HybPipe

Pipeline for processing reads from target enrichment sequencing libraries (Hyb-Seq)

Tomáš Fér
Department of Botany, Charles University in Prague, Czech Republic
27 May 2016

1. Introduction

HybPipe is a set of BASH scripts for UNIX-like environment that is designed for compact and easy-to-use processing of raw Illumina pair-end reads originating from target enrichment libraries, selecting suitable loci and constructing species trees using different methods. Most of the scripts are wrappers around other bioinformatic and statistical software that must be installed prior to analysis. The pipeline is generally based on the analysis proposed by Weitemier et al. (2014). At the beginning reads are formatted to conform to HybPipe requirements and all results are later saved in a synoptic folder structure. In the middle of the pipeline Geneious software is used for read mapping to a reference.

HybPipe is intended for local use on UNIX-based computer systems (it was tested on Linux, MacOS and under Cygwin for Windows) or it can be run on a computer cluster with a job scheduling. All scripts are optimized for Czech National Grid Infrastructure (MetaCentrum) and Smithsonian Institution High Performance Cluster (SI/HPC) Hydra but they can be easily modified for use in other cluster environment.

2. Preparing data and software for the analysis

Before running your analysis following steps are usually necessary: (a) make a directory which is hereafter called `‘homedir’`, (b) download all the HybPipe files (including all folders) from GitHub (https://github.com/tomas-fer/HybPipe) to `‘homedir’`, (c) put your project specific files to `‘HybSeqSource’` directory that is within your `‘homedir’`, (d) install necessary software (see Table 1) and ensure that it is in the path, (e) install appropriate R packages, (f) prepare FASTQ.gz files (two per sample) with Illumina reads to `‘homedir’`, (g) edit analysis settings in the file `‘settings.cfg’` in `‘homedir’`.

2.1. Directory structure

HybPipe is working with dedicated folder structure. Each script creates its own `‘work’` directory within `‘homedir’`, copies all input and other necessary files in it and after finishing calculation copies results back to (newly created) subfolder(s) in home directory. By default `‘work’` directory is deleted after script finishes. Prepare following directory structure (i.e., copy all the data from

GitHub to 'homedir'; Fig.1). This is just an example, there are more HybPipe*.sh files in 'homedir' and more files in 'HybSeqSource':

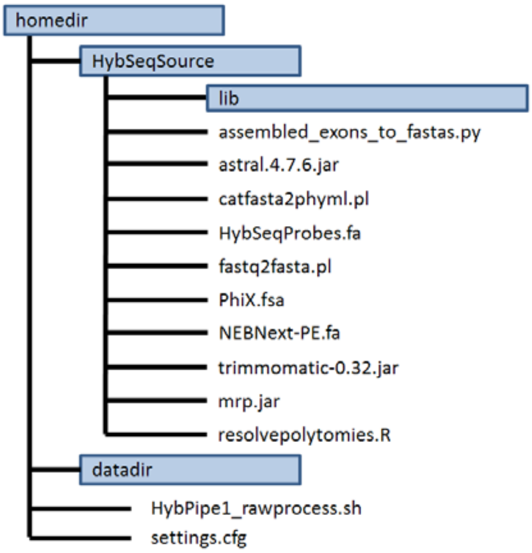


Fig.1: Folder structure before running HybPipe. Create 'homedir' and put all BASH scripts ('HybPipeX*.sh') and 'settings.cfg' into it. In 'HybSeqSource' folder there are all other supporting scripts, sequences of target enrichment probes and plastome coding genes. In 'datadir' there will be all input and output files.

2.2. Project specific files (to be put in 'HybSeqSource')

You must provide sequence references that are specific to your project and put it in 'HybSeqSource' folder. Names of these files must be specified in 'setting.cfg' (see section 2.5.).

2.2.1. Sequences of target enrichment probes

FASTA file with sequential (i.e., not interleaved, only one line per sequence) sequences of target enrichment probes. The names within this file must follow this scheme: >Assembly_geneNumber_exonNumber_whatever (e.g., >Assembly_1_Contig_1_413). The word 'Assembly' is mandatory. Files with the same 'geneNumber' will be later merged to a single file (exon concatenation). Specify name of the file in 'probes=' in 'settings.cfg'.

2.2.2. Sequences of plastome coding genes

FASTA file with coding sequences of chloroplast reference. The names within this file must follow this scheme: Number_number_geneName (e.g. >008854573_1_rps12). Such file can be obtained from GenBank if you extract coding sequences only from reference plastome. Specify name of the file in 'cpDNACDS=' in 'settings.cfg'.

2.3. Installation of all necessary software

Install all the software that is necessary for successful run of HybPipe (Table 1) following instructions on webpages of their developers. Ensure that all the software is in PATH and can be called from anywhere.

2.4. Installation of R packages

HybPipe requires R on several scripts for calculating alignment and tree properties and for plotting boxplots, histograms and correlation plots. After you installed R you need also to install several R packages. Refer to Appendix 5 how to do it locally or for the cluster environment. Without installing appropriate packages some scripts might not work and some plots will not be generated.

2.5. Input FASTQ files

Input Illumina FASTQ files should be pair-end and gzipped. Put these FASTQ.gz files (two files per sample) to `'homedir'`. Prepare file for automatic renaming of FASTQ files that will include two values per line separated by TAB. First value is the desired sample name that must follow this scheme: Genus-species_Code (e.g., Curcuma-longa_S01). Second value is a first part of FASTQ file name for this samples (e.g., Z1065 if FASTQ file name is Z1065_S1_L001_R2_001.fastq.gz). Avoid use of '-' in original FASTQ.gz files. Name the file `'renamelist.txt'` and save it to `'homedir'`.

2.6. Edit analysis settings

Open the file `'settings.cfg'` from your `'homedir'` and edit appropriate values. See Appendix 2 for thorough explanation of all values.

3. Running the pipeline

Now you are ready to run the HybPipe. The whole pipeline consists of several numbered BASH scripts that typically must be run in this order. The initial script is numbered '0' and serves for data preparation and renaming (and optional download from Illumina BaseSpace).

3.0. Prepare input files for analysis

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

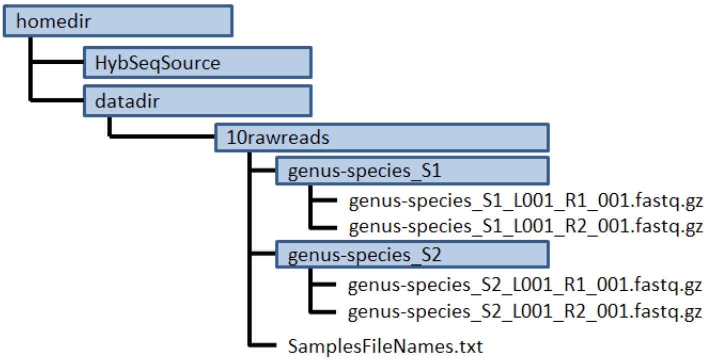

homedir
HybSeqSource
datadir
10rawreads
genus-species_S1
genus-species_S1_L001_R1_001.fastq.gz
genus-species_S1_L001_R2_001.fastq.gz
genus-species_S2
genus-species_S2_L001_R1_001.fastq.gz
genus-species_S2_L001_R2_001.fastq.gz
SamplesFileNames.txt

Fig. 2: Input data structure before running HybPipe. 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'. Also names of *.fastq.gz files must follow these convention. A list of all samples ('SamplesFileNames.txt') must be also provided.

This could be done by hand but the recommended option is to put all fastq.gz files to 'homedir' together with 'renamelist.txt' (see section 2.4.) and run the script 'HybPipe0a_preparedata.sh'. The script will accordingly rename fastq.gz data, create necessary folder structure, move the files to appropriate folder and create a list of files ('SamplesFileNames.txt'). In case you have your data stored in Illumina BaseSpace you can use this script for data download. To access Illumina BaseSpace you need to have a personal access token to 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 don't forget to set the option 'download=yes'. Consult the Appendix 3 how to reach these IDs and how to get a personal token.

3.1. *Raw read filtering*

Once all the data are in '10rawreads' you can start with first analysis running 'HybPipe1_rawprocess.sh'. This script does following operations and creates a subfolder '20filtered' with subfolder for each sample:

- removal of PhiX DNA (PhiX index created using bowtie2-build, reads are mapped to this index using bowtie2 and removed using bam2fastq
- adapter and quality trimming using Trimmomatic
- duplicate removal using fastx_collapser (FASTX-Toolkit)
- create summary table ('reads_summary') with original number of all reads and number of reads after each filtering step (incl. % of reads filtered out)
- all filtered reads without duplicates are saved to a single FASTA per sample and tar gzipped file containing all these files is created in a subfolder 'for_Geneious'

Each sample-specific folder in '20filtered' now contains two files with reads ('*-all.fa' with all reads after filtering and '*-all-no-dups.fas' with reads without duplicates) 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 in '20filtered'. In the subfolder 'for_Geneious' there is tar gzipped file ('*-all-no-dups.tar.gz') containing all sample files to be imported to Geneious for read mapping (see next step).

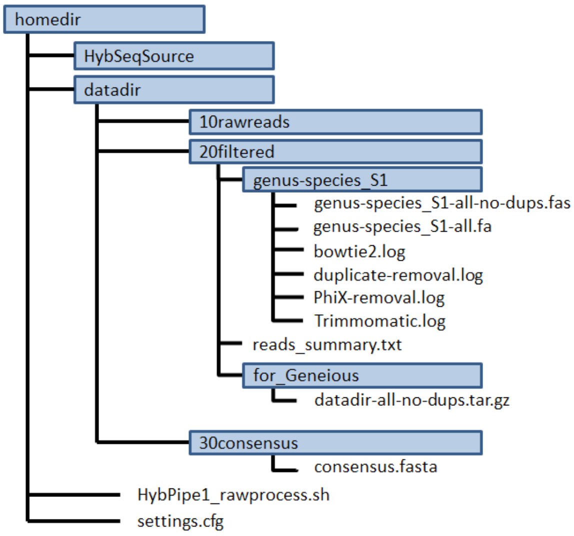


Fig. 3: Folder structure after raw data filtering. Read filtering summary is in 'reads_summary.txt'. A single *.tar.gz file is created in `'20filtered/for_Geneious'` and is ready to download and included files should be imported to Geneious for read mapping. After mapping consensus sequence is exported and copied to `'30consensus'`.

3.2. Read mapping in Geneious

Untar and unzip a file `'*-all-no-dups.tar.gz'` in folder `'20filtered/for_Geneious'` and import all these files to specific folder in Geneious. Now you need to prepare a 'pseudoreference' from all your target exons. 'Pseudoreference' contains all exon sequences separated by stretches of 400 Ns. There are 400 Ns also at the beginning and at the end of the 'pseudoreference'. Run a script `'HybPipe0b_preparereference.sh'` and a new file (`'*_with400Ns_beginend.fas'`) will appear in `'HybSeqSource'` folder. Import this 'pseudoreference' to the same folder in Geneious.

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

a. Data
    i. Refrerence 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 contings
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 like 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 0% majority rule consensus of reads mapped to 'pseudoreference' and is roughly of the same length as 'pseudoreference'. Sequences of individual exons (and adjacent introns before and after each exon) are separated by stretches of '?' due to 400 Ns between each exon in the 'pseudoreference'.

Create a directory `'30consensus'` in `'datadir'` and put `'consensus.fasta'` there (Fig. 3).

3.3. Processing consensus sequences

Run the script `'HybPipe2_generatepslx.sh'` after you put `'consensus.fasta'` to `'30consensus'` subfolder. The script takes this consensus sequence multiple FASTA file (exported from Geneious) and do following:

- split it to individual files (one file per sample)
- in each file the sequence is separated to smaller pieces corresponding to exons (stretches of '?' are replaced by a newline character) and saved to subfolder `'40contigs'` with name `'genus-species_Code_contigs.fas'`.
- each sequence in `'*_contigs.fas'` is then compared to original exon sequences (in a file with target enrichment probes) using BLAT and results are saved as 'pslx' file in subfolder `'50pslx'`.

When searching for similarity between consensus and probes the minimum similarity threshold (minident= in `'settings.cfg'`) highly influence the number of similarity hits. The default value is 90 but it can be lowered to 85 or 80 when trying to do analysis of distantly related species (at the level of whole family or even order, see Fér et al. in prep.)

Before continuing with next step all 'pslx' files should 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' from multiple analyses or just subselect samples and continue analysis with desired samples only.

Note: In this step it is possible to "mine" other data sources (transcriptomes, genome CDS or whole genomes) for sequences similar to target exons. Save FASTA-formatted sequences (important: follow the naming conventions gene-species_Code and add a suffix *.fas, e.g. `'Curcuma-longa_JQCX.fas'`) in a new subfolder in `'homedir'` and specify its name in ‘settings.cfg’ (option ‘othersource=’). They 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’.

3.4. Creating gene alignments

Script ‘HybPipe3_processpslx.sh’ takes all ‘pslx’ files saved in subfolder specified under option ‘otherpslx=’ (in ‘settings.cfg’), e.g. ‘otherpslx=pslx_to_combine’ and process them (Fig. 4):

- using a python script ‘assembled_exons_to_fastas.py’ (Weitemier et al. 2014) the consensus sequences of the same exon from each sample are combined to a single multiple FASTA file
- all FASTA files are aligned using default option with MAFFT (if parallelmafft=yes aligning is done passing through GNU parallel command). MAFFT alignments are saved in subfolder ‘60mafft’ in ‘datadir’.
- exon alignments belonging to the same gene (this is specified in the exon name in 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 together using perl script ‘catfasta2phyml.pl’ and saved in subfolder ‘70concatenated_exon_alignments’ in ‘datadir’. Each ‘Assembly’ is saved in both *.fasta and *.phylip format.

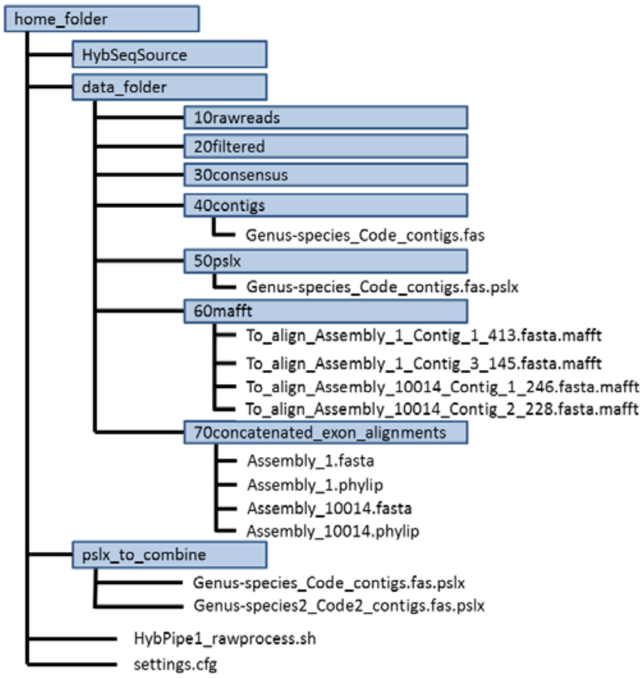

home_folder
HybSeqSource
data_folder
10rawreads
20filtered
30consensus
40contigs
Genus-species_Code_contigs.fas
50pslx
Genus-species_Code_contigs.fas.pslx
60mafft
To_align_Assembly_1_Contig_1_413.fasta.mafft
To_align_Assembly_1_Contig_3_145.fasta.mafft
To_align_Assembly_10014_Contig_1_246.fasta.mafft
To_align_Assembly_10014_Contig_2_228.fasta.mafft
70concatenated_exon_alignments
Assembly_1.fasta
Assembly_1.phylip
Assembly_10014.fasta
Assembly_10014.phylip
pslx_to_combine
Genus-species_Code_contigs.fas.pslx
Genus-species2_Code2_contigs.fas.pslx
HybPipe1_rawprocess.sh
settings.cfg

Fig. 4: Folder structure after processing consensus sequences, generated ‘pslx’ files, aligning exon sequences and concatenating exons to loci using script ‘HybPipe3_processpslx.sh’.

3.5. Deleting sequences and genes with too much missing data

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

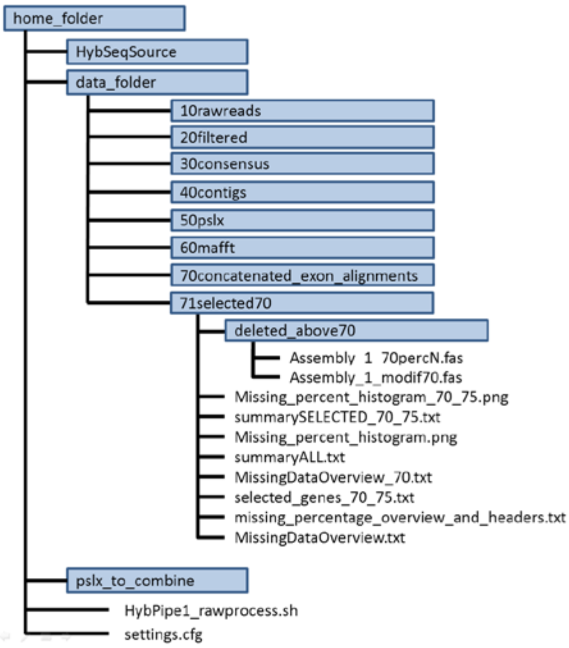


Fig. 5: Folder structure after counting amount of missing data and deleting samples with more missing data than defined in 'MISSINGPERCENT'. Genes selected for subsequent analyses are listed in 'selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt'. Histograms for other alignment characteristics are also created (*.png pictures).

Edit both above mentioned values and run the script 'HybPipe4_missingdataremoval.sh' that loops over all gene alignments in subfolder '70concatenated_exon_alignments' and conducts following analyses and saves all results in folder '71selected' following by number specified in 'MISSINGPERCENT' (e.g., '71selected70'; Fig. 5):

- The amount of missing data per species in each gene alignment is calculated and alignment without samples with excessive missing data are saved in subfolder '71selectedMISSINGPERCENT/deleted_aboveMISSINGPERCENT'. The alignments are named 'Assembly_number_modifMISSINGPERCENT.fas', percentage of missing data per sample is in 'Assembly_number _MISSINGPERCENTpercN.fas'.

- Three tables summarizing the amount of missing data per sample and per 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 – missing data average and number of samples of completely missing data in a particular gene.
  - `‘MissingDataOverview_MISSINGPERCENT.txt’` – genes in rows, species in columns but all values higher that MISSINGPERCENT are replaced by ‘N/A’. Two more columns are added – missing data average (but now calculated only from values below MISSINGPERCENT), 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 specified percentage of all samples (SPECIESPRESENCE) is saved to `’selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt’`. This list is later used in following steps when gene trees are generated.
- Table with summary characteristics for all (`‘summaryALL.txt’`) and for selected (`‘summarySELECTED_MISSINGPERCENT_SPECIESPRESENCE.txt’`) genes are generated using AMAS (Borowiec 2016), MstatX, and TrimAl. These tables include (among others) following values for 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. 6 for an example). These plots allow for easy evaluation of distribution of these values across genes and for potential elimination of some outlier loci (see section 3.9.).

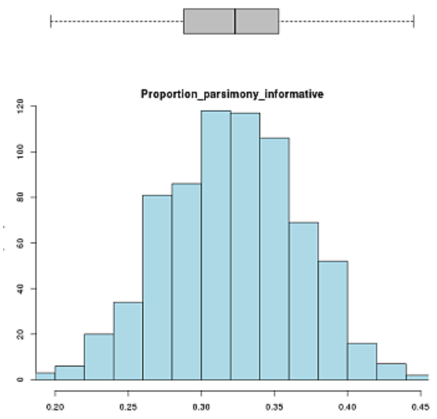


Fig. 6: Box plot and histogram of proportion of parsimony informative characters per alignment calculated with AMAS and plotted using R.

You can run loci selection several times with different settings of MISSINGPERCENT and SPECIESPRESENCE and several folders including above described files will be created. To continue in pipeline with concrete loci selection based on specific amount of missing data just change the appropriate parameters in 'settings.cfg'. You cannot continue before you do this missing data-based selection and a list of selected genes is produced.

3.6. Generate gene trees for selected loci

Phylogenetic trees for alignments specified in 'selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt' in subfolder '71selectedMISSINGPERCENT' are now generated using either FastTree, FastTree with bootstrapping, or RAxML with 100 rapid bootstrap replicates. Generally, FastTree is really very fast even with large datasets, RAxML is more computationally demanding. Both methods produce very similar trees, however, FastTree has a tendency to (highly) overestimate branch support using local support calculations.

Select tree building methods by editing 'tree=' option in 'settings.cfg' (either 'FastTree' or 'RAxML') and in case of 'tree=FastTree' choose whether to use bootstrapping ('FastTreeBoot=yes'). However, bootstrapping substantially increases the time necessary for tree building. In case of 'FastTree' the gene trees are constructed one by one, the 'RAxML' option allows (if run on cluster using option 'parallelraxml=yes') to generate several jobs for a subset of alignment each. All trees are stored in subfolder '72treesMISSINGPERCENT_SPECIESPRESENCE' where 'FastTree' or 'RAxML' subfolders are created (Fig. 7).

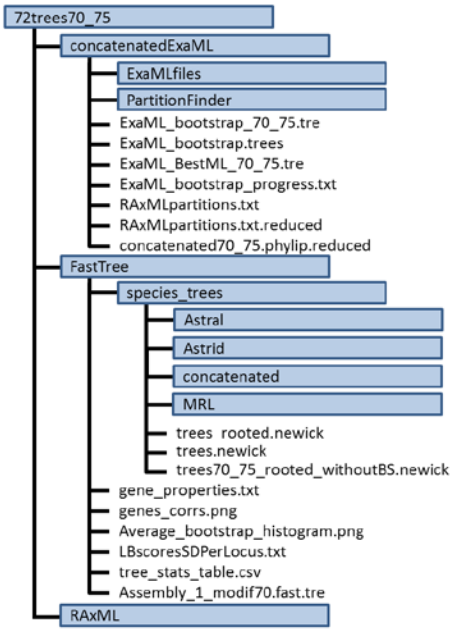


Fig. 7: Structure of subfolder '72trees' for 'MISSINGPERCENT=70' and 'SPECIESPRESENCE=75'. Structure of 'RAxML' subfolder is similar to 'FastTree'. Final species trees are in individuals subfolders (e.g., 'Astral') and not shown.

Several parameters are calculated based on trees (and alignments), saved in `'tree_stats_table.csv'` and visualized using mixed histogram/boxplot diagrams with custom R scripts (with the packages 'ape' and 'phytools'; `'tree_prop.r'` and `'treepropsPlot.R'`; modified from https://github.com/marekborowiec/good_genes):

- 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 coefficient of variation in root to tip distances and returns that value. Lower value is more clock-like, 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 taxon-specific long branch score defined by Struck 2014)

Alignment and tree properties are combined to a single file (`'gene_properties.txt'`) and correlations among all pairs of selected variables are computed and plotted to `'genes_corrs.png'` using R script `'plotting_correlations.R'` (modified from https://github.com/marekborowiec/good_genes; Fig. 8). This helps to recognize genes with extreme values of particular alignment or tree characteristics (e.g., saturated genes) and the summary table (`'gene_properties.txt'`) helps to distinguish, e.g., among slowly and fast evolving genes or among less and more variable genes and select those genes for subsequent phylogenetic analyses only (see 3.9.).

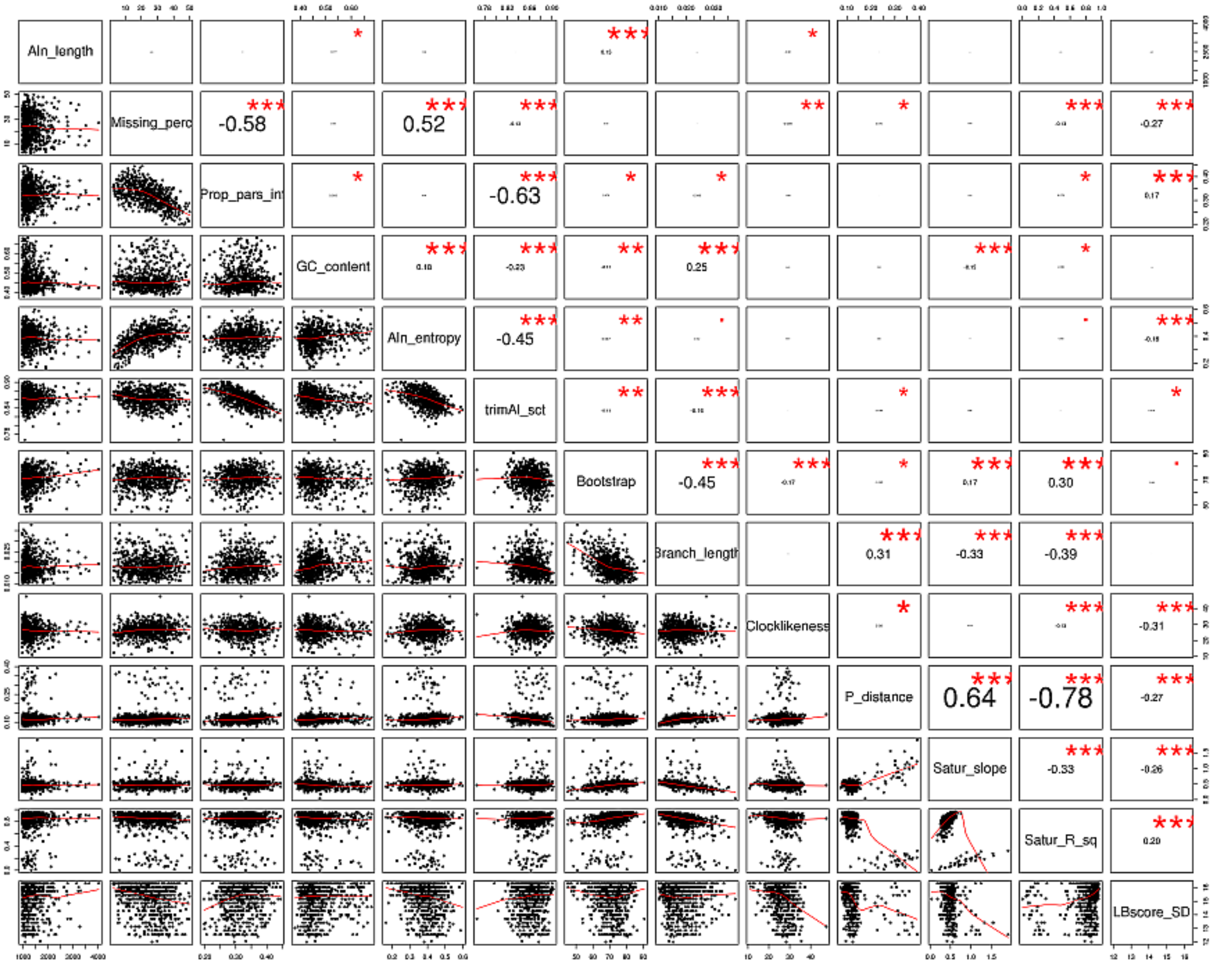


Fig. 8: Correlations among gene/alignment characteristics.

3.7. Root and combine gene trees

Running script `'HybPipe6_roottrees.sh'` all the 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 3.8.). 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. Following files are produced:

- `'trees.newick'` – all trees
- `'trees_rooted.newick'` – all trees rooted with outgroup using nw_reroot command
- `'treesMISSINGPERCENT_SPECIESPRESENCE_rooted_withoutBS.newick'` – all trees rooted with bootstrap support values removed with nw_topology command

3.8. Estimate species trees

Species trees are estimated using several methods including coalescence summary methods (ASTRAL, ASTRID, MP-EST), supertree method (MRL) or concatenation (ML in ExaML). There is one script for each method that either concatenates selected genes or use 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.

3.8.1 ASTRAL species tree

Run the script 'HybPipe7a_astral.sh' and ASTRAL species tree with branch lengths and branch support (local posterior probabilities) with the name 'Astral_MISSINGPERCENT_ SPECIESPRESENCE.tre' is produced using ASTRAL 4.10.2 and saved in subfolder '72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees/Astral'. If RAxML (of FastTree with real bootstrap) gene trees are summarized ASTRAL can also calculate multilocus bootstrapping on bootstrap replicate gene trees. In this case tree with bootstrap support values is saved to 'Astral_ MISSINGPERCENT_SPECIESPRESENCE_withbootstrap .tre'.

3.8.2 ASTRID species tree

Run the script 'HybPipe7b_astrid.sh' and ASTRID species tree (just topology) with the name 'Astrid_MISSINGPERCENT_SPECIESPRESENCE.tre' is produced using ASTRID (Vachaspati and Warnow 2015) and saved in subfolder '72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_trees /Astrid'. If RAxML (of FastTree with real bootstrap) gene trees are summarized ASTRID can also calculate multilocus bootstrapping on bootstrap replicate gene trees. In this case tree with bootstrap support values is saved to 'Astrid_ MISSINGPERCENT_SPECIESPRESENCE_withbootstrap.tre'.

3.8.3 MP-EST species tree

3.8.4 MRL species tree

Run the script 'HybPipe7d_mrl.sh' and a tree using supertree method MRL (Matrix Representation with Likelihood) is produced. First, a randomized binary matrix based on bipartition presences in gene trees is produced using mrp.jar (https://github.com/smirarab/mrpmatrix). Second, the matrix is analysed using heuristics for 2-state maximum likelihood tree ('BINCAT' model in RAxML) and 100 rapid bootstrap replicates are calculated. Final tree with bootstrap support values ('RAxML_bipartitions.MRLresult') is saved to subfolder '72trees_MISSINGPERCENT_ SPECIESPRESENCE/TREE/species_trees/MRP'.

3.8.5 Species tree based on concatenation (FastTree)

The script 'HybPipe7e_concatenatedFastTree.sh' allows running fast analysis of concatenated dataset using FastTree. First, concatenated dataset of selected genes listed in 'selected_genes_MISSINGPERCENT_SPECIESPRESENCE.txt' (in subfolder '71selected MISSINGPERCENT') is prepared using AMAS and saved in both FASTA and PHYLIP format in '72trees_MISSINGPERCENT_SPECIESPRESENCE/TREE/species_ trees/concatenated'. Then FastTree constructs the tree 'concatenated_MISSINGPERCENT _SPECIESPRESENCE.fast.tre'.

3.8.6 Species tree based on concatenation (ExaML)

More reasonable approach how to use concatenated dataset for species tree reconstruction is to apply partitioned analysis which allows modelling parameters for each partition (=gene, position etc.) separately. When running the script 'HybPipe7f_concatenatedExaML.sh' following steps are performed:

- concatenated dataset is prepared similarly to 3.8.5.
- 'partitions.txt' with partition description of concatenated alignment (produced by AMAS) is modified and a configuration file for PartitionFinder 2 ('partition_finder.cfg') is prepared. For simplicity and speed effectivity with large datasets (tens to hundreds of samples, hundreds of genes) following settings are involved: branchlengths = linked, models = GTR+G, model_selection = AICc, search = rclusterf. Consult PartitionFinder manual for other options.
- PartitionFinder is executed in order to find best partitioning scheme. All resulting files are saved to'72trees_MISSINGPERCENT_SPECIESPRESENCE/concatenated ExaML/PartitionFinder'. Check PartitionFinder documentation for information about files in this folder. Best scheme is saved to '72trees_MISSINGPERCENT_SPECIESPRESENCE/concatenatedExaML/ RAxMLpartitions.txt'. The script will check the presence of this file. If the file is found concatenation and PartitionFinder run are skipped and script continues with next step.
- RAxML checks whether concatenated alignment contains any entirely invariable positions and if yes prepare reduced alignment and modify partition file as well. Modified files are saved to 'concatenatedMISSINGPERCENT_SPECIESPRESENCE.phylip. reduced' and 'RAxMLpartitions.txt.reduced' in the subfolder '72trees_MISSINGPERCENT_ SPECIESPRESENCE/concatenatedExaML/'.
- best ML tree is estimated using ExaML and saved to 'ExaML_BestML_MISSINGPERCENT_SPECIESPRESENCE.tre'. This step is extremely computationally demanding and it is recommended to run it on computer cluster. MPI version of ExaML is used.
- 100 bootstrap support 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 time (in mins) necessary for each bootstrap replicates.

3.9. Select & Update

After you build gene trees (see 3.6.) 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. Just open the ‘gene_properties.txt’ in spreadsheet editor (e.g., Excel), sort according desired column and delete unwanted genes. Now, save the table as TAB delimited (or copy the whole table to text editor, e.g. Notepad++ on Windows) and save as `‘gene_properties_update.txt’` to `‘72trees_MISSINGPERCENT_ SPECIESPRESENCE/TREE/update’`. If on Windows be sure that there are UNIX-style end-of-line characters in this text file.

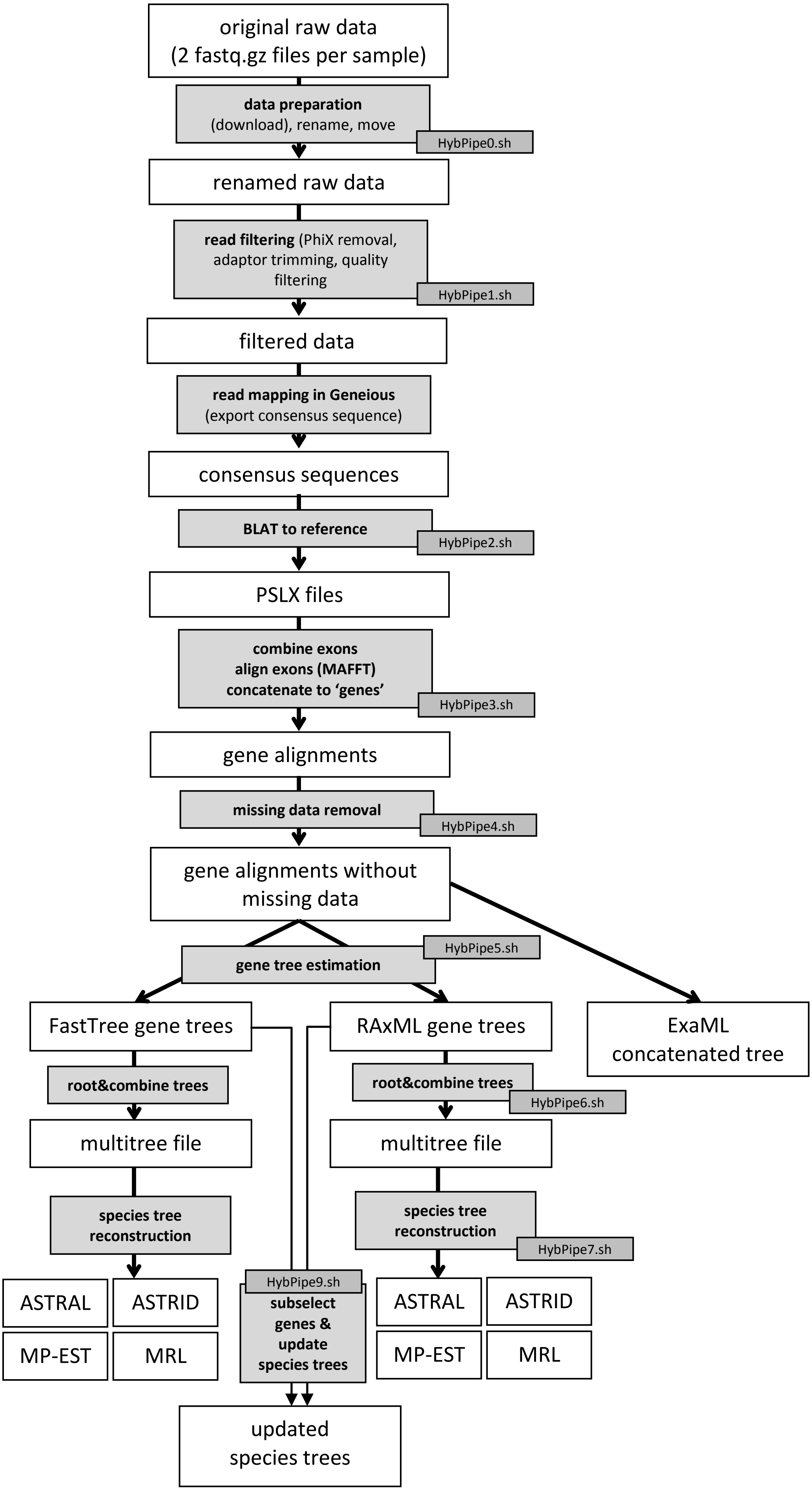
Run script `‘HybPipe9_update_trees.sh’` and following files are generated:

- `‘genes_corrs_update.pdf’` – plot with correlations among pairs of selected variables for updated selection of genes
- `‘selected_genes_70_75_update.txt’` in automatically created subfolder `‘/71selected MISSINGPERCENT/updatedSelectedGenes’`

Now you are ready to build species trees based on subselected genes only. First, change the option ‘update=’ to ‘update=yes’ in `‘settings.cfg’` and then (re)run `‘HybPipe6_roottrees.sh’` and all desired `‘HybPipe7*.sh’` scripts. Species trees are now in subfolder `‘72trees_MISSINGPERCENT_ SPECIESPRESENCE/TREE/update/species_trees’`.

3.10. Working with plastome and rDNA data

Appendix 1: HybPipe flowchart



Appendix 2: Explanation of settings in the file settings.cfg

1. GENERAL SETTINGS

location= Select whether you are running HybPipe locally or at the Czech National Grid (MetaCentrum)
0=locally
1=MetaCentrum

server= If running on MetaCentrum, select a server for input/output data. See Appendix 3 for advice how to run HybPipe on MetaCentrum. Possible values: 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=myanalysis

2. TREE SETTINGS

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

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

OUTGROUP= specify outgroup for rooting trees
e.g., OUTGROUP=Curcuma-longa_S01

3. MISSING DATA SETTINGS

MISSINGPERCENT= All samples with more than specified % of missing data (0-100) will be deleted from gene alignment
e.g., MISSINGPERCENT=70

SPECIESPRESENCE= Only loci with at least specified % of species (0-100) will be included
e.g., SPECIESPRESENCE=75

4. TYPE OF DATA

corrected= Put exons to correct reading frame (yes/no). Exon alignments are translated with all three possible reading frames, number of stop codons is calculated and the reading frame with least (usually 0) number of stop codons is used.

cp= Whether working with cpDNA (yes/no)
yes=working with cpDNA
no=working with LCN exons

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

5. REFERENCE FILES

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

pseudoref= Name of your pseudoreference (name used in FASTA file with pseudoreference, not filename!). This is used to filter out this name from Geneious output

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. PARALLELIZATION SETTINGS

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

parallelraxml= Whether to use parallelization of RAxML gene trees reconstruction (yes/no).
yes=more parallel jobs will be submitted to the cluster (fast), see next option
no=all RAxML calculations will be done serially (slow)

raxmlperjob= A number defining how many RAxML calculation will be calculated 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 cluster.

8. DATA DOWNLOAD SETTINGS

download= Whether data will be at the beginning downloaded from Illumina BaseSpace (yes/no). Requires ‘token_header.txt’ in ‘homedir’ with your specific access code to Illumina BaseSpace. See Appendix 3 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 3 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.

9. OTHER SETTINGS

bedfile= Name of BED file with specification of exons in ‘pseudoreference’ (must be in HybSeqSource folder). In this file there are beginning and end positions of exons. This is used to calculate per exon read coverage using JVARKIT (Bamstat04.jar).

binningsupport= Support value in statistical binning (not yet implemented…)

Appendix 3: How to obtain personal access token for BaseSpace and use it for download FASTQ files within HybPipe

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

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 text file (‘token_header.txt’) with 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 looks 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’ as well.

Appendix 4: How to run HybPipe on MetaCentrum (useful tips)

Appendix 5: How to install R packages before running HybPipe

HybPipe uses R 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. Following packages are necessary: ‘ape’, ‘sequinr’, ‘data.table’. Be sure that you have installed most recent version of ‘ape’ (3.5), some scripts are not working with version 3.4.

1. local use
   - run R
   - type `install.packages(c("ape", "phytools", "seqinr", "data.table"))`
   - follow instructions
   - after finishing try `packageVersion("ape")`and you should get answer `'3.5'`

2. MetaCentrum
   - run script `‘HybPipe0c_Rsetup_MetaCentrum.sh’` which does everything for you (packages are installed into writable library `‘Rpackages’` on your data server and later using

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", "phytools", "seqinr", "data.table"))`
   - select a CRAN mirror by typing its number
   - after finishing try `packageVersion("ape")`and you should get answer `'3.5'`