Applications and pitfalls of multilocus amplicon sequencing with Oxford Nanopore technology

Martin Kapun / NHMW
Sonja Steindl / NHMW
Astra Bertelli / UniPV, NHMW

NOBIS Meeting, Vienna, 28/11/2024

TETTRIs

naturhistorisches museum wien

Wien 2024 NOBIS AUSTRIA 18

Funded by the European Union

Welcome & Important Info

Duration: 13:00 - 16:00

Wifi: NHM-Wien / KapLan

Password: jY5ixSuZnG / setudaki2664

Github repos:

https://github.com/nhmvienna/AmpliPiper

https://github.com/nhmvienna/AmpliPiper_Workshop_2024

Agenda of the day

1h Part I - AmpliPiper: Concepts and Modules

15’ break

45’ Part II - **Hands on:** Installation & data preparation

15’ break

45’ Part III - Hands on II: Advanced datasets, Limitations and Benchmarking

Team NHMW

Team: Collection

Nikola Szucsich

Helge Heimburg (LM Kärnten)

Oliver Macek

collecting; preserving; identification; ABOL database

Team NHMW

Team: Collection

Nikola Szucsich

Helge Heimburg (LM Kärnten)

Oliver Macek

collecting; preserving; identification; ABOL database

Team: Laboratory

Luise Krucken-hauser

Paula Schwahofer

Sandra Kirchner

Elisabeth Haring

Johannes Süß

DNA-extraction; QC; PCR; sequencing

Team NHMW

Team: Collection

Nikola Szucsich

Helge Heimburg (LM Kärnten)

Oliver Macek

collecting; preserving; identification; ABOL database

Team: Laboratory

Luise Krucken-hauser

Paula Schwahofer

Sandra Kirchner

Elisabeth Haring

Johannes Süß

DNA-extraction; QC; PCR; sequencing

Team: Bioinformatics

Astra Bertelli (Uni Pavia)

Sonja Steindl

Martin Kapun

basecalling; QC; Pipeline development

Addressing the biodiversity crisis

1 in 4 species are at risk of extinction

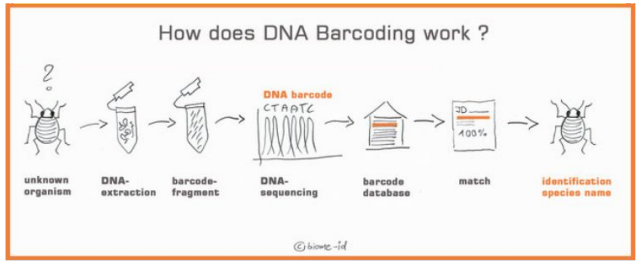
Wildlife populations reduced by **73%**

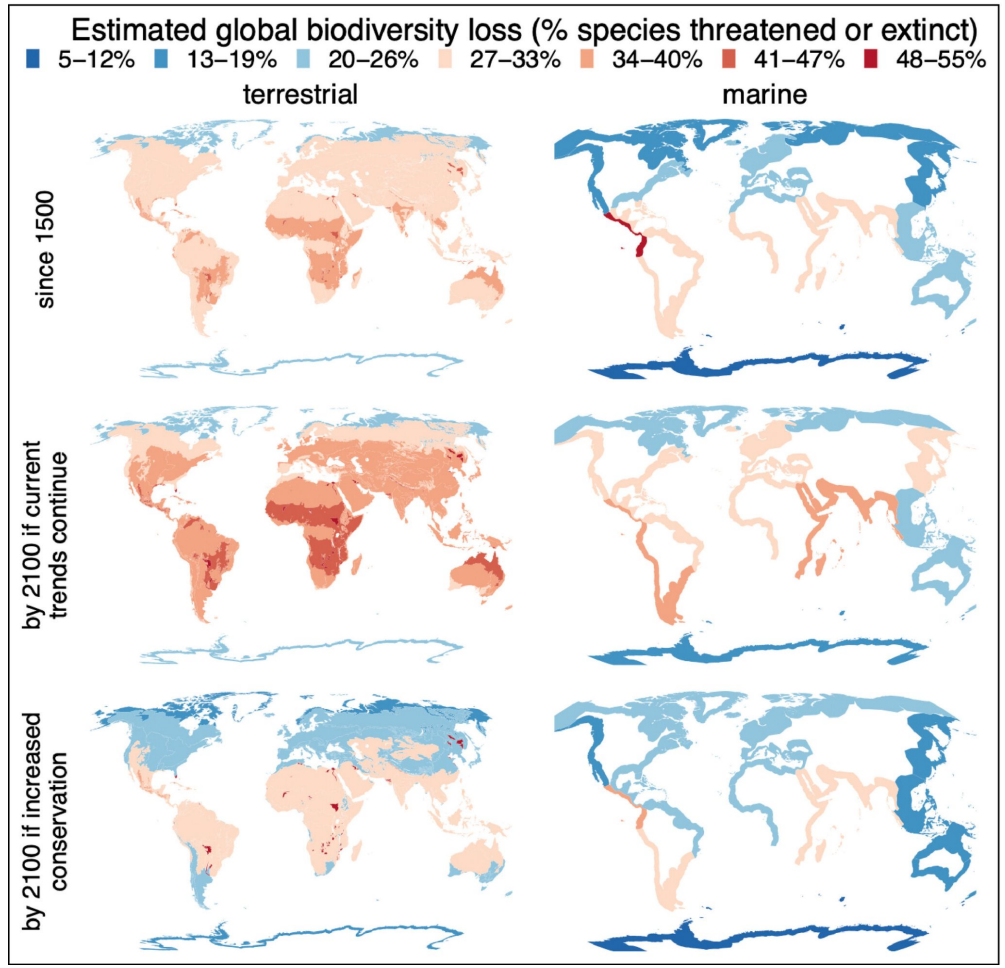
Dramatic loss is expected to happen in the next decades **if current trends continue**

Increased conservation efforts are key in lowering the loss

Quantifying biodiversity is fundamental

DNA Barcoding is a valid tool for quantification





Firest, I., et al. (2023) *Front Ecol Environ*

From single to multilocus barcoding

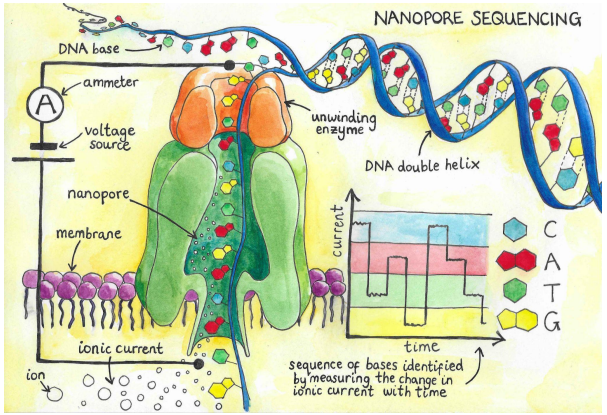
Most barcoding approaches based on **single loci** (COI, ITS, etc.) with primers that **amplify a broad set of taxa**

+ **Well-established protocols mostly for haploid (mt) markers**
+ **Many databases** built for these markers (BOLD, etc.)

- **Detection of hybridization** not possible
- **Incomplete lineage sorting** may confound species delineation
- single gene trees often do not reflect the **genealogy of species**
- **Missing scalability** with classic SANGER sequencing

ONT sequencing

+ allows **In-house sequencing**
+ Relatively **easy-to-use** NGS technology
+ **Cost-efficient** even with low sample throughput
+ **Fragment size** of amplicons is **not limited**
+ Growing set of **well-curated bioinformatic tools**
- **Error-prone**
- **Bioinformatics skills** necessary



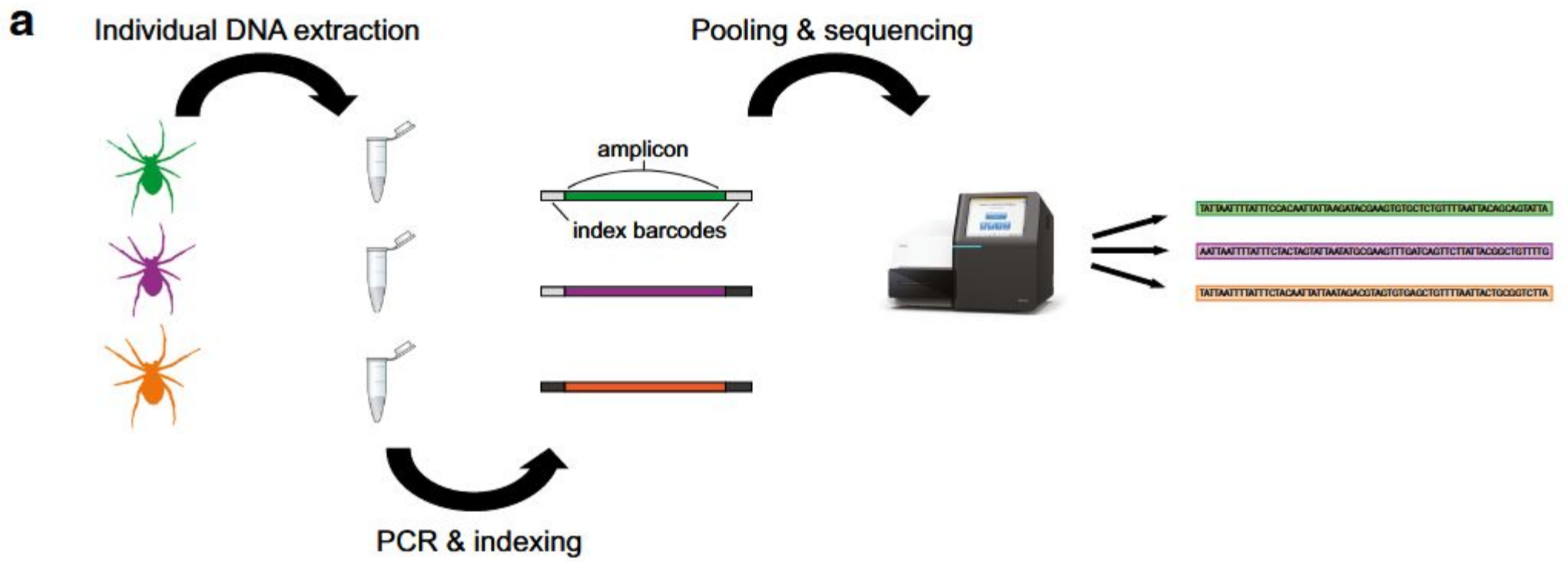
https://oxsci.org/pore-over-this-advances-in-dna-sequencing/

Multilocus DNA barcoding with NGS technology

+ **Multiple** mitochondrial and nuclear **loci** can be jointly analyzed
+ Obtain information of **haplotypes** for di- and polyploid specimens
+ **Number** of examined specimens is **well scalable**
+ Compatibility and extensibility of available **databases** (BOLD, GenBank, etc.)
+ Benefit from **genomic resources**, e.g. BUSCO (Benchmarking Universal Single-Copy Orthologs) genes
- **High error rates**
- **Multiple Marker genes not (yet) well established**
- **not clear if single-copy**
- Pools of reads need to be **demultiplexed**

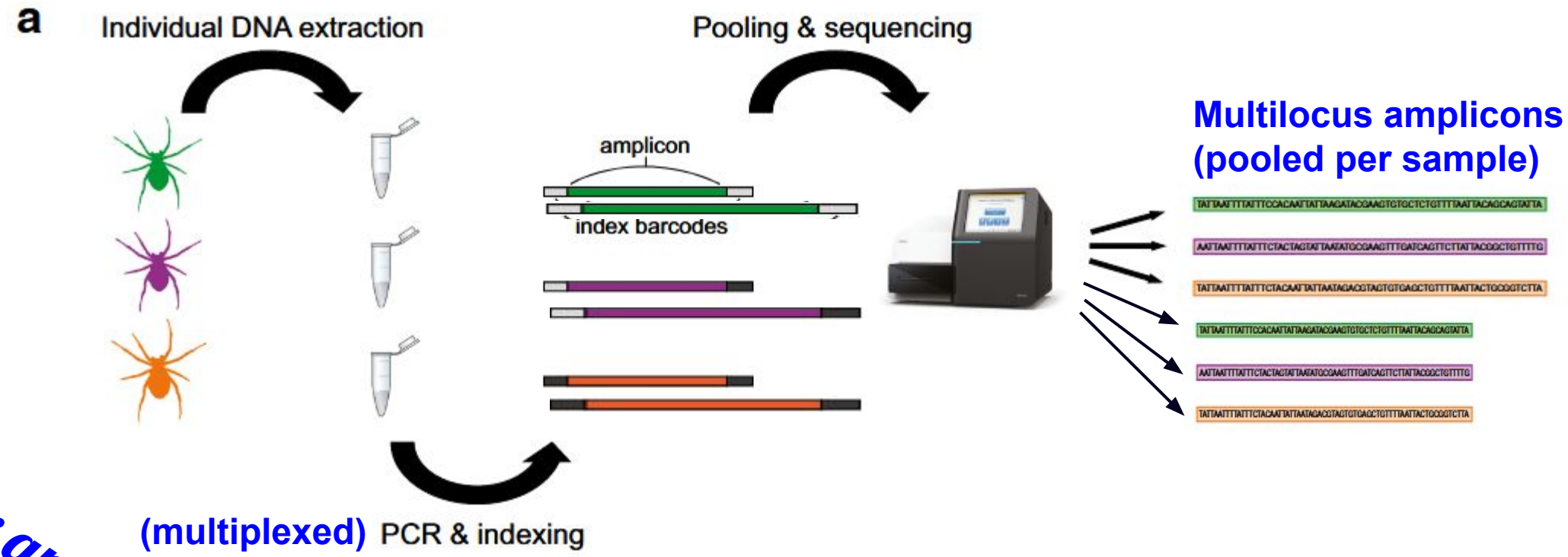
Funded by the European Union

From single locus pooled amplicon sequencing…



Kennedy, et al. (2020). *Dev Genes Evol* **230**: 185–201.

… to multilocus pooled amplicon sequencing



Our approach!

Kennedy, et al. (2020). *Dev Genes Evol* **230**: 185–201.

Our Aim

- A simple yet powerful **combination** of **wet lab** and **bioinformatics protocols**
- Cost-efficiency through **multiplexed PCRs and pooled sequencing**
- **Easy-to-use bioinformatics** pipeline for
  - Demultiplexing of loci
  - Consensus sequence reconstruction
  - Species identification with BOLD
  - Basic Phylogenetic Analyses
  - (Species delimitation with ASAP)
  - Comprehensive and detailed logging, documentation and output
- **Use Case - hover flies (Syrphidae)**

AmpliPiper

- **Open Source**
- **Simple installation** of dependencies
- **Simple input** file formats
- Comprehensive **documentation**
- Computational **efficiency**
- **Versatile** analyses
- **Verbose** logging
- **Comprehensive** output

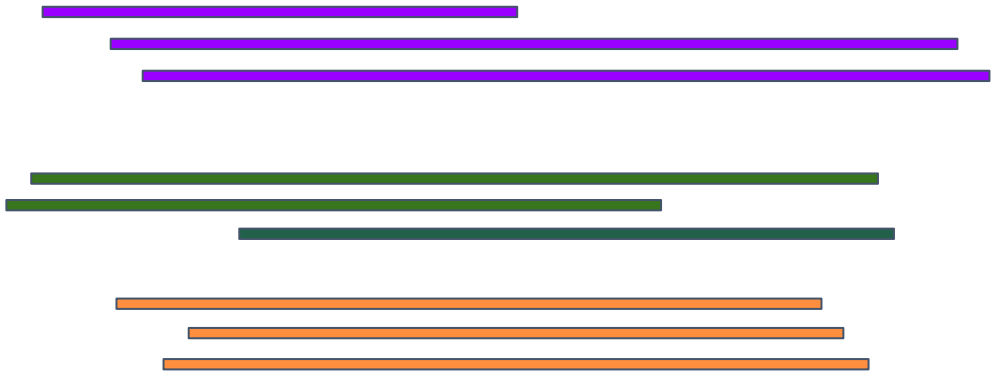
Part I - AmpliPiper

Preparation of input NGS data

Raw multiplexed ONT sequencing data
(tagged with ONT barcodes)

Basecalling with *GUPPY* (v.6.2.1 GPU)

- Super high accuracy mode (SUP)
- Demultiplexing by barcode (EXP-PBC096)
- Gzip compression of all files



Preparation of input NGS data - samples.csv

samples.csv

- **Comma-separated** file format
- Contains
  - sample **IDs**
  - **paths** to **demultiplexed gzipped** FASTQ files
- **One sample per line**
- **May contain header** “ID,PATH”

```
ID,PATH
SAMPLE1,/PATH/TO/SAMPLE1.fastq.gz
SAMPLE2,/PATH/TO/SAMPLE2.fastq.gz
```

Preparation of input NGS data - primers.csv

primers.csv

- **Comma-separated** file format
- Contains
  - primer **IDs**,
  - **fwd** and **rev** primer **sequences**
  - Expected **ploidy**
  - Expected **fragment size**
- **One sample per line**
- **May contain header**

```
ID,FWD,REV,SIZE
Locus1,AGAGGA,GTATTAGA,650
Locus2,AGTATT,GAGGAA,800
```

Preparing AmpliPiper

Requirements:

- **Linux** (tested on CentOS, Ubuntu)
- **(Ana)Conda and Mamba** installed and in path
- ***setup.sh*** will install all dependencies when the pipeline is started the first time.

Running AmpliPiper

Minimal commandline:

```
AmpliPiper.sh \
  --samples /path/to/samples.csv \
  --primers /path/to/primers.csv \
  --output /path/to/output-folder
```

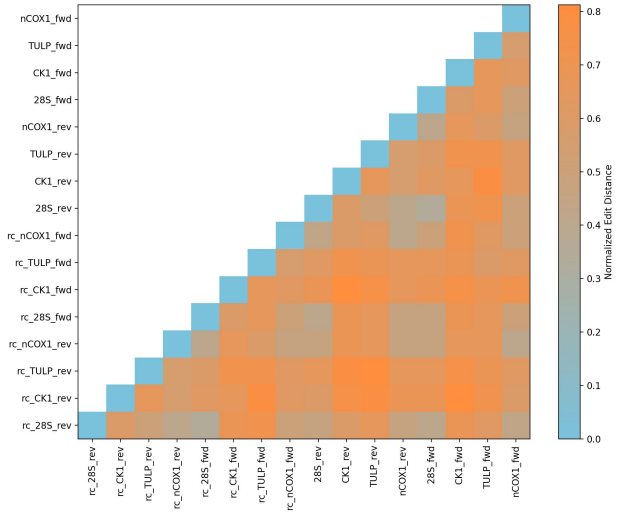
AmpliPiper - optional arguments

Optional Arguments

- `-b` or `--blast` : Enable BLAST search for species identification. When setting this parameter, you need to provide an email address (e.g., `--blast your@email.com`) for using NCBI entrez to retrieve taxonomic information for the BLAST hits (default: disabled).
- `-c` or `--similar_consensus` : Change the minimum similarity threshold (in percent) of amplicon_sorter. If the similarity of two clusters are smaller or equal to the threshold, they are considered separartely otherwise they are collapsed prior to consensus reconstruction (default: 96)
- `-e` or `--exclude` : Provide a text file with samples and loci to exclude from the analysis. Each row should contain the ID of a sample to be excluded. Names need to be identical to the IDs in `samples.csv`
- `-f` or `--force` : Force overwrite the output folder if it already exists (default: cowardly refusing to overwrite).
- `-i` or `--partition` : Use partition model for iqtree with combined dataset. ⚠ may take very long ⚠ (default: disabled)
- `-k` or `--kthreshold` : Define the threshold *k* for the maximum allowed proportion of mismatches for primer alignment during demultiplexing (default: 0.05).
- `-m` or `--minreads` : Set the minimum number of reads required for consensus sequence reconstruction (default: 100).
- `-n` or `--nreads` : Provide the absolute number or percentage of top-quality reads to consider for consensus sequence generation and variant calling (default: 500).
- `-q` or `--quality` : Specify the minimum PHRED quality score for read filtering (default: 10).
- `-r` or `--sizerange` : Define the allowed size buffer in basepairs around the expected locus length (default: 100).
- `-t` or `--threads` : Specify the number of threads to be used for parallel processing (default: 10).
- `-w` or `--nowatermark` : Remove the watermark from the tree figures
- `-y` or `--freqthreshold` : Retain consensus sequences for further analyses which are supported by raw reads, whose frequency in the total pool of reads is larger or equal to this threshold (default: 0.1).

Running AmpliPiper

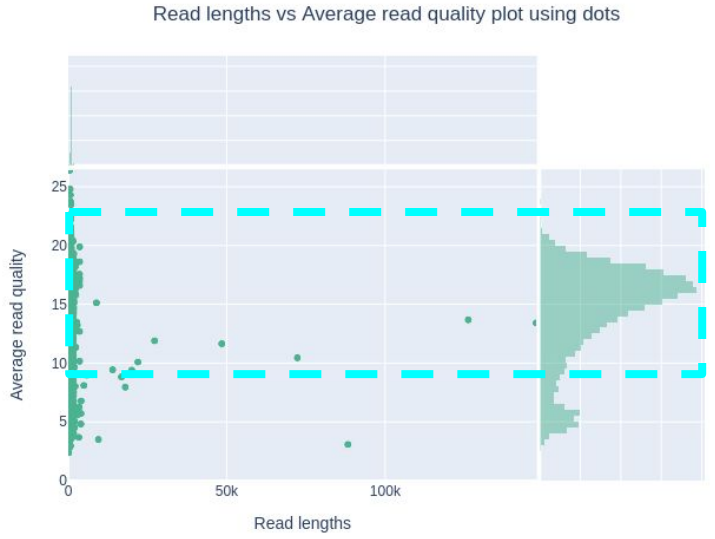
Testing sequence similarity among primers



Custom python script

edlib distances

Filtering raw FASTQ by min PHRED quality

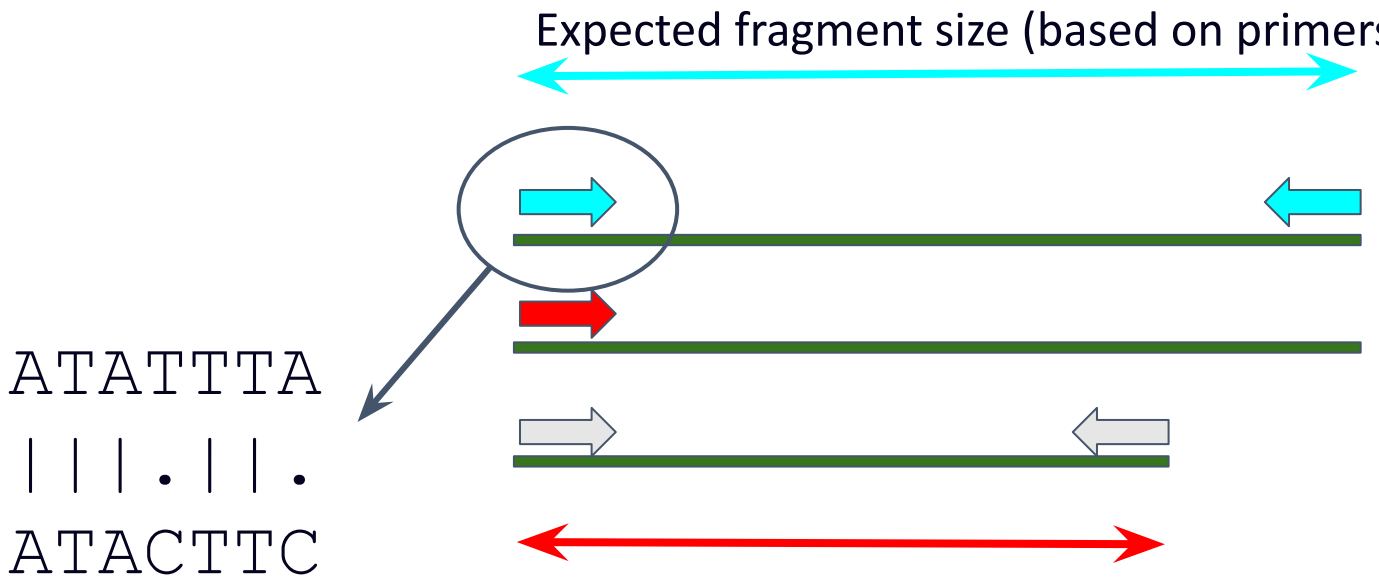


Chopper

`--quality 10`

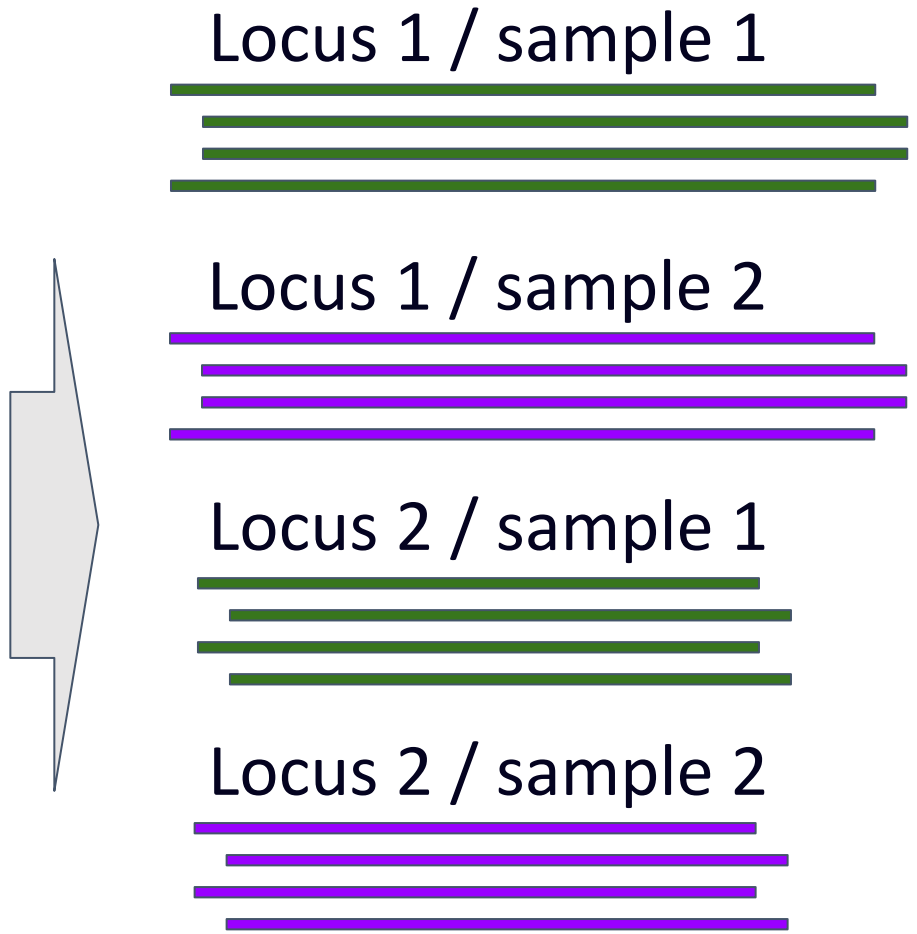
Running AmpliPiper

Demultiplexing by locus (custom script)



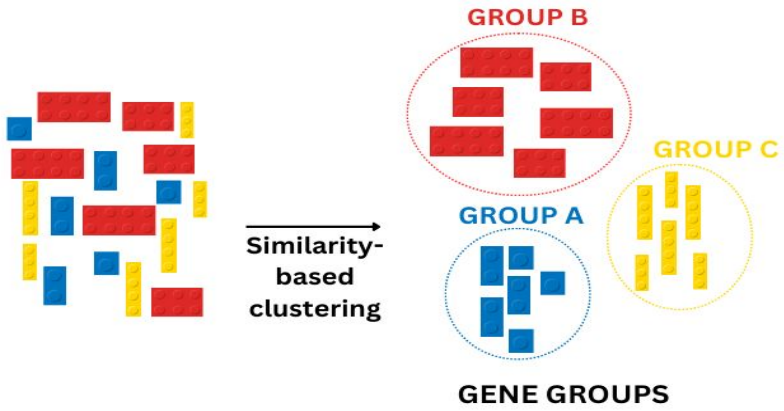
Edlib distance (#mismatches / primerlength)

`--kthreshold`



Running AmpliPiper

Consensus sequence reconstruction (amplicon_sorter)

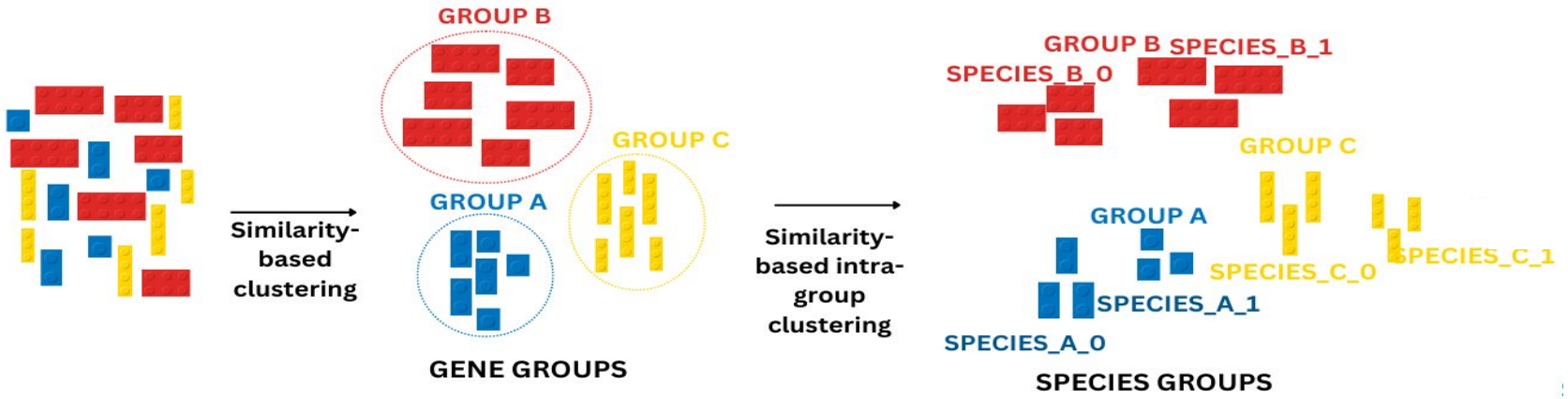


AmpliconSorter

- Step 1: Similarity-based clustering (group your legos by color)

Running AmpliPiper

Consensus sequence reconstruction (amplicon_sorter)

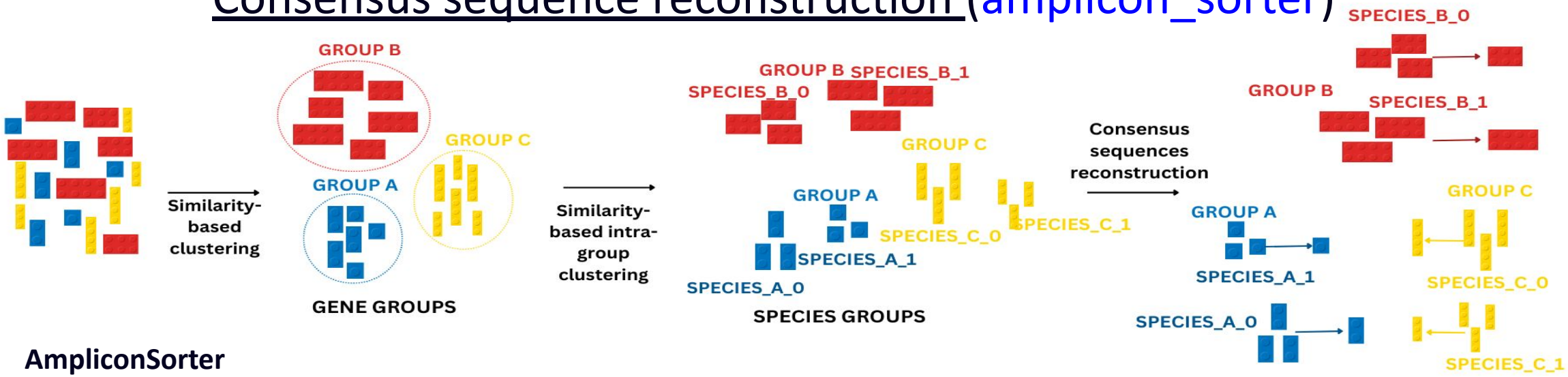


AmpliconSorter

- Step 1: Similarity-based clustering (group your legos by color)
- Step 2: Similarity-based intra-group clustering (group same-color legos by size)

Running AmpliPiper

Consensus sequence reconstruction (amplicon_sorter)

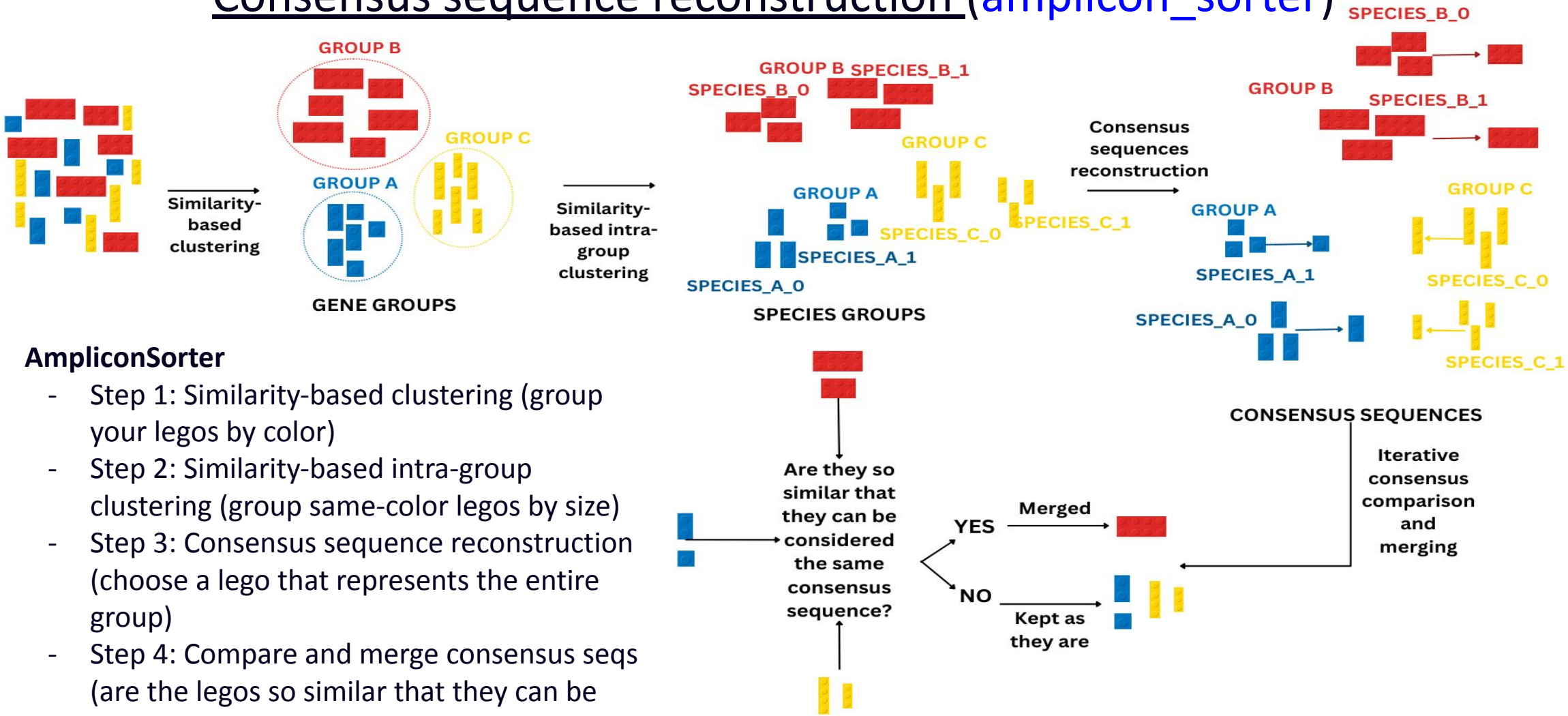


AmpliconSorter

- Step 1: Similarity-based clustering (group your legos by color)
- Step 2: Similarity-based intra-group clustering (group same-color legos by size)
- Step 3: Consensus sequence reconstruction (choose a lego that represents the entire group)

Running AmpliPiper

Consensus sequence reconstruction (amplicon_sorter)



AmpliconSorter

- Step 1: Similarity-based clustering (group your legos by color)
- Step 2: Similarity-based intra-group clustering (group same-color legos by size)
- Step 3: Consensus sequence reconstruction (choose a lego that represents the entire group)
- Step 4: Compare and merge consensus seqs (are the legos so similar that they can be represented by one piece?)

Running AmpliPiper

Modeling Ploidy (custom script)

Observed:

Hap1: 264 reads -> 26%

Hap2: 734 reads -> 74%

Running AmpliPiper

Modeling Ploidy (custom script)

Observed:

Hap1: 264 reads -> 26%

Hap2: 734 reads -> 74%

Expectations:

```
TEST = {2:
        {1: [np.array([0.0, 1.0])],
         2: [np.array([1/2, 1/2])],
         3: [np.array([1/3, 2/3])],
         4: [np.array([1/2, 1/2]), np.array([1/4, 3/4])]},
        3:
        {1: [np.array([0.0, 0.0, 1.0])],
         2: [np.array([0.0, 1/2, 1/2])],
            3: [np.array([1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/2])]},
        4: {1: [np.array([0.0, 0.0, 0.0, 1.0])],
            2: [np.array([0.0, 0.0, 1/2, 1/2])],
            3: [np.array([0, 1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/4, 1/4])]}}
```

Running AmpliPiper

Modeling Ploidy (custom script)

$$P(X = x) = \frac{n!}{x_1!x_2!...x_k!} \cdot p_1^{x_1} \cdot p_2^{x_2} \cdot \ldots \cdot p_k^{x_k}$$

Observed:

Hap1: 264 reads -> 26%

Hap2: 734 reads -> 74%

Expectations:

```
TEST = {2:
        {1: [np.array([0.0, 1.0])],
         2: [np.array([1/2, 1/2])],
         3: [np.array([1/3, 2/3])],
         4: [np.array([1/2, 1/2]), np.array([1/4, 3/4])]},
        3:
        {1: [np.array([0.0, 0.0, 1.0])],
         2: [np.array([0.0, 1/2, 1/2])],
            3: [np.array([1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/2])]},
        4: {1: [np.array([0.0, 0.0, 0.0, 1.0])],
            2: [np.array([0.0, 0.0, 1/2, 1/2])],
            3: [np.array([0, 1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/4, 1/4])]}}
```

Running AmpliPiper

Modeling Ploidy (custom script)

$$P(X = x) = \frac{n!}{x_1!x_2!...x_k!} \cdot p_1^{x_1} \cdot p_2^{x_2} \cdot \ldots \cdot p_k^{x_k}$$

Observed:

Hap1: 264 reads -> 26%

Hap2: 734 reads -> 74%

Ploidy	Expectation	Likelihood
2	[0.5, 0.5]	2.7E-52
3	[0.33, 0.66]	4.5E-07
4	[0.5, 0.5]	2.7E-52
4	[0.25, 0.75]	0.016

Expectations:

```
TEST = {2:
        {1: [np.array([0.0, 1.0])],
         2: [np.array([1/2, 1/2])],
         3: [np.array([1/3, 2/3])],
         4: [np.array([1/2, 1/2]), np.array([1/4, 3/4])]},
        3:
        {1: [np.array([0.0, 0.0, 1.0])],
         2: [np.array([0.0, 1/2, 1/2])],
            3: [np.array([1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/2])]},
        4: {1: [np.array([0.0, 0.0, 0.0, 1.0])],
            2: [np.array([0.0, 0.0, 1/2, 1/2])],
            3: [np.array([0, 1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/4, 1/4])]}}
```

Running AmpliPiper

Modeling Ploidy (custom script)

$$P(X = x) = \frac{n!}{x_1!x_2!...x_k!} \cdot p_1^{x_1} \cdot p_2^{x_2} \cdot \ldots \cdot p_k^{x_k}$$

Observed:

Hap1: 264 reads -> 26%

Hap2: 734 reads -> 74%

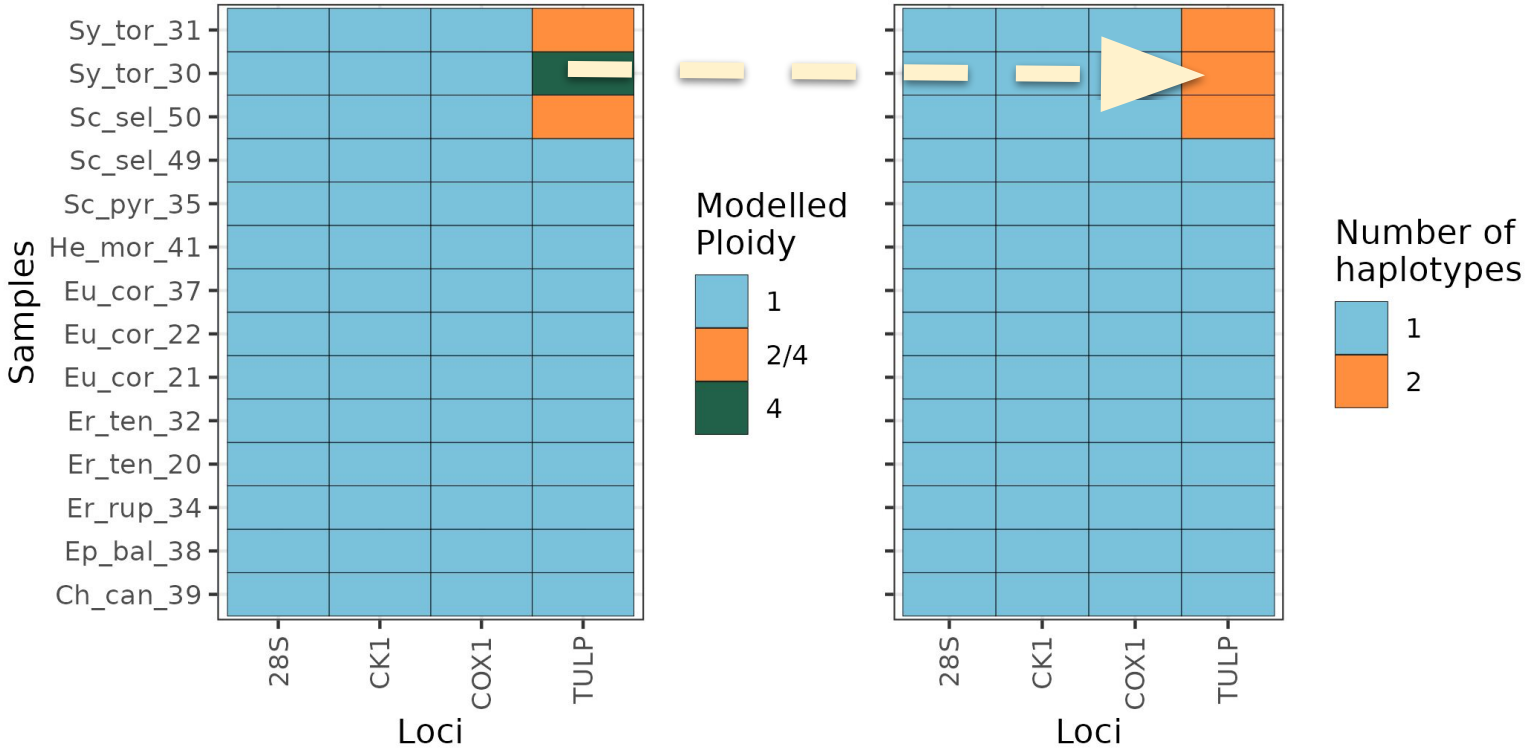
Ploidy	Expectation	Likelihood
2	[0.5, 0.5]	2.7E-52
3	[0.33, 0.66]	4.5E-07
4	[0.5, 0.5]	2.7E-52
4	[0.25, 0.75]	0.016

Expectations:

```
TEST = {2:
        {1: [np.array([0.0, 1.0])],
         2: [np.array([1/2, 1/2])],
         3: [np.array([1/3, 2/3])],
         4: [np.array([1/2, 1/2]), np.array([1/4, 3/4])]},
        3:
        {1: [np.array([0.0, 0.0, 1.0])],
         2: [np.array([0.0, 1/2, 1/2])],
            3: [np.array([1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/2])]},
        4: {1: [np.array([0.0, 0.0, 0.0, 1.0])],
            2: [np.array([0.0, 0.0, 1/2, 1/2])],
            3: [np.array([0, 1/3, 1/3, 1/3])],
            4: [np.array([1/4, 1/4, 1/4, 1/4])]}}
```

Running AmpliPiper

Modelled Ploidy and number of Haplotypes (custom script)

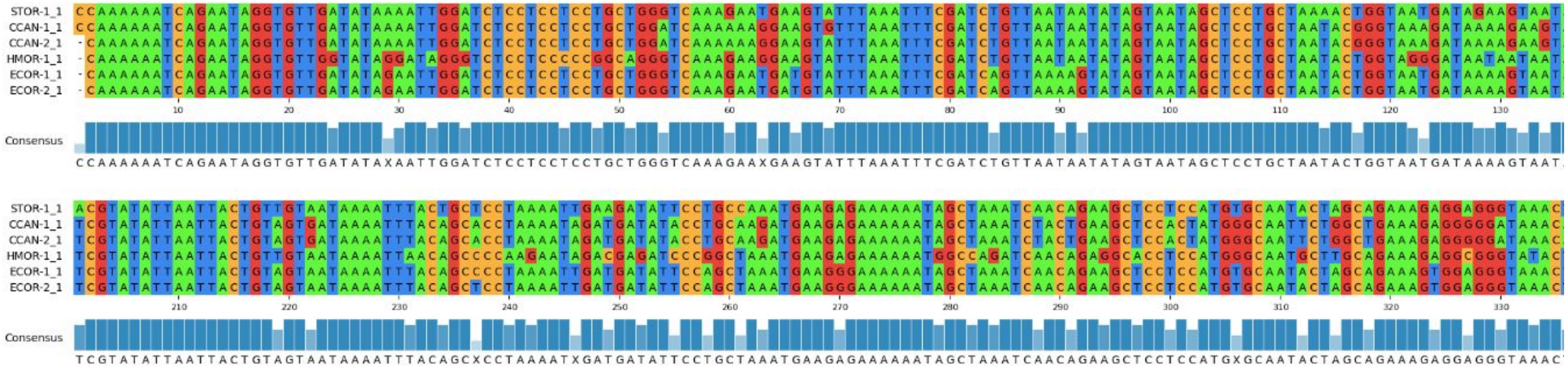


- By default: consensus sequences with < 10% frequencies are excluded
- `--freqthreshold 0.1`

Visualization with ggplot

Running AmpliPiper

Multiple Alignment (Mafft)



Visualization with PyMSAviz

Running AmpliPiper

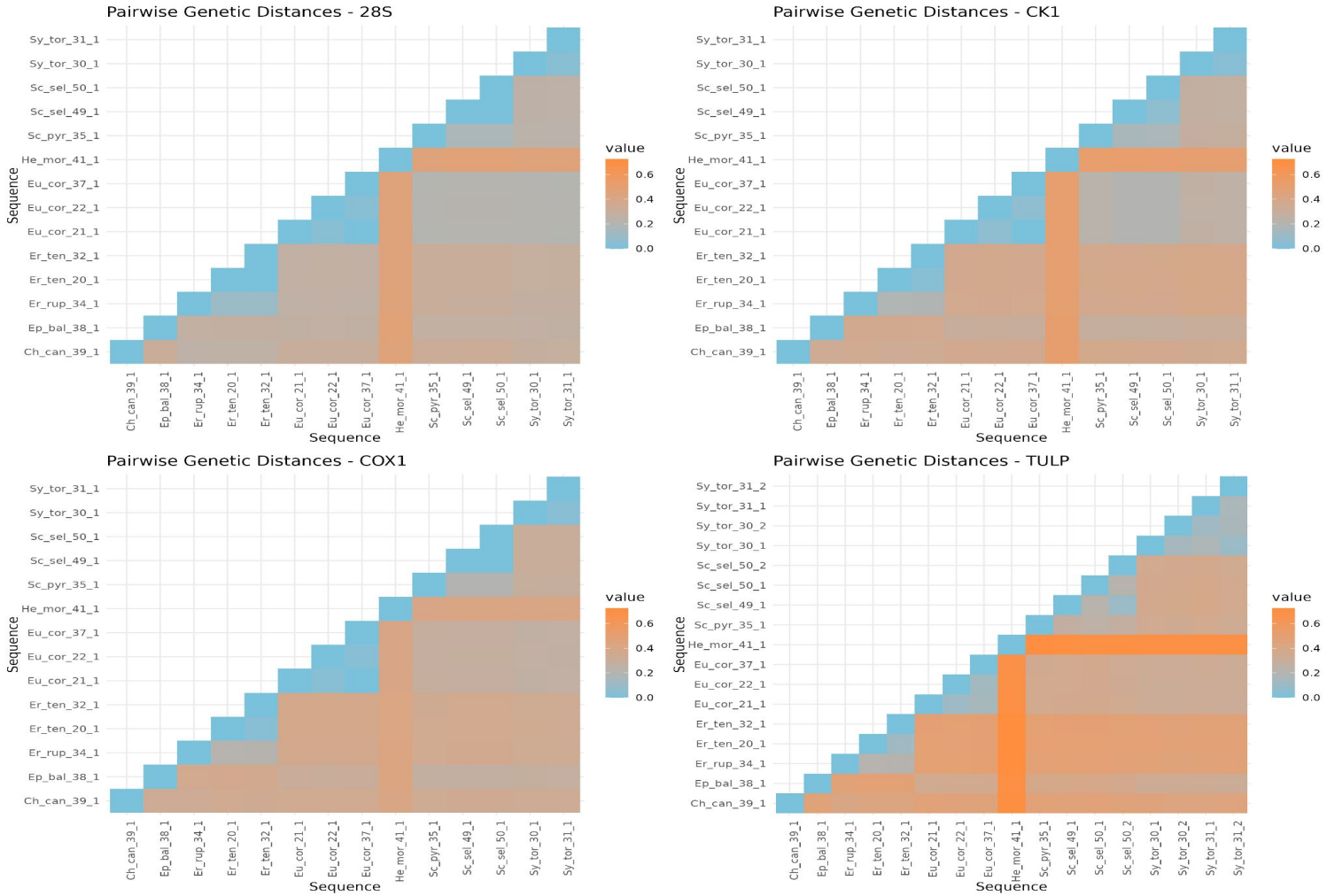
Genetic Distance

Dist=diff(S1,S2)/Length

ape::dist.gene

Visualization with

ggplot::ggplot



Running AmpliPiper

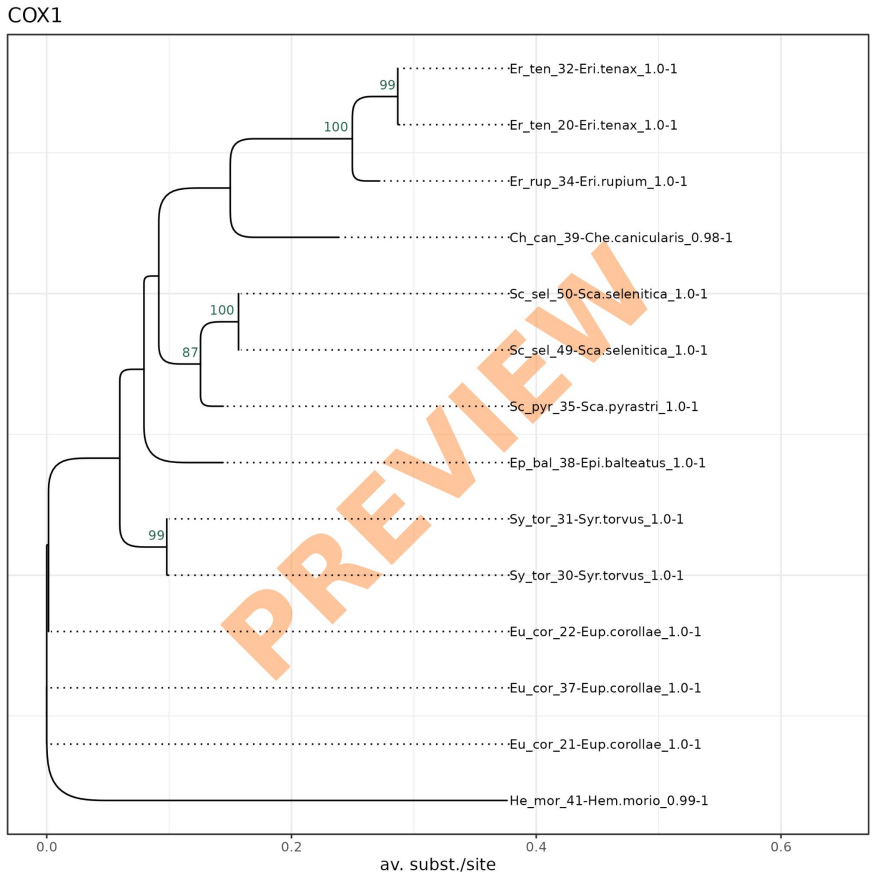
Species Identification using the BOLD or BLAST API (custom script)

SAMPLE	Taxon1 (sim%)	Taxon2 (sim%)	Taxon3 (sim%)	Taxon4 (sim%)	Taxon5 (sim%)	Taxon6 (sim%)
Ch_can_39_1	Cheilosia canicularis (97.99%); count = 3	Cheilosia oblonga (94.38%); count = 1	Cheilosia variabilis (94.37%); count = 4	Cheilosia vulpina (92.76%); count = 1	Cheilosia albitarsis (92.76%); count = 1	
Ep_bal_38_1	Episyrphus balteatus (99.56%); count = 10					
Er_rup_34_1	Eristalis rupium (99.7%); count = 8	Eristalis hirta (99.09%); count = 1	Eristalis himalayensis (96.39%); count = 1			
Er_ten_20_1	Eristalis tenax (99.85%); count = 10					
Er_ten_32_1	Eristalis tenax (100.0%); count = 10					
Eu_cor_21_1	Eupeodes corollae (99.7%); count = 9	Eupeodes luniger (98.84%); count = 1				
Eu_cor_22_1	Eupeodes corollae (99.7%); count = 8	Eupeodes luniger (99.39%); count = 2				

- If locus name is COX1, ITS or MATK_RBCL (Standard barcoding markers)

Running AmpliPiper

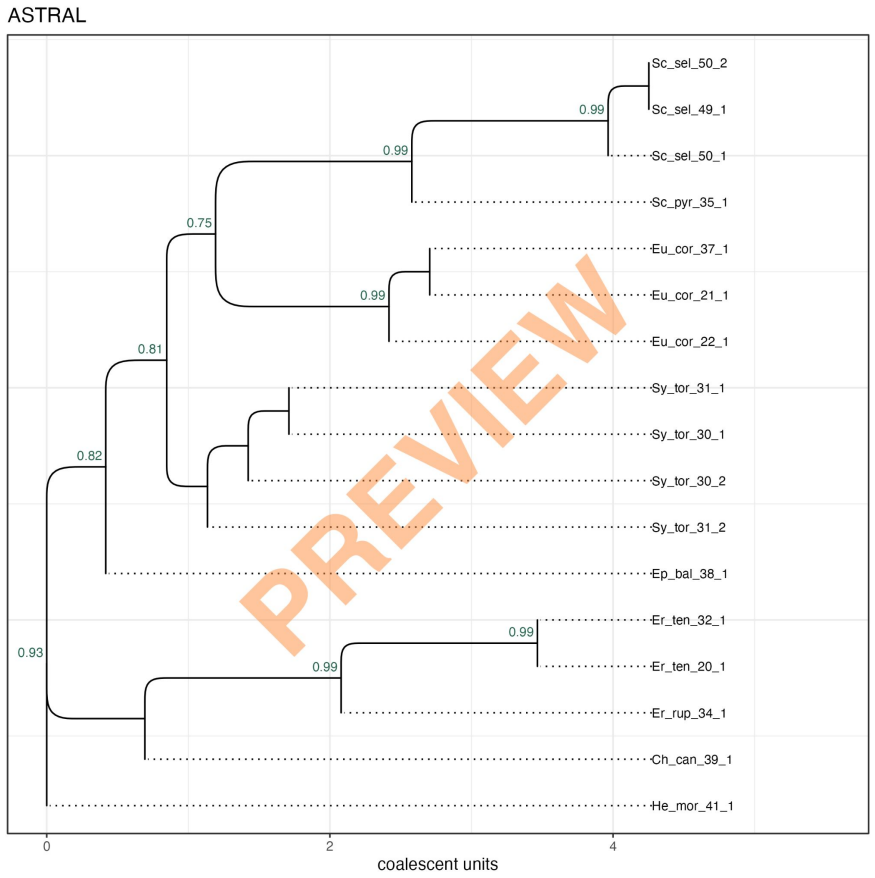
Phylogenetic analyses based on loci-specific maximum likelihood (iqtree)



- If species ID successful, adjusted sample_IDs in tree (custom script)
- Tree plotting with ggtree
- Watermark to remind the necessity of integrating results with other taxonomical evidence

Running AmpliPiper

Phylogenetic analyses based on (SUPER) trees - **integrating across all loci**



Concatenated dataset (iqtree)

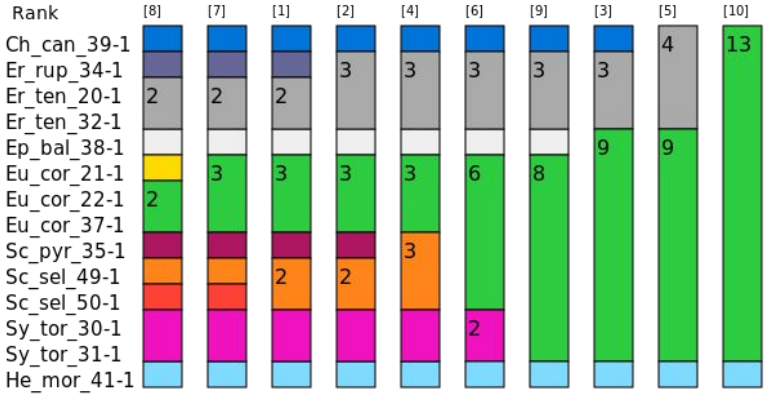
- **ML with (optional) separate substitution rates** for each locus in tree reconstruction
- Samples only included if **< 30% gaps**

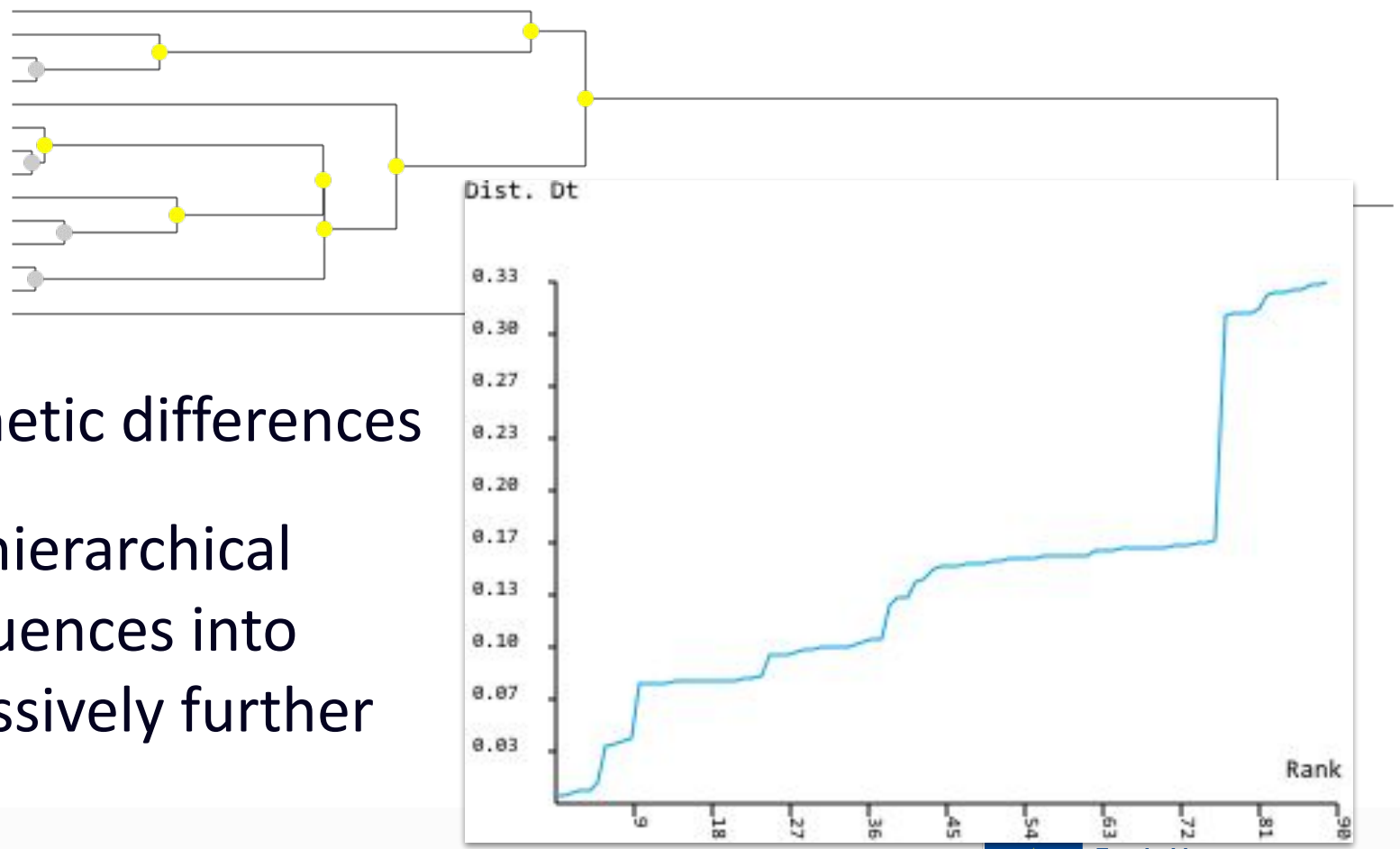
wASTRAL (aster)

- seeks the species tree that maximizes the number of shared quartets (unrooted fourtaxon subtrees) between gene trees and the species tree
- takes informations such as branch lengths and branch support into account

Running AmpliPiper

Species Delimitation (ASAP)



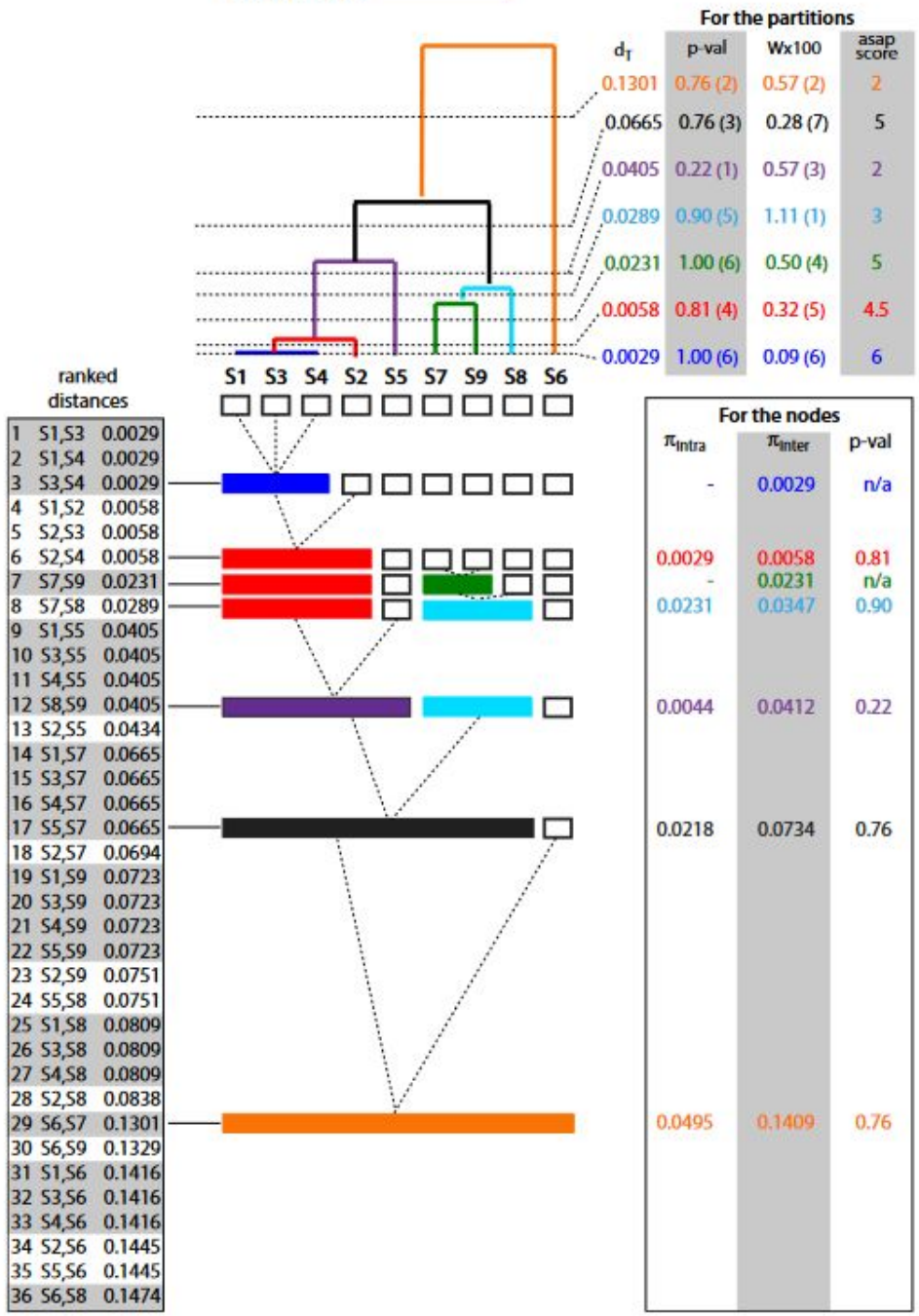


- Ranking all pairwise genetic differences
- Subsequent ascending hierarchical clustering, merging sequences into “groups” that are successively further merged

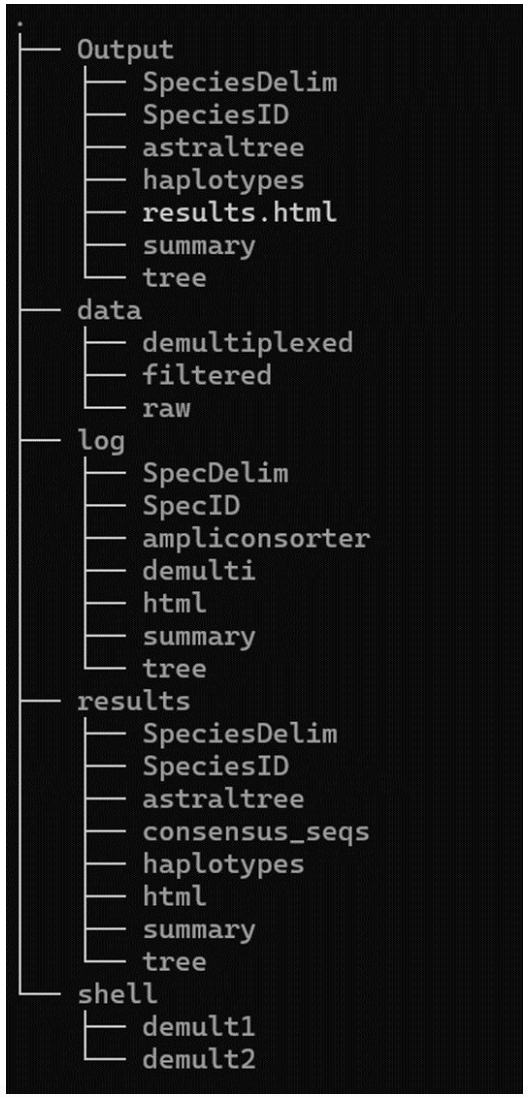
Running AmpliPiper

Species Delimitation (ASAP)

- Ranking all pairwise genetic differences
- Subsequent ascending hierarchical clustering, merging sequences into “groups” that are successively further merged



AmpliPiper - output



Final output folder with all the results on display in `results.html`

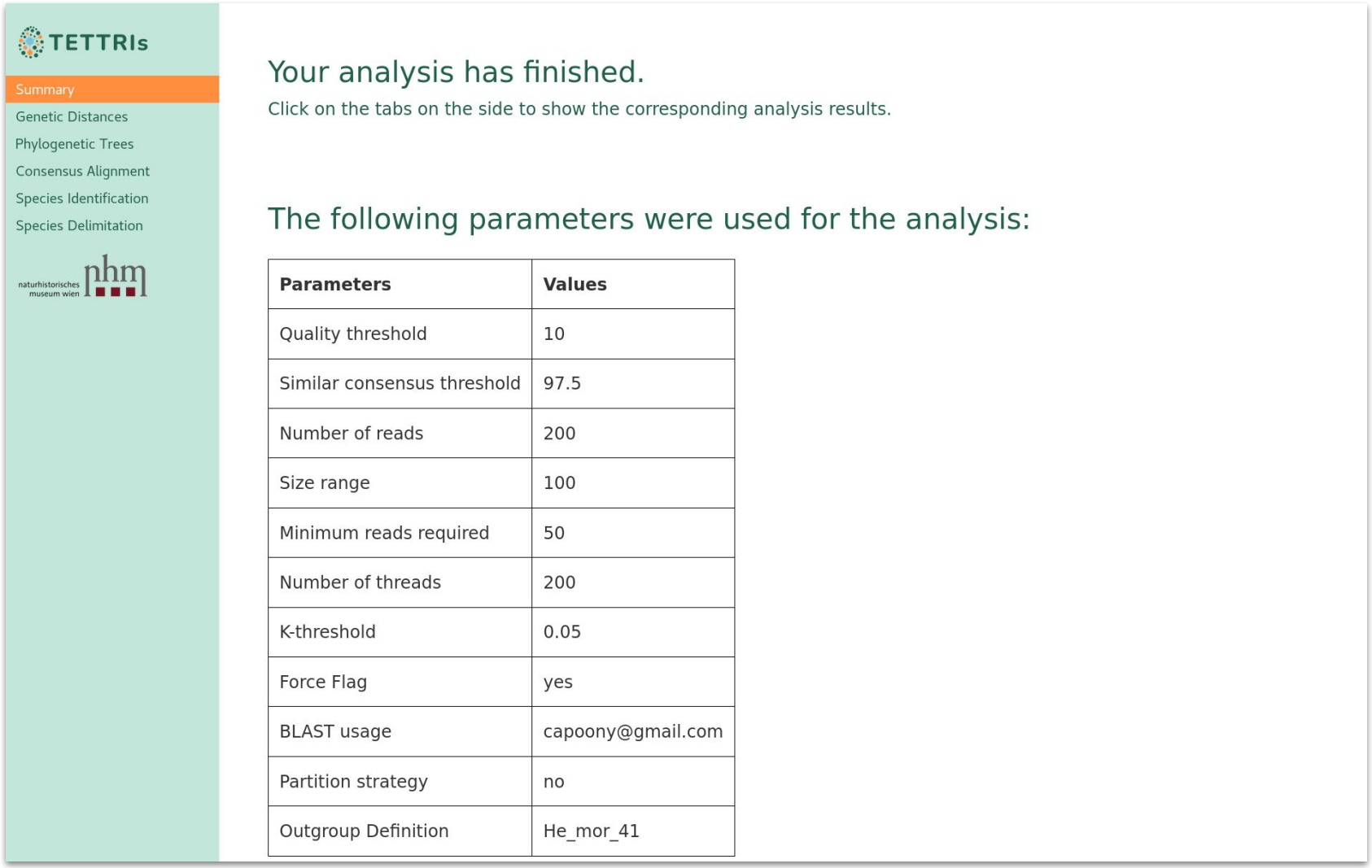
Folder with raw, quality/length-filtered and demultiplexed data

Folder with log outputs from every executed step inside the pipeline

Folder with intermediate and raw results file

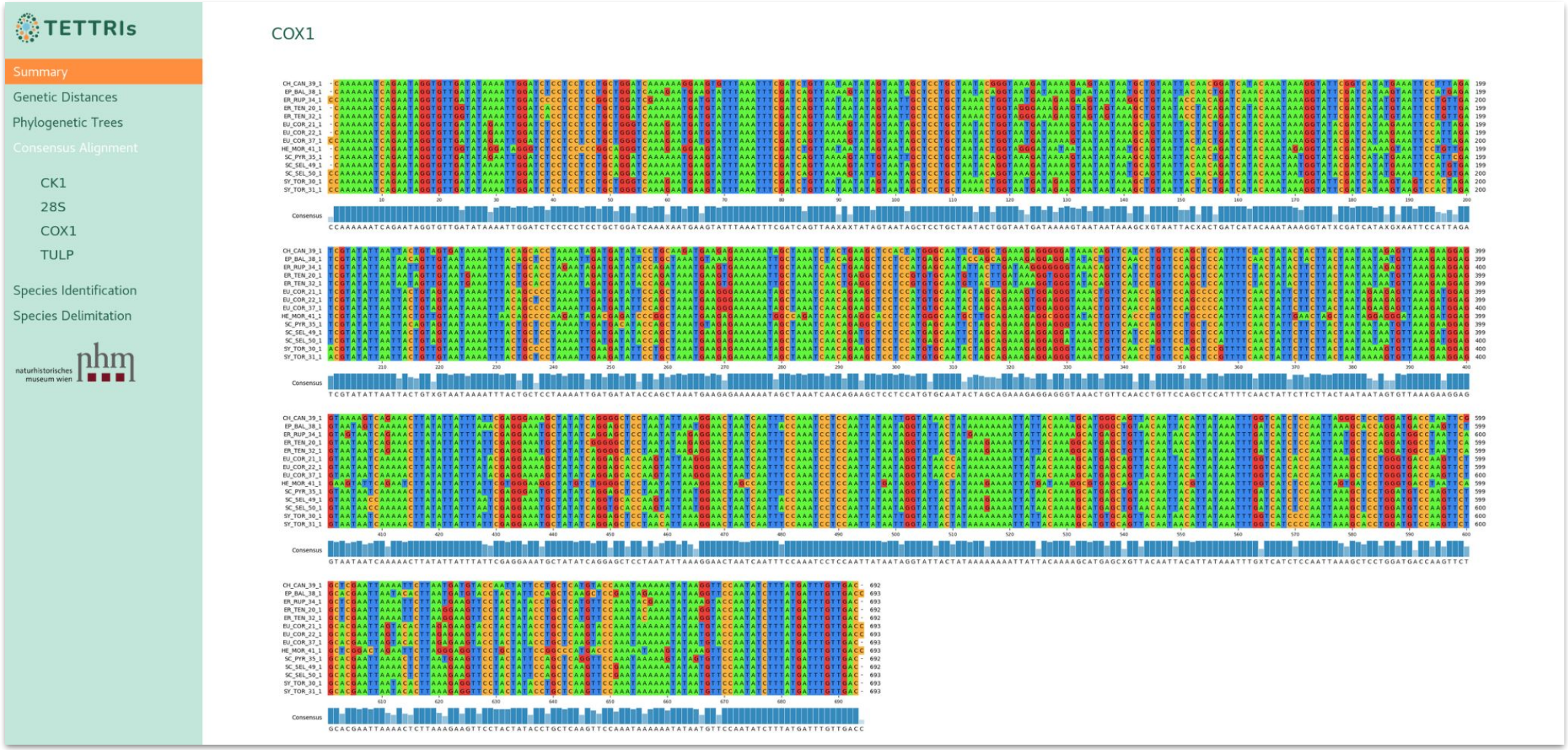
Folder with shell scripts run in parallel during the pipeline execution

AmpliPiper - output



Parameters	Values
Quality threshold	10
Similar consensus threshold	97.5
Number of reads	200
Size range	100
Minimum reads required	50
Number of threads	200
K-threshold	0.05
Force Flag	yes
BLAST usage	capoony@gmail.com
Partition strategy	no
Outgroup Definition	He_mor_41

AmpliPiper - output



AmpliPiper - output

TETTRIs

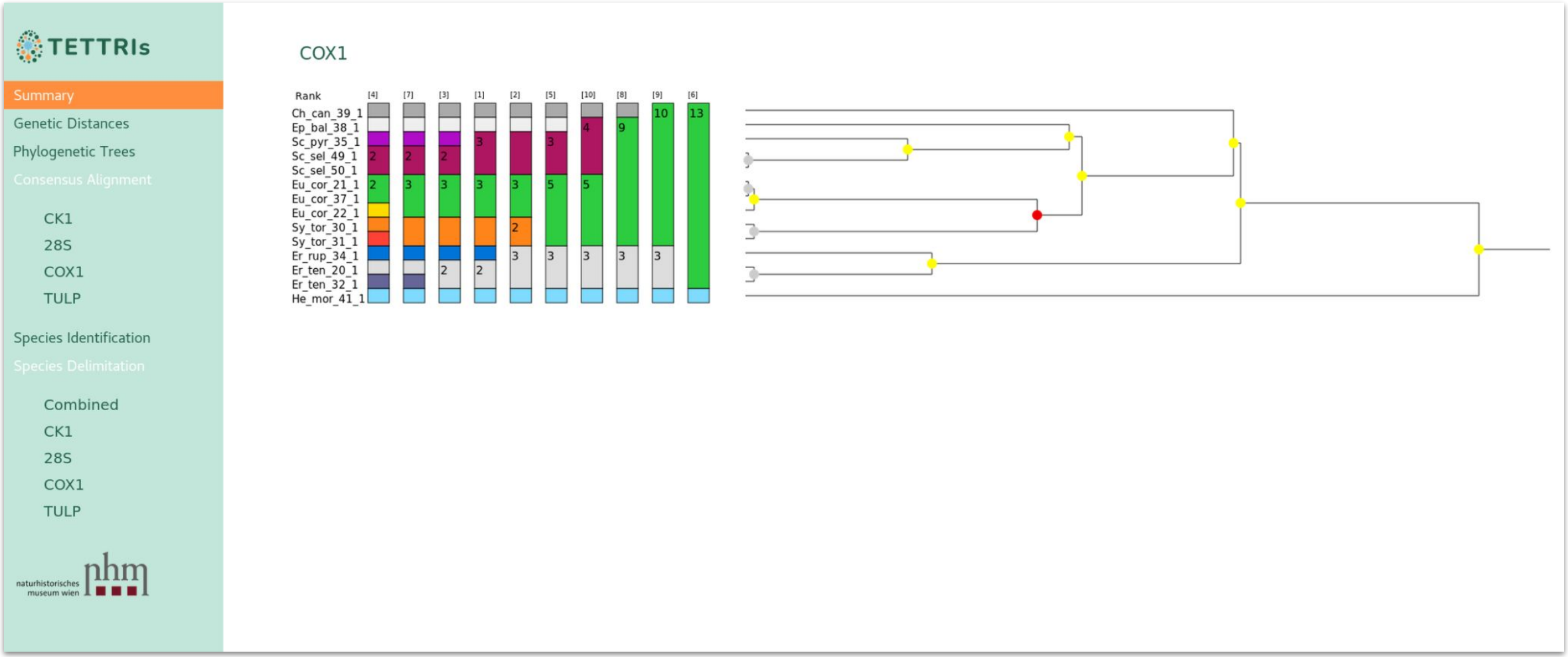
- Summary
- Genetic Distances
- Phylogenetic Trees
- Consensus Alignment
  - CK1
  - 28S
  - COX1
  - TULP
- Species Identification
- Species Delimitation

naturhistorisches museum wien

Summarized results for the Species Identification approach with BLAST.

SAMPLE	Taxon1 (sim%)	Taxon2 (sim%)	Taxon3 (sim%)	Taxon4 (sim%)	Taxon5 (sim%)	Taxon6 (sim%)
Ch_can_39_1	Cheilosia canicularis (97.99%); count = 3	Cheilosia oblonga (94.38%); count = 1	Cheilosia variabilis (94.37%); count = 4	Cheilosia vulpina (92.76%); count = 1	Cheilosia albitarsis (92.76%); count = 1	
Ep_bal_38_1	Episyrphus balteatus (99.56%); count = 10					
Er_rup_34_1	Eristalis rupium (99.7%); count = 8	Eristalis hirta (99.09%); count = 1	Eristalis himalayensis (96.39%); count = 1			
Er_ten_20_1	Eristalis tenax (99.85%); count = 10					
Er_ten_32_1	Eristalis tenax (100.0%); count = 10					
Eu_cor_21_1	Eupeodes corollae (99.7%); count = 9	Eupeodes luniger (98.84%); count = 1				
Eu_cor_22_1	Eupeodes corollae (99.7%); count = 8	Eupeodes luniger (99.39%); count = 2				
Eu_cor_37_1	Eupeodes corollae (99.7%); count = 9	Eupeodes luniger (98.84%); count = 1				
He_mor_41_1	Hemipenthes morio (99.39%); count = 1	Bombyliidae sp. (89.35%); count = 9				
Sc_pyr_35_1	Scaeva pyrastri (99.7%); count = 10					
Sc_sel_49_1	Scaeva selenitica (100.0%); count = 4	Scaeva dignota (99.85%); count = 1	Scaeva komabensis (98.97%); count = 1	Scaeva mecogramma (98.38%); count = 2	Scaeva affinis (96.24%); count = 1	Scaeva pyrastri (95.66%); count = 1
Sc_sel_50_1	Scaeva selenitica (100.0%); count = 4	Scaeva dignota (99.85%); count = 1	Scaeva komabensis (98.97%); count = 1	Scaeva mecogramma (98.38%); count = 2	Scaeva affinis (96.24%); count = 1	Scaeva pyrastri (95.66%); count = 1
Sy_tor_30_1	Syrphus torvus (99.7%); count = 10					
Sy_tor_31_1	Syrphus torvus (99.7%); count = 10					

AmpliPiper - output



AmpliPiper - GitHub repository

README GPL-3.0 license

Language Bash | Production status Beta | Release v0.2.0 beta | Requires Mamba and Conda | Supported platforms linux

AmpliPiper

AmpliPiper

This README provides a quickstart to the pipeline: if you want to dive deeper, check out the complete_documentation.

Installation

The installation of dependencies requires mamba and conda to be already installed on your system.

To install the pipeline program:

1. Clone this repository:

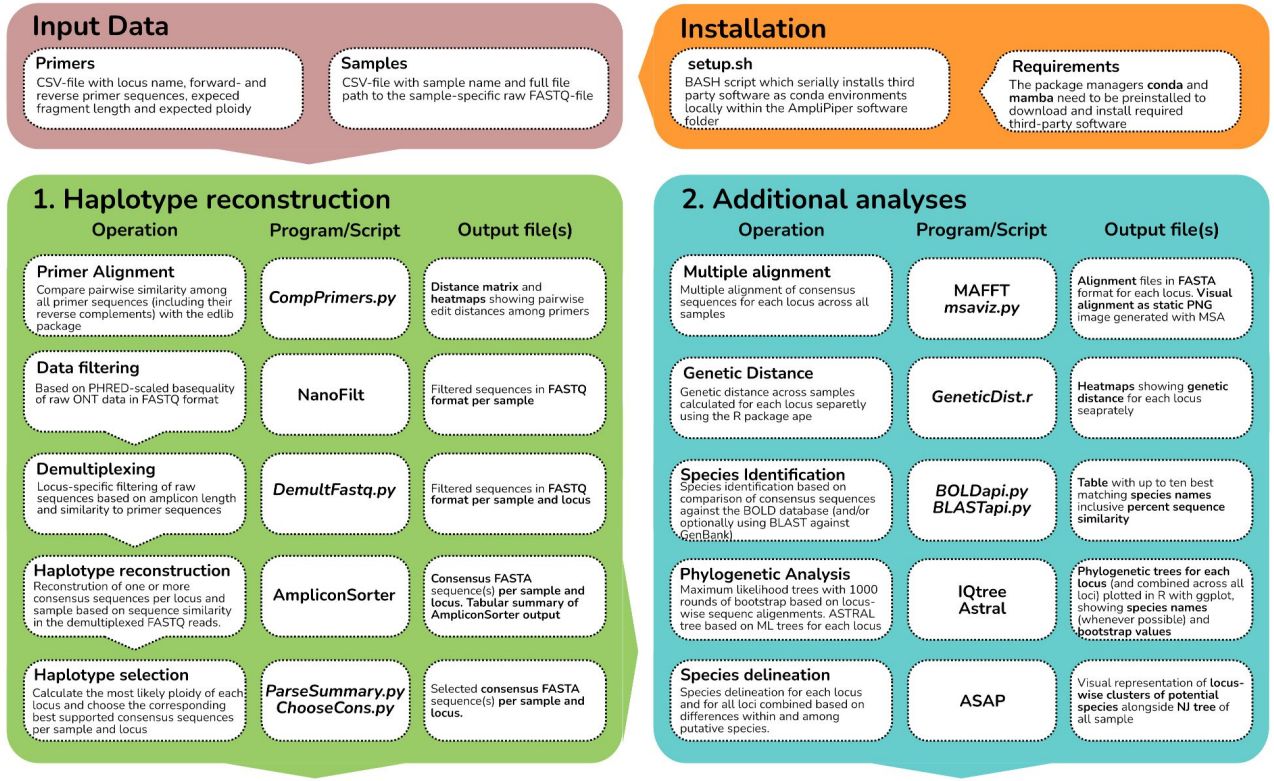
```
git clone https://github.com/nhmvienna/AmpliPiper.git
```

2. Go to the cloned directory:

```
cd AmpliPiper
```

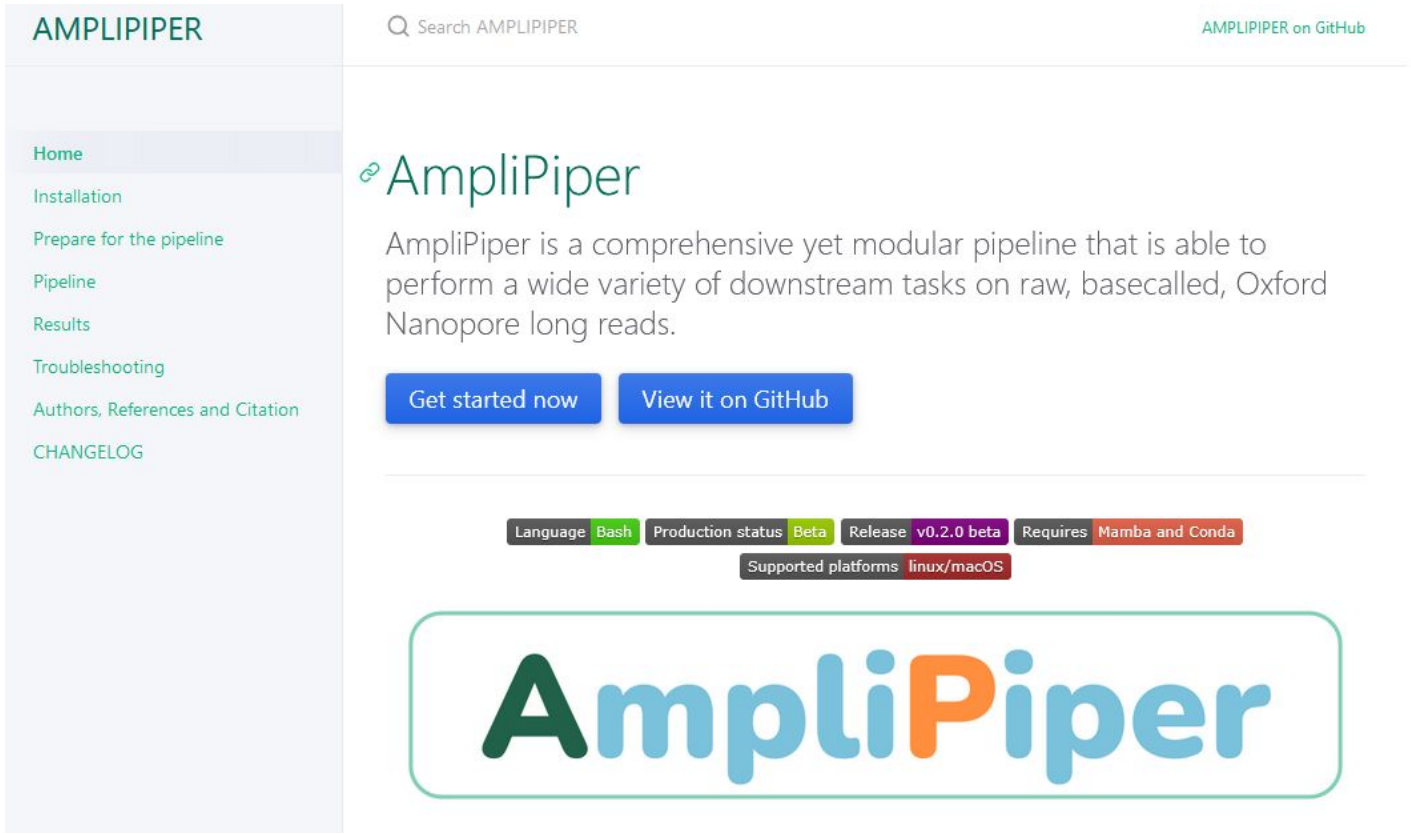
3. Run the setup script:

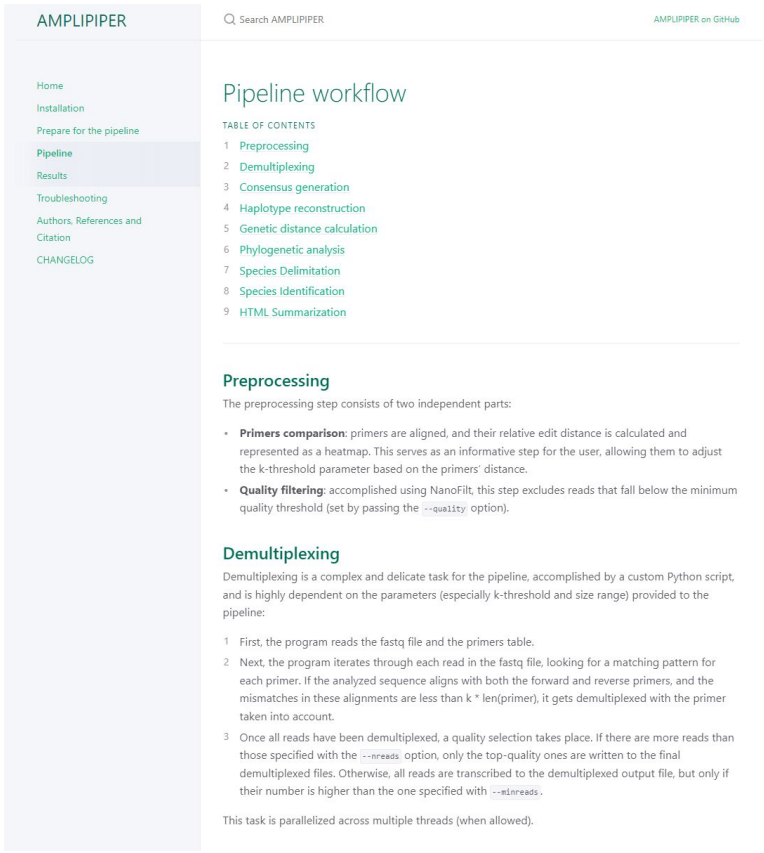
```
bash shell/setup.sh
```



https://github.com/nhmvienna/AmpliPiper

AmpliPiper - documentation





https://astrabert.github.io/AmpliPiper-docs/

Part II - Hands on

AmpliPiper - Use Case

Syrphidae

14 samples
4 loci
(COX1, 28s, TULP, CK1)

- Biological System /Relevance (Pollinators)
  - Visitors of > 72% of global food crops
  - many larva important in biological pest control (predators of aphids)
- Taxonomy/Phylogeny
  - 2 of 4 subfamilies included (Eristalinae & Syrphinae)
- Where does the data come from?
  - Specimens sampled in the course of ABOL-BioBlitz
- 4 Loci (1 mt / 3 nuclear) from recent phylogenetic study (Moran et al. (2021))

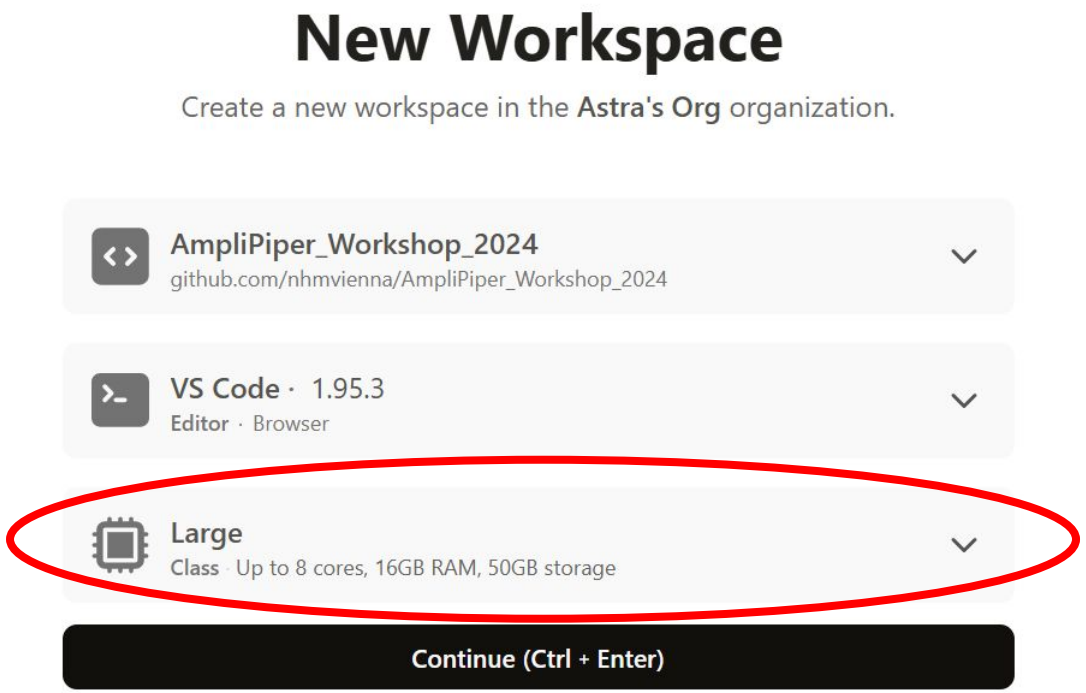
Let’s test AmpliPiper

We prepared a virtual workspace where everyone can run AmpliPiper:

You will find it on the GitHub repository we shared with you!

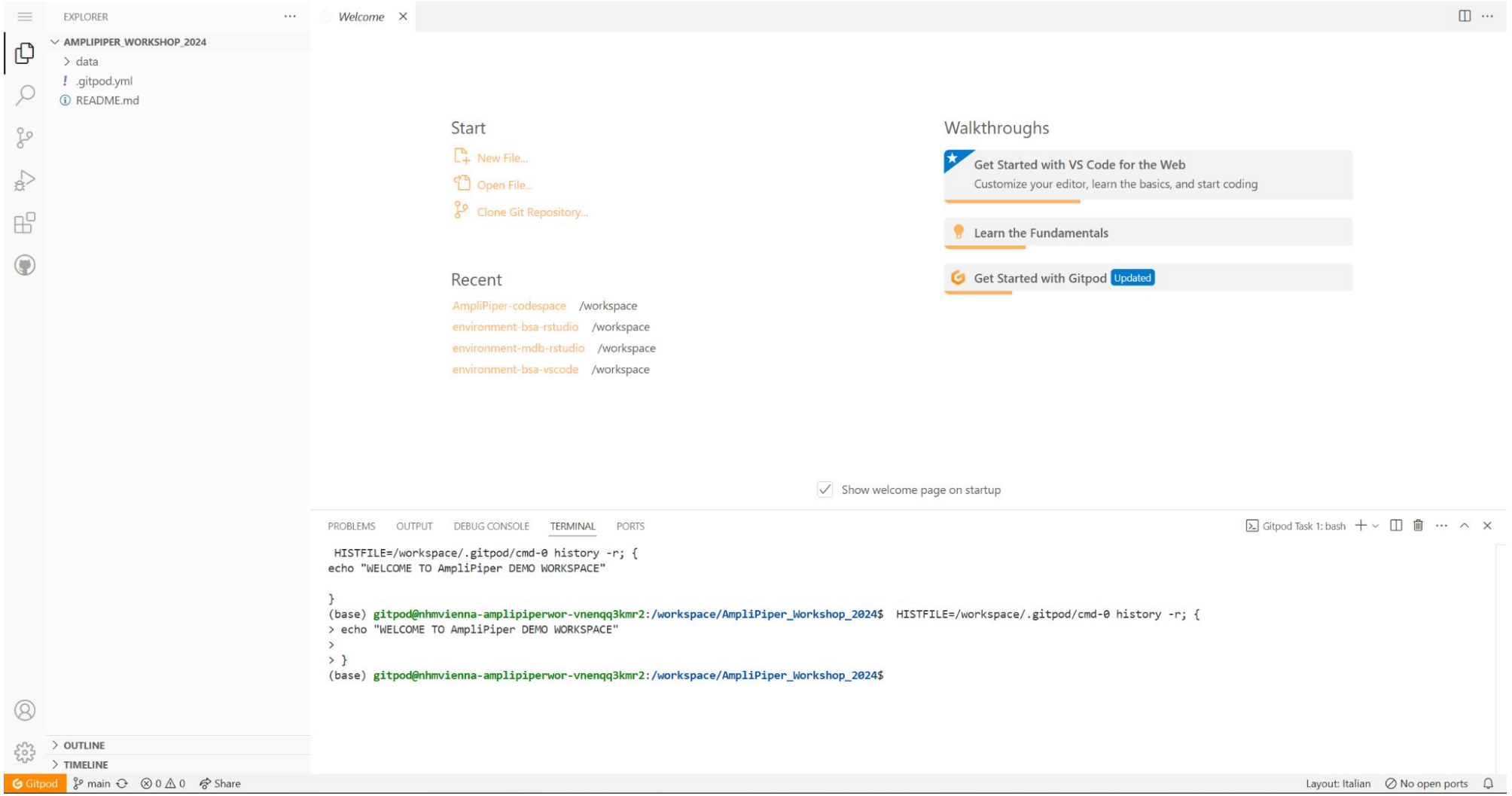
Let’s test AmpliPiper

Choose the **Large** virtual machine instance!



Let’s test AmpliPiper

You will land on a **VSCode** based environment

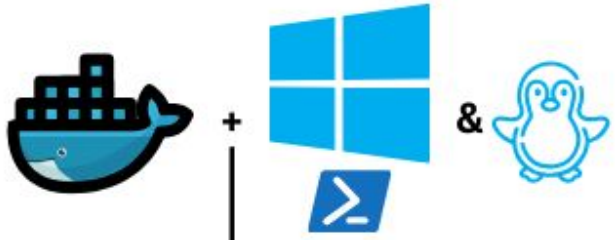


Feeling fancy? Other ways to get AmpliPiper!

Clone the GitHub repository

Execute the installation script

```
(base)user@machine:/proj$ git clone
https://github.com/nhmvienna/AmpliPiper.git
(base)user@machine:/proj$ cd AmpliPiper/
(base)user@machine:/proj$ bash shell/setup.sh
```



Pull the Docker image

Run it directly mounting a volume

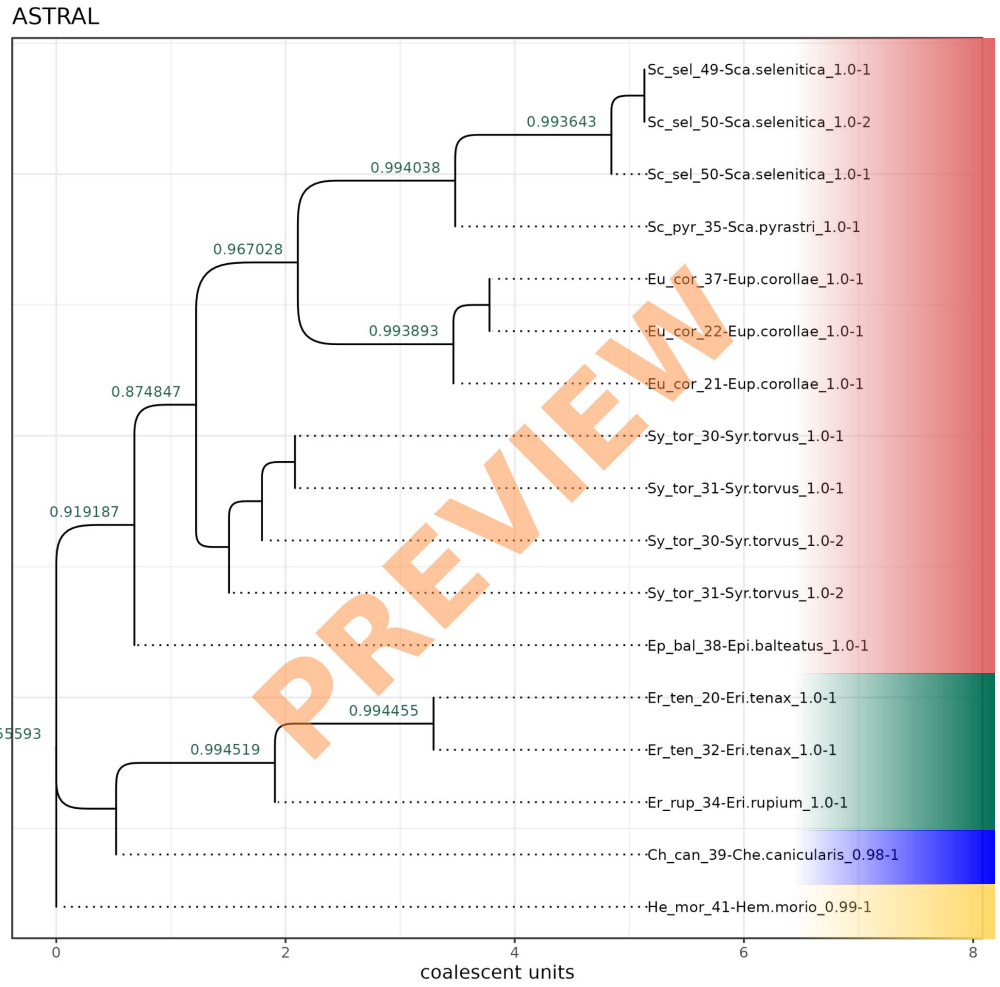
```
PS C:\Users\User> docker pull ghcr.io/nhmvienna/amplipiper:main
PS C:\Users\User> docker run -v c:/path/to/data:/app/userdata
ghcr.io/nhmvienna/amplipiper:main /bin/bash
```

Clone the Workshop repository

```
(base)user@machine:/proj$ git clone
https://github.com/nhmvienna/AmpliPiper_Workshop_2024.git
(base)user@machine:/proj$ cd AmpliPiper_Workshop_2024/
```

Use Docker compose scripts

Summary Syrphidae



Syrphini

Eristalini

Rhingiini

Bombyliidae

- Full agreement with morphological **species identification**
- Correct **phylogenetic clustering**
- High **branch support**
- **Super/concatenated trees** superior to individual loci

Part III - Advanced Datasets, Benchmarking and Limitations

AmpliPiper - optional arguments

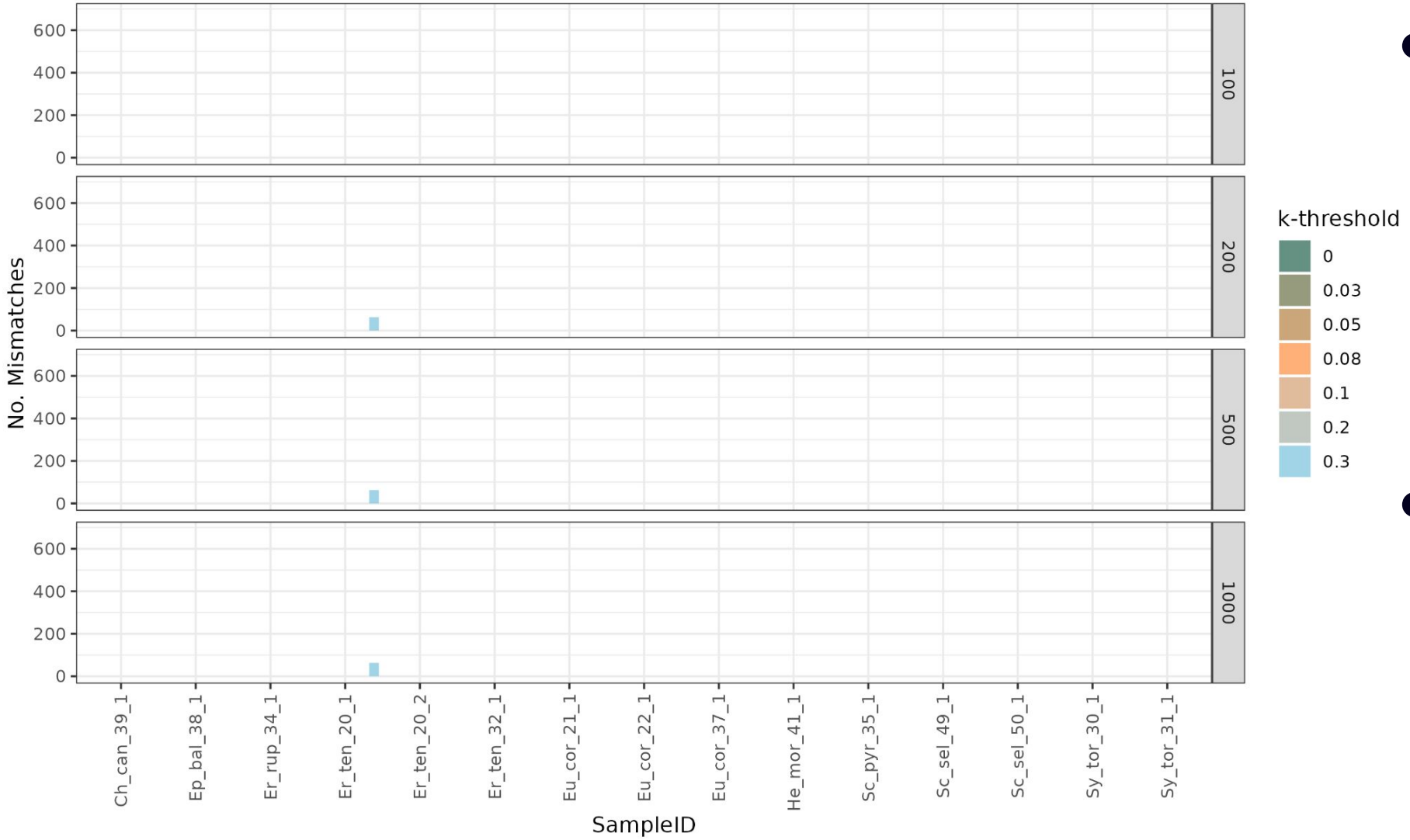
Optional Arguments

- `-b` or `--blast` : Enable BLAST search for species identification. When setting this parameter, you need to provide an email address (e.g., `--blast your@email.com`) for using NCBI entrez to retrieve taxonomic information for the BLAST hits (default: disabled).
- `-c` or `--similar_consensus` : Change the minimum similarity threshold (in percent) of amplicon_sorter. If the similarity of two clusters are smaller or equal to the threshold, they are considered separartely otherwise they are collapsed prior to consensus reconstruction (default: 96)
- `-e` or `--exclude` : Provide a text file with samples and loci to exclude from the analysis. Each row should contain the ID of a sample to be excluded. Names need to be identical to the IDs in `samples.csv`
- `-f` or `--force` : Force overwrite the output folder if it already exists (default: cowardly refusing to overwrite).
- `-i` or `--partition` : Use partition model for iqtree with combined dataset. ⚠ may take very long ⚠ (default: disabled)
- `-k` or `--kthreshold` : Define the threshold *k* for the maximum allowed proportion of mismatches for primer alignment during demultiplexing (default: 0.05).
- `-m` or `--minreads` : Set the minimum number of reads required for consensus sequence reconstruction (default: 100).
- `-n` or `--nreads` : Provide the absolute number or percentage of top-quality reads to consider for consensus sequence generation and variant calling (default: 500).
- `-q` or `--quality` : Specify the minimum PHRED quality score for read filtering (default: 10).
- `-r` or `--sizerange` : Define the allowed size buffer in basepairs around the expected locus length (default: 100).
- `-t` or `--threads` : Specify the number of threads to be used for parallel processing (default: 10).
- `-w` or `--nowatermark` : Remove the watermark from the tree figures
- `-y` or `--freqthreshold` : Retain consensus sequences for further analyses which are supported by raw reads, whose frequency in the total pool of reads is larger or equal to this threshold (default: 0.1).

Benchmarking - comparison to Sanger sequences

- COX1: Pairwise **alignment of Sanger sequences and consensus sequences** from AmpliPiper with edlib
- **Different combination of parameters**
  - k-threshold (The maximum divergence allowed between primer sequences in primers file and read)
  - Minimum number of reads

Benchmarking - comparison to Sanger sequences



- Sequences identical for almost all samples and parameter combinations
- 63 mismatches (~10%) only with k-threshold 30% in single sample!!

Benchmarking - Simulations

Sanger

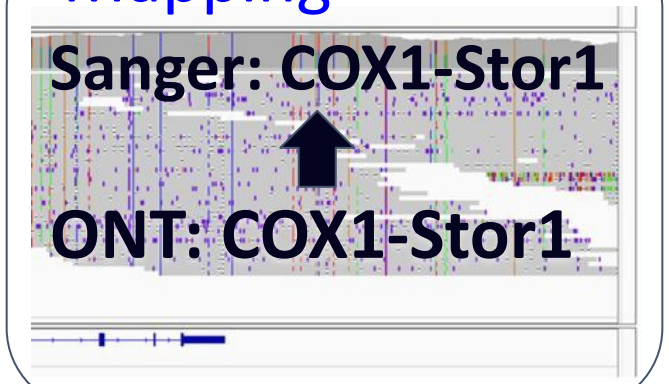
COX1 - Stor_1

Rand. Mutations (1%)

Rand. Mutations (5%)

Benchmarking - Simulations

Mapping



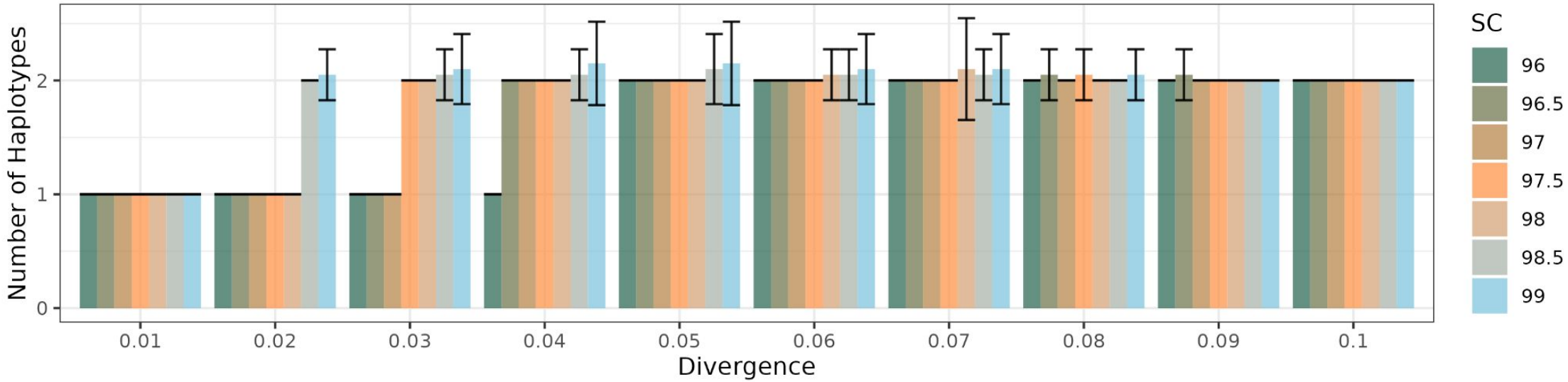
NANOsim

Error - Profile → Simulate based on FASTA → 10,000 x COX1 - Stor_1; 10,000 x Orig + mutations

- 20 x technical replication
- Pooling of simulated reads from original sequence and from mutated sequences (1% - 5% divergence)
- Benchmarking AmpliPiper (No. of Haplotypes, Mismatches)

AmpliPiper - Benchmarking

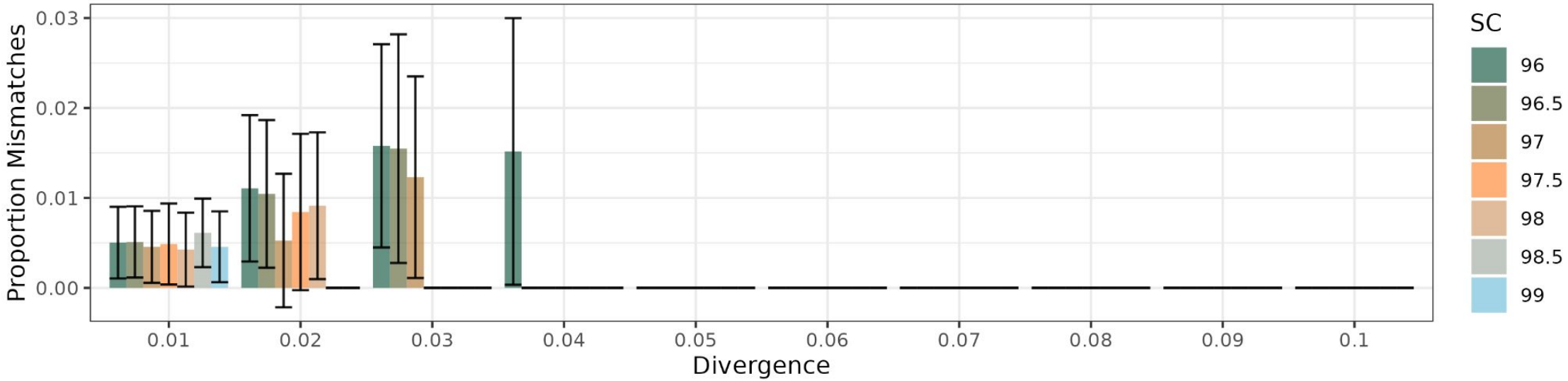
Effects of consensus similarity threshold - Sensitivity



- 50:50 ratio of reads from original and mutated sequences
- **Number of reconstructed consensus sequences**

AmpliPiper - Benchmarking

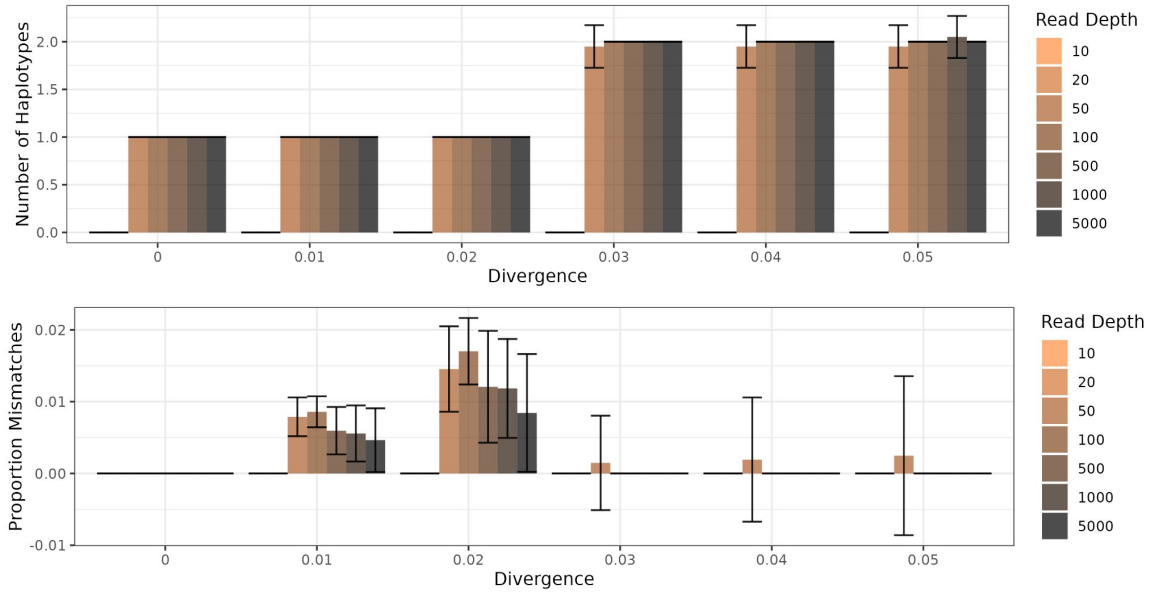
Effects of consensus similarity threshold - Accuracy



- 50:50 ratio of reads from original and mutated sequences
- **Average proportion of mismatches to both original seq.**

AmpliPiper - Benchmarking

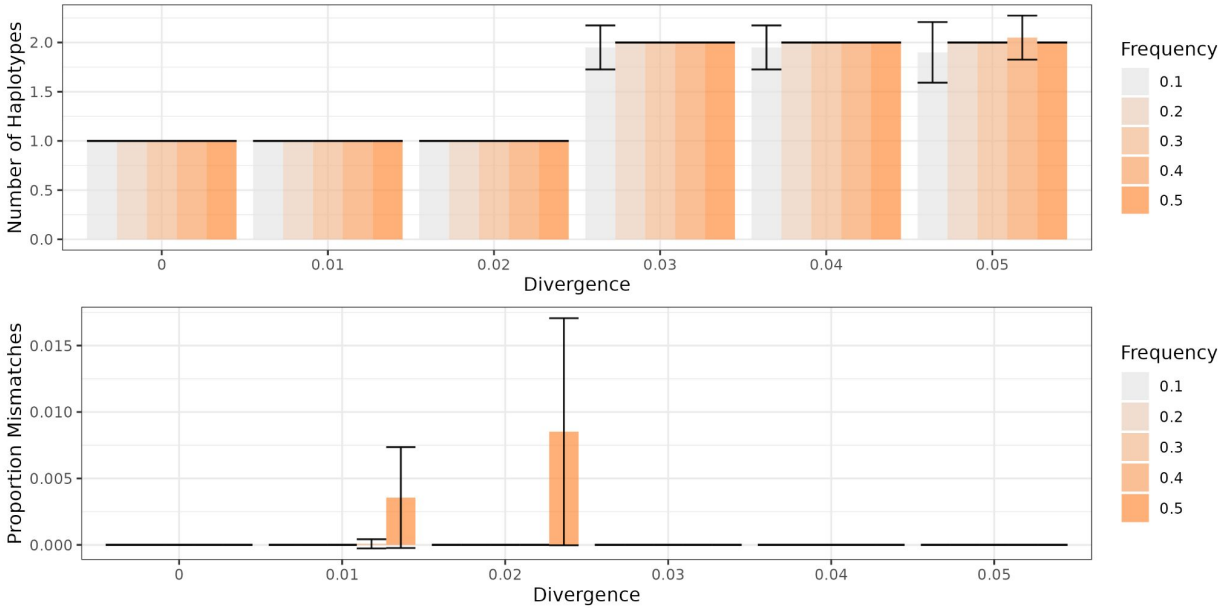
Effect of read depth



Locus reconstruction per sample: unstable or null when using RD≤50

Mismatches with refseq: irrespective of RD if divergence of the reconstructed sequences from the refseq ≤2%

Effects of haplotypes deviating from 1:1 ratio



Locus reconstruction per sample: production of chimeras if divergence of the reconstructed sequences from the refseq ≤2%

Mismatches with refseq: found only with 40-60 and 50-50 ratios & only if the divergence of the reconstructed sequences from refseq ≤2%

Let’s test some more complex datasets

Mercurialis

8 out of 23 samples
10 out of 28 loci

Gerchen, J.F., Veltsos, P., and Pannell, J.R. (2022). *Philos Trans R Soc Lond B Biol Sci* **377**: 20210224.

Opiliones

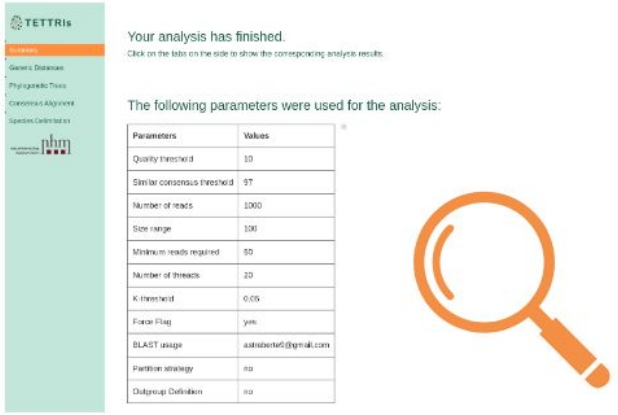
8 samples
COX1 locus

Let’s test some more complex datasets

Pick and prepare a dataset

→

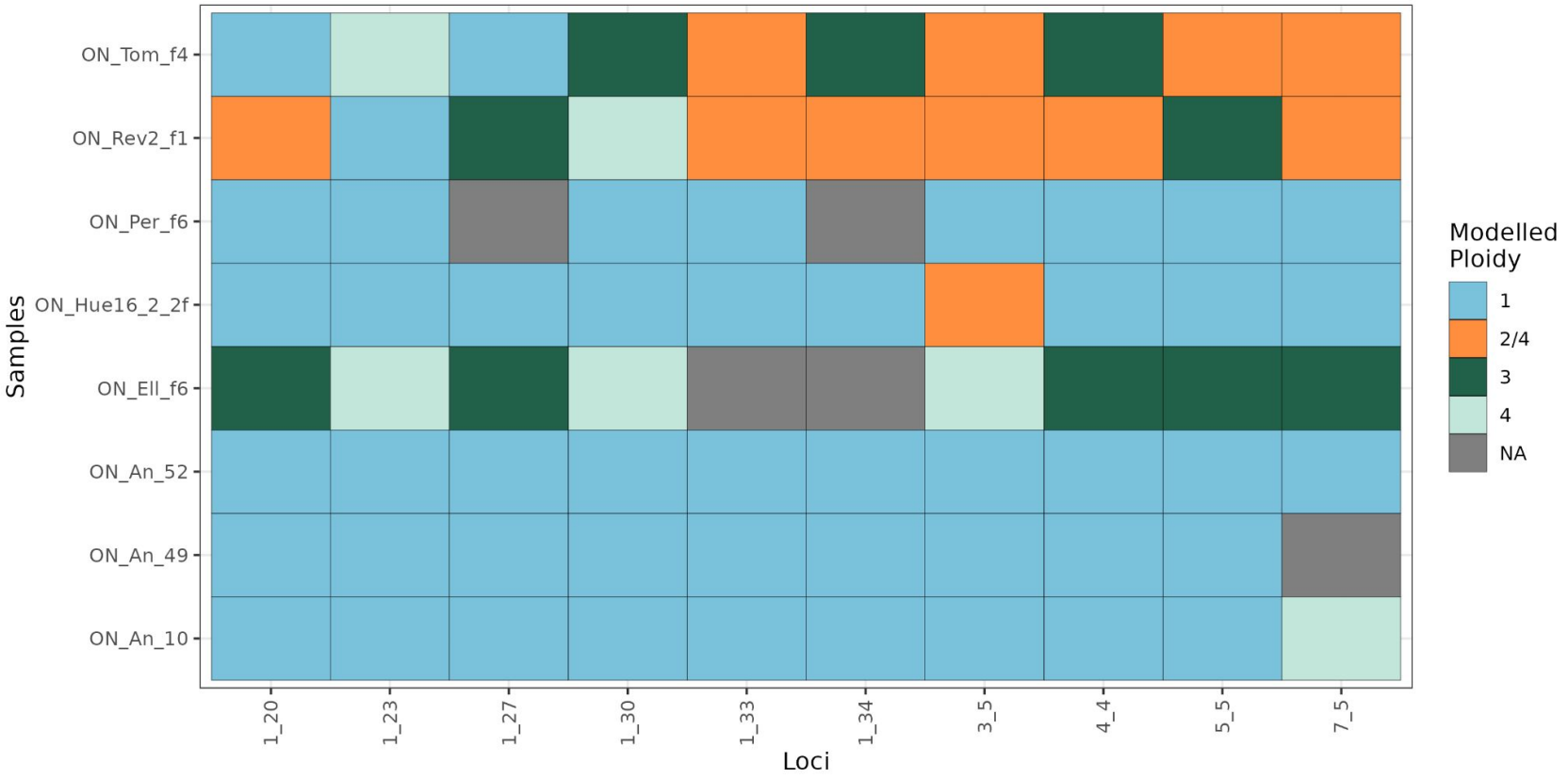
Investigate results



→

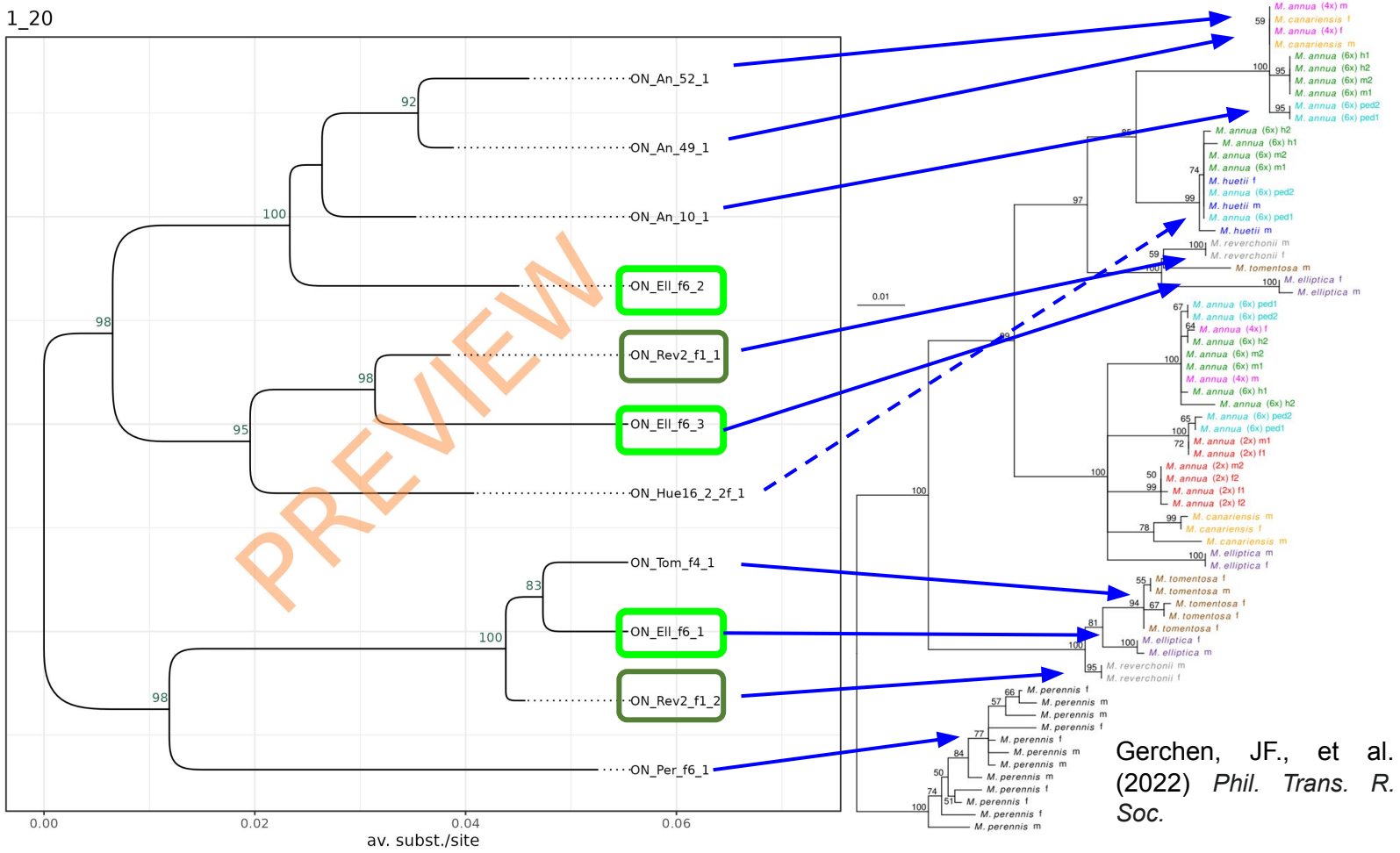
Repeat with different settings

Summary *Mercurialis*



- **Complex ploidy / inbreeding patterns**
  - Limited resolution (hexaploids!!)

Summary *Mercurialis*

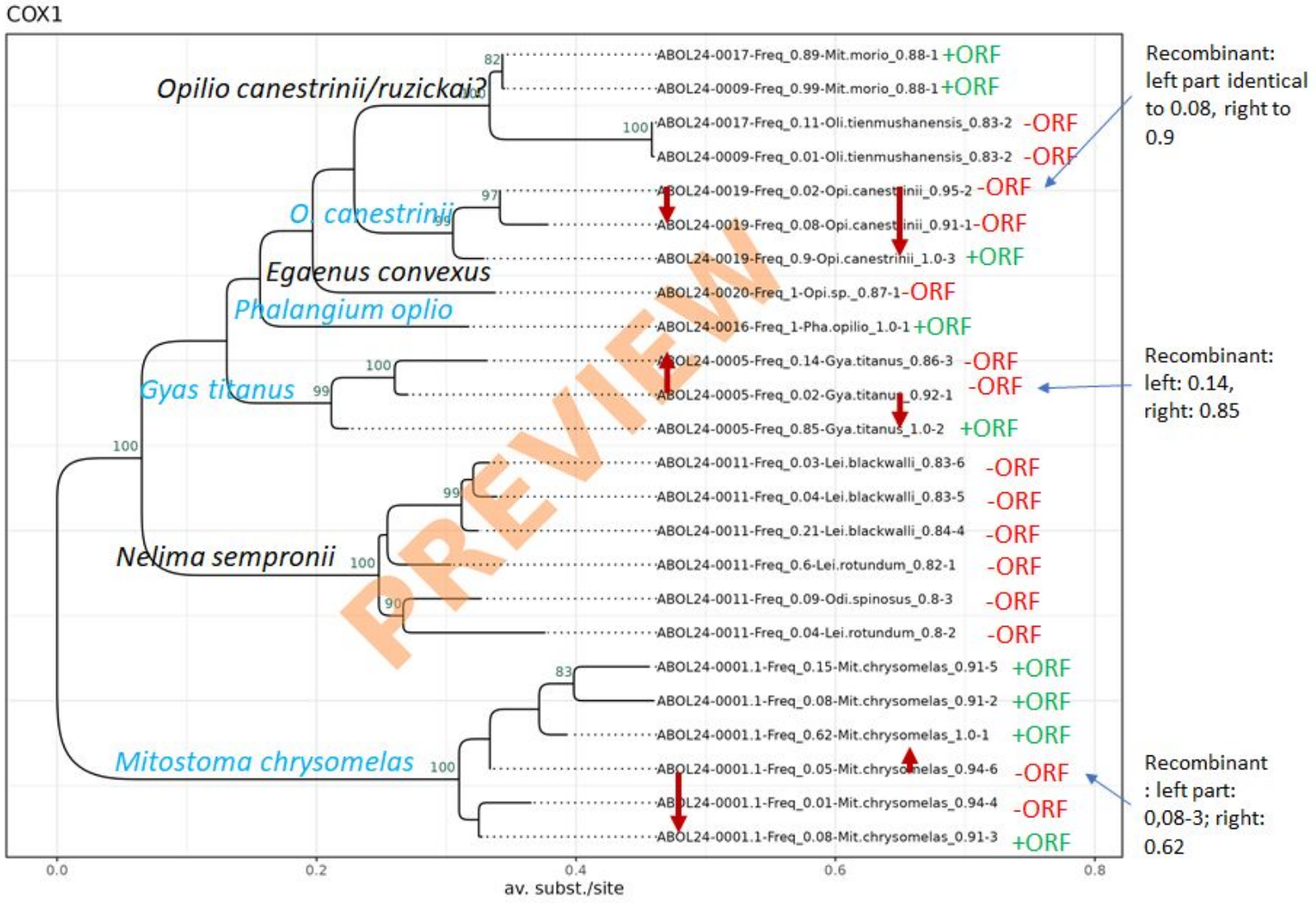


Gerchen, JF., et al. (2022) *Phil. Trans. R. Soc.*

- **Complex ploidy / inbreeding patterns**
  - Limited resolution (hexaploids!!)
- **Allopolyploids**
- **Good congruence** with published data (for some loci)

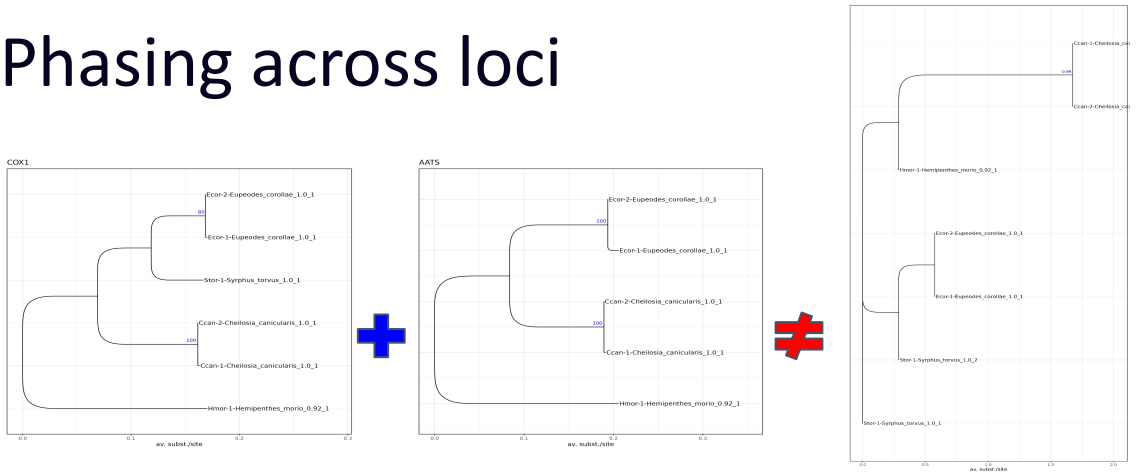
Summary Opiliones

- **Species ID not perfect** (Database bias?)
- Massive amount of **NUMTs**
- Good agreement with **SANGER sequences of clones**
- **Recombinants** !!!
- Additional **manual curation** necessary (ORFs)

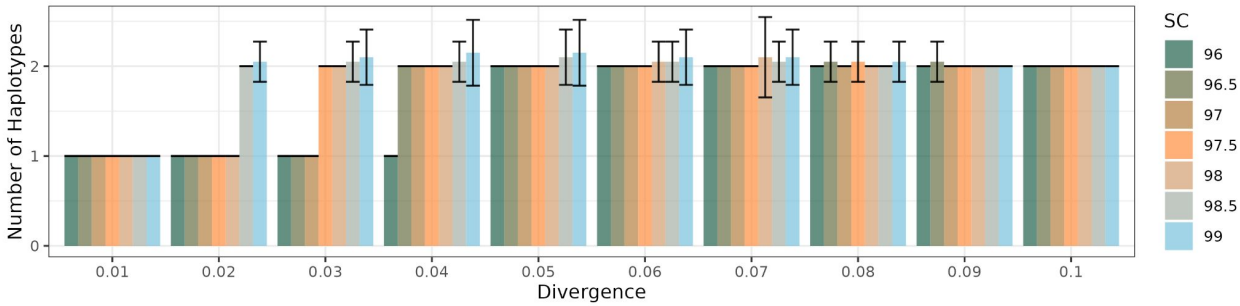


Technical Limitations

Phasing across loci



To distinguish highly similar haplotypes (>97%) within samples

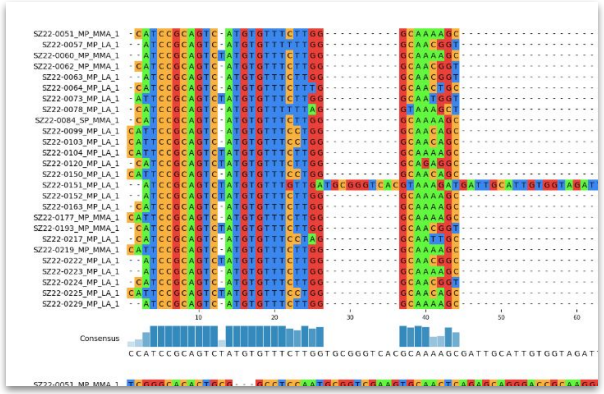


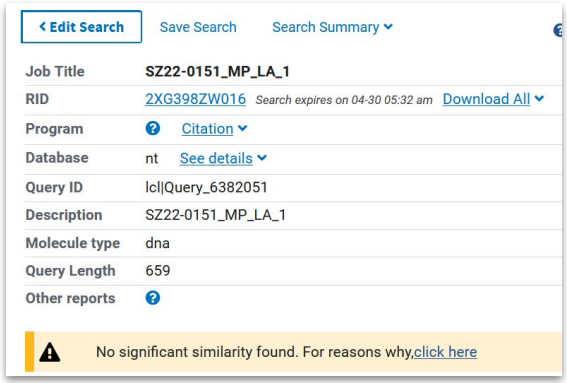
Tricks & Tips

- Try tweaking `--similar_consensus`
- Remember that setting the parameter >0.98 can result in an excess of haplotypes

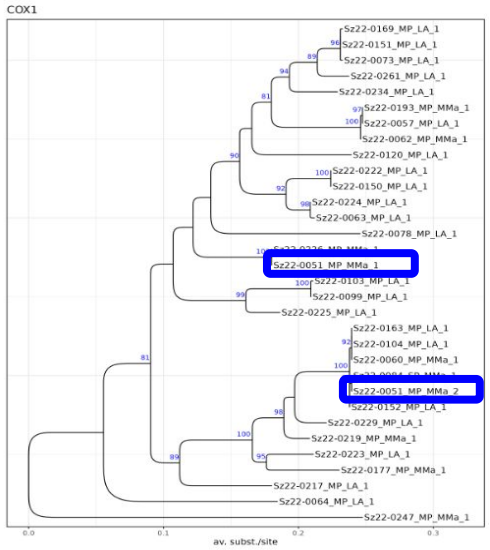
Technical Limitations

PCR artefacts





Contamination



Tricks & Tips

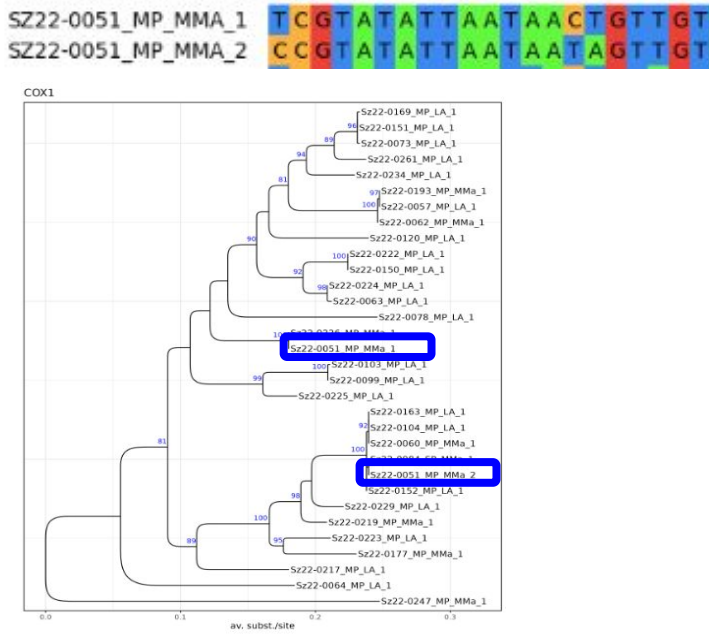
- First run with `--freqthreshold 0` to include all consensus sequences and test for inconsistencies

Tricks & Tips

- Setting a strict `--sizerange` avoids chimeras
- Check species identification
- Exclude the sample(s)/locus(i) that show artefacts

Biological Limitations

Numts/Duplications



Polyploidy/paralogues (multiple haplotypes)

Sample	Locus	#Haplotypes	Read Counts	Frequencies	Ploidy??
ON_A29_2	1_20	3	216/137/147	0.432/0.274/0.294	3
ON_A30_28	1_20	3	168/154/178	0.336/0.308/0.356	3
ON_A4_2h	**1_20**	**2**	**274/226**	**0.548/0.452**	**2/4/6**

Tricks & Tips

- Inspect the frequencies of the consensus sequences
- Compare across multiple loci

Tricks & Tips

- First run with `--freqthreshold 0` to include all consensus sequences and test for inconsistencies
- Check frequency of consensus and compare sequence similarity among consensus sequences

Outlook

- Web interface + HTML I/O
- Markers based on BUSCO genes
- Modularity
- Integration with other databases and web services

Feedback

- Please let us know if…
  - something is not working
  - Could be improved
  - You would love to see an additional feature

https://github.com/nhmvienna/AmpliPiper/issues

Thank you!

Funded by the European Union