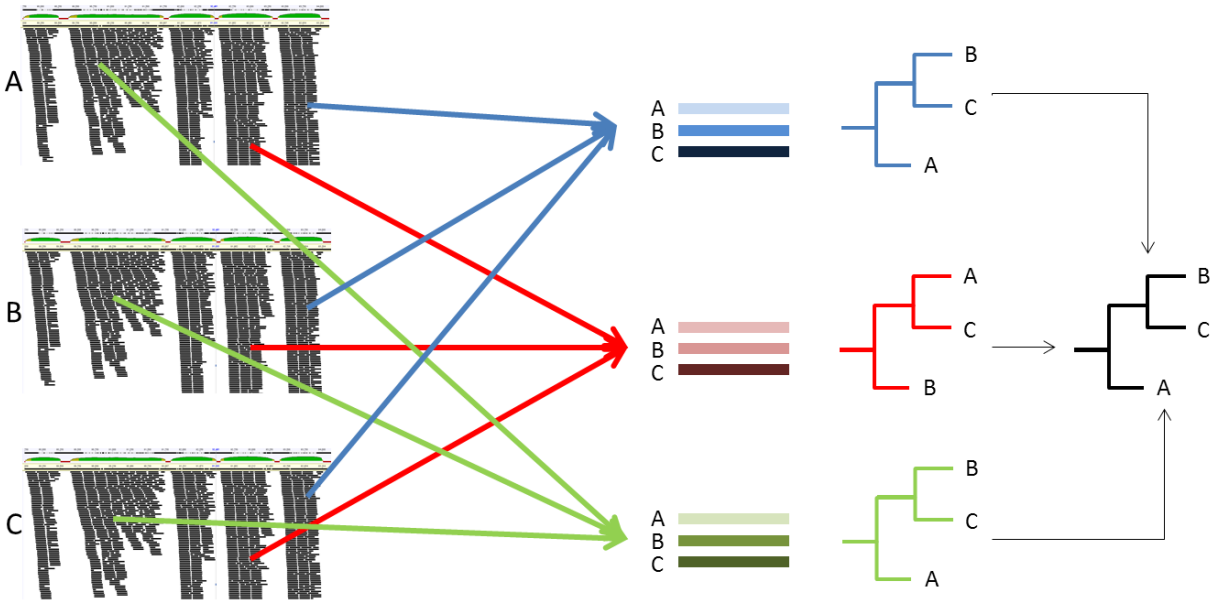
HybPhyloMaker

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

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



Version 1.3

5th December 2016

Tomáš Fér

tomas.fer@natur.cuni.cz

Department of Botany
Charles University, Prague
Czech Republic

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 distribution, MacOS X and under 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.

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. If you are running HybPhyloMaker on Linux, you might try to run `'install_software.sh'` which will install, download and compile all the necessary software and checks whether appropriate binaries are in PATH. Open the script in a text editor and set your distribution and package management tool. Automatic software installation was tested on 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. 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'` **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. After installing R you also need to install three R packages: ape (Paradis et al., 2004), seqinr (Charif et al., 2007) and data.table (https://cran.r-project.org/web/packages/data.table/). Refer to Appendix 3 how to do that locally or in the cluster environment. The script `'install_software.sh'` will do it for you.

IMPORTANT: Without installation of the appropriate R packages some scripts might not work properly and some plots 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 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'`). This structure is ready if you run `'install_software.sh'`.

IMPORTANT: This is just an example, there are more `HybPhyloMaker*.sh` **files in** `'homedir'` **and more files in** `'HybSeqSource'`**. 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.

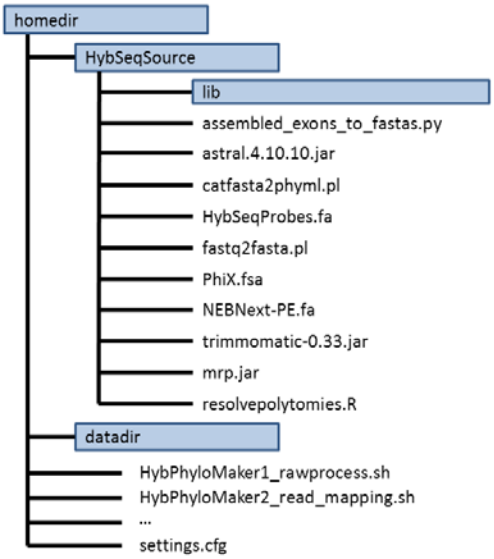


Fig.1: Folder structure before running HybPhyloMaker. Create `'homedir'` and put all BASH scripts (`'HybPhyloMakerX*.sh'`) and `'settings.cfg'` to it. In `'HybSeqSource'` folder there are all other supporting scripts, sequences of target enrichment probes, plastid coding genes, adapters and PhiX. In '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 2.6.).

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'`.

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 to use 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. 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 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'`.

2.6. 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 (and how) will be calculated 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'` 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).

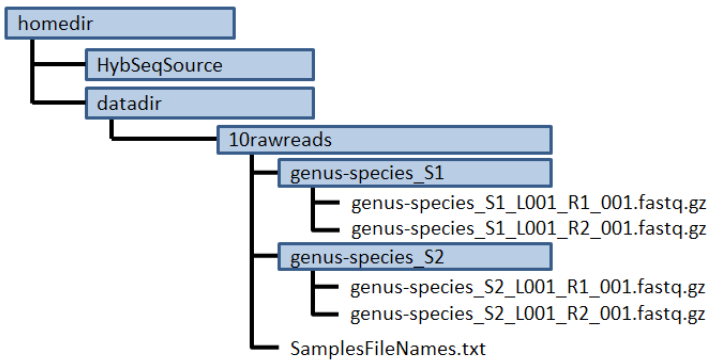


Fig. 2: Input data structure before running ‘HybPhyloMaker1_rawprocess.sh’. All input data (two *.FASTQ.gz files per sample) must be in a specific folder structure within ‘datadir/10rawreads’. Files from each sample must be in a separate folder named ‘Genus-species_Code’. The names of the *.FASTQ.gz 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’). 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 for the first and the last file you want to download 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.

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’) with the original number of all reads and the number of reads after each filtering step (stating also the percentage of reads that were filtered out).

Each sample-specific folder in ‘20filtered’ now contains six files with filtered reads:

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

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) 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 (see http://assets.geneious.com/documentation/geneious/GeneiousReadMapper.pdf). However, the mapping within HybPhyloMaker using Bowtie2 allow users to continue with the analysis from command line and without the necessity to buy a licence for Geneious. Both approaches were tested on several datasets and almost identical results (species trees) were obtained.

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. Bowtie is run with settings that are similar to the '--very-sensitive-local' preset option. The user might change these setting directly in the script. The resulting SAM file is converted to BAM file using samtools and saved to `'21mapped'` folder. Finally, a 0% majority rule consensus sequence is produced from the BAM file using OCOCO (Břinda et al. 2016) just from mapped reads, i.e., without considering 'pseudoreference' sequence. It means that in the case of two (or multiple) bases per site the base which is present in more than 50% reads is called and no ambiguities are introduced to the consensus sequence. In case of lower than minimum coverage (specified in 'mincov=') 'N' is called (and not the reference). The resulting sequence is saved in `'21mapped'` folder as `'Genus-species_Code.fasta'`. Consensus sequences from all samples are combined into the single multiFASTA file (`'consensus.fasta'`) and saved to `'30consensus'` folder within `'datadir'`.

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

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 `'exon'` subfolder must be also created) and put `'consensus.fasta'` there (Fig. 4).

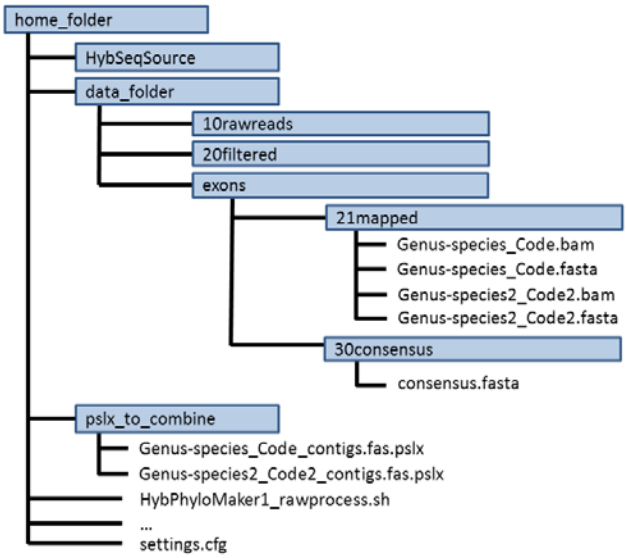


Fig. 4: Folder structure after running `'HybPhyloMaker2_read_mapping.sh'`.

4.3. Processing consensus sequences

Run the script `'HybPhyloMaker3_generatepslx.sh'` after putting `'consensus.fasta'` to the `'exons/30consensus'` subfolder (in case of you were mapping reads in Geneious). The script takes this consensus sequence multiple 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; Fér et al., in prep.).

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. Creating gene alignments

The script ‘HybPhyloMaker4_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 multiple 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 the Perl script ‘catfasta2phyml.pl’ and saved in the subfolder ‘exons/70concatenated_exon_alignments’ in ‘datadir’. Each ‘Assembly’ is saved in both *.fasta and *.phylip format.

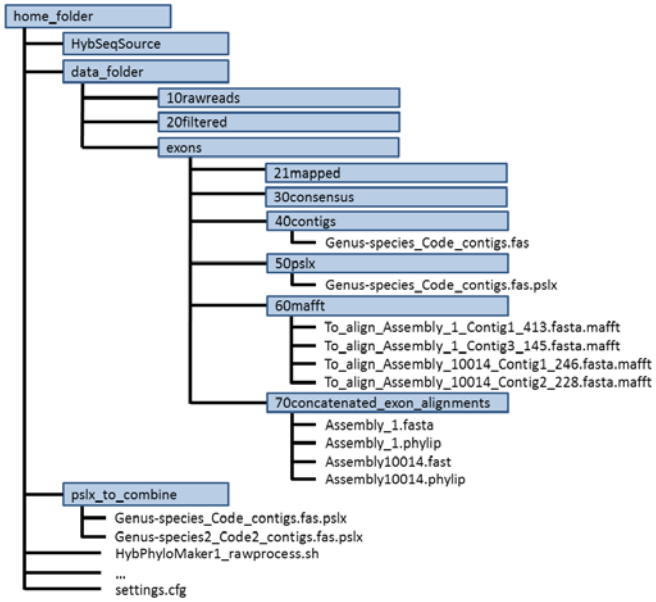


Fig. 5: Folder structure after running ‘HybPhyloMaker4_processpslx.sh’ (after processing the consensus sequences, generating PSLX files, aligning exon sequences and concatenating exons to genes). From now all results are written to the subfolder ‘exons’ (unless you specify ‘cp=yes’ in ‘settings.cfg’ – in this case a subfolder ‘cp’ is created and all the results are saved there; see 4.10.).

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.

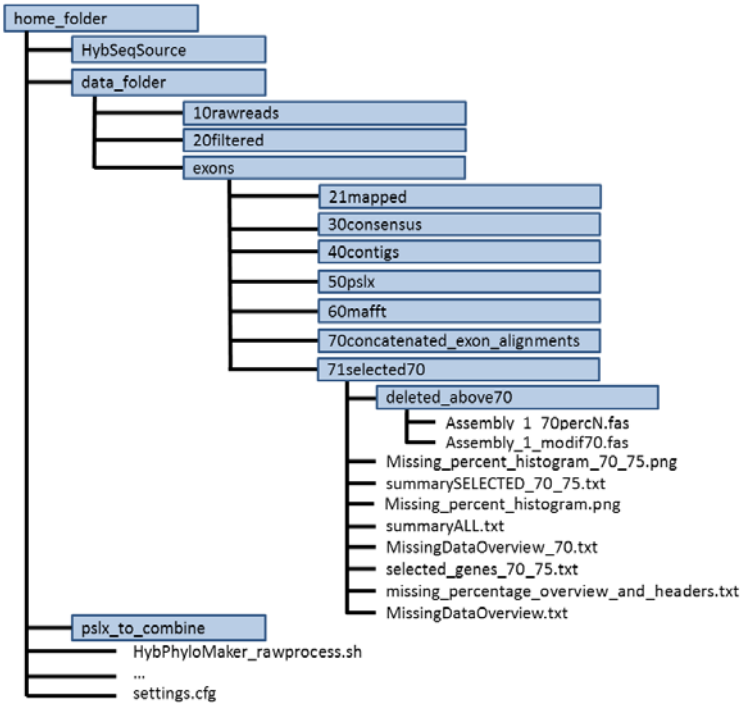


Fig. 6: Folder structure after running `'HybPhyloMaker5_missingdataremoval.sh'` (after counting the amount of missing data and deleting sequences and genes samples with more missing data than defined in 'MISSINGPERCENT' and 'SPECIESPRESENCE'). Genes selected for subsequent analyses are listed in `'selected_genes_MISSINGPERCENT_ 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. 6):

- 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. 7 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.).

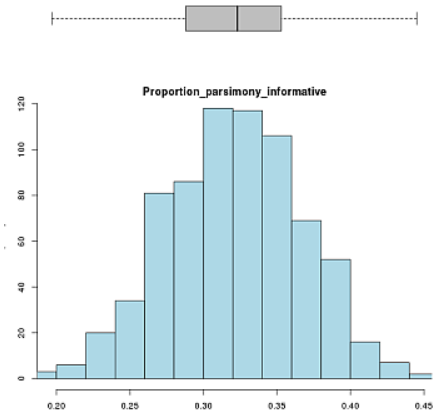


Fig. 7: Boxplot and histogram of the proportion of parsimony informative characters per gene alignment calculated with AMAS and plotted using R.

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.

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 (SH-like support) – very fast approach even on large datasets. However, we consider local branch support (highly) overestimated compared to bootstrapping approach (personal observation).
- 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 100 rapid bootstrap replicates; computationally demanding approach.

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 '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. 8). 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.

In case of 'FastTree' up to three files per gene are created (the last two files are created only in 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'`).

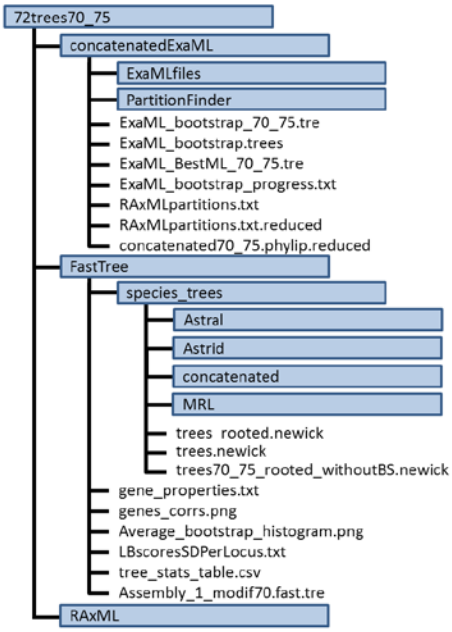


Fig. 8: Structure of the subfolder '72trees' for 'MISSINGPERCENT=70' and 'SPECIESPRESENCE=75'. The structure of the 'RAxML' subfolder is similar to that of the 'FastTree' subfolder. The final species trees are in individual subfolders (e.g., 'Astral') and not shown.

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', 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. 9). 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'.

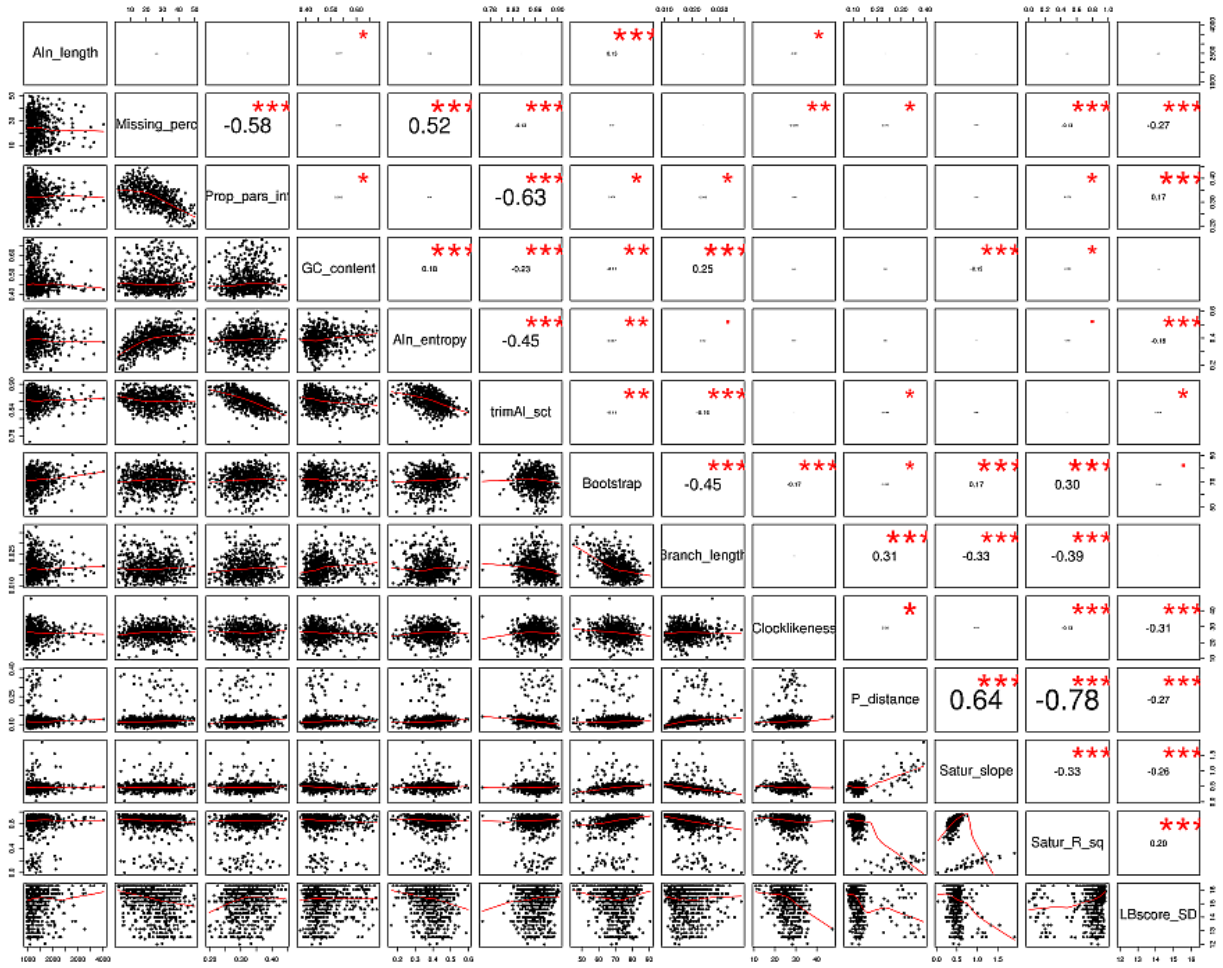


Fig. 9: Correlations among alignment and gene tree properties obtained by running `'HybPhyloMaker6b_FastTree_for_selected.sh'` respective `'HybPhyloMaker6a2_RAxML_trees_summary.sh'`.

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 multiple 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) and concatenation (ML in FastTree and ExaML). 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.

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'.

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. 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 will run only on the computer cluster and is not optimized for local run.

4.9.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:

- ‘genes_corrs_update.pdf’ – plot with correlations among pairs of selected properties for the updated selection of genes,

- ‘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 ‘HybPhyloMaker6_roottrees.sh’ and all desired ‘HybPhyloMaker7*.sh’ scripts. Species trees are now in the subfolder ‘exons/72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/update/species_trees’.

4.10. 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.

References

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.

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.

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

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.

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

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

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.

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.

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.

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

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

Sayyari E & Mirarab S (2016): Fast coalescent-based computation of local branch support from quartet frequencies. *Molecular Biology and Evolution*. doi: 10.1093/molbev/msw079

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.

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.

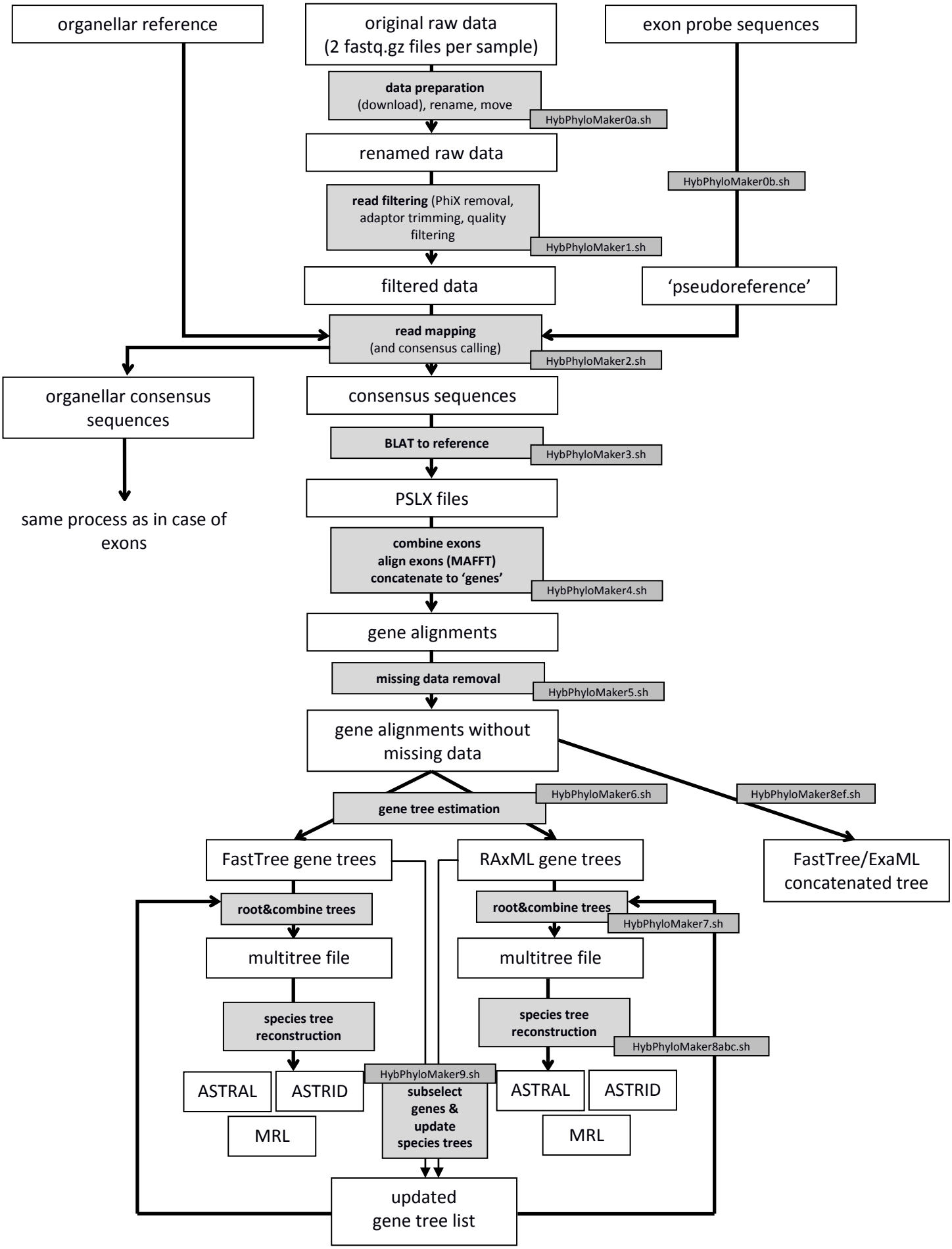
Tange O (2011): GNU Parallel - The Command-Line Power Tool. *;login: The USENIX Magazine* 1(36): 42–47.

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

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.

Appendix 1: HybPhyloMaker flowchart.



Appendix 2: Software to be installed prior to running HybPhyloMaker (in alphabetical order).

(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. **BLAT suite** (https://genome.ucsc.edu/goldenpath/help/blatSpec.html)
3. **Bowtie2** (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml)
4. **ExaML** (http://sco.h-its.org/exelixis/web/software/examl/index.html)
5. **FastTree** (http://www.microbesonline.org/fasttree/)
6. **FastUniq** (https://sourceforge.net/projects/fastuniq/)
7. **GNU parallel** (http://www.gnu.org/software/parallel/)
8. **JDK/JRE** (http://www.oracle.com/technetwork/java/javase/overview/index.html)
9. **MAFFT** (http://mafft.cbrc.jp/alignment/software/)
10. **MstatX** (https://github.com/gcollet/MstatX)
11. **Newick Utilities** (http://cegg.unige.ch/newick_utils)
12. **OCOCO** (https://github.com/karel-brinda/ococo)
13. **p4** (http://p4.nhm.ac.uk)
14. **Perl** (https://www.perl.org/)
15. **Python** (https://www.python.org/)
16. **Python3** (https://www.python.org/download/releases/3.0/)
17. **R** (https://www.r-project.org/) – at least v.3.1.
18. **RAxML** (http://sco.h-its.org/exelixis/web/software/raxml/index.html)
19. **SAMtools** (http://samtools.sourceforge.net/)
20. **TrimAl v1.4** (http://trimal.cgenomics.org/)

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

HybPhyloMaker uses R (at least v.3.1) 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'. Be sure that you have the most recent version of 'ape' (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"))`
    - Follow instructions
    - After finishing try `packageVersion("ape")` and you should get the answer `'3.5'`

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 add tools/R/3.2.1`
    - Run R by typing R
    - Type `install.packages(c("ape", "seqinr", "data.table"))`
    - Select a CRAN mirror by typing its number
    - After finishing try `packageVersion("ape")` and you should get the answer `'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. **assembled_exons_to_fastas.py** (https://github.com/listonlab/Hyb-Seq_protocol/blob/master/assembled_exons_to_fastas/assembled_exons_to_fastas.py)
3. **astral.4.10.10.jar** (https://github.com/smirarab/ASTRAL)
4. **ASTRID** (https://github.com/pranjalv123/ASTRID)
5. **catfasta2phyml.pl** (https://github.com/nylander/catfasta2phyml)
6. **combineboot.py** (*original part of HybPhyloMaker* based on http://p4.nhm.ac.uk/tutorial/combine_supports.html)
7. **CompareToBootstrap.pl, CompareTree.pl, MOTree.pm** (http://meta.microbesonline.org/fasttree/treecmp.html)
8. **fastq2fasta.pl** (http://brianknaus.com/software/srtoolbox/fastq2fasta.pl)
9. **histogram.r** (*original part of HybPhyloMaker*)
10. **LBscores.R** (*original part of HybPhyloMaker* and https://github.com/marekborowiec/metazoan_phylogenomics/blob/master/gene_stats.R)
11. **mrp.jar** (https://github.com/smirarab/mrpmatrix)
12. **NEBNext-PE.fa** (Oligonucleotide sequences © 2006-2010 Illumina, Inc. All rights reserved.)
13. **PhiX.fsa** (http://www.ncbi.nlm.nih.gov/nuccore/9626372)
14. **plotting_correlations.R** (*original part of HybPhyloMaker* and https://github.com/marekborowiec/good_genes/blob/master/plotting_correlations.R)
15. **tree_props.r** (*original part of HybPhyloMaker* and https://github.com/marekborowiec/good_genes/blob/master/tree_props.R)
16. **treepropsPlot.r** (*original part of HybPhyloMaker*)
17. **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: brno2, praha1, plzen1, budejovice1, brno6, brno3-cerit, brno9-ceitec, ostrava1.
data=	Name of the folder with data. This folder is within 'homedir'. e.g., data=testdata

2. 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)
OUTGROUP=	Specify outgroup for rooting both gene and species trees. e.g., OUTGROUP=Curcuma-longa_S01
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'.

3. MISSING DATA SETTINGS

MISSINGPERCENT=	All samples with $\geq$ 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 $\geq$ specified percentage (0-100%) of species per gene will be included in the final locus selection. e.g., SPECIESPRESENCE=75

4. TYPE OF DATA

cp=	Whether working with cpDNA. yes=working with cpDNA no=working with exons only

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).

5. 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).

6. 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'`.

7. 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.
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).

8. 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.

9. MAPPING AND CONSENSUS SETTINGS

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'`.
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). Not yet implemented due to bug in OCOCO. Set to 0.51 (OCOCO default).

10. DATA DOWNLOAD SETTINGS

download=	Whether data will at the beginning be downloaded from Illumina BaseSpace (yes/no). 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.
first=	ID for the FASTQ file of the first sample you want to download from Illumina BaseSpace. See Appendix 6 how to locate it.
last=	ID for the FASTQ file of the last sample to download. All samples with ID between 'first' and 'last' will be downloaded.

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

- forward read (R1) of the first sample in a run
- reverse read (R2) of the last sample in a run
- Go to (via clicking) Projects -> <project-name> -> Samples -> <sample-name> -> <file>.fastq.gz
- 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

4. Save these two IDs to `'settings.cfg'` as ‘first’ and ‘last’ in section ‘DATA DOWNLOAD SETTINGS’ and enable BaseSpace data download by setting ‘download=yes’.

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

To be added…