified in 'mincov=') 'N' is called (and not the reference) in both cases. The resulting sequence is saved in `'21mapped'` folder as `'genus-species_Code.fasta'`. Consensus sequences from all samples are combined into the single multi-FASTA file (`'consensus.fasta'`) and saved to `'30consensus'` folder within `'datadir/exons'`. In case the consensus was called using ConsensusFixer two additional files, `'ambigperc.txt'` and `'ambigbaseperc.txt'`, summarizing the proportion of ambiguous bases are created and saved to `'30consensus'` folder. The file `'ambigperc.txt'` summarizes (per sample) total sequence length, length without '?', number of base characters only without Ns (i.e., called bases) and number/percentage of ambiguous bases. The file `'ambigbaseperc.txt'` gives number and percentage of all ambiguous bases per sample.

IMPORTANT: Before running `'HybPhyloMaker2_read_mapping.sh'` **the 'pseudoreference must be prepared from the target enrichment probe sequences using** `'HybPhyloMaker0b_preparereference.sh'`.

COMMENT: If you wish just to do consensus calling with different above-mentioned approaches and/or with different 'mincov=' and 'majrule=' settings (and mapping is already done) set 'mapping=no', delete/rename `'30consensus'` **folder and run** `'HybPhyloMaker2_read_mapping.sh'` **again. The script will skip the mapping step and will just call the consensus sequences. Similarly you can do it if you would like to change between kindel and ConsensusFixer.**

4.2.2. Read mapping in Geneious

An alternative to read mapping described in 4.2.1. is to utilize the effective read mapper in Geneious. Untar and unzip the file `'*-all-no-dups.tar.gz'` from the folder `'20filtered/for_Geneious'` and import all these files to a new folder in Geneious. Import 'pseudoreference' (see 2.4.2.) to the same folder in Geneious.

Mark all files in this folder (e.g., Ctrl+A) and click Tools -> Align/Assemble -> Map to Reference. Use the following settings for mapping or modify them according to your needs:

a. Data
   i. Reference sequence: <*your pseudoreference*>
   ii. Assemble each sequence list separately
b. Methods
   iii. Sensitivity: Custom Sensitivity
c. Trim Sequences: Do not trim
d. Results
   iv. Save assembly report
   v. Save contigs
e. Advanced
   vi. Allow gaps: Maximum Per Read 15%
   vii. Word Length 14
   viii. Ignore words repeated more than 20 times

ix. Maximum Mismatches Per Read 30%
x. Maximum Gap Size 10
xi. Index Word Length 12
xii. Maximum Ambiguity 4

When the mapping is done (this can take up to several hours) select all files with mapping to the ‘pseudoreference’ (marked by three red oblique lines in Geneious and called ‘genus-species_nr-all-no-dups assembled to …’) and File -> Export -> Consensus sequence(s):

i. Threshold: 0% - Majority
ii. Do not select ‘Ignore Gaps’
iii. If No Coverage Call: ?
iv. Do not select ‘Trim to reference sequence’
v. Append text to name of alignment: ‘_consensus_sequence’
vi. After OK… Create sequence list
vii. Save as `'consensus.fasta'`

This `'consensus.fasta'` contains as many FASTA records as is number of your samples. Each sequence is a 0% majority rule consensus of reads mapped to the ‘pseudoreference’ and is roughly of the same length as the ‘pseudoreference’. Sequences of individual exons are separated by strings of ‘?’ due to several hundreds of Ns between each exon in the ‘pseudoreference’.

Create a directory `'30consensus'` in `'datadir/exons'` (the `'exons'` subfolder must be also created) and put `'consensus.fasta'` there (Fig. 4).

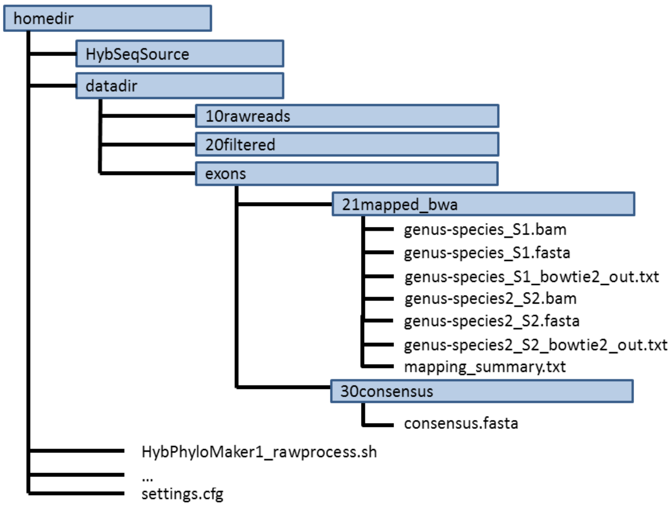


Fig. 4: Folder structure after running `'HybPhyloMaker2_read_mapping.sh'`. In this case the mapping was done using BWA.

4.3. Processing consensus sequences

Run the script `'HybPhyloMaker3_generatepslx.sh'` after putting `'consensus.fasta'` to the `'exons/30consensus'` subfolder (only in case of you were mapping reads in Geneious). The script takes this consensus sequence multi-FASTA file (produced by `'HybPhyloMaker2_read_mapping.sh'` or exported from Geneious) and does the following:

- splits it into individual files (one file per sample),
- each individual file is split into smaller pieces corresponding to the exons (strings of ‘?’ are replaced by a newline character) and saved to the subfolder `'exons/40contigs'` with the name `'genus-species_Code_contigs.fas'`,

- each sequence in `'*_contigs.fas'` is then compared to the original exon sequences (from the target enrichment probes file) using BLAT (Kent, 2002) and the results are saved as PSLX file in the subfolder `'exons/50pslx'`.

When searching for similarity between consensus and probe sequences with BLAT the minimum similarity threshold ('minident=' in `'settings.cfg'`) highly influences the number of similarity hits. The default value is 90 but it can be lowered to 85 or 80 in analyses of distantly related species (at the level of a whole family or even order; Carlsen et al., 2018).

COMMENT: If consensus sequences were called using ConsensusFixer (and ambiguous codes are present) the resulting PSLX generated by BLAT will contain only Ns instead of these original ambiguities. However, they are automatically modified to get the ambiguities back within the script `'HybPhyloMaker3_generatepslx.sh'`.

IMPORTANT: Before continuing with the next step all PSLX files need to be copied to a folder within `'homedir'` **and the name of this folder should be specified in** `'settings.cfg'` **('otherpslx='). In this step you can combine PSLX files from multiple analyses or subselect samples and continue with the analysis based on the desired samples only. Example:** `mkdir pslx_to_combine && cp testdata/exons/50pslx/* pslx_to_combine`

It is possible to "mine" other data sources (transcriptomes, genome CDS or whole genomes) for sequences similar to the targeted exons. Save these FASTA-formatted sequences (important: follow the file naming convention 'gene-species_Code' and add a suffix *.fas, e.g., `'Curcuma-longa_JQCX.fas'`) in a new subfolder in `'homedir'` and specify the name of the new subfolder in `'settings.cfg'` ('othersource='). The sequences from other data sources will be processed in the same way as Hyb-Seq samples. Never leave the option 'othersource=' empty; if you do not intend to use other data sources write 'othersource=NO'.

4.4.a Creating gene alignments

The script `'HybPhyloMaker4a_processpslx.sh'` takes all PSLX files that are saved in the subfolder specified under the option 'otherpslx=' (in `'settings.cfg'`), e.g., 'otherpslx=pslx_to_combine' and processes them (Fig. 5):

- the consensus sequences of the same exon from each sample are combined to a single multi-FASTA file using the Python script `'assembled_exons_to_fastas.py'` (Weitemier et al., 2014),
- all FASTA files are aligned with MAFFT using the default option; if 'parallelmafft=yes' the alignment process is passed through the GNU parallel command (Tange, 2011); the MAFFT alignments are saved in the subfolder `'exons/60mafft'` in `'datadir'`,
- exon alignments belonging to the same gene (this is specified in the exon name in the target enrichment probes file, e.g., '>Assembly_1_Contig_1_413' and '>Assembly_1_Contig_3_608' are parts of the same gene 'Assembly_1') are then concatenated using AMAS (Borowiec, 2016) and saved in the subfolder `'exons/70concatenated_ exon_alignments'` in `'datadir'`. Each 'Assembly' is saved in both *.fasta and *.phylip format.

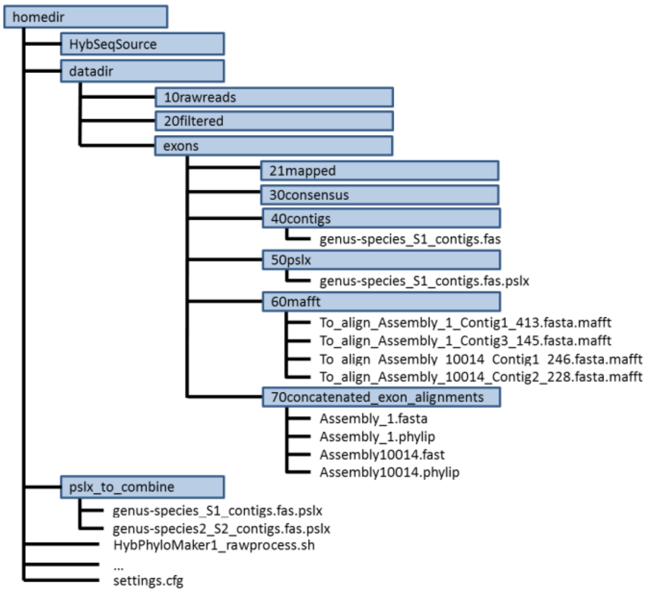

homedir
HybSeqSource
datadir
10rawreads
20filtered
exons
21mapped
30consensus
40contigs
genus-species_S1_contigs.fas
50pslx
genus-species_S1_contigs.fas.pslx
60mafft
To_align_Assembly_1_Contig1_413.fasta.mafft
To_align_Assembly_1_Contig3_145.fasta.mafft
To align Assembly 10014 Contig1 246.fasta.mafft
To_align_Assembly_10014_Contig2_228.fasta.mafft
70concatenated_exon_alignments
Assembly_1.fasta
Assembly_1.phylip
Assembly10014.fast
Assembly10014.phylip
pslx_to_combine
genus-species_S1_contigs.fas.pslx
genus-species2_S2_contigs.fas.pslx
HybPhyloMaker1_rawprocess.sh
...
settings.cfg

Fig. 5: Folder structure after running `'HybPhyloMaker4_processpslx.sh'` (after processing the consensus sequences, generating PSLX files, aligning exon sequences and concatenating exons to genes). From now all results are written to the subfolder `'exons'` (unless you specify 'cp=yes' in `'settings.cfg'` – in this case a subfolder `'cp'` is created and all the results are saved there; see 4.10.). Moreover, before running this script PSLX files have to be manually moved to newly created folder `'pslx_to_combine'` which must be specified under 'otherpslx=' in `'settings.cfg'`.

4.4.a2 Selecting low-heterozygosity loci

In case the probes do not really target single-copy genes or the mapping options are less stringent the mapping to the 'pseudoreference' could result in (many) heterozygous sites. Loci with many heterozygous sites could actually be a result of paralogous loci mapping to the same locus. In this case usually the 'correct' sequence (homologous to the probe) is obtained using majority rule consensus (orthologous reads are dominating in the mapping; Supplement Figure 1 in Fér & Schmickl, 2018). However, if this is not the case one could obtain a paralogous or chimeric sequence which will probably behave strange during phylogeny reconstruction. **This script is optional; if you do not intend to filter loci for excessive heterozygosity (which might indicate paralogy) you can skip this part and continue with section 4.5).**

Using a script `'HybPhyloMaker4a2_selectNonHet.sh'` you can filter out the exons that have a maximum specified number of heterozygous sites and thus address only the variability within putative orthologues. Select 'nohetcalculation=yes' and set maximum number of heterozygous sites allowed per exon ('maxhet=') in `'settings.cfg'` before running the script `'HybPhyloMaker4a2_selectNonHet.sh'`. It creates the folder `'70concatenated_exon_alignments_NoHet'` which includes the subfolder `'summary'` with the file `'summary.txt'` that contains for each exon: its name, start and stop position in the 'pseudoreference', a number of heterozygous sites in each sample, and sum and maximum number of heterozygous sites over all samples. The script uses SAMTOOLS and BCFTOOLS to call variants from BAM files and extract only variants with heterozygous genotype call (i.e., 0/1 or 1/2; only variant with mapping quality higher than 36 are considered). Then it calculates number of selected variants per each exon.

Based on the `'summary.txt'` a list of loci in which the number of heterozygous sites does not exceeds {maxhet} in any sample is created (`'exonsWithoutHet${maxhet}.txt'`) and saved to `'70concatenated_exon_alignments_NoHet/${maxhet}'`. Based on this list exon alignments from `'exons/60mafft'` are selected, exon alignments belonging to the same gene are concatenated and saved to the subfolder `'exons/70concatenated_` exon_alignments_NoHet/${maxhet}'. These alignments are ready to use when continuing with the pipeline.

If you want to generate another set of genes with different value of {maxhet} and the calculation of number of heterozygous sites per exon and sample was already done (and 'summary.txt' already exists) you can just select 'nohetcalculation=no' and change the 'maxhet=' value. Another subfolder '70concatenated_exon_alignments_NoHet/ ${maxhet}' is created without the need to calculate number of heterozygous sites again.

IMPORTANT: Before continuing with the next steps the resulting files from the folder 'exons/70concatenated_ exon_alignments_NoHet/${maxhet}' **must be copied to** 'exons/70concatenated_ exon_alignments'**. If this folder already exists it has to be deleted or renamed before copying the files.**

4.4.b Reading frame correction for gene alignments

The script 'HybPhyloMaker4b_correctframe_translate.sh' takes exon alignments in 'exons/60mafft' and sets them to the correct reading frame. **This script is optional; if you do not intend to set the correct reading frame in your alignments you can skip this part and continue with section 4.5).** The script successively translates each nucleotide alignment to amino acids with all three reading frames (using the command 'transeq' from EMBOSS; Rice et al., 2000). The number of introduced stop codons is recorded for each reading frame. If there is just one possible translation with zero stop codons this reading frame is treated as correct. In case there are more than one translations with zero stop codons this exons is not included in further analysis because the correct reading frame can't be assessed. If there is no translation with zero stop codons (i.e., all three possible reading frames introduced some stop codons) the exon is either not considered for further analysis or the translation with the lowest number of stop codons is accepted if it is below a specified threshold value ('maxstop=' in 'settings.cfg'). This allows the user accept also exons with a few stop codons introduced, e.g., by errors induced via sequencing or read mapping. Those stop codons are converted to Ns in both nucleotide and amino acid alignments. Furthermore, incomplete triplets are removed from both the beginning and the end of the alignment, i.e., all the alignments are set to frame 1 and their length is divisible by 3. The following folders and files are produced:

- folder '61mafft_corrected' includes exon nucleotide alignments in the corrected reading frame and trimmed to complete triplets
- folder '62mafft_translated' includes exon amino acid alignments and following exon lists and summary table:
    - 'selected_exons.txt' – list of exons selected for further analyses (to be concatenated)
    - 'removed_lowest_number_of_stop_codons_exceeded_maxstop.txt' – list of exons that were removed due to too many stop codons in the entire alignment (translation with each reading frame produced number of stop codons exceeding 'maxstop' in 'settings.cfg')
    - 'removed_more_than_1_possible_reading_frame.txt' – list of exons that were removed because translation with more than one reading frame returned alignment with zero stop codons

  - ‘stop_codons_by_frame.txt’ – summary table showing for each exon its name, number of stop codons after translation with each reading frame (3 values), lowest number of stop codons (lowest number out of those three values), and number of reading frame translations producing zero stop codons. This table is used to produce above mentioned lists.

- `‘80concatenated_exon_alignments_corrected’` includes concatenated nucleotide alignments from the same gene. Four file are generated for each gene:
  - `‘CorrectedAssembly_{nr}.fasta’` – concatenated gene alignment in fasta format
  - `‘CorrectedAssembly_{nr}.phylip’` – concatenated gene alignment in phylip format
  - `‘CorrectedAssembly_{nr}.part’` – partition file (per exon partitioning)
  - `‘CorrectedAssembly_{nr}.codonpart.file’` – partition file (per codon and per exon partitioning)
- `‘90concatenated_exon_alignments_translated’` includes concatenated amino acid alignments from the same gene (in fasta and phylip format) and a partition file

IMPORTANT: if you want to continue the work with alignments in corrected reading frame you have to set ‘corrected=yes’ in `‘settings.cfg’`.

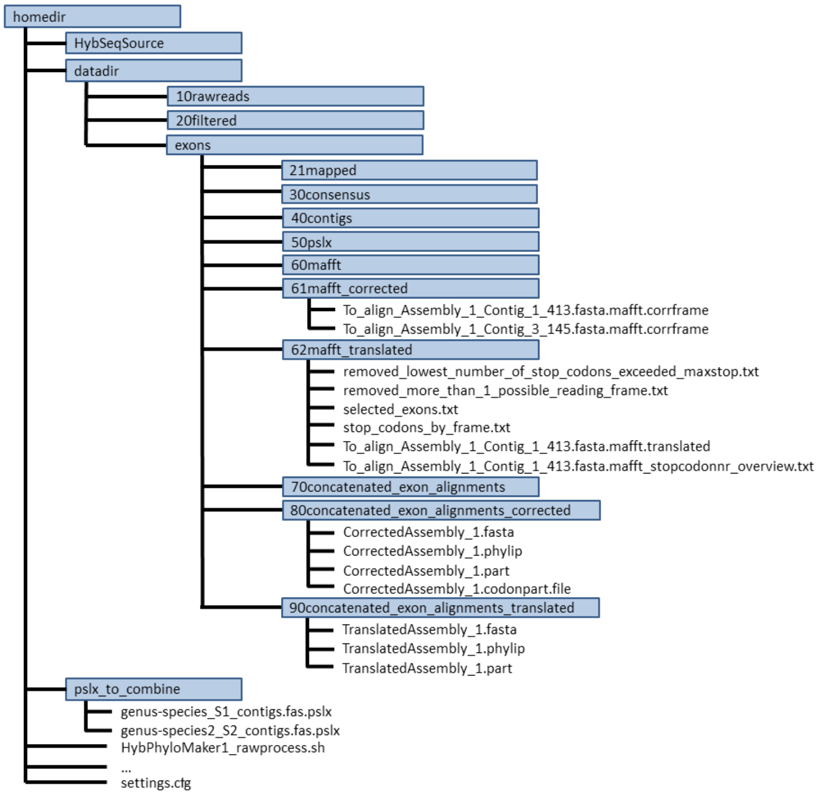

homedir
HybSeqSource
datadir
10rawreads
20filtered
exons
21mapped
30consensus
40contigs
50pslx
60mafft
61mafft_corrected
To_align_Assembly_1_Contig_1_413.fasta.mafft.corrframe
To_align_Assembly_1_Contig_3_145.fasta.mafft.corrframe
62mafft_translated
removed_lowest_number_of_stop_codons_exceeded_maxstop.txt
removed_more_than_1_possible_reading_frame.txt
selected_exons.txt
stop_codons_by_frame.txt
To_align_Assembly_1_Contig_1_413.fasta.mafft.translated
To_align_Assembly_1_Contig_1_413.fasta.mafft_stopcodonnr_overview.txt
70concatenated_exon_alignments
80concatenated_exon_alignments_corrected
CorrectedAssembly_1.fasta
CorrectedAssembly_1.phylip
CorrectedAssembly_1.part
CorrectedAssembly_1.codonpart.file
90concatenated_exon_alignments_translated
TranslatedAssembly_1.fasta
TranslatedAssembly_1.phylip
TranslatedAssembly_1.part
pslx_to_combine
genus-species_S1_contigs.fas.pslx
genus-species2_S2_contigs.fas.pslx
HybPhyloMaker1_rawprocess.sh
...
settings.cfg

Fig. 6: Folder structure after running `‘HybPhyloMaker4b_correctframe_translate.sh’` (after setting alignments to the correct reading frame and translating into amino acids). From now the pipeline works either with ‘uncorrected’ or ‘corrected’ alignments according to ‘corrected=yes/no’ in `‘settings.cfg’`.

4.5. Deleting sequences and genes with too much missing data

Missing data can largely influence phylogenetic analyses, and samples with an excessive amount of missing data should be deleted from further analyses. In HybPhyloMaker there are two levels how you can filter samples and genes based on the amount of missing data. First, sequences of a sample with more than a certain percentage of missing data per gene (‘MISSINGPERCENT=’ in ‘settings.cfg’) will be deleted from a gene alignment. Second, the number of samples per gene alignment that is left after this first step of missing data removal is calculated and only genes with more than the specified percentage of samples per gene (‘SPECIESPRESENCE=’ in ‘settings.cfg’) are retained.

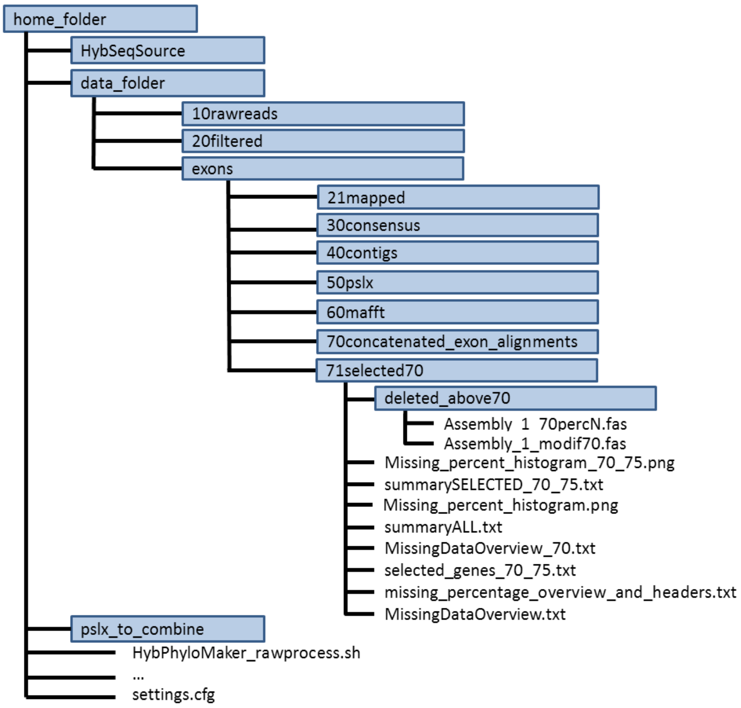


Fig. 7: Folder structure after running ‘HybPhyloMaker5_missingdataremoval.sh’ (after counting the amount of missing data and deleting sequences and genes samples with more missing data than defined in ‘MISSINGPERCENT’ and ‘SPECIESPRESENSE’). Genes selected for subsequent analyses are listed in ‘selected_genes_MISSINGPERCENT_ SPECIESPRESENCE.txt’. Histograms for other alignment characteristics are also created (*.png pictures).

Edit both above mentioned missing data parameter options and run the script ‘HybPhyloMaker5_missingdataremoval.sh’ that loops over all gene alignments in the subfolder ‘exons/70concatenated_exon_alignments’ and conducts the following analyses and saves all results in the folder ‘exons/71selected’, whose name also contains the first number specified in ‘MISSINGPERCENT’ (e.g., ‘71selected70’; Fig. 7):

- The amount of missing data per species in each gene alignment is calculated and alignments without samples with excessive missing data are saved in the subfolder ‘exons/71selectedMISSINGPERCENT/deleted_aboveMISSINGPERCENT’. The alignments are named ‘Assembly_number_modifMISSINGPERCENT.fas’, percentage of missing data per sample can be found in ‘Assembly_number _MISSINGPERCENTpercN.fas’.
- Three tables summarizing the amount of missing data per sample and gene are generated:
    - ‘missing_percentage_overview_and_headers.txt’ – species in rows, genes in columns.

  - ‘MissingDataOverview.txt’ – genes in rows, species in columns. Two more columns are added to the end of the table – average missing data across all genes of each sample and number of samples with completely missing data in a particular gene.
  - ‘MissingDataOverview_MISSINGPERCENT.txt’ – genes in rows, species in columns, but all values higher than ‘MISSINGPERCENT’ are replaced by ‘N/A’. Two more columns are added to the end of the table – average missing data across all genes of each sample (but now calculated only from values below ‘MISSINGPERCENT’) and percentage of samples with less than ‘MISSINGPERCENT’ missing data in a particular gene (i.e., percentage of values that were not replaced by ‘N/A’).

- Based on the percentage of samples left in each gene (last column in ‘MissingDataOverview_MISSINGPERCENT.txt’), the list of genes with more than the specified minimum percentage of all samples per gene (‘SPECIESPRESENCE’) is saved to ’selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt’. This list is used in the following step of gene tree reconstruction.

- Tables with summary statistics of alignment properties for all (‘summaryALL.txt’) and selected (‘summarySELECTED_MISSINGPERCENT_SPECIESPRESENCE.txt’) genes are generated using AMAS (Borowiec, 2016), MstatX (https://github.com/gcollet/MstatX), and TrimAl (Capella-Gutiérrez et al., 2009). These tables include (amongst others) the following characteristics of each gene: number of taxa, alignment length, proportion of variable sites, proportion of parsimony informative sites, GC content, alignment entropy and conservation distribution.

- Mixed histogram/boxplot diagrams (in *.png format) are generated for selected alignment characteristics for both all and selected genes using ‘alignmentSummary.R’ in R (see Fig. 8 for an example). These plots allow an easy evaluation of the distribution of these properties across genes and give a support for potential elimination of outlier loci (see 4.9.).

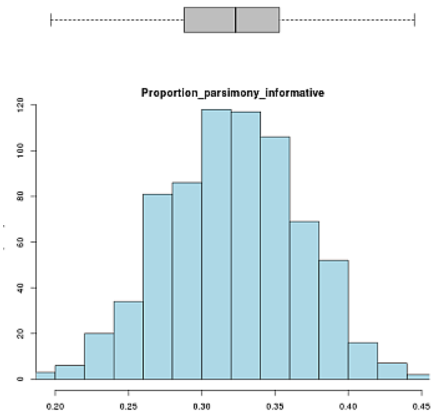


Fig. 8: Boxplot and histogram of the proportion of parsimony informative characters per gene alignment calculated with AMAS and plotted using R.

You can run gene selection several times with different settings of ‘MISSINGPERCENT’ and ‘SPECIESPRESENCE’ and several folders that contain the above described files will be created. In order to continue in the pipeline after performing a concrete gene selection based on a specific amount of missing data, just enter your desired parameter options for missing data in `'settings.cfg'`. Continue with gene tree reconstruction.

IMPORTANT: You cannot continue with the pipeline before you do this missing data-based gene selection, which produces a list of selected genes for subsequent gene tree building.

IMPORTANT: If you work with data in corrected reading frame (see 4.4.b) you have to set 'corrected=yes' in `'settings.cfg'`. In this case the data from `'exons/80concatenated_exon_alignments_corrected'` are considered and the results are saved in `'exons/81selected_correctedMISSINGPERCENT'`.

4.6. Generate gene trees for selected loci

Phylogenetic trees for alignments specified in `'selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt'` in the subfolder `'exons/71selectedMISSINGPERCENT'` are now generated using one of the three approaches:

- FastTree (Price et al., 2010) with local branch support values (Shimodaira-Hasegawa (SH)-like support) – very fast approach even on large datasets. However, these support values tends to be higher compared to bootstrapping approach (see, e.g., Guindon et al. 2010).
- FastTree with bootstrapping. First, 100 bootstrap replicates are generated for each gene using RAxML; then FastTree is applied to each of the replicates and the presence of groups in bootstrap trees is mapped onto the tree based on original alignment.
- RAxML (Stamatakis, 2014) with variable number of standard/rapid bootstrap replicates and with GAMMA or CAT model. Bootstopping option is also available. Computationally demanding approach. [Note: Combination of CAT model approximation with rapid bootstrapping should be avoided, see Simmons & Norton (2014)].

In case RAxML is used for gene tree building one of the partitioning schemes is utilized (according to 'genetreepart' in `'settings.cfg'`):

- none – trees are produced without partitioning
- exon – by exon partitioning is used
- codon – by exon and codon partitioning is used (this is only possible for data with corrected reading frame, see 4.4.b)

Select the tree building method by editing the 'tree=' option in `'settings.cfg'` (either 'FastTree' or 'RAxML') and in case of 'tree=FastTree' choose whether to use bootstrapping ('FastTreeBoot=yes' in case of bootstrapping). However, bootstrapping substantially increases the time necessary for tree building. In case of 'tree=RAxML' set also the evolutionary model ('model='), type of generating bootstrap replicates ('raxmlboot='), number of bootstrap replicates ('bsrep='), or whether to use a bootstopping ('bootstop='). Setting bootstopping to 'bootstop=yes' automatically generates as many bootstrap replicates as necessary to converge and previously set 'bsrep=' is ignored. MRE is used as bootstrap convergence criterion with cut-off threshold set to 0.03. Also select the partitioning scheme by editing 'genetreepart='. In case of 'FastTree' (running the script `'HybPhyloMaker6b_FastTree_for_selected.sh'`) the gene trees are constructed one by one, the 'RAxML' option (running `'HybPhyloMaker6a_RAxML_for_selected.sh'`) allows to generate several jobs for a subset of alignments (only if run on a computer cluster; use the option 'parallelraxml=yes'). If RAxML is run locally, the trees are also produced one by one and the whole computation might take very long, especially with a higher number of genes/samples (several hundreds and more). All trees are stored in the subfolder `'exons/72treesMISSINGPERCENT_`

SPECIESPRESENCE', where 'FastTree' or 'RAxML' subfolders are created (Fig. 9). In case of 'RAxML' five files per gene are created:

- 'RAxML_bestTree.Assembly_name_modifMISSINGPERCENT.result' – best ML tree,
- 'RAxML_bipartitions.Assembly_name_modifMISSINGPERCENT.result' – best ML tree with bootstrap values (this tree is later used for subsequent species tree reconstructions),
- 'RAxML_bipartitionsBranchLabels.Assembly_name_modifMISSINGPERCENT.result' – best ML tree with bootstrap values as branch labels,
- 'RAxML_bootstrap.Assembly_name_modifMISSINGPERCENT.result' – all bootstrap trees,
- 'RAxML_info.Assembly_name_modifMISSINGPERCENT.result' – information to the analysis.

There is also a 'bootstop_summary.txt' file generated showing how many bootstrap replicates were done before the convergence was achieved (if 'HybPhyloMaker6a_RAxML_for_selected.sh' was run in a cluster and several jobs were automatically submitted there are multiple 'bootstop_summary_{JOBname}.txt' files which are later summarized by running 'HybPhyloMaker6a2_RAxML_trees_summary.sh' (see below).

In case of 'FastTree' up to three files per gene are created (the last two files are created only if bootstrapping is requested):

- 'Assembly_geneName_modifMISSINGPERCENT.fast.tre' – tree with local support values,
- 'Assembly_geneName_modifMISSINGPERCENT.boot.fast.tre' – tree with bootstrap support values,
- 'Assembly_geneName_modifMISSINGPERCENT.boot.fast.trees' – all bootstrap trees.

The outputs of 'RAxML' and 'FastTree' runs are redirected to logfiles ('raxml.log', 'FastTree.log', and 'FastTreeBoot.log').

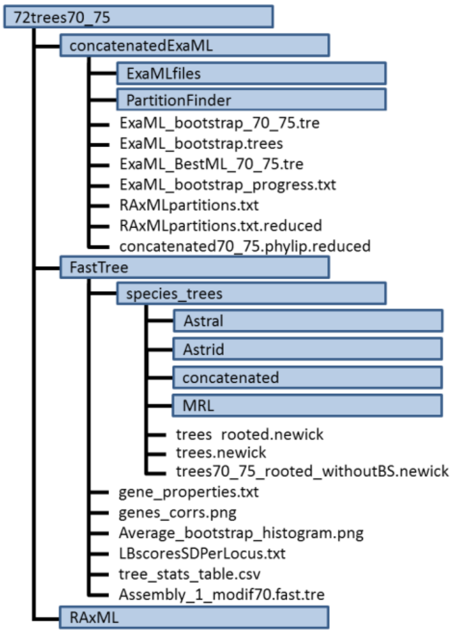


Fig. 9: Structure of the subfolder `'72trees'` for 'MISSINGPERCENT=70' and 'SPECIESPRESENCE=75'. The structure of the `'RAxML'` subfolder is similar to that of the `'FastTree'` subfolder. The final species trees are in individual subfolders (e.g., `'Astral'`) and not shown. If you are working with alignments in corrected reading frame ('corrected=yes') all the data are in the folder `'82trees_corrected'`.

Several summary statistics of properties are calculated based on the gene trees, saved in `'tree_stats_table.csv'` and visualized using mixed histogram/boxplot diagrams with the custom R scripts `'tree_props.r'` modified from https://github.com/marekborowiec/good_genes and `'treepropsPlot.r'` (with the packages 'ape' and 'seqinr'). In case of RAxML gene trees the calculation of these statistics is implemented in the separate script `'HybPhyloMaker6a2_RAxML_trees_summary.sh'` (this script also summarizes all job-specific `'bootstop_summary_{JOBname}.txt'` files into a single `'bootstop_summary.txt'` file), in case of FastTree gene tree reconstruction calculation of the summary statistics is implemented in the same script, `'HybPhyloMaker6b_FastTree_for_selected.sh'`. In the following the gene tree characteristics are listed:

- average bootstrap support,
- average branch length,
- average uncorrected p-distance,
- clocklikeness (a measure how close to ultrametric a tree is: the algorithm finds a root that minimizes the coefficient of variation in root to tip distances and returns that value; a lower value is more clock-like, an ultrametric tree has a score of 0),
- simple linear regression on uncorrected p-distances against inferred distances, i.e., branch length (slope and R^2; higher values mean lower saturation potential),
- long-branch score (standard deviation from the taxon-specific long branch score defined by Struck, 2014).

Alignment and gene tree properties are combined to a single file (`'gene_properties.txt'`) and correlations among all pairs of selected characteristics are computed and plotted to `'genes_corrs.png'` using the R script `'plotting_correlations.R'` (modified from https://github.com/marekborowiec/good_genes; Fig. 10). This helps to recognize genes with extreme values of particular alignment or gene tree characteristics (e.g., saturated genes), and the summary table (`'gene_properties.txt'`) helps to distinguish among, e.g., slowly and quickly evolving genes or less and more variable genes and select specific genes for subsequent phylogenetic analyses (see 4.9.). Screen outputs of all `R` runs are redirected to `'R.log'`.

IMPORTANT: If you work with data in corrected reading frame (see 4.4.b) you have to set 'corrected=yes' in `'settings.cfg'`. In this case the alignments from `'exons/81selected_corrected/deleted_aboveMISSINGPERCENT'` are considered and the trees are saved in `'exons/82trees_correctedMISSINGPERCENT_SPECIESPRESENCE'`.

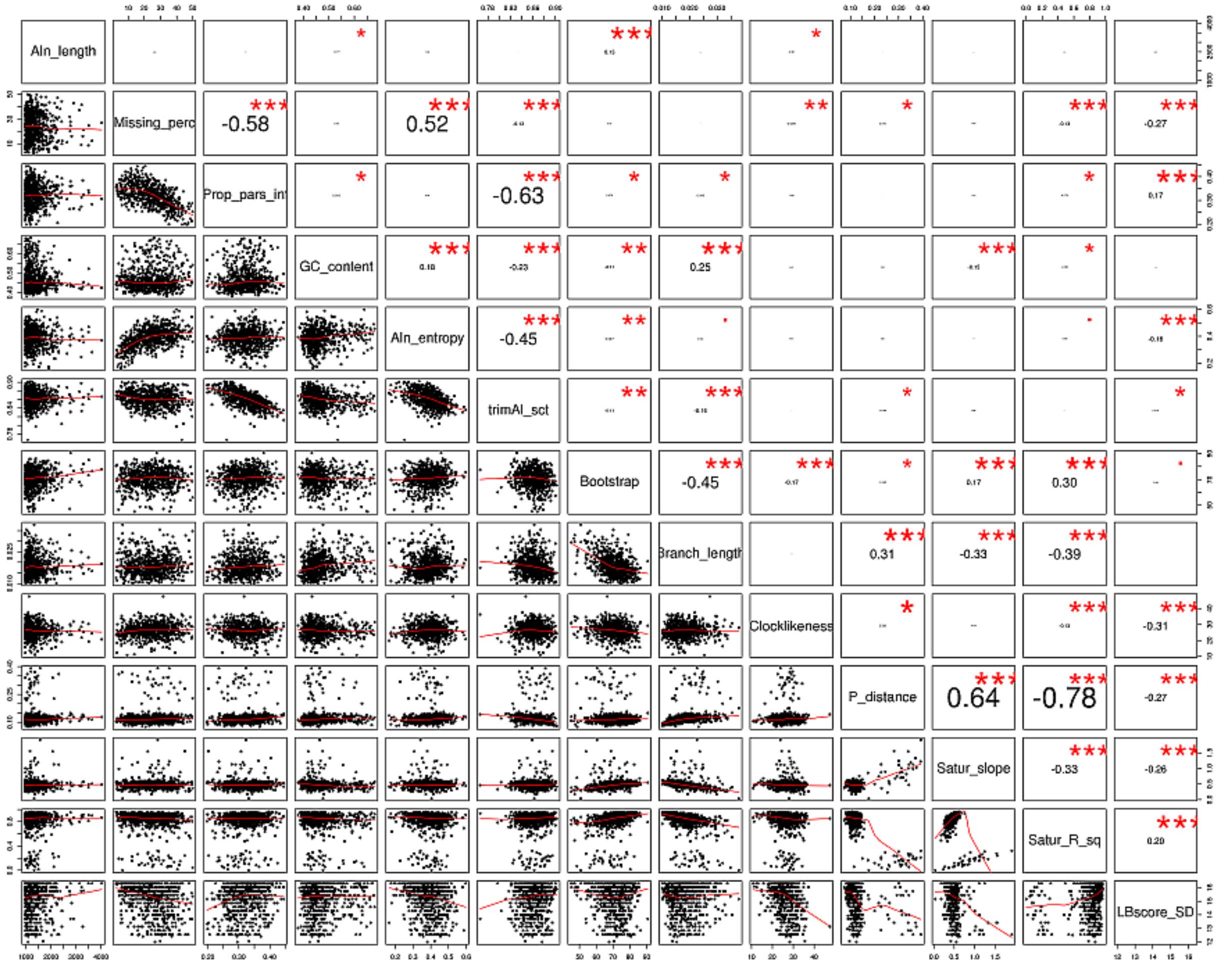


Fig. 10: Correlations among alignment and gene tree properties obtained by running `'HybPhyloMaker6b_FastTree_for_selected.sh'` respective `'HybPhyloMaker6a2_RAxML_trees_summary.sh'`.

4.7. Root and combine gene trees

By running the script `'HybPhyloMaker7_roottrees.sh'` all gene trees (produced either with 'FastTree' or 'RAxML') are combined into a single multi-NEWICK file that is later used for species tree estimation (see 4.8.). Optionally, trees are rooted (define a root with 'OUTGROUP=' in `'settings.cfg'`; trees will not be rooted if this option is empty) and bootstrap support values are removed using Newick Utilities. The following files are produced:

- `'trees.newick'` – all trees,
- `'trees_rooted.newick'` – all trees rooted with the outgroup using the 'nw_reroot' command (only if 'OUTGROUP=' is not empty),
- `'treesMISSINGPERCENT_SPECIESPRESENCE_rooted_withoutBS.newick'` – all rooted trees with bootstrap support values removed utilizing the 'nw_topology' command (only if 'OUTGROUP=' is not empty),
- `'treesMISSINGPERCENT_SPECIESPRESENCE_withoutBS.newick'` – all trees with bootstrap support values removed utilizing the 'nw_topology' command (only if 'OUTGROUP=' is not specified).

The script reports if all gene trees were rooted or how many gene trees were not rooted (all gene trees might not include specified outgroup taxon). Many species tree building methods (incl. ASTRAL, ASTRID, MRL; see 4.8.) do not require gene trees to be rooted and you can ignore this warning. However, MP-EST method (not implemented in HybPhyloMaker) requires all gene trees to be rooted and thus the resulting 'trees_rooted.newick' is not suitable for such analysis.

4.8. Estimate species trees

Species trees are estimated utilizing several methods including coalescence summary methods (ASTRAL, ASTRID), a supertree method (MRL), concatenation (ML in FastTree and ExaML), and a Bayesian concordance method (BUCKy). There is one script for each method, and, depending on the script, either concatenates the selected genes or uses the gene trees in 'treesMISSINGPERCENT_SPECIESPRESENCE_rooted_withoutBS.newick' for species tree inference. Based on the 'tree=' setting in 'settings.cfg', species trees will be estimated based on gene trees previously produced by FastTree or RAxML.

4.8.1. ASTRAL species tree

ASTRAL (Accurate Species TRee Algorithm; Mirarab et al., 2014) is a program for estimating species tree that is consistent under multi-species coalescent model. ASTRAL finds the species tree that has the maximum number of shared induced quartet trees with the set of gene trees.

Run the script 'HybPhyloMaker8a_astral.sh' and a species tree with branch lengths (in coalescent units) and branch support (local posterior probabilities based on quartet frequencies; Sayyari & Mirarab 2016) with the name 'Astral_MISSINGPERCENT_SPECIESPRESENCE.tre' is produced and saved in subfolder 'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/Astral'. The progress of ASTRAL run is written to the file 'Astral.log'. If bootstrapped RAxML gene trees (or FastTree gene trees with real bootstrap support) are summarized, ASTRAL can perform multilocus bootstrapping on the bootstrap replicate gene trees (100 bootstrap replicates). Several trees are produced:

- 'Astral_MISSINGPERCENT_SPECIESPRESENCE_withbootstrap.tre' – the species tree with bootstrap support values,
- 'Astral_MISSINGPERCENT_SPECIESPRESENCE_allbootstraptrees.tre' –all bootstrap replicates,
- 'Astrid_MISSINGPERCENT_SPECIESPRESENCE_bootmajorcons.tre' – the majority rule consensus tree of bootstrap replicate trees.

Progress of ASTRAL bootstrapping is saved to 'Astral_boot.log'. If 'combine=yes' is set in 'settings.cfg' another tree named 'Astral_MISSINGPERCENT_SPECIESPRESENCE_mainANDbootANDcons.tre' with combined support values (local posterior probabilities, multilocus bootstrap support, majority rule) is created.

IMPORTANT: When constructing ASTRAL species tree each sample is treated as a different species. If you wish to reconstruct a species tree with multiple individuals per species (force them to be together in the species tree) you should take the respective multiple NEWICK file out of the HybPhyloMaker and run ASTRAL with '-a' option providing a mapping file (see ASTRAL manual for details).

4.8.2. ASTRID species tree

ASTRID (Accurate Species TRee Reconstruction with Internode Distances; Vachspati & Warnow 2015) is another species tree reconstruction program that is consistent under multi-species coalescent model. It implements NJst method (Liu & Yu 2011) for datasets with missing entries. ASTRID is much faster than ASTRAL on large datasets.

Run the script `'HybPhyloMaker8b_astrid.sh'` a species tree (just topology) with the name 'Astrid_MISSINGPERCENT_SPECIESPRESENCE.tre' is produced and saved in the subfolder `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/Astrid'`. The progress of ASTRID run is written to the file `'Astrid.log'`. If bootstrapped RAxML gene trees (or FastTree gene trees with real bootstrap support) are summarized, ASTRID can perform multilocus bootstrapping on the bootstrap replicate gene trees (100 bootstrap replicates). Several trees are produced:

- `'Astrid_MISSINGPERCENT_SPECIESPRESENCE_withbootstrap`.tre' – the species tree with bootstrap support values,
- `'Astrid_MISSINGPERCENT_SPECIESPRESENCE_allbootstraptrees.tre'` – all bootstrap replicates,
- `'Astrid_MISSINGPERCENT_SPECIESPRESENCE_bootmajorcons`.tre' – majority rule consensus tree of bootstrap replicate trees.

Progress of ASTRID bootstrapping is saved to `'Astrid_boot.log'`. If 'combine=yes' is set in `'settings.cfg'` another tree named `'Astrid_MISSINGPERCENT_SPECIESPRESENCE_bootANDcons.tre'` with combined support values (multilocus bootstrap support and majority rule) is created.

4.8.3. MRL species tree

MRL (Matrix Representation with Likelihood; Nguyen et al. 2012) is a supertree method that combines trees on subsets of the full taxon set together to produce a tree on the entire set of taxa. First it encodes a set of gene trees by a large randomized matrix (the "MRL matrix") over {0,1, ?} (using `mrp.jar`; https://github.com/smirarab/mrpmatrix) and then analyzes the matrix using heuristics for 2-state Maximum Likelihood (implemented in, e.g., as 'BINCAT' model in RAxML).

Run the script `'HybPhyloMaker8c_mrl.sh'` and a MRL species tree with the name `'MRL_MISSINGPERCENT_SPECIESPRESENCE.tre'` is generated and saved in the subfolder `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/MRP'`. All bootstrap replicates are saved to `'MRL_MISSINGPERCENT_SPECIESPRESENCE_allbootstraptrees.tre'`, information about RAxML run to `'RAxML_MRL_info.log'` and MRL matrix to the file `'MRLmatrix_MISSINGPERCENT_SPECIESPRESENCE.phylip'`.

4.8.4. Species tree based on concatenation (FastTree)

The script `'HybPhyloMaker8e_concatenatedFastTree.sh'` allows running a fast analysis of the concatenated dataset using FastTree. First, the concatenated dataset of the selected genes listed in `'selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt'` (in the subfolder `'71selected MISSINGPERCENT'`) is prepared using AMAS and saved in both FASTA and PHYLIP format in `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/concatenated'`. Then FastTree constructs the tree `'concatenated_MISSINGPERCENT_SPECIESPRESENCE.fast.tre'`.

IMPORTANT: If you work with data in corrected reading frame (see 4.4.b) you have to set 'corrected=yes' in `'settings.cfg'`. In this case all the species trees are saved to `'exons/82trees_correctedMISSINGPERCENT_SPECIESPRESENCE'`.

4.8.5. Species tree based on concatenation (ExaML)

A more reasonable approach how to use a concatenated dataset for constructing a species phylogeny is to apply a partitioned analysis, which allows modelling parameters for each partition (=gene/position/etc.) separately. The script requires the file 'partitionfinder-2.1.1.tar.gz' in `'HybSeqSource'`. This file can be downloaded from PartitionFinder GitHub (https://github.com/brettc/partitionfinder/releases/tag/v2.1.1) When running the script `'HybPhyloMaker8f_ concatenatedExaML.sh'` the following steps are performed:

- the concatenated dataset is prepared similarly to 4.8.4.,
- `'partitions.txt'` with a partition description of the concatenated alignment (produced by AMAS) is modified and a configuration file (`'partition_finder.cfg'`) for PartitionFinder2 (Lanfear et al. 2014) is prepared. For simplicity and speed efficiency with large datasets (tens to hundreds of samples, hundreds of genes) the following settings are involved: branchlengths = linked, models = GTR+G, model_selection = AICc, search = rclusterf. Consult the PartitionFinder manual for other options.
- PartitionFinder is executed in order to find the best partitioning scheme. All resulting files are saved to `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/concatenatedExaML/PartitionFinder'`. Check the PartitionFinder documentation for information about files in this folder. The best scheme is saved to `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/concatenatedExaML/RAxMLpartitions.txt'`. The script will check for the presence of this file. If the file is found, the concatenation and PartitionFinder run are skipped and the script continues with the next step.
- RAxML checks whether the concatenated alignment contains any entirely invariable positions and, if yes, prepares a reduced alignment and modifies the partition file as well. The modified files are saved to `'concatenatedMISSINGPERCENT_SPECIESPRESENCE.phylip.reduced'` and `'RAxMLpartitions.txt.reduced'` in the subfolder `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/concatenatedExaML/'`.
- The best ML tree is estimated using ExaML (Kozlov et al., 2015) and saved to `'ExaML_BestML_MISSINGPERCENT_SPECIESPRESENCE.tre'`. This step is extremely computationally demanding, and it is recommended to run it on a computer cluster. The MPI version of ExaML is used.
- 100 bootstrap replicates are calculated and the tree with support values is saved to `'ExaML_bootstrap_MISSINGPERCENT_SPECIESPRESENCE.tre'`. All 100 bootstrap trees are in `'ExaML_bootstrap.trees'`. Progress of the calculation of bootstrap replicates is continuously written to `'ExaML_bootstrap_progress.txt'` together with the time (in min) necessary for each bootstrap replicates.

IMPORTANT: This script is optimised for the cluster environment but might also run locally.

4.8.6. BUCKy concordance tree

BUCKy (Bayesian Untangling of Concordance Knots; Larget et al., 2010) estimates the concordance factor (CF) of a clade, which is defined as the proportion of genes that truly have the clade in their trees. It uses a Bayesian approach to distinguish between gene tree estimation error and gene tree incongruence (Ané et al., 2007). Typically, a sample of trees from MrBayes analysis is used as the input for BUCKy. However, similarly to Yang & Warnow (2011), bootstrap trees from FastTree or RAxML are used here. Because BUCKy requires full-sample datasets (i.e., no missing samples in all gene trees) it will work only if you run it over genes filtered for no missing samples (i.e., with 'SPECIESPRESENCE=100'). Otherwise some samples will be pruned from the dataset. You need to consult BUCKy manual to treat missing data in a different way.

First, review the specific settings for BUCKy in `'settings.cfg'`, i.e., 'nrbucky=', 'nrruns=', 'nrchains=', and 'alpha=' (consult BUCKy manual for appropriate settings). Then run the script `'HybPhyloMaker8g_BUCKy.sh'` which produces standard outputs of the program BUCKy (see manual for the interpretation). These files are written to `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/BUCKy/output'`. Moreover, primary concordance tree and population tree are extracted from the '*.concordance' file and save as separate tree files to `'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/BUCKy/`.

4.8.7. Divergence dating of species tree

The species tree with branch lengths estimated in substitution-per-site units can be converted to the tree with divergence time estimates for every node. Here, this task is achieved with penalized likelihood implemented in treePL (). TBA

4.8.8. Support value combination from multiple trees

TBA

4.9. Networks and other analyses

When the gene alignments are generated it is possible to do several other analyses. Currently, two network-building approaches are implemented (neighbor-net and SuperQ network). Moreover, D-statistics and related admixture evidence can be inferred using Dsuite, quartet sampling support values can be calculated, dissimilarity heatmap based on filtered SNPs can be plotted, and number of conflicting gene trees can be inferred and plotted using PhyParts.

4.9.1. Neighbor-Net network

Neighbor-Net (Bryant & Moulton 2004) is a distance-based method for constructing phylogenetic networks that is based on the neighbor-joining (NJ) algorithm. It permits the representation of conflicting signal or alternative phylogenetic histories based on an arbitrary distance matrix.

Run the script `'HybPhyloMaker8h_neighbourNetwork.sh'` and the neighbor-net network is build using 'neighborNet' function from the 'phangorn' package in R and saved to `'species_trees/NeighbourNetwork'`. The network file can be opened with SplitsTree4 (Huson & Bryant 2006) and it should be make planar by selecting *Draw* -> *EqualAngle* and clicking on '*Apply*'.

4.9.2. SuperQ network

SuperQ (Grünewald et al. 2013) is a method for constructing supernetworks from trees (a network analogy to super trees). It breaks the input trees into quartet trees and stitching these together to a split network with an adaptation of the QNet method (Grünewald et al. 2007) and using the branch lengths from the input trees to estimate the branch lengths in the resulting network.

Run the script `'HybPhyloMaker8j_SuperQNetwork.sh'` and the SuperQ network is build using SPECTRE (Bastkowski et al. 2018) and saved to `'species_trees/SuperQ'`. Again, the network file can be opened with SplitsTree4.

4.9.3. SNaQ network

SNaQ algorithm (Species Networks applying Quartets; Solís-Lemus & Ané, 2016) allows inference of explicit phylogenetic networks which is done with maximum pseudolikelihood from gene trees. It incorporates uncertainty in estimated gene trees and gene tree discordance due to incomplete lineage sorting (ILS). SNaQ is a part of PhyloNetworks Julia package (https://github.com/JuliaPhylo/PhyloNetworks.jl; Solís-Lemus et al., 2017). It uses a list of unrooted gene trees and starts the search from the topology obtained with ASTRAL. Then it first calculates the network with zero reticulations (hmax=0) and successively continues to the number of reticulations defined in 'hmax=' in '`HybSeqSource`'. Calculation for every 'h' is repeated n-times and on a cluster, computations are parallelized (number of repetitions is set by number of available computer cores). All the resulting networks are then re-rooted with OUTGROUP and plotted as SVG/PDF figures (both rooted and unrooted networks are plotted). Maximum pseudolikelihoods for best networks for every 'h' are summarized and plotted as a graph.

Running the script `'HybPhyloMaker8l_SNaQ.sh'` creates these files in `'species_trees/SNaQ'`:

- `'runSNaQ.log'` – log of all steps of the Julia script,
- `'netH_XXruns.log'` – log for calculation of the network with H reticulation(s) with XX repetitions,
- `'netH_XXruns.out'` – results with best network in extended NEWICK format as well as in the format for Dendroscope, and list of all suboptimal networks sorted by decreasing loglik,
- `'netH_XXruns.networks'` – all estimated networks with their loglik (best network first),
- `'netH_XXruns_{rooted,unrooted}.{svg,pdf}'` – rooted and unrooted networks, if the file with rooted network is missing rooting was not possible, see 'runSNaQ.log',
- `'scores.txt'` – loglik summary for best network for every 'h',
- `'NetworkScoresPlot.pdf'` – plot of values in 'scores.txt',
- `'tableCF.csv'` – a table of concordance factors (CF) for each 4-taxon subset

4.9.4. PhyloNet network

PhyloNet (Than et al., 2008) is a tool for analysing reticulate evolutionary relationships, known as phylogenetic networks. Only the maximum pseudolikelihood (InferNetwork_MPL) approach is implemented in HybPhyloMaker (`'HybPhyloMaker8m_PhyloNet.sh'`). First, NEXUS input file is prepared from NEWICK list of gene trees and PhyloNet command is added. Second, submission jobs for cluster environment are prepared – one for each number of allowed reticulations from `'hstart='` to `'hmax='` with number of runs (parameter '-x', `'numruns='`), controlled from 'settings.cfg'. Job files are saved to 'species_trees/PhyloNet' and can be submitted by running 'submitPhyloNetjobs.sh'. After all jobs are finished, a second script ('HybPhyloMaker8m2_PhyloNet_summary.sh') can be run. It summarizes the results in multiple ways:

- creates a summary table ('data_PhyloNet_summary.txt') – this includes statistics of all estimated networks (number of allowed reticulation events (h), number of branches, number of gene trees, ki – sum of previous three numbers, logLikelihood, AIC, deltaAIC) sorted by increasing deltaAIC, i.e., best networks first. AIC is calculated using approach given in Kandziora et al. (2022), i.e., AIC=-2loglik + 2ki.
- re-roots and plots all networks using Julia PhyloNetworks package (using readTopology, rootatnode! and plot functions), all rooted/unrooted networks are then merged into a single PDF ('data_PhyloNet_{rooted,unrooted}.pdf')

IMPORTANT: This script is only optimised for the cluster environment and cannot be run locally. However, it can be used to prepare input NEXUS files.

4.9.5. Dsuite

Dsuite (Malinsky et al. 2021; https://github.com/millanek/Dsuite) is a set of scripts that is used to estimate admixture/hybridization evidence from variable sites (SNPs) in the genome. First, Patterson's D (also known as ABBA-BABA statistics) and the related estimate of admixture fraction *f* (referred to as the f4-ratio) are computed for all possible trios of samples/species (Dtrios command). Even though they were proposed in a population genetic framework the methods can be applied to investigate hybridisation/introgression within groups of closely related species. Second, *D* and f statistics are calculated for branches on a tree that relates the samples/species (Fbranch command). The results are plotted using python and ruby scripts as described in Dsuite GitHub and in a tutorial (https://github.com/millanek/tutorials/tree/master/analysis_of_introgression_with_snp_data).

Running the script 'HybPhyloMaker8i_Dsuite.sh' does the following steps and creates these files in 'species_trees/Dsuite':

- the concatenated dataset is prepared similarly to 4.8.4. ('concatenated.fasta' and 'partitions.txt'),
- ASTRAL tree is used as a tree for the analysis ('sptree.tre'),
- OUTGROUP taxon is renamed to 'Outgroup' in both tree and alignment file,
- SNP-sites (Page et al. 2016; https://github.com/sanger-pathogens/snp-sites) is used to generate a VCF file from the alignment ('FULLconcatenated.vcf.gz'),
- resulting VCF file is filtered with BCFTOOLS (Danecek et al. 2021) to remove site with more than 20% missing data ('FILTEREDconcatenated.vcf.gz'),
- SNPs are filtered (reduced) using the strategy specified under 'SNPs=' in 'settings.cfg'
    - first – only first SNP per locus is used ('firstSNP.vcf.gz')
    - random – single random SNP per locus is used ('randomSNP.vcf.gz')
    - thinning – one SNP per 100bp window is used (this is done using VCFTOOLS with '--thin' option ('thinned.vcf.gz'),
- SNP statistics is created ('SNPstat.txt' and 'nrSNPsPerExon.txt'),
- Dsuite is run as described above and it creates following results:
    - 'SpeciesSet_gene_flow_BBAA.txt' – results of *D*-statistics calculations for all possible species trios

   - `'SpeciesSet_gene_flow_tree.txt'` – as the previous, with trios arranged according to the species tree
   - `'SpeciesSet_gene_flow_Dmin.txt'` – as the previous, with trios arranged so that the *D*-statistics is minimised
   - `'SpeciesSet_gene_flow_BBAA_D.{svg,pdf}'` – heatmap of the most significant *D*-statistics found between all pairs of species (P2 and P3) across all possible P1 species in trios
   - `'SpeciesSet_gene_flow_BBAA_f4ratio.{svg,pdf}'` – as the previous, but for f4-ratio (proportion of the genome affected by geneflow)
   - `'SpeciesSet_gene_flow_Fbranch.txt'` – results of f-branch metric (gene flow assigned to specific branch on phylogeny; Malinsky et al. 2018)
   - `'gene_flow.{png,svg,pdf}'` – figure of f-branch results

4.9.6. Quartet Sampling

Quartet Sampling (QS; Pease et al. 2018; https://github.com/FePhyFoFum/quartetsampling) is a method to evaluate existing phylogenetic topology by calculating branch support using repeated sampling of quartets. QS characterizes discordance in large-sparse and genome-wide data sets distinguishing strong conflict from weak support. It offers more comprehensive and specific information on branch support by calculating three values per every node: quartet concordance (QC), quartet differential (QD) and quartet informativness (QI); as well as quartet fidelity (QF) for terminal nodes. See the above mentioned publication and software documentation for background and interpretation of these values.

Running the script `'HybPhyloMaker8k_quartetsampling.sh'` does the following steps and creates these files in `'species_trees/quartetsampling'`:

- already available concatenated alignment is used (script `'HybPhyloMaker8e_concatenatedFastTree.sh'` should be successfully run before running Quartet Sampling,
- one of available species trees is used for scoring, it depends on 'qstree=' in `'settings.cfg'` (one of FastTree, ExaML or ASTRAL),
- Quartet Sampling python script is run with default settings and several files are produced:
   - RESULT.node.scores.csv – frequency, QC, QD and QI for every node, QF for terminal branches,
   - RESULT.node.counts.csv – number of QS replicates for concordant and both discordant quartet arrangements for every node,
   - RESULT.labeled.tre – a tree with each internal branch labelled with their QS## identifier and with QF for terminals,
   - RESULT.labeled.tre.freq/qc/qd/qu – a tree with each internal branch labelled with frequency of concordant replicates or QC/QD/QI scores,
   - RESULT.labeled.tre.figtree – FigTree format tree that contains all QS scores and a "score" field with QC/QD/QI for internal branches.
- some of the trees are modified and used for plotting the final figure with a custom R script to resemble trees in the original publication (Pease et al. 2018; Fig. 11)
   - QS.pdf – a tree with QC/QD/QI for internal branches and with colour-coded nodes based on QC value according above-mentioned publication (dark green QC > 0.2, light green 0.2 ≥ QC > 0, orange 0 ≥ QC ≥ −0.05, red QC < −0.05). The coloring is done with the help of the 'node.support' function from 'phyloch' R package (Heibl 2008).

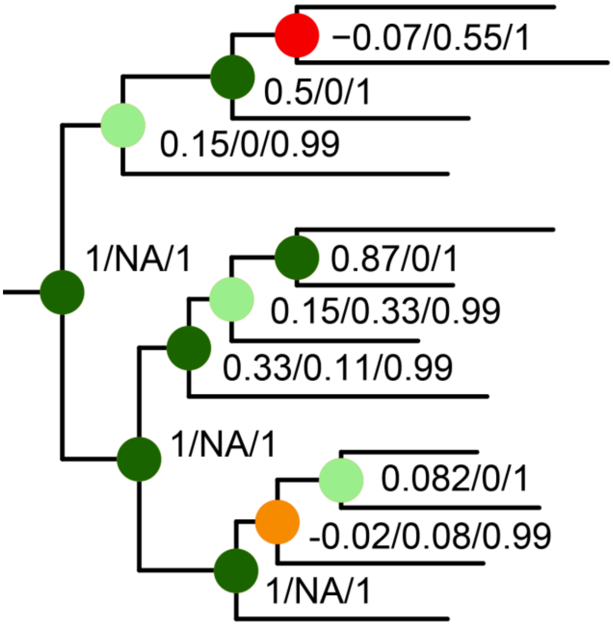


Fig. 11: A species tree scored with Quartet Sampling (`'HybPhyloMaker8k_quartetsampling.sh'`). Node labels are three values corresponding to quartet concordance (QC), quartet differential (QD) and quartet informativness (QI). Coloration of nodes follows QC values (dark green QC > 0.2, light green 0.2 ≥ QC > 0, orange 0 ≥ QC ≥ −0.05, red QC < −0.05).

4.9.7. SNP heatmap

A heatmap based on pair-wise distances derived from filtered variable sites (SNPs) is plotted. The script `'HybPhyloMaker8n_SNPheatmap.sh'` takes all (not only selected) gene alignments, concatenates them and uses SNP-sites (Page et al. 2016) to generate a VCF file which is further filtered with BCFTOOLS (Danecek et al. 2021) to remove site with more than 20% missing data. The genotype calls (0s or 1s) are extracted as binary matrix and a pair-wise distance matrix is calculated using simple-matching coefficient (implemented in PERMANOVA R package; Vicente-Gonzalez & Vicente-Villardon 2021). The matrix is then plot as a heatmap using `heatmap.2` function in R (package `gplots`). The resulting PDF might require some tuning of parameters (script `'SNPheatmap.R'` in `'HybSeqSource'`). Following files are created in '`species_trees/SNPheatmap`':

- `'FULLconcatenated.vcf.gz'` – VCF of all SNPs from concatenated alignment,
- `'FILTEREDconcatenated.vcf.gz'` – VCF of filtered SNPs only,
- `'SNPmatrix.txt'` – binary matrix of filtered SNPs,
- `'SNPmatrixTransposed.txt'` – transposed matrix of filtered SNPs (samples as rows),
- `'DATA_SNPdistmat.txt'` – pair-wise distance matrix (taken from `$carpet` of `heatmap.2` function results),
- `'DATA_SNPheatmap.pdf'` – heatmap figure with dendrograms (at the left side and at the top) and dissimilarity colour scale.

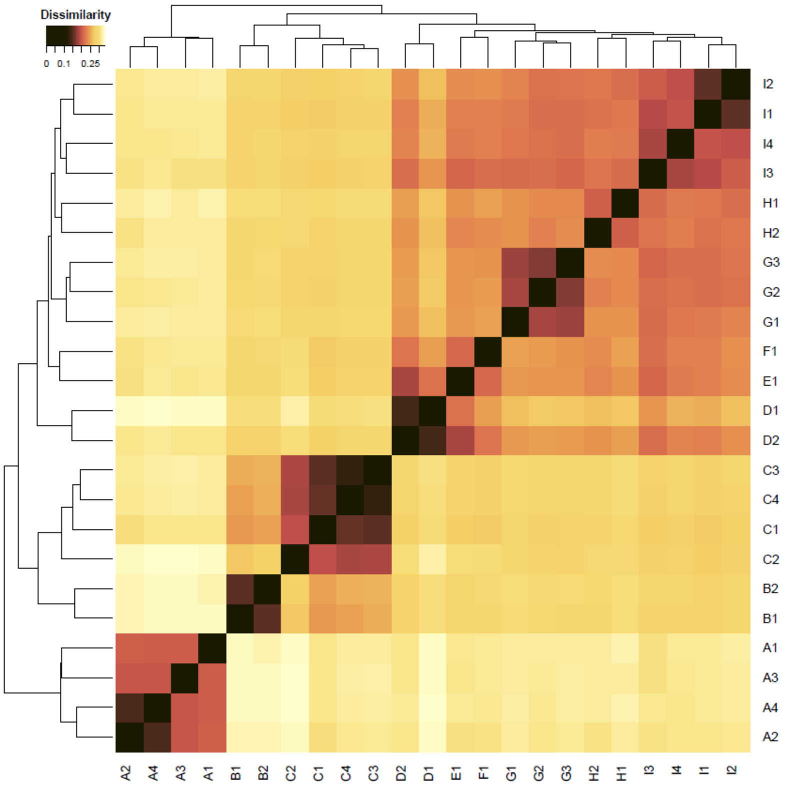


Fig. 12: A heatmap based on simple-matching (dis)similarity calculated on binary matrix of filtered SNPs of all concatenated genes. Lowest dissimilarity (dark colours) is between more related samples (i.e., samples sharing more SNPs).

4.9.8. PhyParts

PhyParts (Smith et al. 2015; https://bitbucket.org/blackrim/phyparts) is an approach testing whether individual gene tree topologies are concordant or conflicting with a species topology. It summarizes the number of concordant/discordant gene trees for each node on a species tree. PhyParts requires rooted gene trees and rooted species tree. The script PhyParts PieCharts (M. Johnson; https://github.com/mossmatters/MJPythonNotebooks/blob/master/PhyParts_PieCharts.ipynb) is used to visualize PhyParts output and produces pie charts on each node of the species tree phylogeny corresponding to the proportion of gene trees that:

- blue: support the shown topology,
- green: conflict with the shown topology (most common conflicting bipartition),
- red: conflict with the shown topology (all other supported conflicting bipartitions),
- gray: have no support for conflicting bipartion.

Currently, only ASTRAL species tree can be scored within HybPhyloMaker. However, you can select several options in `'settings.cfg'` before running PhyParts script: 'phypartsbs=' to set the cut-off for bootstrap support in gene trees, 'ppcolors=' to set the colours for piecharts, or 'nrpptrees=' to set number of trees for which PhyParts will be calculated (i.e., limit the number of trees for, e.g., large datasets). See Appendix 5 for more details.

Running the script `'HybPhyloMaker11_phyparts.sh'` creates these files/folders in `'species_trees/Astral/phyparts_PHYPARTSBS'`:

- `'phyparts_Astral_MISSINGPERCENT_SPECIESPRESENCE_BSPHYPARTSBS_XXXtrees.pdf'` – ASTRAL tree with piecharts (e.g., Fig. 13) and number of concordant (above branches) and discordant (below branches) gene trees for every bipartition

- 'phyparts_Astral_MISSINGPERCENT_SPECIESPRESENCE_BSPHYPARTSBS _XXXtrees.svg' – the same in SVG format
- 'phypartsinfo_ BSPHYPARTSBS _XXXtrees.txt' – info about the name of species and gene tree files, number of gene trees, outgroup name and number of rooted gene trees (this is the 'XXX')
- folder 'trees_res_XXXtrees' containing results of PhyParts run (log, hist, alts, key, concord.node, and conflict.node files; see PhyParts documentation for more details)

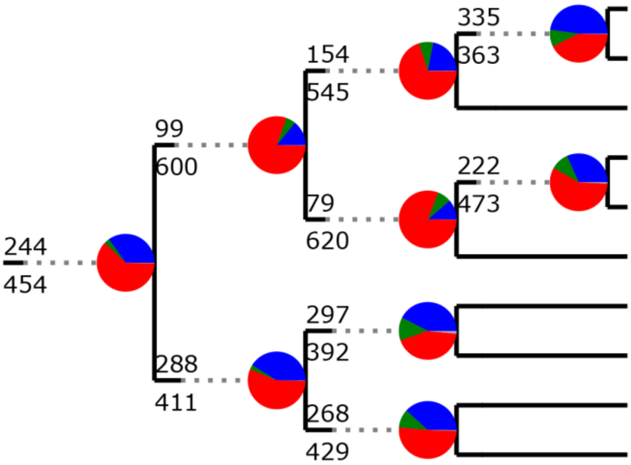


Fig. 13: A species tree with PhyParts results and PhyParts PieCharts plots showing number of gene trees concordant with the species tree at that node (top number), the bottom number indicates the number of gene trees in conflict with that clade in the species tree. Pie charts show the proportion of gene trees that support the shown topology (blue), conflict with the shown topology – most common conflicting bipartition (green), conflict with the shown topology – all other supported conflicting bipartitions (red), or have no support for conflicting bipartion (gray).

4.10. Select & Update

After the gene trees are built (see 4.6.) and a table with summary characteristics for all selected loci ('gene_properties.txt') is generated there is an easy possibility to subselect only some of the genes based on those characteristics. Open the 'gene_properties.txt' in a spreadsheet editor (e.g., Excel), sort it according to your desired column(s) and delete unwanted genes. Now save the table as TAB delimited (or copy the whole table to a text editor, e.g. Notepad++ in Windows) under the name 'gene_properties_update.txt' to 'exons/72trees_ MISSINGPERCENT_SPECIESPRESENCE/TREE/update'. If in Windows, be sure that there are UNIX-style end-of-line characters in this text file.

Run the script 'HybPhyloMaker9_update_trees.sh'. The following files are generated for the set of selected genes:

- 'genes_corrs_update.pdf' – plot with correlations among pairs of selected properties for the updated selection of genes,
- histograms showing distribution of characteristics for selected genes (similar to that created in 4.7.)
- 'selected_genes_70_75_update.txt' in the automatically created subfolder 'exons/71selectedMISSINGPERCENT/updatedSelectedGenes'

Now you are ready to build species trees based on these subselected genes only. First, change the option 'update=' to 'update=yes' in 'settings.cfg' and then (re)run 'HybPhyloMaker7_roottrees.sh' and all desired 'HybPhyloMaker8*.sh' scripts.

Species trees are now in the subfolder 'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/update/species_trees' (Fig. 13).

4.11. Subselect trees (loci) containing requisite taxa & collapse unsupported branches

If you want to include only trees (loci) containing samples of your choice (e.g., outgroup taxon) and/or collapse unsupported branches in gene trees before running species tree estimation there is a possibility to run 'HybPhyloMaker10_requisite_collapse.sh'. Set the values in 'settings.cfg': (1) option 'requisite=yes' will generate a modified multi-NEWICK file containing only trees with samples specified in 'requisitetaxa='. You can also specify more samples by separating them with '|', e.g. 'requisitetaxa="speciesA|speciesB"'. This will select trees containing 'speciesA' OR 'speciesB'. (2) option 'collapse=' is used if you wish to create modified multi-NEWICK file with trees that have collapsed all branches with support lower than specified here.

After running the script new subfolder(s) is/are created in the subfolder 'exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/update/species_trees': subfolder 'with_requisite' containing subfolders 'collapsedCOLLAPSE' (if 'requisite=yes' and 'collapse' was not set to '0') or subfolder 'collapsedCOLLAPSE' (if 'requisite=no' and 'collapse' was not set to '0'). Those (sub) folders contain modified multi-NEWICK file(s). Now you are ready to re-run all desired 'HybPhyloMaker8*.sh' scripts to reconstruct species trees with subset of gene trees and/or with collapsed gene trees. Respective folder(s) appear in subfolders specified above (Fig. 13).

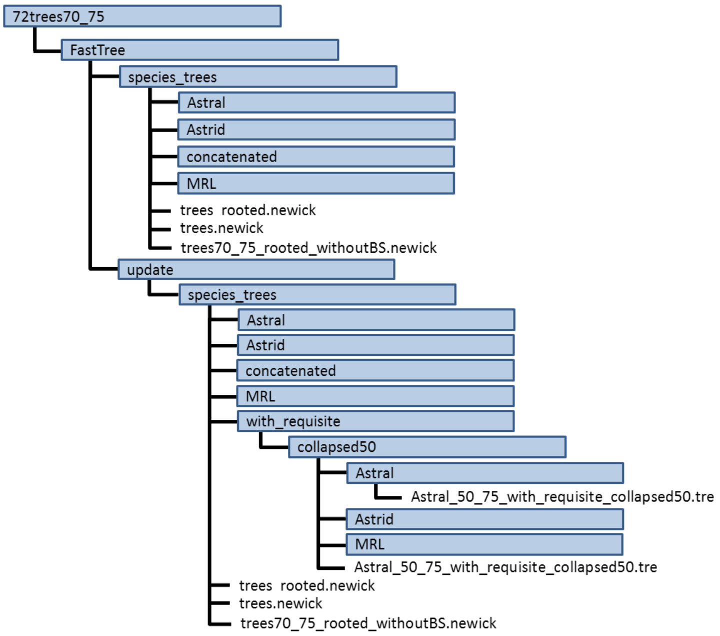


Fig. 13: Folder structure after running 'HybPhyloMaker9_update_trees.sh' (creates an 'update' folder), 'HybPhyloMaker7_roottrees.sh' (creates a 'species_trees' folder), 'HybPhyloMaker10_requisite_collapse.sh' (creates a 'with_requisite' folder) and scripts for species tree estimation (ASTRAL, ASTRID, MRL) that were run before and after running script 'HybPhyloMaker10_requisite_collapse.sh'.

4.12. Exclude samples from alignment

TBA...

4.13. Working with organellar data

HybPhyloMaker also allows working with organellar reads that are often obtained in sufficient quantity as off-target reads when sequencing enriched HybSeq libraries (Weitemier et al., 2014; Schmickl et al., 2016). Usually you will obtain 5-15% of plastid reads and 1-2% of mitochondrial reads. This quantity (even with a high multiplex ratio) allows you to perform a *de novo* plastome/chondriome assembly, however, this approach is usually unsuccessful when less reads are available. Nevertheless, even with a lower number of organellar reads a sufficient sequencing depth is usually achieved, especially for coding regions. Therefore, we implemented the possibility to work with coding organellar regions in HybPhyloMaker.

First, you need to set 'cpDNA=yes' in `'settings.cfg'`. This tells all the HybPhyloMaker scripts that they should use chloroplast reference (coding sequences) defined in 'cpDNACDS=' and a pseudoreference created from it (using `'HybPhyloMaker0b_preparereference.sh'`, see 2.4.3.) and work with files originated from chloroplast-related reads. Then you could start running scripts similar as in the case of nuclear exons. The script `'HybPhyloMaker2_read_mapping.sh'` maps the filtered, duplicate-free reads (obtained by running `'HybPhyloMaker1_rawprocess.sh'` which is not necessary to run again!) to the organellar 'pseudoreference'. All results are now saved to the newly created `'cp'` folder within `'datadir'`. Alternatively, you might use Geneious to do the read mapping. In this case follow the general recommendations from chapter 4.2.2. and export the consensus sequences. Save this file as `'consensus_cpDNA.fasta'` and copy it to the folder `'cp/30consensus'`. Now you are ready to run the script `'HybPhyloMaker3_generatepslx.sh'` and generate PSLX files with sequences that are homologous to the coding cpDNA regions. Copy desired PSLX files from `'cp/50pslx'` to a specific folder within `'homedir'` and specify its name as 'otherpslxcp=' in `'settings.cfg'`. Then you can run HybPhyloMaker scripts 4 to 9 similarly as described for exons (see 4.4. – 4.9.). HybPhyloMaker will recognize that you are working with organellar DNA and will save all results to `'datadir/cp'`.

COMMENT: The folder `'70concatenated_exon_alignments'` is not created for organellar data because chloroplast genes are not concatenating before gene tree building.

4.14. Additional scripts

HybPhyloMaker includes also some additional scripts that help with results formatting and download. These scripts are not numbered and are usually run when some analyses are finished.

4.14.1. Generate reports

The script 'HybPhyloMaker_reports.sh' generates formatted *.xlsx file (to be opened, e.g., in Microsoft Excel) with some additional statistics (sum, average) from the following tables:

- filtering summary ('reads_summary.txt' in '20filtered')
- mapping summary ('mapping_summary.txt' in '21mapped')
- missing data overview for all genes ('MissingDataOverview.txt' in '71selectedMISSINGPERCENT')
- missing data overview for selected genes ('MissingDataOverview_MISSINGPERCENT.txt' in '71selectedMISSINGPERCENT')
- alignment characteristics for all genes ('summaryALL.txt' in '71selectedMISSINGPERCENT')
- alignment/tree characteristics for selected genes ('gene_properties.txt' in '72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE')

This script takes all these tables and combine them into a single multi-sheet *.xlsx file using 'HybPhyloMaker_reports.R' and save as '${data}_summary.xlsx' in the 'data' folder. The table is generated for the specified 'MISSINGPERCENT=' and 'SPECIESPRESENCE=' values. If any of the tables are missing no summary table is produced.

4.14.2. Generate test dataset

TBA…

4.14.3. Generate plastome reference

TBA…

References

Ané C, Larget B, Baum DA, Smith SD & Rokas A (2007): Bayesian estimation of concordance among gene trees. *Molecular Biology and Evolution* 24: 412–426.

Barrett CF, Specht CD, Leebens-Mack J, Stevenson DW, Zomlefer WB & Davis JI (2014): Resolving ancient radiations: can complete plastid gene sets elucidate deep relationships among the tropical gingers (Zingiberales)? *Annals of Botany*, 113: 119–133.

Constantinides B & Robertson DL (2017): Kindel: indel-aware consensus for nucleotide sequence alignments. *Journal of Open Source Software*, 2: 282

Bastkowski S, Mapleson D, Spillner A, Wu T, Balvočiūtė M & Moulton V (2018): SPECTRE: a suite of phylogenetic tools for reticulate evolution. *Bioinformatics*, 34: 1056–1057.

Bolger AM, Lohse M & Usade B (2014): Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*, 30: 2114–2120.

Borowiec ML (2016): AMAS: a fast tool for alignment manipulation and computing of summary statistics. *PeerJ*, 4: e1660.

Bryant D & Moulton V (2004): Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks. *Molecular Biology and Evolution*, 21: 255–265.

Břinda K, Boeva V & Kucherov G (2016): Dynamic read mapping and online consensus calling for better variant detection. *arXiv*:1605.09070.

Carlsen MM, Fér T, Schmickl R, Leong-Škorničková J, Newman M & Kress WJ (2018): Resolving the rapid plant radiation of early diverging lineages in the tropical Zingiberales: Pushing the limits of genomic data. *Molecular Phylogenetics and Evolution*, 128: 55–68.

Capella-Gutiérrez S., Silla-Martínez JM & Gabaldón T (2009): trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics*, 25: 1972–1973.

Charif D & Lobry JR (2007): SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In: Structural approaches to sequence evolution: Molecules, networks, populations (Bastolla U et al. Eds.), *Biological and Medical Physics, Biomedical Engineering*, pp 207–232.

Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM & Li H (2021): Twelve years of SAMtools and BCFtools. *GigaScience*, 10: giab008.

Dowle M & Srinivasan A (2017): data.table: extension of `data.frame`. R package version 1.10.4. https://CRAN.R-project.org/package=data.table

Fér T & Schmickl RE (2018): HybPhyloMaker: target enrichment data analysis from raw reads to species trees. *Evolutionary Bioinformatics*, 14: 1–9. doi: 10.1177/1176934317742613.

Gálvez-Merchán A, Min KHJ, Pachter L & Booeshaghi AS (2023): Metadata retrieval from sequence databases with ffq. *Bioinformatics*, 39: btac667.

Gordon A & Hannon GJ (2010): FASTX-Toolkit. FASTQ/A short-reads pre-processing tools. http://hannonlab.cshl.edu/fastx_toolkit/ [accessed 29th August 2016].

Grünewald S, Forslund K, Dres, AW & Moulton V (2007): QNet: an agglomerative method for the construction of phylogenetic networks from weighted quartets. *Molecular Biology and Evolution*, 24: 532-538.

Grünewald S, Spillner A, Bastkowski S, Bögershausen A & Moulton V (2013): SuperQ: computing supernetworks from quartets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10: 151-60.

Guangchuang Y (2018): treeio: Base classes and functions for phylogenetic tree input and output. R package version 1.4.3. https://guangchuangyu.github.io/software/treeio

Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W & Gascuel O (2010): New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Systematic Biology*, 59: 307–321.

Heibl C (2008): PHYLOCH: R language tree plotting tools and interfaces to diverse phylogenetic software packages. http://www.christophheibl.de/Rpackages.html.

Huson DH & Bryant D (2006): Application of phylogenetic networks in evolutionary studies. *Molecular Biology and Evolution*, 23: 254–267.

Junier T & Zdobnov EM (2010): The Newick utilities: high-throughput phylogenetic tree processing in the UNIX shell. *Bioinformatics*, 26: 1669–1670.

Kandziora M, Sklenář P, Kolář F & Schmickl R (2022): Discordance in recent and rapidly radiating groups? Developing a workflow using *Loricaria* (Asteraceae) as an example. *Frontiers in Plant Science*, 12: 765719.

Katoh K & Toh H (2008): Recent developments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, 9: 286–298.

Kearse M, Moir R, Wilson A, Stones-Havas S, Cheung M, Sturrock S, Buxton S, Cooper A, Markowitz S, Duran C, Thierer T, Ashton B, Meintjes P & Drummond A. (2012): Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics*, 28, 1647–1649.

Kent WJ (2002): BLAT - the BLAST-like alignment tool. *Genome Research*, 12: 656–64.

Kozlov AM, Aberer AJ & Stamatakis A (2015): ExaML version 3: a tool for phylogenomic analyses on supercomputers. *Bioinformatics*, 31: 2577-2579.

Lanfear R, Calcott B, Ho SY & Guindon S (2012): PartitionFinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses. *Molecular Biology and Evolution*, 29: 1695–1701.

Lanfear R, Calcott B, Kainer D, Mayer C & Stamatakis A (2014): Selecting optimal partitioning schemes for phylogenomic datasets. *BMC Evolutionary Biology* 14: 82.

Langmead B & Salzberg S (2012): Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9: 357–359.

Larget BR, Kotha SK, Dewey CN & Ané C (2010): BUCKy: Gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics* 26: 2910–2911.

Li H & Durbin R (2009): Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25: 1754–1760.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R & 1000 Genome Project Data Processing Subgroup (2009): The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25: 2078–2079.

Liu L & Yu L (2011): Estimating species trees from unrooted gene trees. *Systematic Biology*, 60: 661–667.

Malinsky M, Svardal H, Tyers AM, Miska EA, Genner MJ, Turner GF & Durbin R (2018): Whole-genome sequences of Malawi cichlids reveal multiple radiations interconnected by gene flow. *Nature Ecology & Evolution*, 2: 1940–1955.

Malinsky M, Matschiner M & Svardal H (2021): Dsuite - fast D-statistics and related admixture evidence from VCF files. *Molecular Ecology Resources*, 21: 584–595.

Mirarab S, Reaz R, Bayzid MS, Zimmermann T, Swenson MS & Warnow T (2014): ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30: i541–i548.

Nguyen N, Mirarab S & Warnow T (2012): MRL and SuperFine+MRL: new supertree methods. *Algorithms for Molecular Biology*, 7: 3.

Page AJ, Taylor B, Delaney AJ, Soares J, Seemann T, Keane JA & Harris SR (2016): SNP-sites: rapid efficient extraction of SNPs from multi-FASTA alignments. *Microbial Genomics*, 2: e000056.

Paradis E, Claude J & Strimmer K (2004): APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20: 289–290.

Pease JB, Brown JW, Walker JF, Hinchliff CE & Smith SA (2018): Quartet Sampling distinguishes lack of support from conflicting support in the green plant tree of life. *American Journal of Botany*, 105: 385–403.

Price MN, Dehal PS & Arkin AP (2010): FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS ONE*, 5: e9490.

Rice P, Longden I & Bleasby A (2000): EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics*, 16: 276–277.

Sayyari E & Mirarab S (2016): Fast coalescent-based computation of local branch support from quartet frequencies. *Molecular Biology and Evolution*, 33: 1654–1668.

Schliep K (2011): phangorn: phylogenetic analysis in R." *Bioinformatics*, 27: 592–593.

Schmickl R, Liston A, Zeisek V, Oberlander K, Weitemier K, Straub SC, Cronn RC, Dreyer LL & Suda J (2016): Phylogenetic marker development for target enrichment from transcriptome and genome skim data: the pipeline and its application in southern African *Oxalis* (Oxalidaceae). *Molecular Ecology Resources*, 16: 1124–35.

Simmons MP & Norton AP (2014): Divergent maximum-likelihood-branch-support values for polytomies. *Molecular Phylogenetics and Evolution*, 73: 87–96.

Smith SA, Moore MJ, Brown JW & Yang Y (2015): Analysis of phylogenomic datasets reveals conflict, concordance, and gene duplications with examples from animals and plants. *BMC Evolutionary Biology*, 15: 150.

Solís-Lemus C & Ané C (2016): Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS Genetics*, 123: e1005896.

Solís-Lemus C, Bastide P & Ané C (2017): PhyloNetworks: A package for phylogenetic networks. *Molecular Biology and Evolution*, 34: 3292–3298.

Stamatakis A (2014): RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30: 1312–1313.

Struck TH, Wey-Fabrizius AR, Golombek A, Hering L, Weigert A, Bleidorn C, Klebow S, Iakovenko N, Hausdorf B, Petersen M, Kück P, Herlyn H & Hankeln T (2014): Platyzoan paraphyly based on phylogenomic data supports a noncoelomate ancestry of Spiralia. *Molecular Biology and Evolution,* 31: 1833–49.

Tange O (2011): GNU Parallel - the command-line power tool. *;login: The USENIX Magazine* 1(36): 42–47.

Than C, Ruths D & Nakhleh L (2008): PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9: 322.

Vachaspati P & Warnow T (2015): ASTRID: Accurate Species TRees from Internode Distances. *BMC Genomics*, 16: S3.

Vicente-Gonzalez L, Vicente-Villardon JL (2021): PERMANOVA: Multivariate analysis of variance based on distances and permutations. R package version 0.2.0. https://CRAN.R-project.org/package=PERMANOVA

Walker A (2017): openxlsx: read, write and edit XLSX files. R package version 4.0.17. https://CRAN.R-project.org/package=openxlsx

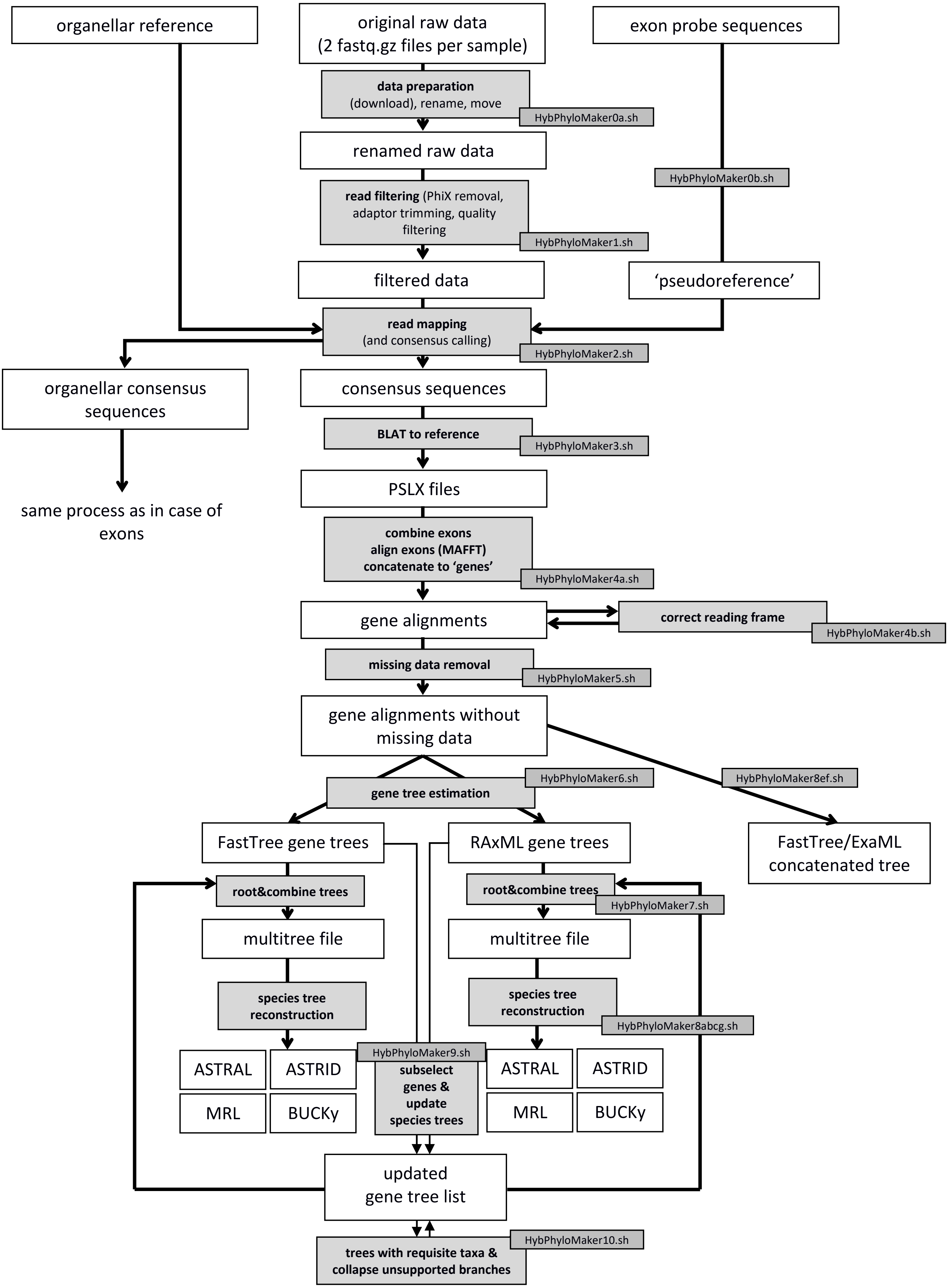
Warnes G, Bolker B, Bonebakker L, Gentleman R, Huber W, Liaw A, Lumley T, Maechler M, Magnusson A, Moeller S, Schwartz M, Venables B (2024): gplots: Various R Programming Tools for Plotting Data. R package version 3.1.3.1. https://CRAN.R-project.org/package=gplots

Weitemier K, Straub SC, Cronn RC, Fishbein M, Schmickl R, McDonnell A & Liston A (2014): Hyb-Seq: combining target enrichment and genome skimming for plant phylogenomics. *Applìcations in Plant Sciences*, 2: 1400042.

Xu H, Luo X, Qian J, Pang X, Song J, Qian G., Chen J & Chen S (2012): FastUniq: a fast de novo duplicates removal tool for paired short reads. *PLoS ONE*, 7: e52249.

Yang J & Warnow T (2011): Fast and accurate methods for phylogenomic analyses. *BMC Bioinformatics* 12 (Suppl 9): S4.

Appendix 1: HybPhyloMaker flowchart.



Appendix 2: Software to be installed prior to running HybPhyloMaker (in alphabetical order). On Linux this software can be automatically installed by running the script `'install_software.sh'`.

(see also table on GitHub https://github.com/tomas-fer/HybPhyloMaker/blob/master/docs/HybPhyloMaker_software.pdf)

1. **bam2fastq** (https://gsl.hudsonalpha.org/information/software/bam2fastq)
2. **BCFtools** (https://samtools.github.io/bcftools/)
3. **BLAT suite** (https://genome.ucsc.edu/goldenpath/help/blatSpec.html)
4. **Bowtie2** (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml)
5. **BUCKy** (https://www.stat.wisc.edu/~ane/bucky/)
6. **BWA** (http://bio-bwa.sourceforge.net/)
7. **EMBOSS** (http://emboss.open-bio.org/)
8. **ExaML** (http://sco.h-its.org/exelixis/web/software/examl/index.html)
9. **FastTree** (http://www.microbesonline.org/fasttree/)
10. **FastUniq** (https://sourceforge.net/projects/fastuniq/)
11. **GNU parallel** (http://www.gnu.org/software/parallel/)
12. **JDK/JRE** (http://www.oracle.com/technetwork/java/javase/overview/index.html)
13. **MAFFT** (http://mafft.cbrc.jp/alignment/software/)
14. **MstatX** (https://github.com/gcollet/MstatX/)
15. **Newick Utilities** (http://cegg.unige.ch/newick_utils/)
16. **OCOCO** (https://github.com/karel-brinda/ococo/)
17. **p4** (http://p4.nhm.ac.uk/)
18. **Perl** (https://www.perl.org/)
19. **Python** (https://www.python.org/)
20. **Python3** (https://www.python.org/download/releases/3.0/)
21. **R** (https://www.r-project.org/) – at least v.4.0
22. **RAxML** (http://sco.h-its.org/exelixis/web/software/raxml/index.html)
23. **SAMtools** (http://samtools.sourceforge.net/)
24. **seqtk** (https://github.com/lh3/seqtk/)
25. **TrimAl v1.4** (http://trimal.cgenomics.org/)

Appendix 3: How to install R packages before running HybPhyloMaker.

HybPhyloMaker uses R (at least v.3.4) to calculate some alignment and tree characteristics and also to produce plots in PNG and PDF formats. It is absolutely necessary to install several R packages before running scripts that utilize R. The following packages are necessary: 'ape', 'seqinr', 'data.table', 'openxlsx', 'phangorn' and 'treeio'. Be sure that you have recent version of 'ape' (at least 3.5) installed, some scripts do not work with version 3.4.

1. Local use

- Run R
- Type `install.packages(c("ape", "seqinr", "data.table", "openxlsx", "phangorn", "BiocManager"))`
- Follow instructions
- After finishing try `packageVersion("ape")` and you should get the answer `'5.1'` (i.e., at least `'3.5'`)
- installation of 'treeio' requires BiocManager
- Type `library (BiocManager)`
- Type `BiocManager::install("treeio")`
- However, running `'install_software.sh'` should install all the R packages automatically

2. MetaCentrum

- Run the script `'HybPhyloMaker0c_Rsetup_MetaCentrum.sh'`, which does everything for you (the packages are installed into the writable library `'Rpackages'` on your data server).

3. Hydra

- Login to any login node
- Load R using `module load tools/R/3.4.1`
- Run R by typing `R`
- Type `install.packages(c("ape", "seqinr", "data.table", "openxlsx"))`
- Type `'y'` when asked to use your personal to install packages (twice)
- Select a CRAN mirror by typing its number
- After finishing try `packageVersion("ape")` and you should get the answer `'5.1'` (i.e., at least `'3.5'`.

Appendix 4: HybPhyloMaker support files and scripts (in alphabetical order).

In the `'HybSeqSource'` folder there are all necessary files and scripts that are called by the main HybPhyloMaker BASH scripts. Consider proper citations of these sources, e.g., as follows:

1. **alignmentSummary.R** (*original part of HybPhyloMaker*)
2. **AMAS** (https://github.com/marekborowiec/AMAS)
3. **assembled_exons_to_fastas.py** (https://github.com/listonlab/Hyb-Seq_protocol/blob/master/assembled_exons_to_fastas/assembled_exons_to_fastas.py)
4. **astral.5.7.7.jar** (https://github.com/smirarab/ASTRAL)
5. **astralt4.R** (*original part of HybPhyloMaker*)
6. **ASTRID** (https://github.com/pranjalv123/ASTRID)
7. **catfasta2phyml.pl** (https://github.com/nylander/catfasta2phyml)
8. **combineboot.py** (*original part of HybPhyloMaker* based on http://p4.nhm.ac.uk/tutorial/combine_supports.html)
9. **CompareToBootstrap.pl, CompareTree.pl, MOTree.pm** (http://meta.microbesonline.org/fasttree/treecmp.html)
10. **ConsensusFixer** (https://github.com/cbg-ethz/ConsensusFixer)
11. **fastq2fasta.pl** (http://brianknaus.com/software/srtoolbox/fastq2fasta.pl)
12. **heatmap.R** (*original part of HybPhyloMaker*)
13. **histogram.r** (*original part of HybPhyloMaker*)
14. **HybPhyloMaker_reports.R** (*original part of HybPhyloMaker*)
15. **LBscores.R** (*original part of HybPhyloMaker* and https://github.com/marekborowiec/metazoan_phylogenomics/blob/master/gene_stats.R)
16. **mrp.jar** (https://github.com/smirarab/mrpmatrix)
17. **NEBNext-PE.fa** (Oligonucleotide sequences © 2006-2010 Illumina, Inc. All rights reserved.)
18. **PhiX.fsa** (http://www.ncbi.nlm.nih.gov/nuccore/9626372)
19. **plotNetworks.jl** (*original part of HybPhyloMaker*)
20. **plotQStree.R** (*original part of HybPhyloMaker*)
21. **plotting_correlations.R** (*original part of HybPhyloMaker*) and https://github.com/marekborowiec/good_genes/blob/master/plotting_correlations.R)
22. **runSNaQ.jl** (*original part of HybPhyloMaker*)
23. **tree_props.r** (*original part of HybPhyloMaker* and https://github.com/marekborowiec/good_genes/blob/master/tree_props.R)
24. **treepropsPlot.r** (original *part of HybPhyloMaker*)
25. **TreeCollapseCL4.jar** (http://emmahodcroft.com/TreeCollapseCL.html)
26. **trimmomatic-0.33.jar** (http://www.usadellab.org/cms/?page=trimmomatic)

Appendix 5: Explanation of HybPhyloMaker general settings, parameters and parameter options in the file ‘settings.cfg’.

1. GENERAL SETTINGS

location= Select whether you are running HybPhyloMaker locally, at the Czech National Grid (MetaCentrum) or the Smithsonian Institution HPC (Hydra).
0=locally
1=MetaCentrum
2=Hydra

server= If running on MetaCentrum, select a server for input/output data. See Appendix 7 for advice on how to run HybPhyloMaker on MetaCentrum. Possible options e.g.: brno1-cerit, brno2, brno3-cerit, brno6, brno12-cerit, budejovice1, plzen1.

data= Name of the folder with data. This folder is within ‘homedir’.
e.g., data=testdata

adapterfile= File name of fasta file of adapters to be trimmed using Trimmomatic (use one of the files distributed with Trimmomatic for TruSeq or Nextera libraries or the file ‘NEBNext-PE.fa’ provided with HybPhyloMaker for NEBNext Ultra libraries). The file must be located in ‘HybSeqSource’ folder.

2. GENE TREE SETTINGS

tree= Which software is used for gene tree building (FastTree/RAxML).
FastTree (with local support calculations) – fast
RAxML (with 100 rapid bootstrap replicates) – slow

FastTreeBoot= Whether trees generated by FastTree should be bootstrapped (yes/no).
yes=tree with true bootstrap support values are produced (slow)
no=trees with local supports values are produced (fast)

raxmlboot= Type of bootstrap generated by RAxML (standard/rapid). Rapid bootstrapping is (much) faster but produces (slightly) inflated values.

bsrep= Number of bootstrap replicates for RAxML gene trees.

bootstop= Use bootstopping strategy for RAxML gene trees, i.e., bootstrapping until convergence (yes/no). If this option is set to ‘yes’ number of bootstrap replicates (‘bsrep=’) is ignored.

model= Evolutionary model for RAxML gene trees (GTRGAMMA, GTRGAMMAI, GTRCAT, GTRCATI). Do not use CAT model for datasets with less than 50 taxa. Combination of GTRCAT and rapid bootstrapping is not recommended.

genetreepart= Which partitioning scheme is used for gene tree building with RAxML
no=trees are produced without partitioning
exon=by exon partitioning
codon=by exon and by codon partitioning (only works for data with corrected reading frame)

OUTGROUP= Specify outgroup for rooting both gene and species trees.
e.g., OUTGROUP=Curcuma-longa_S01

3. SPECIES TREE SETTINGS

examlboot= Whether to calculate bootstrap analysis for concatenated dataset using ExaML (yes/no).

mlbs= Multilocus bootstrap for ASTRAL and ASTRID trees (yes/no). Trees with multilocus bootstrap support values are produced when running ASTRAL/ASTRID species tree methods. Only for RAxML and bootstrapped FastTree trees. Can be very slow with large datasets.

combine= Whether to combine support values from main, bootstrap and bootstrap consensus trees to one tree (yes/no) – for ASTRAL and ASTRID trees only. Works only if ‘mlbs=yes’.

collapse= Collapse branches in gene trees with bootstrap support below this value, set to '0' if no collapsing is requested (0-99).

astralt4= Calculate ASTRAL tree with local posterior probabilities of three alternative hypotheses using '-t 4' (yes/no).

requisite= Whether to select only gene trees containing requisite samples (yes/no).

requisitetaxa= List of requisite taxa (taxon names separated by "|", the whole expression must be within quotes!). The gene tree is selected if at least one taxon is present in the particular gene tree.

nrbucky= Number of MCMC steps for BUCKy.

nrruns= Number of runs for BUCKy (parameter '-k').

nrchains= Number of chains for BUCKy (parameter '-c').

alpha= Parameter 'alpha' for BUCKy (parameter '-a').

phypartsbs= Parameter '-s' for PhyParts (cut-off for bootstrap support in gene trees). Set this 0-100 for RAxML and 0-1 for FastTree gene trees.

ppcolors= Colours for PhyParts pie charts (four named colours separated by spaces, the whole expression must be within quotes). See https://en.wikipedia.org/wiki/Web_colors for colour's names. Leave empty for original colours.

nrpptrees= Get only first XXX gene trees for PhyParts (leave empty if all trees should be used).

qstree= Species tree used for Quartet Sampling (Astral, FastTree or ExaML)

tpltree= Species tree for treePL (ExaML, Astral4 or FastTree)

tplbs= Make treePL also on bootstrapped trees (yes/no)

constree= Name of the constraint tree for ExaML. The tree should be in `‘HybSeqSource’` folder and needs to contain all taxa of the alignment!

SNPs= SNP thinning option for Dsuite
first – only first SNP per locus is used
random – single random SNP per locus is used
thinning – one SNP per 100bp window is used

hstart= Parameter ‘hstart’ for SNaQ/PhyloNet, i.e. minimum number of hybrid nodes. Keep ‘0’ when starting the analysis, otherwise it expects results for this ‘hstart’ in the results folder.

hmax= Parameter ‘hmax’ for SNaQ/PhyloNet. Maximum number of hybrid nodes.

numruns= Number of runs of the PhyloNet search (parameter '-x')

tree1= First (master) tree for bootstrap support value combination (Astral, Astral4, Astrid, MRL, FastTree, ExaML) tree2= Second tree for bootstrap support value combination (Astral, Astral4, Astrid, MRL, FastTree, ExaML)

prec= Number of decimals for rounding bootstrap support values in combined tree (default is 3)

4. MISSING DATA SETTINGS

MISSINGPERCENT= All samples with ≥ specified percentage (0-100%) of missing data per gene will be deleted from those particular gene alignment.
e.g., MISSINGPERCENT=70

SPECIESPRESENCE= Only loci with ≥ specified percentage (0-100%) of species per gene will be included in the final locus selection.
e.g., SPECIESPRESENCE=75

noallgaps= Removing gap and 'n' only positions in exon alignments using trimAl (yes/no).

gappyout= Trimming exon alignments using trimAl gappyout function (yes/no).

5. TYPE OF DATA

cp= Whether working with cpDNA.
yes=working with cpDNA
no=working with exons only
full=working with full plastome reference

update= Whether working with an updated list of genes (yes/no). After running the analysis with all selected genes there is an option to do a narrower selection of genes (see manual).

corrected= Whether working with alignments corrected for reading frame (yes/no).

trimmed= Working with data trimmed by trimAl (yes/no).

maxstop= Maximum number of stop codons allowed per alignment (i.e., considered as errors) to be accepted for further analyses.

selection= Name of the sample selection folder (to be created with script 12)

6. REFERENCE FILES

nrns= Number of Ns for separating exons in the pseudoreference (400 is recommended for 2x150 bp reads and 800 for 2x250 bp reads).

probes= Name of the FASTA file with exonic probe sequences (must be stored in ‘HybSeqSource’ folder).

minident= Minimum sequence identity between probe and sample used in BLAT when generating PSLX files (default is 90).

cpDNACDS= Name of the FASTA file with cpDNA CDS sequences (must be stored in ‘HybSeqSource’ folder). However, also non-coding sequences can be defined here.

cpDNA= File name with full plastome cpDNA reference in FASTA format (one IR should be removed).

cpGBfile= File name with full plastome cpDNA (GenBank flat file) (must be stored in HybSeqSource folder)

cpGBnr= GenBank accession number of full plastome cpDNA record (e.g. KR967361)

mincplength=200 Minimum sequence length in plastome reference (to be generated by script 0f)

7. PATH TO DATA

othersource= Name of the folder with other transcriptomes/genomes to combine with Hyb-Seq data. This folder must be in ‘homedir’.

otherpslx= Name of the folder with PSLX files to combine. This folder must be in ‘homedir’.

otherpslxcp= Name of the folder with cpDNA PSLX files to combine. This folder must be in ‘homedir’.

8. SOFTWARE BINARIES AND NUMBER OF CORES

raxmlseq= Name of the binary for sequential version of RAxML (raxmlHPC, raxmlHPC-SSE3, or raxmlHPC-AVX).

raxmlpthreads= Name of the binary for Pthreads version of RAxML (raxmlHPC-PTHREADS, raxmlHPC-PTHREADS-SSE3, or raxmlHPC-PTHREADS-AVX).

fasttreebin= Name of the binary for FastTree (e.g., fasttree, fastreemp, fasttreeMP…).

astraljar= Name of the ASTRAL jar file. This file must be in ‘HybSeqSource’ folder together with ‘lib’ folder - see Astral homepage.

astridbin= Name of the binary for ASTRID (ASTRID, ASTRID-linux, or ASTRID-osx). This file must be in ‘HybSeqSource’ folder.

examlbin= Name of the binary for ExaML (examl, examl-AVX, or examl-OMP-AVX).

numbcores= Number of cores/threads available (not applicable for clusters where number of cores is set using PBS and passed through env variables).

9. PARALLELIZATION SETTINGS

parallelmafft= Whether to compute MAFFT alignments in parallel using GNU ‘parallel’ command (yes/no).

parallelraxml= Whether to use parallelization of RAxML gene tree reconstruction (for cluster environment only).
yes=parallel jobs will be submitted to the cluster (fast), see next parameter
no=all RAxML calculations will be done serially (slow)

raxmlperjob= A number defining how many RAxML calculations will be done per single submitted job (number of jobs = number of genes / raxmlperjob). E.g., with 600 genes and raxmlperjob=20, 30 jobs will be submitted to the cluster.

10. MAPPING AND CONSENSUS SETTINGS, HETEROZYGOSITY

mappingmethod= Which method is used to map reads to ‘psudoreference’ (bowtie2/bwa).

mapping= Whether to do mapping to ‘psudoreference’ or consensus calling only (yes/no).
yes=mapping and consensus calling is done
no=only consensus calling is done (this allows to try effect of different coverage). Works only if mapping was already done and BAM files are present in ‘21mapped’.

conscall= Whether OCOCO, kindel v.0.1.4 or ConsensusFixer is used for consensus calling (ococo/kindle/consensusfixer). OCOCO and kindel are used for majority rule consensus sequence whereas ConsensusFixer produces consensus with ambiguous bases.

mincov= Minimum site coverage for SNP calling (N will be in consensus for sites with lower coverage).

majthres= Majority threshold for consensus calling (0-1). Works only with kindel, not OCOCO (probably due to a bug in OCOCO?).

plurality= Minimal relative frequency of alternative base to call as ambiguity (0-1). Only for ConsensusFixer.

nohetcalculation= Whether to do calculation of number of heterozygous sites per exon and sample (yes/no).

maxhet= Maximum number of heterozygous sites per exon to include it.

11. DATA DOWNLOAD SETTINGS

download= Whether data will at the beginning be downloaded from Illumina BaseSpace (yes/no) or SRA/ENA (sra). For BaseSpace it requires ‘token_header.txt’ in ‘homedir’ with your specific access code to Illumina BaseSpace. See Appendix 6 for advice how to obtain your personal token.

projectID= ID of the Illumina BaseSpace project from which the samples should be downloaded.

runID= ID of the Illumina BaseSpace run (only if a basic run statistics is required).

bsserver= Illumina BaseSpace API server (see https://developer.basespace.illumina.com/docs/content/documentation/cli/cli-overview#SpecifyAPIserverandAccessToken). E.g., ‘https://api.basespace.illumina.com’.

12. OTHER SETTINGS

AMAS= The way how to use AMAS when concatenating exons (fast/slow). Try 'slow' only if you observe 'Argument list too long' error when running scripts utilizing AMAS.

nrexons= Number of exons for generating test dataset (script 0d)

Appendix 6: How to obtain a personal access token for BaseSpace and use it for downloading FASTQ files within HybPhyloMaker.

Illumina BaseSpace is a cloud platform for storage of NGS runs and performing analyses. It allows web-based access to files that were generated during sequencing runs including resulting FASTQ files. However, BaseSpace also allows communication via its own API and download of files from command line. This is a useful feature, as downloads can be parallelized and data quickly downloaded directly to a computer cluster. You can do this using HybPhyloMaker:

1. Obtain access 'token' from Illumina BaseSpace (see steps 1-5 at https://support.basespace.illumina.com/knowledgebase/articles/403618-python-run-downloader)

- Register at http://basespace.illumina.com
- Go to https://developer.basespace.illumina.com and login
- Click on the “My Apps” link in the tool bar.
- In the applications tab, click on the “Create a new Application” button
- Fill out the Applications Details and then click the “Create Application” button
- In the Credentials tab, there is your “Access Token”

2. Save the token to a text file (`‘token_header.txt’`) with a one line text:
header = "x-access-token: <your-token-here>"
, e.g.,
header = "x-access-token: 127fg65dt57307q43we67fxf247i290h"

3. Login to BaseSpace via web browser and get IDs for

- project - Go to (via clicking) Projects -> <project-name>. Look at the address which should look like https://basespace.illumina.com/sample/28555179/files/tree/Z001_S1_L001_R1_001.fastq.gz?id=2016978377. Desired ID is the last number.
- run - Go to (via clicking) Runs -> <run-name>. Look at the address which should look like

4. Save these two IDs to `‘settings.cfg’` as ‘projectID’ and ‘runID’ in section ‘DATA DOWNLOAD SETTINGS’ and enable BaseSpace data download by setting ‘download=yes’. If you only set ‘projectID’ basic information about the sequencing run won’t be collected. However, all the FASTQ.gz file are downloaded.

Appendix 7: How to run HybPhyloMaker on MetaCentrum (useful tips).

1. Apply for MetaCentrum account
   - at the page http://metavo.metacentrum.cz/en/index.html click on *Getting an account* – Registration form and to your institution
   - login with your institutional credentials (CAS password at UK) and fill in the application. You get login information for MetaCentrum.
2. Get familiar with the documentation of the infrastructure (https://docs.metacentrum.cz/)
   - from Windows computer you can access (upload/download) data using, e.g. WinSCP or Total Commander with SFTP plugin installed (my preferred choice). See http://www.totalcmd.net/plugring/G_SFTP.html how to do that.
3. Access the frontend server and prepare the scripts & data for analysis
   - open terminal (Linux) or Windows PowerShell (Win+R, type powershell, Enter) and type this command to access frontend skirit (aka brno2)
     ```
     ssh metausername@skirit.ics.muni.cz
     ```
     and type password and press Enter
   - now you are in `/storage/brno2/home/$LOGNAME/`, where `$LOGNAME` is your login
   - clone HybPhyloMaker from GitHub
     ```
     git clone https://github.com/tomas-fer/HybPhyloMaker
     ```
   - move HybSeqSource folder to your home folder (IMPORTANT: mandatory)
     ```
     mv HybPhyloMaker/HybSeqSource/ .
     ```
   - test data are now in `HybPhyloMaker/testdata`, however, you can put the datafolder wherever you want, even on a different server (storage)
   - modify `settings.cfg` within HybPhyloMaker folder
     ```
     cd HybPhyloMaker
     nano settings.cfg
     ```
     now edit these settings

     location=1

     server=brno2

     data=HybPhyloMaker/testdata

     now Ctrl+O and Enter to save changes, Ctrl+X to exit editor
4. Now you need to install all required `R` and `julia` packages (all other software is already available via modules). Run these commands
   - `qsub HybPhyloMaker0c_Rsetup_MetaCentrum.sh`
   - `qsub HybPhyloMaker0d_julia_setup_MetaCentrum.sh`
5. You are now ready to run the scripts which is done with `qsub` command
   - e.g. `qsub HybPhyloMaker0b_preparereference.sh`
6. If you would like to put your data to a different server (storage), e.g. brno12-cerit you need
   - change the option 'server=' in the file '`settings.cfg`'
   - and then run R and Julia installation script again
   - and copy '`HybSeqSource`' folder to that server too (all required scripts/files are used from '`HybSeqSource`' folder on the server set in '`settings.cfg`', e.g. use this command
     ```
     cd && cp -r HybSeqSource /storage/brno12-
     cerit/home/$LOGNAME/
     ```

7. Before submitting a job you should check (and eventually change) general requirements (as time, memory and number of computer cores). Every script starts with a header, i.e. few lines starting with `#PBS`, where
    - `walltime` is maximum planned time in hh:mm:ss
    - `select` is number of machines (usually keep this 1)
    - `ncpus` is number of cores (can be increased if script calls software allowing multithreading/parallelization
    - `mem` is reserved memory (in mb or gb), this should be increased if working with large files
    - `scratch_local` is reserved size of the disk space, increase in case of large files

   In some cases, you might need also to modify such settings deeper in the script, see documentation for particular scripts.
8. The status (qeued/running) of jobs submitted using `qsub` command can be monitored
    - either by typing this command
      ```
      qstat -u $LOGNAME
      ```
    - or using this webpage (after logging)
      https://metavo.metacentrum.cz/pbsmon2/person and clicking on 'list of my jobs'
9. After finishing, every job creates an output file in the folder from which it was submitted
    - output file has a suffix starting with 'o'
    - the file contains both standard and error outputs of all software called by scripts as well as some specific comments
    - you should always control output file to check whether the job finished successfully
    - output file can be also used for troubleshooting and putative reasons of errors