This README document describes the bioinformatic workflow for Cunha Neto et al. “Rampant convergent evolution of vascular oddities and a synnovation characterize the rapid radiation of Paullinieae lianas”.

**MAIN FOLDERS.**

* **Tree Inference**. This folder houses subfolders accounting for the full tree inference workflow: cleaning raw data 🡪 HybPiper Assembly 🡪 HybPiper Downstream Analyses 🡪 Sequence Alignments 🡪 Maximum Likelihood Tree Building 🡪 Chronogram.
* **Phylogenetic Comparative Methods**. This folder houses subfolders accounting for all phylogenetic comparative methods analyses: tests of correlated evolution, state-dependent diversification analyses, ancestral state reconstructions of vascular variant evolution.

**General notes about the content of each folder**

* Files ending in “.sh” are bash scripts
* Bash scripts are ordered from 00–09, indicating the order in which each script was performed throughout the bioinformatic pipeline
* Files ending in “.txt” are tab delimited text files
* Files ending in “.fa” and “.FNA” are fasta files
* Files ending in “.tre” are phylogenetic tree files

See below for a description of the contents and details of the workflow for each of the two main folders.

**Tree Inference Subfolders**

* **00–01.Batch1\_Trimmomatic&HybPiper\_Assembly**
* **00–01.Batch2\_Trimmomatic&HybPiper\_Assembly**
* **00–01.Batch3\_Trimmomatic&HybPiper\_Assembly**

*Description*: Given the computational constraints on the NYU High Performance Computer (HPC), the cleaning of raw data and the subsequent HybPiper assemblies of those cleaned data were processed separated into three batches. These three batches are contained in the following subfolders within the main folder “Tree Inference”.

* **02–05.HybPiper\_Downstream**

*Description:* This folder contains scripts and results to evaluate the HybPiper assembly, including creating a heatmap, sequence statistics, removing paralogs for downstream analysis, and the retrieval of sequences to generate alignments for tree inference.

* **06–07.Sequence\_Alignments**

*Description:* This folder contains scripts to align and trim exons. Also within this subfolder is the concatenated alignment that is used for the iqtree analysis (pau\_333s\_351g.fasta).

* **08.IQtree**

*Description*: This folder contains the script to run a partitioned maximum likelihood tree inference using IQ-TREE v. 1.6.12 on the mafft & post-trimal exon concatenated alignment. The iqtree results are presented in a subfolder named “iqtree\_results”. A script to change the tree tip names from sample ID (e.g., “PAU24”) to species names (“Serjania\_atrolineata\_b”) is available (ChangeSampleNames\_iqtree.sh)

* **09.ASTRAL**

*Description:* This folder contains the script to run individual gene trees for each of the 351 mafft + trimal exon alignments. The subfolders content: “trimmed\_exons” contains 351 post mafft +trimal alignments; “exon\_trees” contain 351 gene trees; “ASTRAL\_analysis” contains the input trees and result of an ASTRAL analysis.

* **10.CHRONOGRAM**

Given the size of the dataset (351 genes and 333 samples) a BEAST analysis was not feasible, so we opted to use treePL. Here we used the iqtree consensus tree as a topological constraint to make 1000 bootstrap trees with varying branch lengths. Then we subsampled 100 random bootstrap trees to individually time calibrate using treePL. Those 100 chronograms were collapsed using TreeAnnotator to generate a chronogram where the height 95%HPD is equivalent to the uncertainty given the variability of branch lengths among the 100 bootstrap chronogram trees.

**00–01.Batch1\_Trimmomatic&HybPiper\_Assembly**

**Scripts:**

*00.trimmomatic\_batch1.sh*

* This bash script used trimmomatic 0.39 to remove adapters and low-quality sequences from the raw data in batch 1.

*01.hybpiper\_batch1.sh*

* This bash script used HybPiper to generate consensus assemblies of Angiosperms353 loci data in batch 1.

**Files:**

*namelist\_batch1.txt*

* this is the list of sample names within batch 1.

*TruSeq3-PE.fa*

* This is the TruSeq adapter fasta file necessary to run 00.trimmomatic\_batch1.sh.

**00–01.Batch2\_Trimmomatic&HybPiper\_Assembly**

**Scripts:**

*00.trimmomatic\_batch2.sh*

* This bash script used trimmomatic 0.39 to remove adapters and low-quality sequences from the raw data in batch 2.

*01.hybpiper\_batch2.sh*

* This bash script used HybPiper to generate consensus assemblies of Angiosperms353 loci data in batch 2.

**Files:**

*namelist\_batch2.txt*

* This is the list of sample names within batch 2.

*TruSeq3-PE.fa*

* This is the TruSeq adapter fasta file necessary to run 00.trimmomatic\_batch2.sh.

**00–01.Batch3\_Trimmomatic&HybPiper\_Assembly**

**Scripts:**

*00.trimmomatic\_batch3.sh*

* This bash script used trimmomatic 0.39 to remove adapters and low-quality sequences from the raw data in batch 3.

*01.hybpiper\_batch3.sh*

* This bash script used HybPiper to generate consensus assemblies of Angiosperms353 loci data in batch 3.

**Files:**

*namelist\_batch3.txt*

* This is the list of sample names within batch 3.

*TruSeq3-PE.fa*

* This is the TruSeq adapter fasta file necessary to run 00.trimmomatic\_batch3.sh.

**02–05.HybPiper\_Downstream**

**Scripts:**

*02.hybpiper\_stats.sh*

* This command will summarize target enrichment and gene recovery efficiency for a set of samples. creating *seq\_length.csv* and *hybpiper\_stats.tsv.*

*04.hybpiper\_heatmap.sh*

* This bash script inputs *seq\_length.csv* to generate *recovery\_heatmap.png* of gene coverage per sample.

*04.hybpiper\_paralog.sh*

* This bash script searches for sequences flagged as paralogs and deletes them from the folders containing putative orthologous loci, so they are not included in downstream analyses

*05.hybpiper\_retrieve.sh*

* This bash script extracts the exons and supercontigs (introns and exons) and places them into separate directories.

**Files:**

*seq\_length.tsv*

* This file is the product of *02.hybpiper\_stats.sh*, detailing the length of each gene per sample.

*hybpiper\_stats.tsv*

* This file is the product of *02.hybpiper\_stats.sh*, detailing read mapping statistics per sample.

*recovery\_heatmap.png*

* This image file is the product of *04.hybpiper\_heatmap.sh,* showing the percentage coverage for each sample for each gene.

*namelist\_all.txt*

* This is the sample namelist file needed to run *02.hybpiper\_stats.sh*, *04.hybpiper\_paralog.sh*, and *05.hybpiper\_retrieve.sh*.

*Paullinieae\_140s\_352g\_target\_fixed.fasta*

* This is a custom fasta file composed of published assembled Angiosperms353 genes from Sapindaceae/Paullinieae used by HybPiper for sequence assembly.

**06–07.Sequence\_Alignments**

**Scripts:**

*06.mafft\_exons.sh*

* This bash script inputs the unaligned exons resulting from *05.hybpiper\_retrieve.sh*,then produces maaft alignments for each exon.

*07.trimal\_exons.sh*

* This bash script inputs the aligned exons from *06.mafft\_exons.sh*, then exports trimal-trimmed alignments.

**Files:**

*namelist\_regions\_order.txt*

* This text file lists all gene names. This file is needed to run *06.mafft\_exons.sh* and *07.trimal\_exons.sh*

*pau\_333s\_351g.fasta*

* This is the concatenated alignment of 333 samples and 351 mafft aligned and post trimal exons. To create this file, the alignments produced by *07.trimal\_exons.sh* were imported into Geneious and concatenated using the “Concatenate Sequence or Alignment” tool, where exons were order from lowest (clean\_4691) to highest number (clean\_7628).

**08.IQ-TREE**

**Scripts:**

*08.iqtree\_exons.sh*

* This bash script will generate the maximum likelihood tree of the concatenated alignment, using the partition gene file (pau\_333s\_351g\_partitions)

*iqtree.tre*

* This is the maximum likelihood tree with bootstrap support at the nodes with the tips assigned their species names (e.g., Paullinia\_atrolineata\_b)

*pau\_333s\_351g\_partitions.contree*

* This is the maximum likelihood tree with bootstrap support at the nodes with the sample ID numbers (PAU24)

*pau\_333s\_351g.fasta*

* This is the concatenated alignment of 351 genes across 333 samples. The sample ID names are still present.

*pau\_333s\_351g\_partitions*

* This is the partition gene file, showing the coordinates of each of 351 genes across the concatenated alignment.

*ChangeSampleNames\_iqtree.sh*

* This script uses the function “sed” to find and replace sample ID (e.g., PAU24) in *pau\_333s\_351g\_partitions.contree* species names (e.g., “Serjania\_atrolineata\_b”) in a file called *iqtree.tre*

**09.ASTRAL**

**Scripts:**

*09.astral\_exons.sh*

* This bash script will generate one gene tree for each of the alignments that was created by *07.trimal\_exons.sh.*

**Files**

*namelist\_regions\_order.txt*

* This text file lists all gene names. This file is needed to run *09.astral\_exons.sh*

**Subfolders: ASTRAL\_analysis:** This subfolder has files and scripts needed to generate a species tree using ASTRAL 5.7.8.

* + Tree Preparation scripts and files:
    - *exon\_trees.tre*
      * this is all “.contrees” (maximum likelihood consensus trees generated for each exon gene alignment)
    - *exon\_trees\_names.tre*
      * this is the exon\_trees.tre, but with the names corrected from sample ID (e.g.,“PAU24”) to species name (e.g., “Serjania\_atrolineata\_b”) by the ChangeSampleNames\_astral.txt script
    - *ChangeSampleNames\_astral.sh*
      * This uses the function “sed” to find and replace sample ID (e.g.,“PAU24”) in *exon\_trees.tre* to species name (e.g., “Serjania\_atrolineata\_b”) in a file called *exon\_trees\_names.tre*
  + ASTRAL Analysis files
    - astral.5.7.8.jar –– this is the astral program
    - lib –– these are the libraries associated with the astral.5.7.8.jar
    - README.md–– this is the readme for astral.5.7.8.jar
    - astral.tre–– this is the result of the astral analysis
    - astral.pdf–– this is the result of the astral analysis as a pdf

***The workflow to generate the ASTRAL tree is outlined below:***

1. Tree Preparation for ASTRAL:
   * In terminal….
     + navigate to the directory of exon trees
       - cd exon\_trees
     + combine all consensus trees
       - cat \*.contree > exon\_trees.tre
     + move exon\_trees.tre to ASTRAL\_analysis subfolder
       - mv exon\_trees.tre ../ ASTRAL\_analysis
     + Change sample ID names to species names to make exon\_trees\_names.tre
       - ./ChangeSampleNames\_astral.sh”
2. Run ASTRAL:
   * + In terminal…
       - In the same directory as the exon\_trees\_names.tre, place the astral.5.7.8.jar and the associated lib/ folder
       - Then run astral:
         * java -jar astral.5.7.8.jar -i exon\_trees\_names.tre -o astral.tre

This is the astral trees

**10.CHRONOGRAM**

**Subfolder: iqtree\_bootstrapTrees.** This subfolder contains input and output files for the iqtree constrained topology bootstrap analysis to generate 1000 trees with the same topology, but varying branch lengths.

* Important Input files/scripts:
  + - *iqtree2*
      * this is the iqtree2 program which has the constrained topology function
    - *iqtree\_bootstraps.sh*
      * this is the script to execute the iqtree bootstrap analysis to generate 1000 trees with varying branch lengths
    - *pau\_333s\_351g.fasta*
      * concatenated alignment of 333 samples and 351 exons alignments
    - *pau\_333s\_351g\_partitions*
      * this file contains the gene partition information
    - *pau\_333s\_351g\_partitions.contree*
      * this is the consensus ML tree generated by *08.iqtree\_exons.sh*, which will serve here as a topological constraint
* Important Output files:
  + - pau\_333s\_351g\_partitions.ufboot
      * result of this constrained iqtree boostrap analysis, containing 1000 bootstrap trees with the same topology, but varying branch lengths.

**Subfolder: treepl\_chronogram\_nosubsp.** This subfolder contains the input and output files associated with a treePL time-calibration analysis on the 100 randomly sampled bootstrap trees from the 1000 trees created through the constrained topology iqtree analysis.

**Scripts:**

* *1. extract\_bootstrap.py*
  + this script will randomly sample 100 bootstrap trees from the 1000 bootstrap tree file “pau\_333s\_351g\_partitions.ufboot” to create a file called “pau\_333s\_351g\_partitions.ufboot\_100.tre”
* *1a.ChangeSampleNames\_treePL.sh*
  + This file uses the function “sed” to find and replace sample ID (e.g., PAU24) in *pau\_333s\_351g\_partitions.ufboot\_100.tre* into species names (“Paullinia\_pinnata\_a”).
* *2.DropTips&RootTreesForChronogram.R*
  + This script uses the chronogram-taxa.txt namelist to prune each of the 100 input tree (*pau\_333s\_351g\_partitions.ufboot\_100.tre)*, and exports each pruned tree into folder called “prunedTrees”
* *3.TreePL\_loop.py*
  + This script uses treePL and the “config\_template.txt” file to loop through each of the pruned trees to time calibrate each of them and export each chronogram into a subfolder called “chronograms”
* 4.Chronogram.R
  + This script was used to take the raw chronogram tree file and make it into a figure for publication.

**Files:**

* *chronogram-taxa.txt*
  + This file list all the names to be included in the chronogram, which are one per taxa. All subspecies and varieties were collapsed under the species name. This file is needed by DropTips&RootTreesForChronogram.R
* *config.txt* 
  + - This file tabulates the parameters for the penalized likelihood analysis on each of the 100 trees, including three calibration points. This file is needed 3.TreePL\_loop.py
* *pau\_333s\_351g\_partitions.ufboot\_100.tre*
* random subset of 100 of the 1000 boostrap trees extracted via 1.extract\_bootstrap.py to be used for treePL analyses.
* *pau\_333s\_351g\_partitions.ufboot\_100\_names.tre*
  + - random subset of 100 of the 1000 boostrap trees extracted via 1.extract\_bootstrap.py to be used for treePL analyses with sample ID (e.g.,“PAU24”) replaced by species name (e.g., “Serjania\_atrolineata\_b”)

***The workflow to generate the chronogram is outlined below:***

1. Generate bootstrap iqtrees with constrated topology and varying branch lengths

* In terminal…
  + Place the following files within a single directory:
    - *Iqtree2*
    - *pau\_333s\_351g.fasta,*
    - *pau\_333s\_351g\_partitions*
    - *pau\_333s\_351g\_partitions.contree*
  + Run *iqtree\_bootstraps.sh* 
    - This will create the output “pau\_333s\_351g\_partitions.ufboot”, which is the bootstrap trees

1. Sample 100 randomly selected trees to root and prune
   * In terminal
     1. python3 1.extract\_bootstrap.py
        1. this will randomly subset 100 trees from the 1000 trees within pau\_333s\_351g\_partitions.ufboot
     2. ./1a.ChangeSampleNames\_treePL.sh
        1. This will change sample ID to species names in each of the 100 trees
   * In R
     1. Run the script 2.DropTips&RootTreesForChronogram.R
        1. This will prune the tips down to the taxa listed in chronogam-taxa.txt, and generate the “prunedTrees” folder
2. Time calibrate trees
   * In terminal (navigate to the directory with “prunedTrees” )
     1. python3 3.TreePL\_loop.py
        1. this will use treePL to time calibrate each of the pruned trees and deposit them into a new folder called “chronograms”
3. Generate chronogram
   * In terminal (within the newly generated “chronograms” folder)
     1. cat \*.tre > combined\_chronograms.tre
     2. Open combined\_chronograms.tre in Figtree and export as a nexus tree 🡪 combined\_chronograms.nexus
   * Download and open TreeAnnotator (https://beast.community/treeannotator)
     1. Settings:
        1. Input Tree “combined\_chronograms.nexus”
        2. Target tree type “Maximum clade credibility tree”
        3. Node heights “Median Heights”
     2. Output Tree CI-combined\_chronograms.tre
        1. This is is the chronogram.

**Phylogenetic Comparative Methods**

**Subfolder: normal-simmaps.** This subfolder contains the code and data for the normal stochastic mapping analyses (Fig. S11 and S12).

* **Script:** simmaps\_vascular-categories.R (will need to source plot\_simmap.R)
* **Data:** DatasetS1.txt
* **Input Tree:** CI-combined\_chronograms
* **Simmap results:** simmap.trees\_vc.RDS
* **Script:** *simmaps\_vascular-patterns.R*  (will need to source plot\_simmap.R)
* **Data:** DatasetS1.txt
* **Input Tree:** CI-combined\_chronograms
* **Simmap results:** *simmap.trees\_vp.RDS*

**Subfolder: paramo-stochastic-maps:** This subfolder contains the code and data for ancestral state reconstruction using PARAMO (Fig. 4).

* **Script:** PARAMO\_stochastic-maps.R
* **Data:** data\_PARAMO.txt
* **Input Tree:** CI-combined\_chronograms
* **Simmap results:** results in subfolder called simmaps

**Subfolder: Phylogenetic signal.** This subfolder contains the code and data to estimate phylogenetic signal with vascular variants.

* **Script:** Phylogenetic\_signal.R
* **Data:** Dataset\_PhylogeneticSignal.csv
  + - Dataset of the presence and absence of vascular variants.
* **Input Tree:** CI-combined\_chronograms
  + - Original chronogram

**Subfolder: MEDUSA.** This subfolder contains the code, data, and input files for estimating diversification rate shifts using MEDUSA.

* **Script:** MEDUSA.R
* **Data:** Richness.csv
  + - The total number of species per genus (Table S5) is equally distributed across the species from the respective genus in the tree. This is the primary input.
* **Input Tree:** CI-combined\_chronograms\_duplicated
  + - The tree was duplicated using a text editor to create a “multiPhylo” object (for plotting reasons)
* **Additional files/Outputs:** *Medusa\_results.rdata*: contains the objects of the analysis.

**Subfolder: BAMM.** This subfolder contains the code, data, and input files for estimating diversification rate shifts using BAMM.

* **Script:** BAMM.R
* **Data:** *sample\_probs.txt*
  + - Sampling fraction is similar to Richness in MEDUSA but formatted for BAMM. This is the primary input.
* **Input Tree:** CI-combined\_chronograms
  + - Original chronogram
* **Additional files:**
  + **Intermediate outputs/inputs**
    - *Paullinieae\_V1.tree*
      * A new Newick tree is created at the beginning of the analysis.
    - *myPriors.txt*
      * Output of the BAMM analysis; used to indicate the parameter of the phylogeny in the *BAMM\_control\_Paullinieae.txt*.
    - *BAMM\_control\_Paullinieae.txt*
      * File used to indicate parameters for BAMM analysis; it describes the name and outputs of the following intermediate files:
        + *chain\_swap.txt*
        + *Paullinieae\_event\_data.txt*
        + *Paullinieae\_mcmc\_out.txt*

**Subfolder: HiSSE.** This subfolder contains the code and data for estimating trait-dependent diversification using HiSSE.

* **Script:** HiSSE\_VascularVariants.R
* **Data:** Dataset\_HiSSE.csv
  + - Dataset of the presence and absence of vascular variants, tendrils, lianas, and zygomorphic flowers.
* **Input Tree:** CI-combined\_chronograms
  + - Original chronogram
* **Additional files:** 
  + **Intermediate outputs/inputs**
    - *Paullinieae\_hisse.rdata*
      * Contains the objects from the model selection step.
    - *Recon\_objects.rdata*
      * Contains the objects from the model averaging step.