UNIVERSITÄT BONN

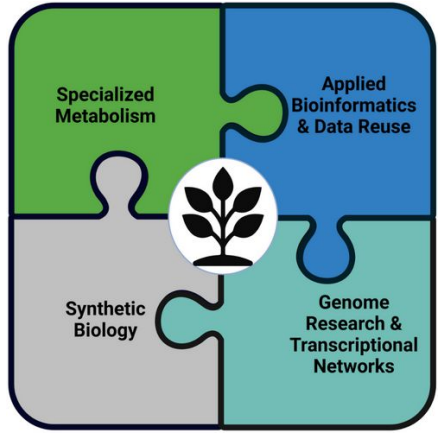
PuckerLab

# de.NBI course 2: From Gene Models to Biological Insights - Functional Annotation

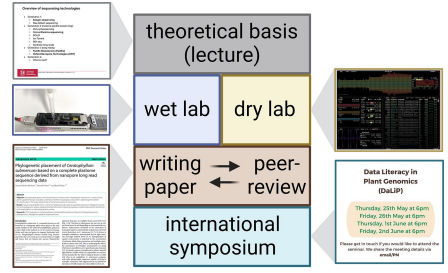
@PuckerLab

@puckerlab

- Flavonoids: Anthocyanins, Flavonols, Flavones, Proanthocyanidins
- Withanolides
- Carotenoids
- …



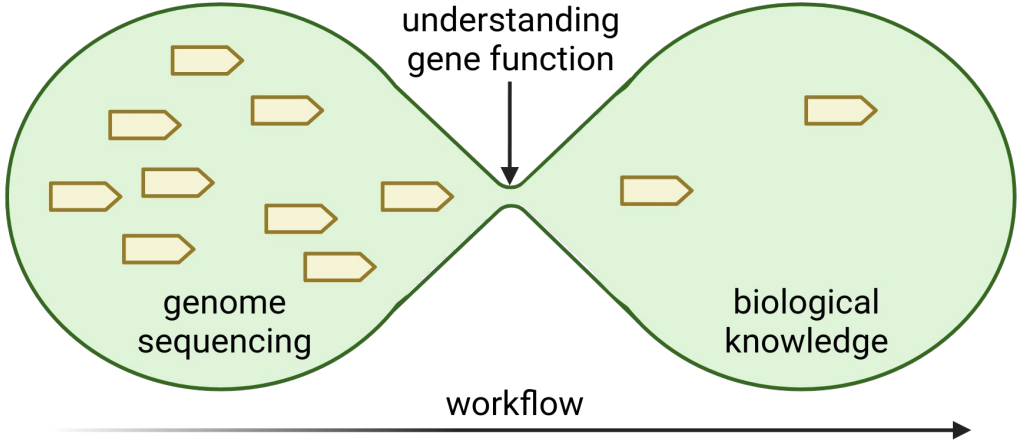

Tools: KIPES, NAVIP, MGSE, MYB/bHLH_annotator, DupyliCate…





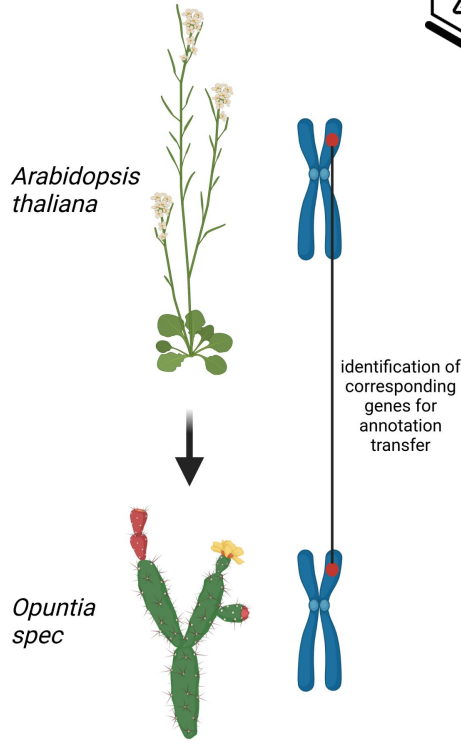

Wolff et al., 2023: 10.1515/jib-2023-0033
Dassow et al., 2025: 10.1101/2025.01.31.635922

# Course motivation




From L to R: Shakunthala, Samuel, Nancy, Julie, and Boas

Picture credits: Jakob Horz






Pucker, 2024: 10.20944/preprints202402.0645.v1

# Availability of Slides

All materials will be shared with participants at the end of this course. Background information is available here:

https://www.preprints.org/manuscript/202508.1176

**Slides** →https://github.com/bpucker/teaching/tree/master/deNBI2025

UNIVERSITÄT BONN

# Start of Day 1


Today you will learn:

- Introduction to Virtual Machine
- Access to Virtual Machine
- Running BUSCO
- Transposable element annotation

# Introduction to Virtual Machines (VM)



- high performance computing => server farm

UNIVERSITÄT BONN

# Introduction to Virtual Machines (VM)


- high performance computing => server farm
- de.NBI offers VMs (VM = virtual machine)
  - running Linux
  - access through SSH

- de.NBI connected LifeScience AAI Account needed

# Introduction to Virtual Machines (VM)

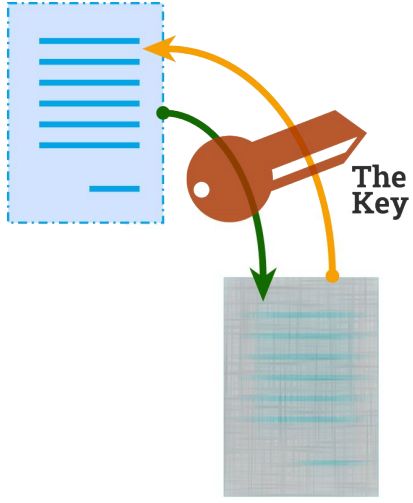
- high performance computing => server farm
- de.NBI offers VMs (VM = virtual machine)
  - running Linux
  - access through SSH

Who is new?

- de.NBI connected LifeScience AAI Account needed

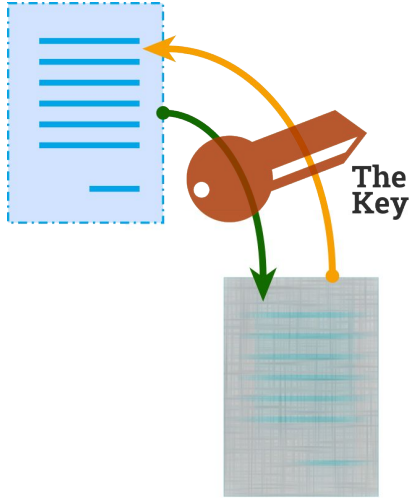
- Tasks:
  1. Register: https://cloud.denbi.de/wiki/registration/
  2. Apply for the course de.NBI project: https://signup.aai.lifescience-ri.eu/fed/registrar/?vo=denbi&group=pbbcourse20251

# SSH keys - asymmetric encryption

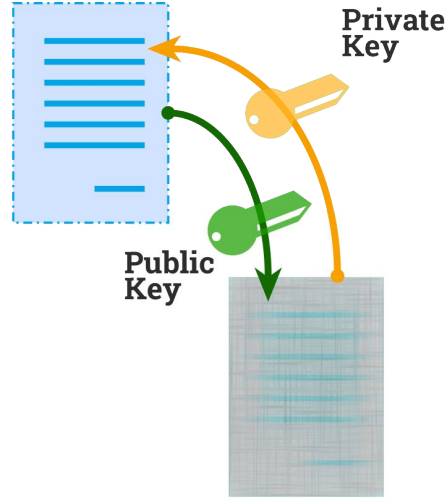



**Symmetric encryption**

# SSH keys - asymmetric encryption

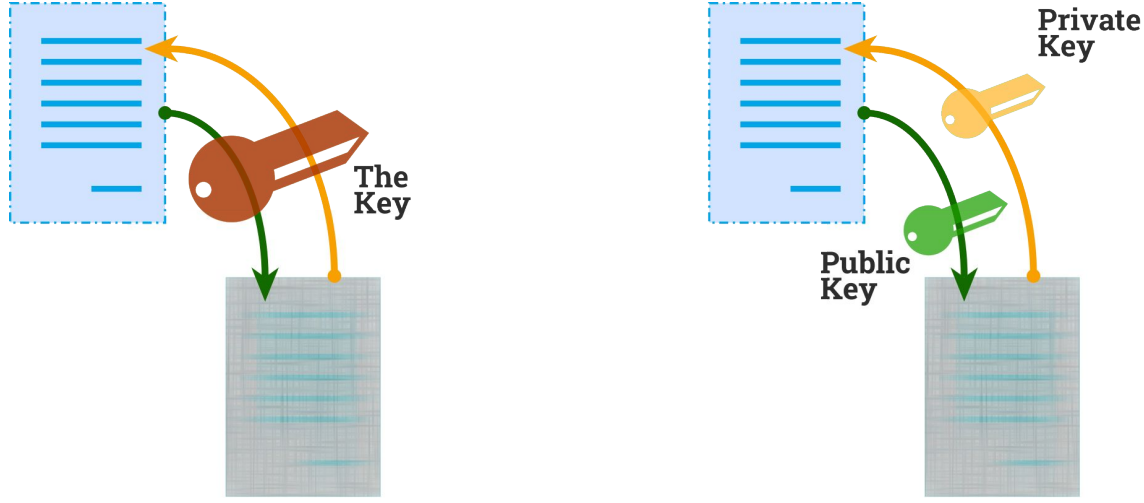



Symmetric encryption




Asymmetric encryption

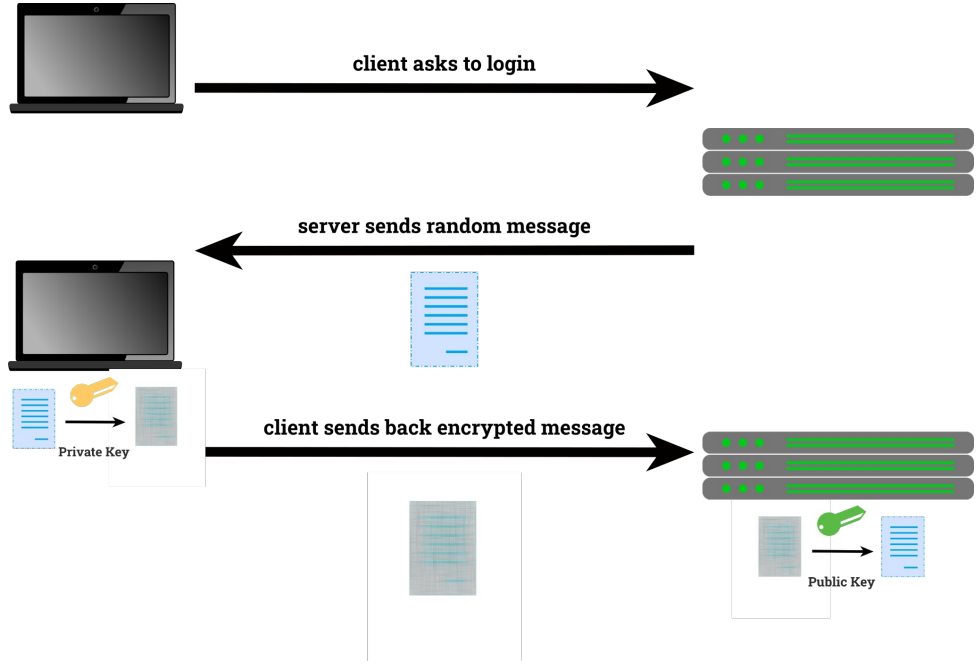
# SSH keys - asymmetric encryption




**Symmetric encryption** **Asymmetric encryption**

ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJHT0aW+Q2bjkfvVM7kN2ucaf8eltoaIV6V0CDBgpWF+dKV1HKkvGWWfjtYMOdLDUKlAzJc5Q6bKs6bkYiqqtUQ=

# SSH keys - asymmetric encryption



# Introduction to Virtual Machines (VM)

- high performance computing => server farm
- de.NBI offers VMs (VM = virtual machine)
  - running Linux
  - access through SSH

- de.NBI connected LifeScience AAI Account needed

- Tasks:
  1. Register: https://cloud.denbi.de/wiki/registration/
  2. Apply for the course de.NBI project: https://signup.aai.lifescience-ri.eu/fed/registrar/?vo=denbi&group=pbbcourse20251
  3. create SSH key pair in de.NBI cloud portal and confirm unix user name

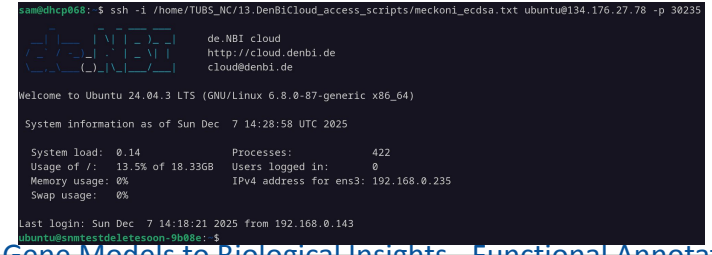
# SSH command and CLI

- SSH key might need the correct permissions
- needed information:
  - username
  - ip address/domain
  - port number (usually optional but mandatory for de.NBI VMs)
  - ssh private key
  - exemplary commands:

https://github.com/bpucker/teaching/tree/master/deNBI2025

ssh -i /path/to/key username@ip-address -p port-number

ssh -i C:\Users\User\de.NBI_course\key.txt ubuntu@134.176.27.78 -p 23457




# BUSCO - evaluating structural annotation

- Good structural annotation needed before functional annotation

UNIVERSITÄT BONN

# BUSCO - evaluating structural annotation


- Good structural annotation needed before functional annotation

- BUSCO = Benchmarking Universal Single-Copy Orthologs

UNIVERSITÄT BONN

# BUSCO - evaluating structural annotation


- Good structural annotation needed before functional annotation
- BUSCO = Benchmarking Universal Single-Copy Orthologs
- Universal genes -> expected to be in the genome
- Singe-Copy -> expected to be present only once per haplophase

UNIVERSITÄT BONN

# BUSCO - evaluating structural annotation


- Good structural annotation needed before functional annotation
- BUSCO = Benchmarking Universal Single-Copy Orthologs
- Universal genes -> expected to be in the genome
- Single-Copy -> expected to be present only once per haplophase
- if all BUSCO genes are present in the assembly => good assembly
- some might be natively missing due to gene loss in evolution

UNIVERSITÄT BONN BUSCO


- exemplary commands:

docker container name (will be automatically downloaded)

list datasets:

```
sudo docker run --rm -v /vol/data:/vol/data ezlabgva/busco:v6.0.0_cv1 busco --list-datasets
```

command that will be run in the docker container starts here

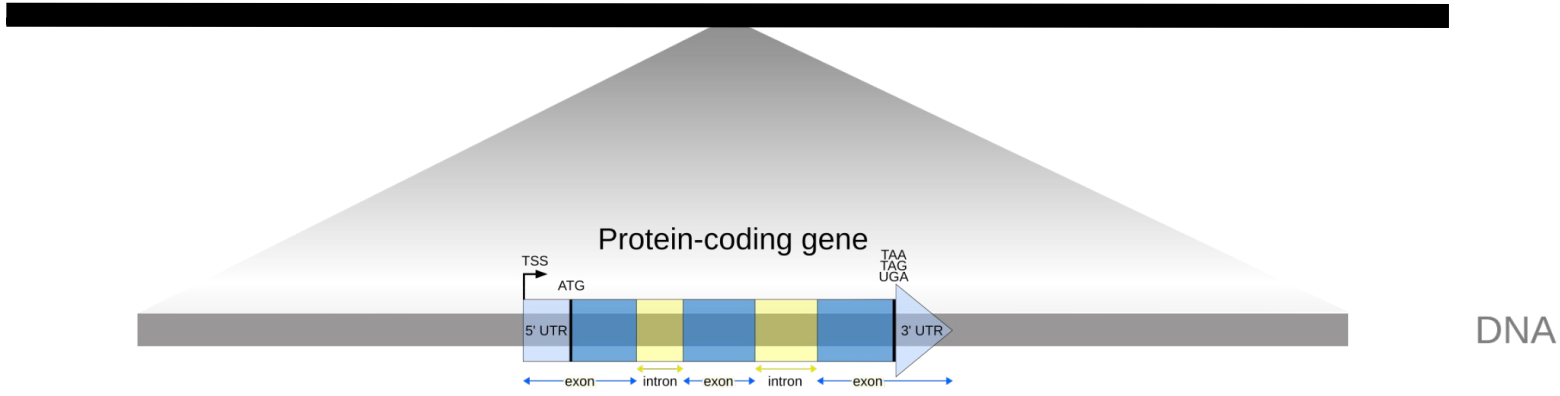
important for the docker container to access files

start the docker container
input file
mode and lineage-dataset
output file

```
sudo docker run --rm -v /vol/data:/vol/data ezlabgva/busco:v6.0.0_cv1 \
busco -i /vol/data/user/protein.fasta \
-m proteins --cpu 13 -l brassicales_odb12 \
--out_path /vol/data/user/BUSCO_pep/ -o busco_run_id --opt-out-run-stats
```

# Structural annotation:

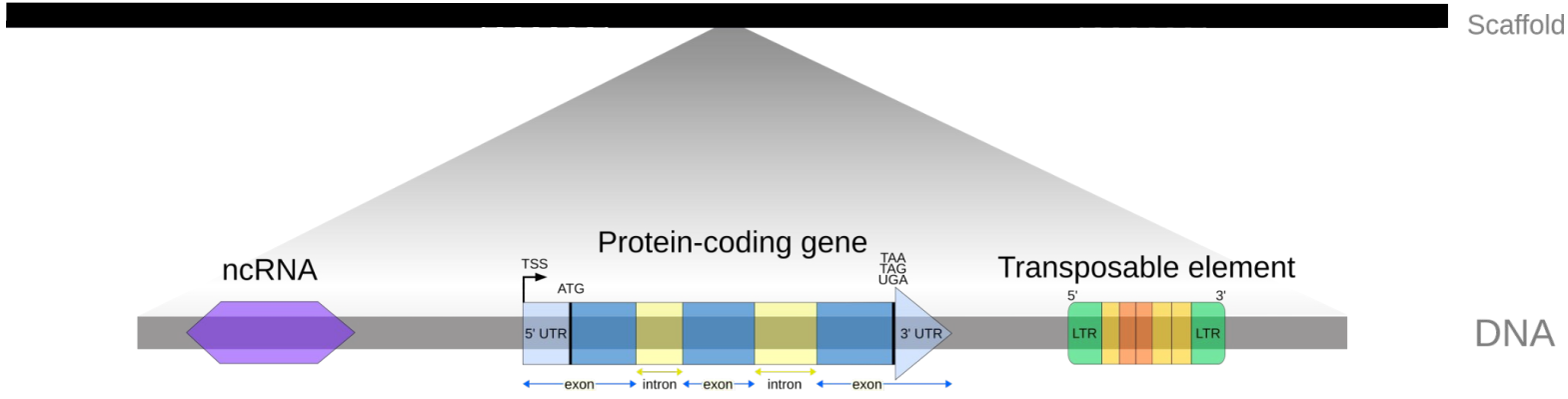
*finding genes in a genome*



UNIVERSITÄT BONN

# Structural annotation:

*finding genes in a genome*

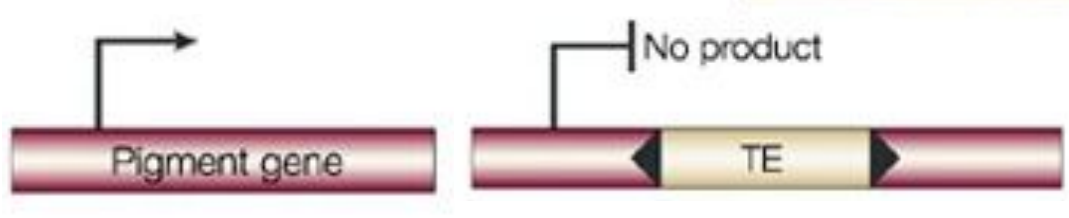



Not just protein-coding genes, but also non-coding DNA like Transposable element (TEs) and non-coding RNAs (ncRNAs)

In plants, only ~1-10% protein-coding genes

# Transposable elements

- Discovered by Barbara McClintock in maize
- TEs responsible for large plant genome sizes
- Based on where they land, they dictate the regulation and impact






Feschotte et al, 2002: 10.1038/nrg793
Jiang et al, 2022: 10.1186/s12864-023-09272-x

# Classes of TEs

| Classification | | Structure | TSD | Code | Occurrence |
|---|---|---|---|---|---|
| Order | Superfamily | | | | |
| *Class I (retrotransposons)* | | | | | |
| LTR | *Copia* | GAG AP INT RT RH | 4–6 | RLC | P, M, F, O |
| | *Gypsy* | GAG AP RT RH INT | 4–6 | RLG | P, M, F, O |
| | *Bel–Pao* | GAG AP RT RH INT | 4–6 | RLB | M |
| | *Retrovirus* | GAG AP RT RH INT ENV | 4–6 | RLR | M |
| | *ERV* | GAG AP RT RH INT ENV | 4–6 | RLE | M |
| DIRS | *DIRS* | GAG AP RT RH YR | 0 | RYD | P, M, F, O |
| | *Ngaro* | GAG AP RT RH YR | 0 | RYN | M, F |
| | *VIPER* | GAG AP RT RH YR | 0 | RYV | O |
| PLE | *Penelope* | RT EN | Variable | RPP | P, M, F, O |
| LINE | *R2* | RT EN | Variable | RIR | M |
| | *RTE* | APE RT | Variable | RIT | M |
| | *Jockey* | ORF1 APE RT | Variable | RIJ | M |
| | *L1* | ORF1 APE RT | Variable | RIL | P, M, F, O |
| | *I* | ORF1 APE RT RH | Variable | RII | P, M, F |
| SINE | *tRNA* | | Variable | RST | P, M, F |
| | *7SL* | | Variable | RSL | P, M, F |
| | *5S* | | Variable | RSS | M, O |
| *Class II (DNA transposons) - Subclass 1* | | | | | |
| TIR | *Tc1–Mariner* | Tase* | TA | DTT | P, M, F, O |
| | *hAT* | Tase* | 8 | DTA | P, M, F, O |
| | *Mutator* | Tase* | 9–11 | DTM | P, M, F, O |
| | *Merlin* | Tase* | 8–9 | DTE | M, O |
| | *Transib* | Tase* | 5 | DTR | M, F |
| | *P* | Tase | 8 | DTP | P, M |
| | *PiggyBac* | Tase | TTAA | DTB | M, O |
| | *PIF–Harbinger* | Tase* ORF2 | 3 | DTH | P, M, F, O |
| | *CACTA* | Tase ORF2 | 2–3 | DTC | P, M, F |
| Crypton | *Crypton* | YR | 0 | DYC | F |
| *Class II (DNA transposons) - Subclass 2* | | | | | |
| Helitron | *Helitron* | RPA Y2 HEL | 0 | DHH | P, M, F |

LTR copia

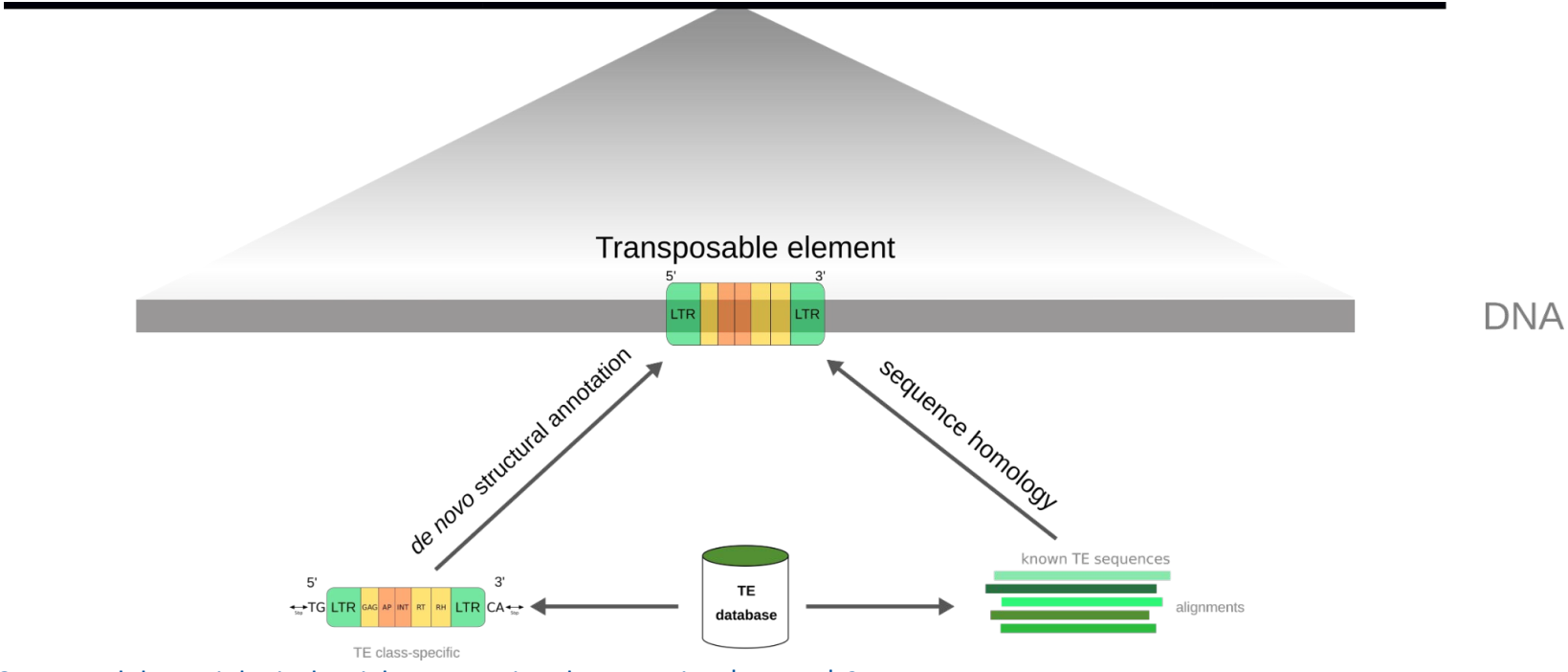
LTR gypsy


LINE L1




Wicker et al, 2007: 10.1038/nrg2165

# Annotating TEs



# Annotating TEs -Tools


## Specific tools

- LTR_FINDER/ LTR_HARVEST ⟶ LTRs
- TIR-LEARNER ⟶ TIRs
- Helitron scanner ⟶ Helitrons

# Annotating TEs -Tools

Specific tools

- LTR_FINDER/ LTR_HARVEST ⟶ LTRs
- TIR-LEARNER ⟶ TIRs
- Helitron scanner ⟶ Helitrons

## EDTA (Extensive *de novo* TE annotator)

- pipeline combining above tools
- appropriate filtering
- Final TE library

# EDTA Installation


**Installing Miniconda**

cd /vol/data/tools ← #Important not in home

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

bash Miniconda3-latest-Linux-x86_64.sh

# the above command will interactively ask 4 questions, first press ENTER to continue
# Do you accept license terms? [yes|no] →yes
#Miniconda3 will be installed in /home/ubuntu/miniconda3 or specify a different location → /vol/data/tools/Miniconda3
#Proceed with initialization [yes|no] → yes

source ~/.bashrc

conda config --add channels conda-forge

conda update -n base --all

conda install -n base mamba

# Installing EDTA

## Installing EDTA through conda/mamba

mamba create -n EDTA2.2 -c conda-forge -c bioconda -c r annosine2 biopython cd-hit coreutils genericrepeatfinder genometools-genometools glob2 tir-learner ltr_finder_parallel ltr_retriever mdust multiprocess muscle openjdk perl perl-text-soundex r-base r-dplyr regex repeatmodeler r-ggplot2 r-here r-tidyr tesorter samtools bedtools LTR_HARVEST_parallel HelitronScanner

## Getting the latest version of EDTA from GitHub

cd /vol/data/tools; git clone https://github.com/oushujun/EDTA.git; cd EDTA; git checkout EDTA2; git branch

## Running EDTA

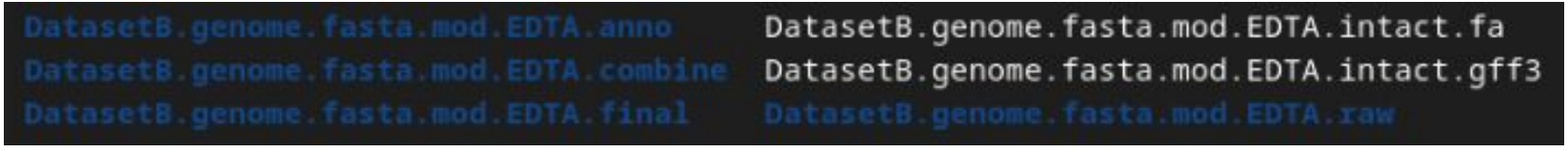
cd /vol/data/user

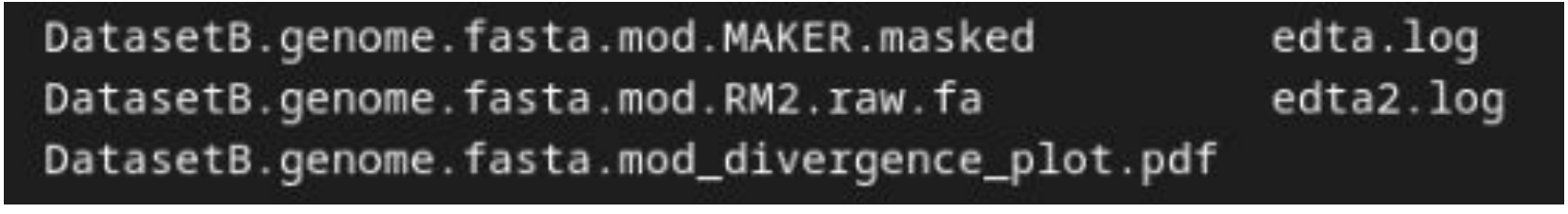
mkdir EDTA_results; cd EDTA_results

conda activate EDTA2.2

perl /vol/data/tools/EDTA/EDTA.pl --genome /vol/data/DatasetB.genome.fasta --overwrite 1 --anno 1 --sensitive 1 --evaluate 1 --threads 13 >>edta.log 2>&1 &

# EDTA- output and results

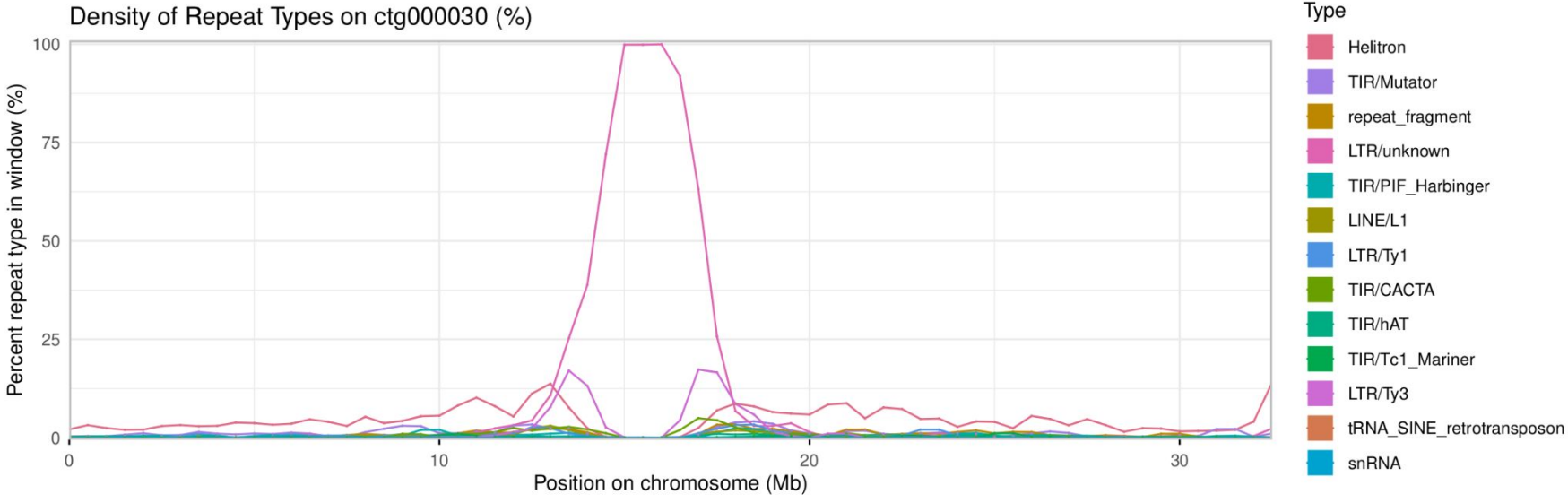
# EDTA- output and results

1. *.EDTA.TEanno.sum
2. *.EDTA.TEanno.gff3
3. Density plots for top contigs

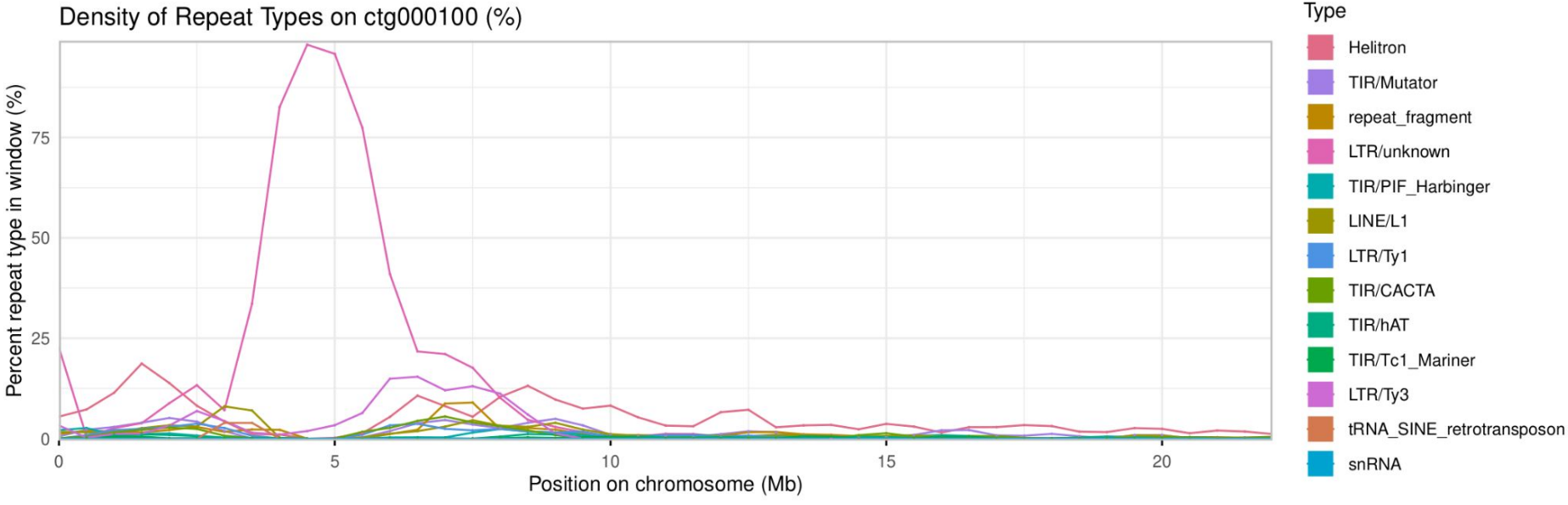
```
Repeat Classes
==============
Total Sequences: 13
Total Length: 134799993 bp
Class                  Count        bpMasked    %masked
=====                  =====        ========     =======
LINE                   --           --           --
    L1                 1239         989157       0.73%
LTR                    --           --           --
    Copia              738          1002959      0.74%
    Gypsy              2081         3195026      2.37%
    unknown            5660         16713457     12.40%
SINE                   --           --           --
    tRNA               143          179011       0.13%
TIR                    --           --           --
    CACTA              1668         918659       0.68%
    Mutator            3170         1767375      1.31%
    PIF_Harbinger      1751         728452       0.54%
    Tc1_Mariner        539          259836       0.19%
    hAT                807          335497       0.25%
nonTIR                 --           --           --
    helitron           11058        5960229      4.42%
repeat_fragment        3879         1580072      1.17%
                       ---------------------------------
    total interspersed 32733        33629730     24.95%

snRNA                  8            859          0.00%
-----------------------------------------------------------
Total                  32741        33630589     24.95%
```

# Density plot



# Density plot



UNIVERSITÄT BONN

# End of day 1


Today you learned:

- Accessing de.NBI VMs via SSH
- Working in a CLI environment
- Running BUSCO
- Identifying transposon elements

Tomorrow you will learn:

- tRNA annotation
- Running InterProScan
- KIPEs3, MYB_annotator
- Tree building

UNIVERSITÄT BONN

# Start of day 2


Yesterday you learned:

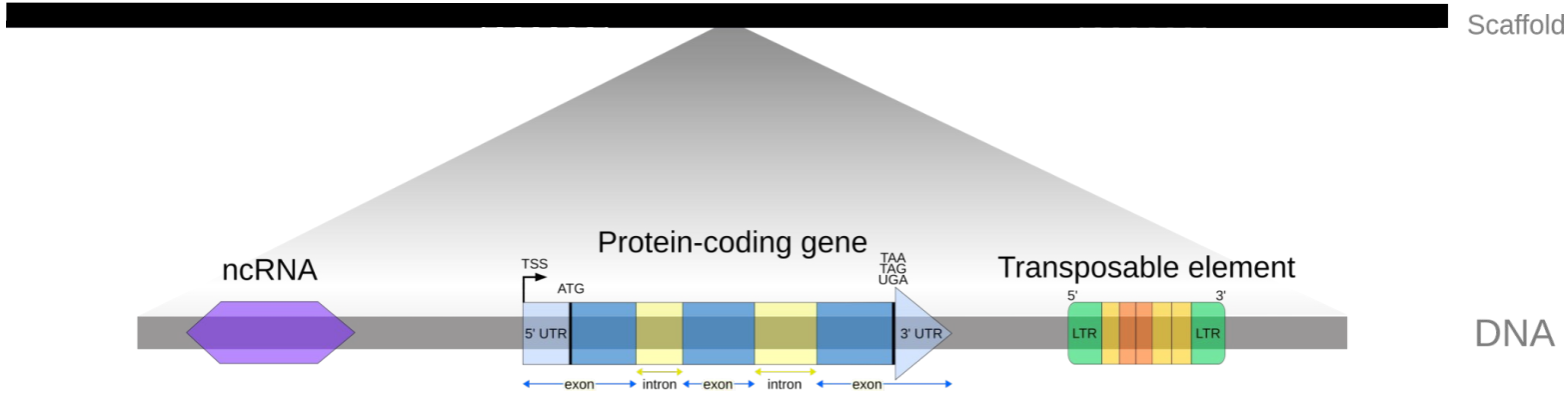
- Accessing de.NBI VMs via SSH
- Working in a CLI environment
- Running BUSCO
- Identifying transposon elements

Today you will learn:

- tRNA annotation
- Running InterProScan
- KIPEs3, MYB_annotator
- Tree building

# Structural annotation:

*finding genes in a genome*




Not just protein-coding genes, but also non-coding DNA like Transposable element (TEs) and non-coding RNAs (ncRNAs)

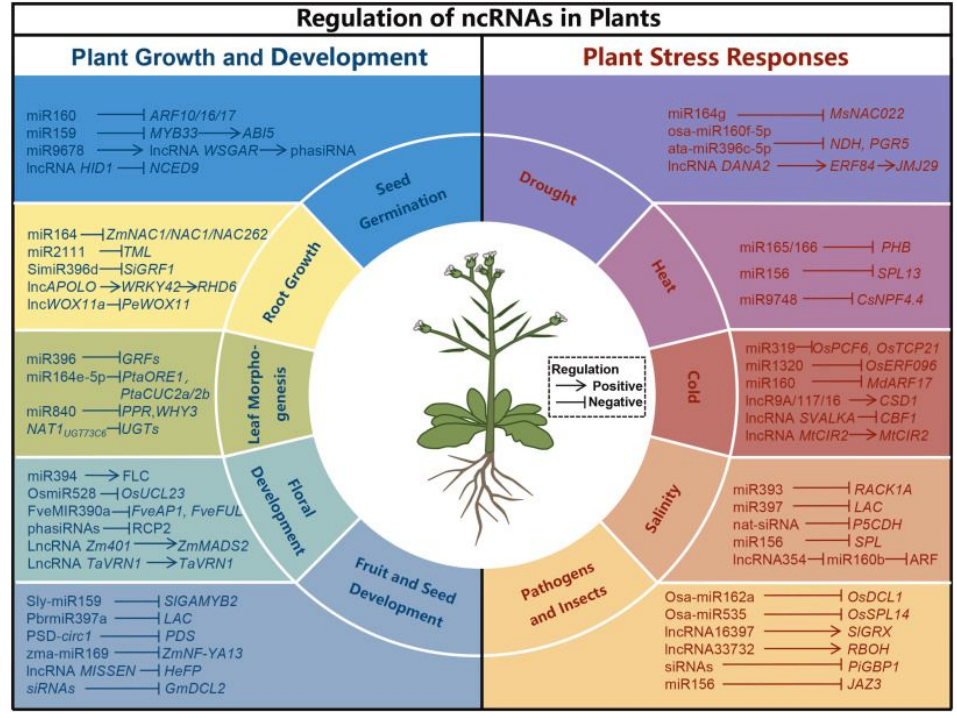
In plants, only ~1-10% protein-coding genes

# Annotating ncRNAs



## Types of Non-coding RNAs (ncRNAs)

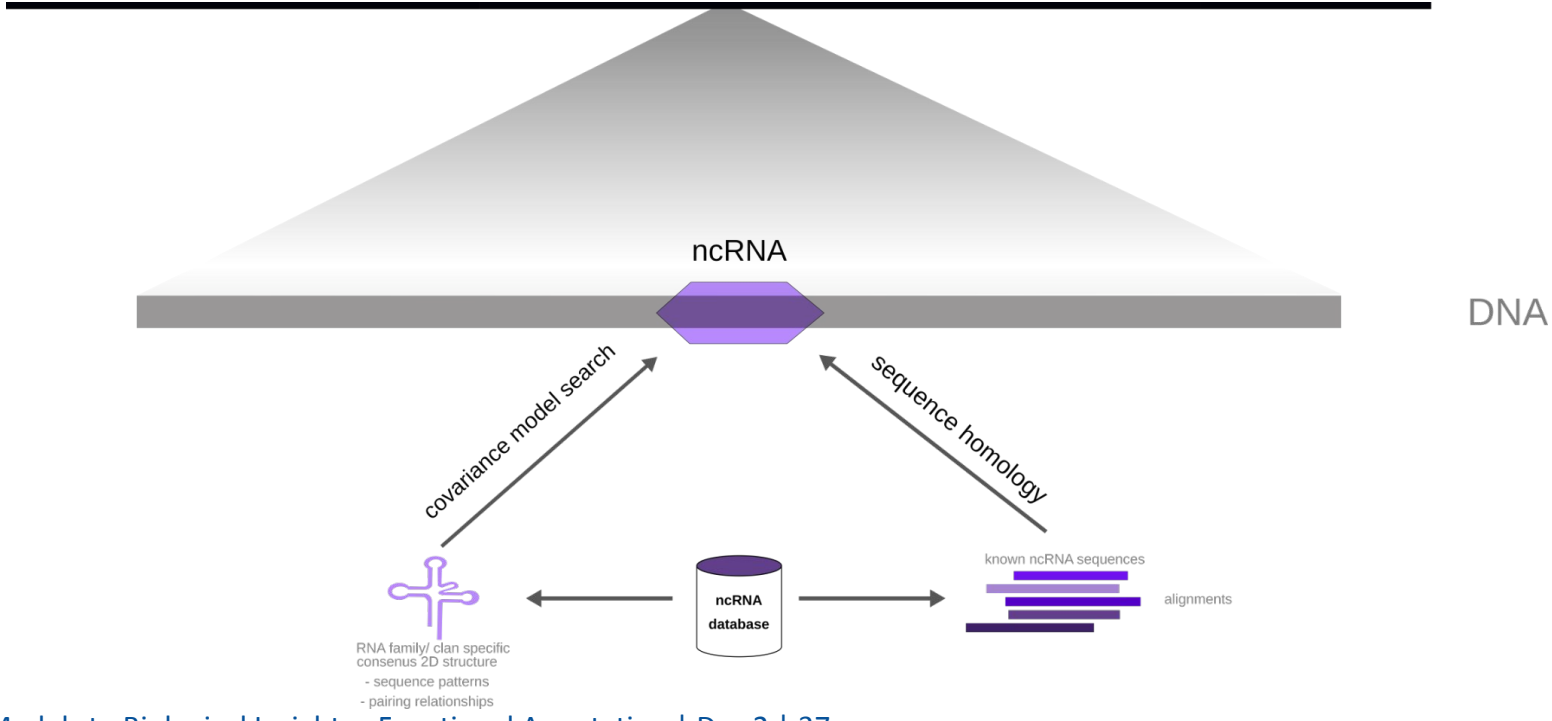
- transfer RNAs (tRNAs)
- ribosomal RNAs (rRNA)
- small RNAs like microRNAs, siRNAs, snoRNAs, snRNAs






Cao et al, 2024: 10.1016/j.plaphy.2024.108435

# Annotating ncRNAs



# Annotating ncRNA - State of Art


- Infernal (INFERence of RNA ALignment)

  Identifies and classifies all ncRNAs in a genome sequence based on Rfam databases' calibrated covariance models

## Specialized tools for different ncRNA classes

- tRNAs ⟶ tRNAscan-SE 2.0
- rRNA ⟶RNAmmer, SSU-Align

# Installing tRNAscan-SE 2.0

cd /vol/data/tools ← location to install all tools

Get the latest release of tRNAscan-SE2.0 from GitHub (https://github.com/UCSC-LoweLab/tRNAscan-SE)



right click and copy link address

wget https://github.com/UCSC-LoweLab/tRNAscan-SE/archive/refs/tags/v2.0.12.tar.gz #paste after wget in tools directory

tar -xvzf v2.0.12.tar.gz #unarchive extract (-x), verbose (-v), unzip (-z), output in a new folder (-f)

cd tRNAscan-SE-2.0.12/

# Installing tRNAscan-SE 2.0

sudo ./configure

sudo make

sudo make install

## Install dependency → Infernal

(http://eddylab.org/infernal/)

*Installation where tRNAscan-SE 2.0 is installed*

wget http://eddylab.org/infernal/infernal-1.1.5.tar.gz

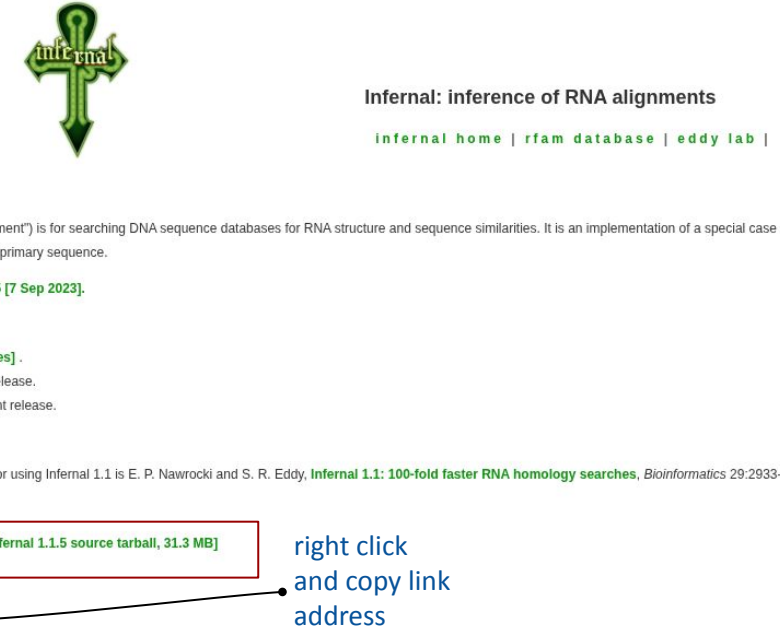
tar -xvzf infernal-1.1.5.tar.gz

cd infernal-1.1.5

sudo ./configure

sudo make

sudo make install



# Running tRNAscan-SE 2.0

```
(base) ubuntu@lilyamir-4e1bc:~$ tRNAscan-SE

tRNAscan-SE 2.0.12 (Nov 2022)

FATAL: No sequence file(s) specified.

Usage: tRNAscan-SE [-options] <FASTA file(s)>

  Scan a sequence file for tRNAs
         -- default: use Infernal & tRNA covariance models
            with eukaryotic sequences
            (use -B, -A, -M, -O or -G to scan other types of sequences)
```

**Example command:**

tRNAscan-SE -o Athaliana_tRNA.out -m Athaliana_tRNA.stats /vol/data/DatasetB.genome.fasta

# tRNAscan-SE 2.0 result files


## 1. Athaliana_tRNA.stats

| | |
|---|---|
| tRNAs decoding Standard 20 AA: | 881 |
| Selenocysteine tRNAs (TCA): | 0 |
| Possible suppressor tRNAs (CTA,TTA,TCA): | 0 |
| tRNAs with undetermined/unknown isotypes: | 9 |
| Predicted pseudogenes: | 18 |
| | ------- |
| Total tRNAs: | 908 |
| tRNAs with introns: | 70 |

## 2. Athaliana_tRNA.out → comprehensive result file with positions of predicted tRNAs in the genome sequence

UNIVERSITÄT BONN

# Functional annotation


- What is the function of a gene
- Knockout experiments for all genes are expensive and time consuming
- Annotation transfer - orthologs are assumed to have the same function

# InterProScan - Functional annotation

- InterPro - database integrating predictive info about protein function from a number of partner resources - CATH, CDD, PANTHER, Pfam and so on
- Hosted and maintained by EMBL-EBI
- InterProScan5 - software to perform functional annotation of proteins
- Integrated with the InterPro database

# InterProScan5 installation


## Installing pre-requisites

- java -version

```
openjdk version "11.0.4" 2019-07-16
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.4+11)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.4+11, mixed mode)
```

If not, do:

sudo apt update

sudo apt install openjdk-11-jdk and check java -version

sudo update-alternatives --config java (select java-11)

# InterProScan5 installation


#Set JAVA_HOME for OpenJDK 11

- readlink -f /usr/bin/java | sed "s:/bin/java::"
- /usr/lib/jvm/java-11-openjdk-amd64
- echo 'export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64' >> ~/.bashrc
- echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc
- source ~/.bashrc

UNIVERSITÄT BONN

# InterProScan5 installation


## Step1: fetch InterProScan tarball

cd /vol/data/tools

wget
https://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.76-107.0/interproscan-5.76-107.0-64-bit.tar.gz

## Step 2: fetch checksum

wget
https://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.76-107.0/interproscan-5.76-107.0-64-bit.tar.gz.md5

## Step 3:#checksum to confirm the download was successful:

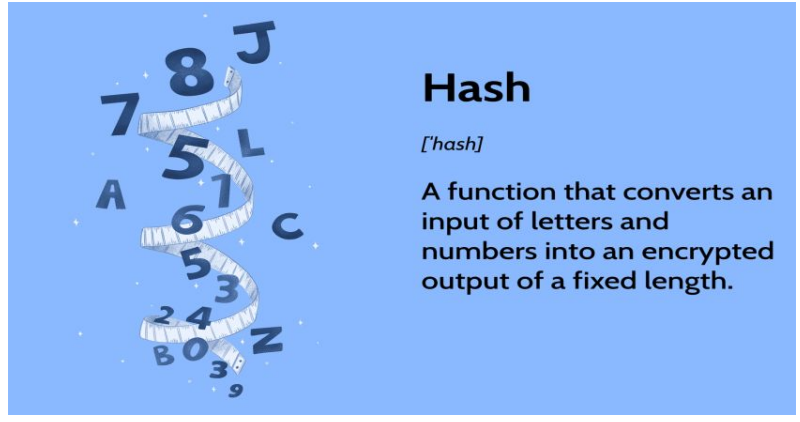
md5sum -c interproscan-5.76-107.0-64-bit.tar.gz.md5

# Must return *interproscan-5.76-107.0-64-bit.tar.gz: OK*

## Step 4:#Extract the tar ball

tar -pxvzf interproscan-5.76-107.0-*-bit.tar.gz

# Hash and md5sum




- Message Digest Algorithm 5
- Verifies file authenticity and integrity
- Simply, to ensure the data fetched is correct and complete

# InterProScan5 setting up

#Run the setup script before the 1st time you run InterProScan

cd interproscan-5.76-107.0

python3 setup.py -f interproscan.properties

#Check

./interproscan.sh

UNIVERSITÄT BONN

# Preprocessing PEP file


- InterProScan complains about *
- Remove * before running InterProScan
- #Command

```
cd /vol/data/InterProScan5_input

sed 's/\*//g' Arabidopsis_thaliana.pep.fasta > Arabidopsis_thaliana_cleaned.pep.fasta
```

UNIVERSITÄT BONN

# Running InterProScan5


#test run

cd /vol/data/tools/interproscan-5.76-107.0

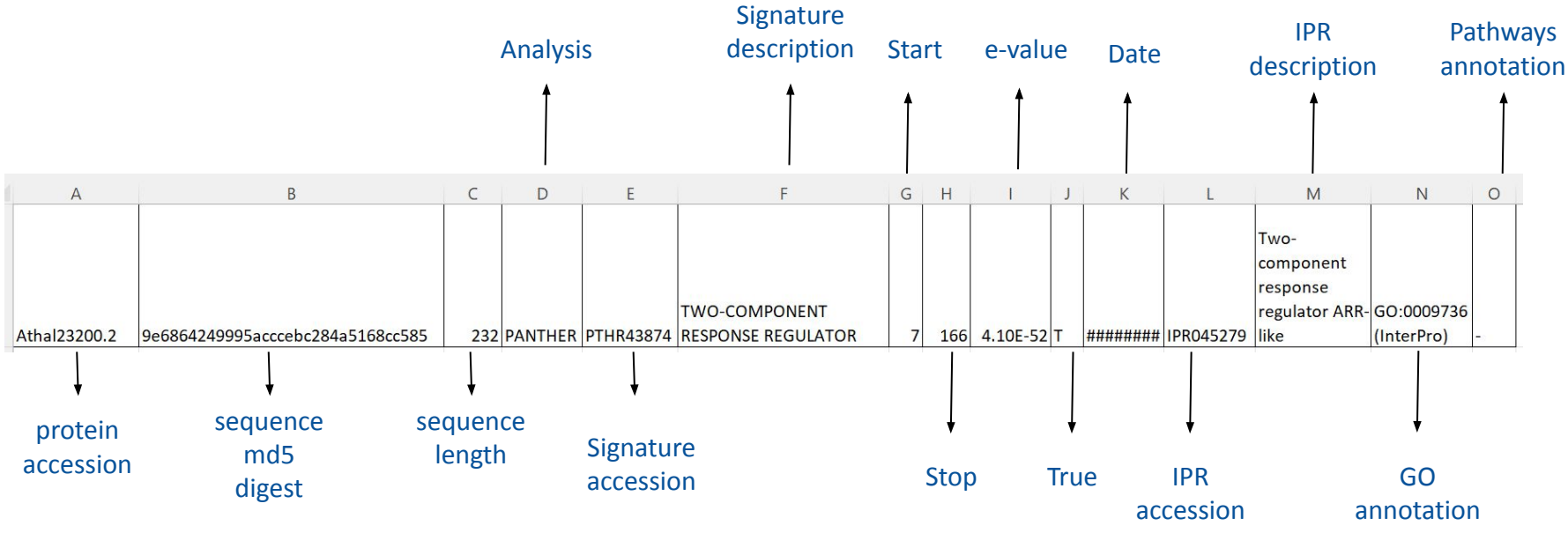
./interproscan.sh -i test_all_appl.fasta -f tsv -dp

./interproscan.sh -i test_all_appl.fasta -f tsv

#model command

./interproscan.sh -i /vol/data/InterProScan5_input/Arabidopsis_thaliana_cleaned.pep.fasta -appl Pfam,PANTHER -b /vol/data/user/Athaliana_interpro -f tsv --goterms --pathways --cpu 12 >> /vol/data/user/interpro.log  2>&1 &

# Interpreting the InterProScan5 result file



# Functional annotation with construct_anno.py

- Python script developed in-house
- Constructs functional annotation file for a given set of peptide sequences
- Uses functional annotation of *Arabidopsis thaliana* and BLAST to transfer the functions across

| *A .thaliana* | Function |
|---|---|
| geneA | functionA |

Orthologs →

| Query species | Function |
|---|---|
| geneA' | ??? |



https://github.com/bpucker/PBBtools/blob/main/collection/README.md#construct_annopy

# BLAST installation

#Step 1: Get the BLAST tar ball

cd /vol/data/tools

wget https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/ncbi-blast-2.17.0+-x64-linux.tar.gz

#Step 2: Decompress the tar ball

tar -xzvf ncbi-blast-2.17.0+-x64-linux.tar.gz

cd ncbi-blast-2.17.0+/bin;ls

# BLAST installation


#Step 3: add mafft to the bash path (OPTIONAL)

nano ~/.bashrc

#Step 4: Add this line at the end of the bashrc file:

export PATH=$PATH:/vol/data/tools/ncbi-blast-2.17.0+/bin

#Step 5: update/ reload the bashrc file

source ~/.bashrc

# Functional annotation with construct_anno.py

#Step 1: Fetch the script from GitHub

wget

https://raw.githubusercontent.com/bpucker/PBBtools/refs/heads/main/collection/construct_anno.py

#Step 2: Read the script usage

python3 construct_anno.py



https://github.com/bpucker/PBBtools/blob/main/collection/README.md#construct_annopy

# Functional annotation with construct_anno.py


#Step 3: Run the script

```
python3 construct_anno.py --in /vol/data/construct_anno_input/Carica_papaya.pep.fasta \

--ref /vol/data/InterProScan5_input/Arabidopsis_thaliana.pep.fasta \

--anno /vol/data/construct_anno_input/Athaliana_functional_annotation.tsv \

--out /vol/data/transfer_annotation --cpu 12 &
```



https://github.com/bpucker/PBBtools/blob/main/collection/README.md#construct_annopy

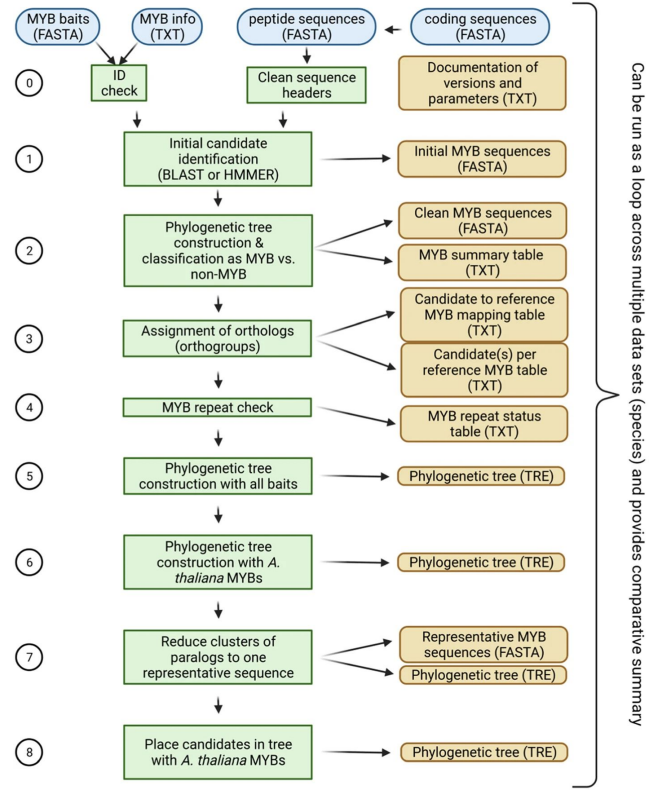
UNIVERSITÄT BONN

# Interested in specific genes or pathway?

UNIVERSITÄT BONN

# MYB_annotator

- Python based pipeline for automatic identification and functional annotation of MYBs
- Basis
  1. Domain and motif identification
  2. Phylogenetic analysis with well-characterized sequences






Pucker, 2022: 10.1186/s12864-022-08452-5

UNIVERSITÄT BONN

# MYB_annotator


## Installation

git clone https://github.com/bpucker/MYB_annotator.git

## Requirements:

dendropy → pip3 install dendropy

mafft → sudo apt install mafft

fasttree → sudo apt install fasttree

hmmer → sudo apt install hmmer

## Print usage:

python3 /vol/data/tools/MYB_annotator/MYB_annotator.py


## Example command:

```
python3 /vol/data/tools/MYB_annotator/MYB_annotator.py \
--baits /vol/data/tools/MYB_annotator/MYB_baits.fasta \
--info /vol/data/tools/MYB_annotator/MYB_baits.txt \
--out /vol/data/MYB_annotator/ \
--subject /vol/data/GeMoMa/filter1/final/proteins.fasta \
 --ath /vol/data/tools/MYB_annotator/AthMYBs.fasta \
--refmybs /vol/data/tools/MYB_annotator/AthRefMYBs.txt
```

# MYB_annotator - Results


## **RESULTS** directory

- 03b_new_2_ref_myb_mapping_file.txt
- *_tree.tre

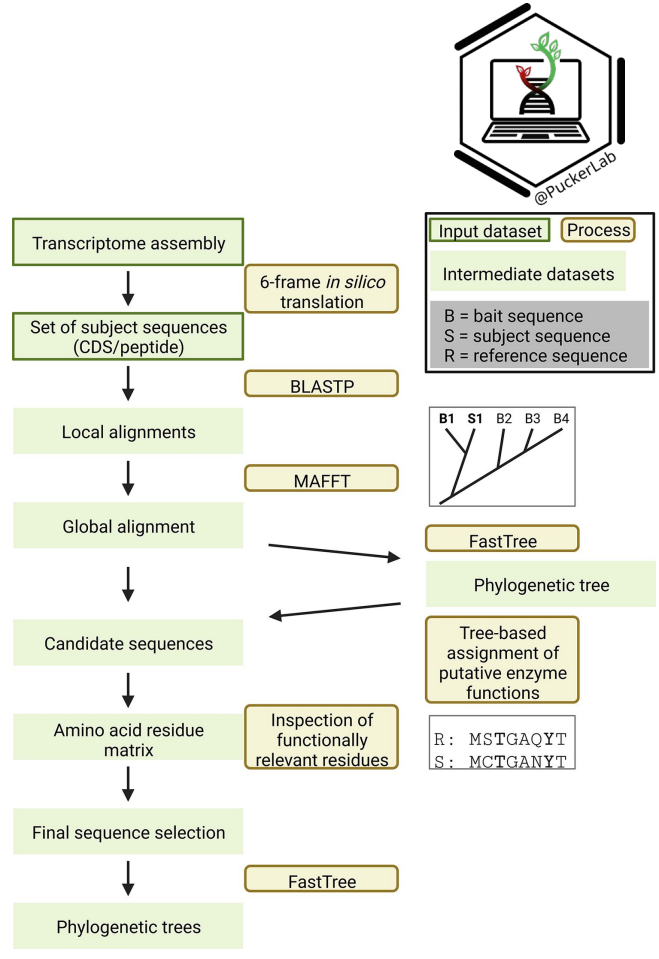
| NewMYB | RefMYB | EdgeDistance | PatristicDistance | Annotation |
|---|---|---|---|---|
| Athal00230.1 | At4g38620-At2R-MYB004 | 2 | 1e-08 | repressor phenylpropanoid, sinapate, lignin |
| Athal00327.1 | At4g37780-At2R-MYB087 | 2 | 1e-08 | axillary meristem, root growth |
| Athal00390.1 | At4g37260-At2R-MYB073 | 2 | 0.0 | stress response, hormone signaling |
| Athal00658.1 | At4g34990-At2R-MYB032 | 2 | 1e-08 | repressor phenylpropanoid, sinapate, lignin |
| Athal00850.1 | At4g33450-At2R-MYB069 | 2 | 0.0 | cell wall, lignin, seed oil, axillary meristem |
| Athal00850.2 | At4g33450-At2R-MYB069 | 2 | 0.0 | cell wall, lignin, seed oil, axillary meristem |
| Athal00928.1 | AT4G32730.1-At3R-MYB1 | 2 | 0.0 | cell cycle control |
| Athal00928.2 | AT4G32730.1-At3R-MYB1 | 2 | 0.0 | cell cycle control |
| Athal01516.1 | At4g28110-At2R-MYB041 | 2 | 1e-08 | defense, stress response |
| Athal01665.1 | At4g26930-At2R-MYB097 | 2 | 1e-08 | anther development, stress response |
| Athal01827.1 | At4g25560-At2R-MYB018 | 2 | 1e-08 | photomorphogenesis |
| Athal02170.1 | At4g22680-At2R-MYB085 | 2 | 1e-08 | SCW, lignin |

UNIVERSITÄT BONN

# Interested in specific pathway?

## Knowledge-based Identification of Pathway Enzymes (KIPEs3)

- Automatic identification and annotation of any biological pathway of interest
- Flavonoid and Carotenoid biosynthesis pathway well established
- Basis: (a) orthology, and (b) conserved domains and residues
- Available as a command line-tool and on webserver: https://pbb-tools.de/

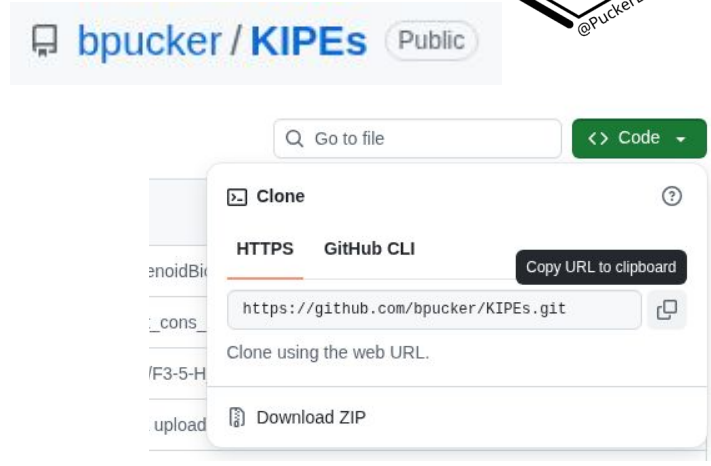





Pucker et al., 2020:10.3390/plants9091103
Rempel, Choudhary & Pucker, 2022: 10.1101/2022.06.30.498365

UNIVERSITÄT BONN KIPEs3


## Installing KIPEs (https://github.com/bpucker/KIPEs)




cd /vol/data/tools

git clone https://github.com/bpucker/KIPEs.git

cd KIPEs

KIPEs3.py carotenoid flavonoid


## cd baits/

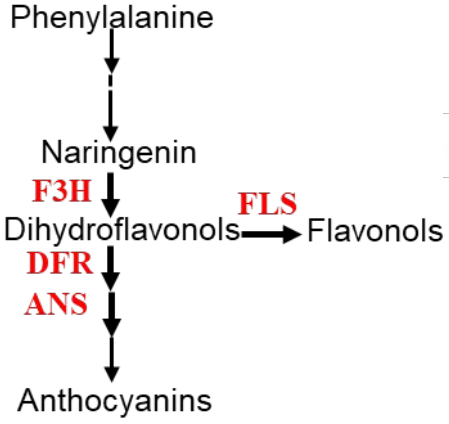
```
(base) ubuntu@pga2024-02ba6:/vol/data/tools/KIPEs/baits$ ls
3AT.fasta   4CL.fasta   A3RT.fasta  ABCC.fasta   ANR.fasta  BZ2.fasta  CHI1.fasta  CHS.fasta  F2H.fasta       F3-5
-H-b.fasta  F3GT.fasta  FLS.fasta   FNS2.fasta  MATE.fasta  PAL.fasta   TT15.fasta  TT9.fasta
3MAT.fasta  5MAT.fasta  A5GT.fasta  AHA10.fasta  ANS.fasta  C4H.fasta  CHI2.fasta  DFR.fasta  F3-5-H-a.fasta  F3-H
.fasta      F3H.fasta   FNS1.fasta  LAR.fasta   OMT.fasta   TT10.fasta  TT19.fasta  UGT72L1.fasta
```

## cd residues/

```
(base) ubuntu@pga2024-02ba6:/vol/data/tools/KIPEs/residues$ ls
3AT.res   A3RT.res   ANR.res   CHI1.res  F2H.res       F3GT.res  FNS2.res  PAL.res   TT9.res
3MAT.res  A5GT.res   ANS.res   CHI2.res  F3-5-H-a.res  F3H.res   LAR.res   TT10.res  UGT72L1.res
4CL.res   ABCC.res   BZ2.res   CHS.res   F3-5-H-b.res  FLS.res   MATE.res  TT15.res
5MAT.res  AHA10.res  C4H.res   DFR.res   F3-H.res      FNS1.res  OMT.res   TT19.res
```

# Basis of KIPEs - (a) Orthology

## Pathway of interest

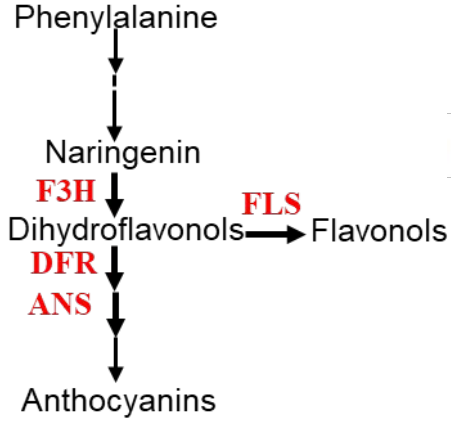



## Collection of baits

| | F3H | FLS | DFR |
|---|---|---|---|
| Bait1 | MWNIILCIVGLV… | …. | …. |
| Bait2 | …. | …. | …. |
| Bait3 | …. | …. | …. |

# Basis of KIPEs - (a) Orthology

## Pathway of interest




## Collection of baits

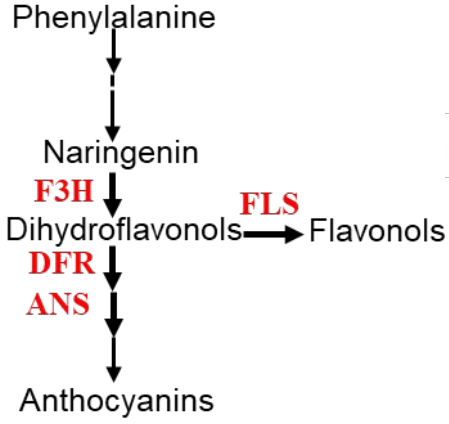
| | F3H | FLS | DFR |
|---|---|---|---|
| Bait1 | MWNIILCIVGLV… | …. | …. |
| Bait2 | …. | …. | …. |
| Bait3 | …. | …. | …. |

## Ortholog sequences

```
Bait 1      1  MWNIILCIVGLVVVGITWWVYRWRNPKCRSVLPPGSMGLPLIG  43
Bait 2
Subject 1   3  MWDIILCILGVVVVGITHWVYRWRNPKCKGVLPPGSMGLPLIG  45   ✓
Subject 2   2  MWNIILCIVGLVVVGITWWVYKLRNPKCRSVLPPGSMGLPLIG  44   ✓
Subject 3   1  MWNIILCIVGLVVVGITHWVYRLSPGECKGVLPPGSMGLPLIG  43   ✓
Subject 4   1  MSSSCPSKVSRLFLSEDSKETISTKCKELIATLPKLDPPHPTY  43   ✗
```

# Basis of KIPEs - (a) Orthology

## Pathway of interest




## Collection of baits

| | F3H | FLS | DFR |
|---|---|---|---|
| Bait1 | MWNIILCIVGLV… | …. | …. |
| Bait2 | …. | …. | …. |
| Bait3 | …. | …. | …. |

## Predicted sequences

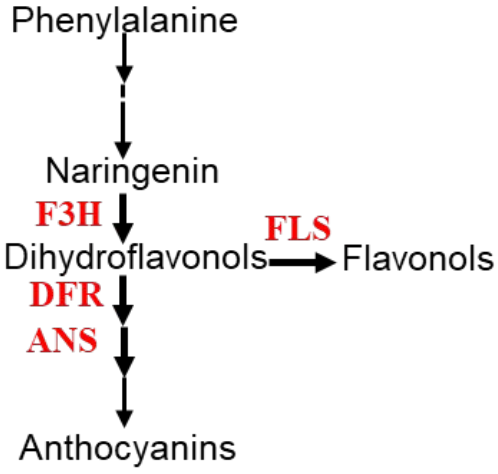
| | F3H | FLS | DFR |
|---|---|---|---|
| Subj1 | MWNIILCIVGLV… | …. | …. |
| Subj2 | MWDIILCILGVV… | …. | …. |
| Subj3 | MWNIILCIVGLV… | …. | …. |

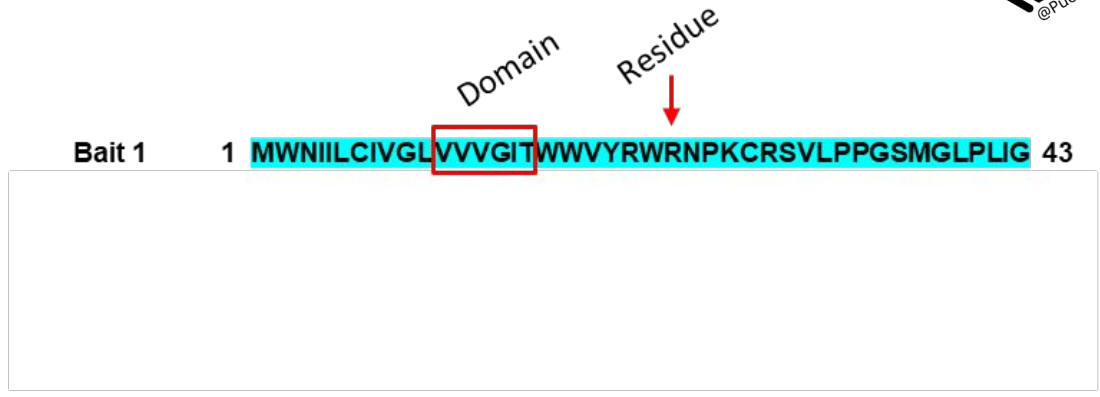
## Ortholog sequences

```
Bait 1      1  MWNIILCIVGLVVVGITWWVYRWRNPKCRSVLPPGSMGLPLIG  43
Bait 2
Subject 1   3  MWDIILCILGVVVVGITHWVYRWRNPKCKGVLPPGSMGLPLIG  45   ✓
Subject 2   2  MWNIILCIVGLVVVGITWWVYKLRNPKCRSVLPPGSMGLPLIG  44   ✓
Subject 3   1  MWNIILCIVGLVVVGITHWVYRLSPGECKGVLPPGSMGLPLIG  43   ✓
Subject 4   1  MSSSCPSKVSRLFLSEDSKETISTKCKELIATLPKLDPPHPTY  43   ✗
```

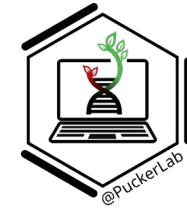
# Basis of KIPEs - (b) Conserved domains and residues

## Pathway of interest

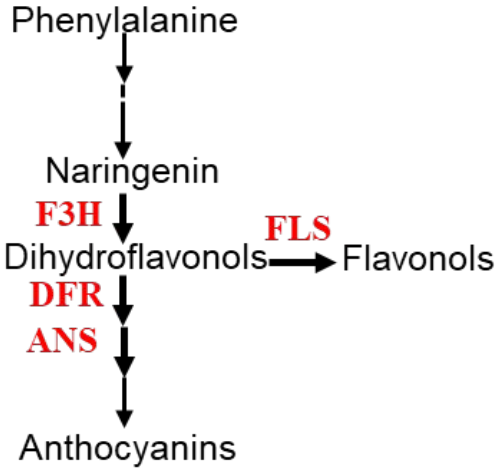


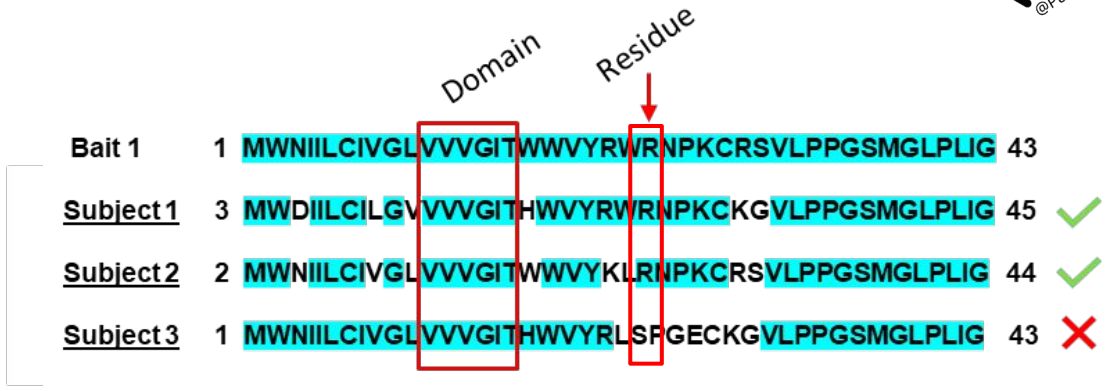



# Basis of KIPEs - (b) Conserved domains and residues




## Pathway of interest




- Confident functional assignment using conserved functional domains and residues

# Running KIPEs


python3 /vol/data/tools/KIPEs/KIPEs3.py

**Requirements:**

dendropy ✅

mafft ✅

ncbi-blast+ ⟶ sudo apt install ncbi-blast+

Subject sequences ⟶ available at /vol/data/dataset/subject/

Example command:

python3 /vol/data/tools/KIPEs/KIPEs3.py --subjectdir /vol/data/dataset/subject/ --baits /vol/data/tools/KIPEs/baits/ --residues /vol/data/tools/KIPEs/residues/ --out /vol/data/user/KIPEs_results/ --cpu 13

UNIVERSITÄT BONN

# KIPEs result Inference

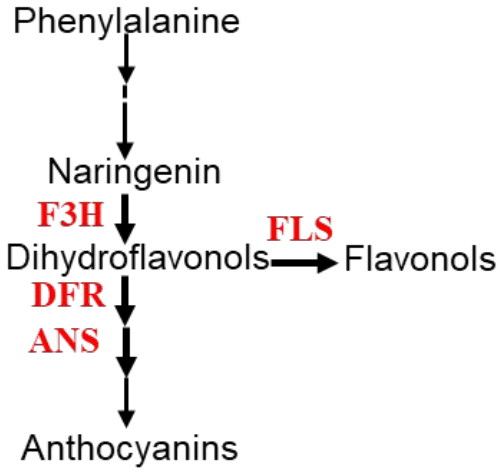

Important files/folders:

- summary.txt
- conserved_residues/
- final_pep_files/

Try :

grep -w “100.0” summary.txt

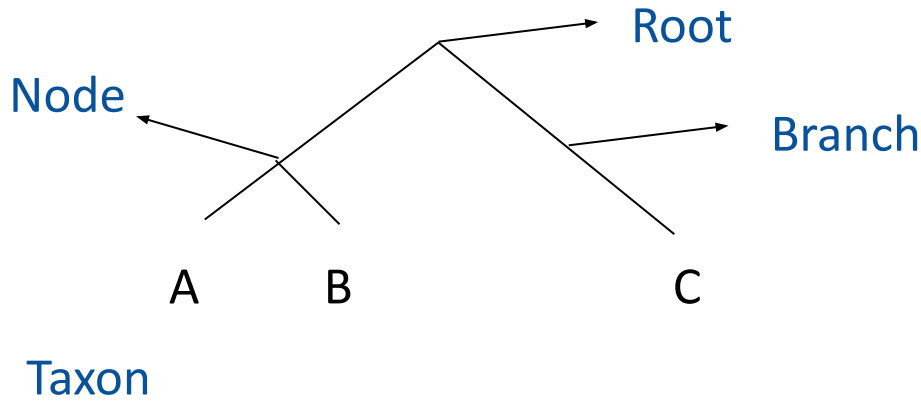
# Flavonoid Pathway




- Plant specialized metabolite pathway producing pigments
- Contains typical specialized metabolite genes like 2-ODDs, CYP450, GTs, SDRs, and so on.
- 2-ODD family members - F3H, FLS, and ANS

# Phylogenetic analysis

- Gene and protein sequence relationships with their ancestral sequences
- An estimate or inference - not absolute fact!



# Global alignment


DEMO TIME!

- Alignment of most similar regions between sequences - Local
- End to end alignment of the sequences - Global

# Installing MAFFT

#Step 1 fetch the link from the web

wget https://mafft.cbrc.jp/alignment/software/mafft-7.526-linux.tgz

#Step 2 decompress

tar xfvz mafft-7.526-linux.tgz

#Step 3 test

cd mafft-linux64

execute mafft using ./mafft.bat

UNIVERSITÄT BONN

# Installing MAFFT


#Step 4: add mafft to the bash path (OPTIONAL)

nano ~/.bashrc

#Step 5: Add this line at the end of the bashrc file:

export PATH=$PATH:path/to/mafft/executable

#Step 6: update/ reload the bashrc file

source ~/.bashrc

# Selecting the MAFFT parameters

cd /vol/data/KIPEs_results_for_tree/

grep “>” -c Final_combined.pep.fasta

https://mafft.cbrc.jp/alignment/software/tips0.html

L-INS-i, E-INS-i, G-INS-i

Optimal mode for our use case?

# Alignment with MAFFT

#G-INS-i

```
mafft --globalpair --maxiterate 1000 --amino /vol/data/KIPEs_results_for_tree/Final_combined.pep.fasta
> /vol/data/user/Athaliana_aligned.aln
```

# Cleaning the alignment

# Step 1: fetch the python script from GitHub

wget https://raw.githubusercontent.com/bpucker/ApiaceaeFNS1/refs/heads/main/algntrim.py

#Step 2: Understand script usage

python3 algntrim.py

#Step 3: Run the script

python3 algntrim.py --in /vol/data/user/Athaliana_aligned.aln --out /vol/data/user/Athaliana_aligned_trimmed.aln

UNIVERSITÄT BONN

# Need to clean the alignment


- Improve the signal from the sequences for phylogeny
- Decrease noise from the alignment
- Reduces computational costs for inferring a phylogenetic tree
- Occupancy - percentage of sequences in an aligned column that don’t have gaps
- We remove columns with occupancy < 10% (0.1) by default in the script (can be changed)

# Installing IQ-TREE3


sudo apt-get install iqtree

UNIVERSITÄT BONN

# Installing IQ-TREE3


#Step 1: fetch the IQ-TREE3 source folder

cd /vol/data/tools

wget https://github.com/iqtree/iqtree3/releases/download/v3.0.1/iqtree-3.0.1-Linux.tar.gz

#Step 2: Decompress the tarball

tar -xvzf iqtree-3.0.1-Linux.tar.gz

cd iqtree-3.0.1-Linux/bin;ls

#Step 3: Open the bashrc file

nano ~/.bashrc

UNIVERSITÄT BONN

# Installing IQ-TREE3


#Step 4: Add this line at the end of the bashrc file:

```
export PATH=$PATH:/vol/data/tools/iqtree-3.0.1-Linux/bin
```

#Step 5: reload the bashrc file

```
source ~/.bashrc
```

UNIVERSITÄT BONN

# Finally tree building!


#Step 1: Look at the IQ-TREE3 tutorial pages

https://iqtree.github.io/doc/Tutorial

#Step 2: Run IQ-TREE3

cd /vol/data/user

mkdir IQ-TREE3_results

iqtree3 -s /vol/data/user/Athaliana_aligned_trimmed.aln -m MFP -wsr --alrt 1000 -B 1000 -pre /vol/data/user/IQ-TREE3_results/2ODD -T 5

UNIVERSITÄT BONN

# Understanding the IQ-TREE3 command


- m - MFP -> ModelFinder Plus (https://iqtree.github.io/doc/Substitution-Models)
- wsr -> write site rates
- alrt -> number of bootstrap replicates for SH-alrt test
- Shimodaira-Hasegawa approximate Likelihood Ratio Test -> how confident should you be about each branch in your tree
- B -> Ultrafast bootstrap
- pre -> full path to output folder with the name of your treefile

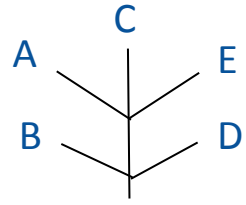
UNIVERSITÄT BONN

# Understanding bootstrapping


- Build an initial tree with the alignment - Original tree
- Randomly resample the alignment with replacement -> pseudo - alignment
- Obtain a tree from this pseudo - alignment
- Repeat for 1000 times (-B 1000)
- Compare original tree with the pseudo-aligned trees
- For each branch count the number of times they are exactly grouped in all the bootstrapped trees
- Get bootstrap support values for each branch

# Understanding the treefile format

Newick standard form - represents trees in computer readable form




Newick FORMAT ->(B,(A,C,E),D);

(B:6.0,(A:5.0,C:3.0,E:4.0):5.0,D:11.0);

# Trivia on trees!


Society for Study of Evolution, New hampshire

Finalized the treefile format at Newick's

over a plate of lobsters!

UNIVERSITÄT BONN

# End of day 2


## Today you learned:

- Annotating tRNAs
- Functional annotation with InterProScan5, construct_anno.py
- Specific functional analyses with KIPEs, MYB_annotator
- Multiple sequence alignment with MAFFT
- Phylogenetic tree building with IQ-TREE3

UNIVERSITÄT BONN

# Start of day 3


**Yesterday you learnt:**

- Functional annotation with InterProScan5, construct_anno.py
- tRNA annotation with tRNAScanSE
- Specific gene family/ pathway enzyme analyses with MYB_annotator, KIPEs
- MAFFT alignment, trimming
- Tree building with IQ-TREE3

**Today you will learn:**

- File transfer between local and remote hosts
- Tree visualization with iTOL
- Gene duplication analysis with DupyliCate

# Secure copy (SCP)


# Syntax

scp [options] user@IPxxx:source/file/folder/path destination/file/folder/path

#Command to transfer tree file from VM to local machine

#Execute from cmd prompt in local machine

scp -P 30xxx -i path/to/private/key/file -r

ubuntu@134.176.27.78:/vol/data/user/IQ-TREE3_results/2ODD.treefile

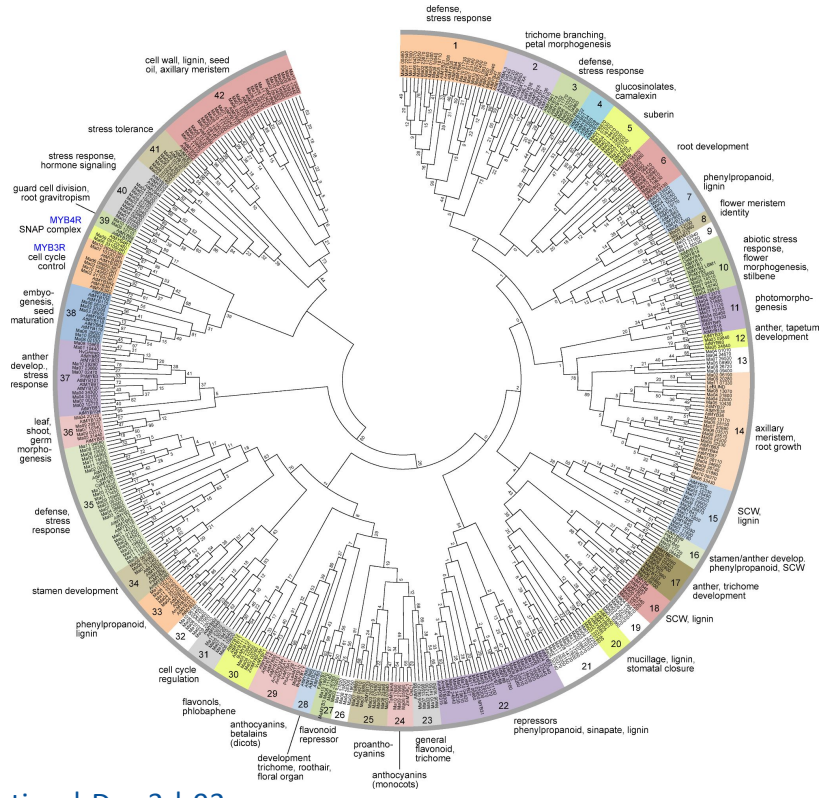
path/to/destination/folder/in/local/machine

UNIVERSITÄT BONN

# Tree visualization and Interpretation

@PuckerLab

- Understanding the function of sequences based on trees
- Generation of nice illustrations for publications
- Exploration of large datasets



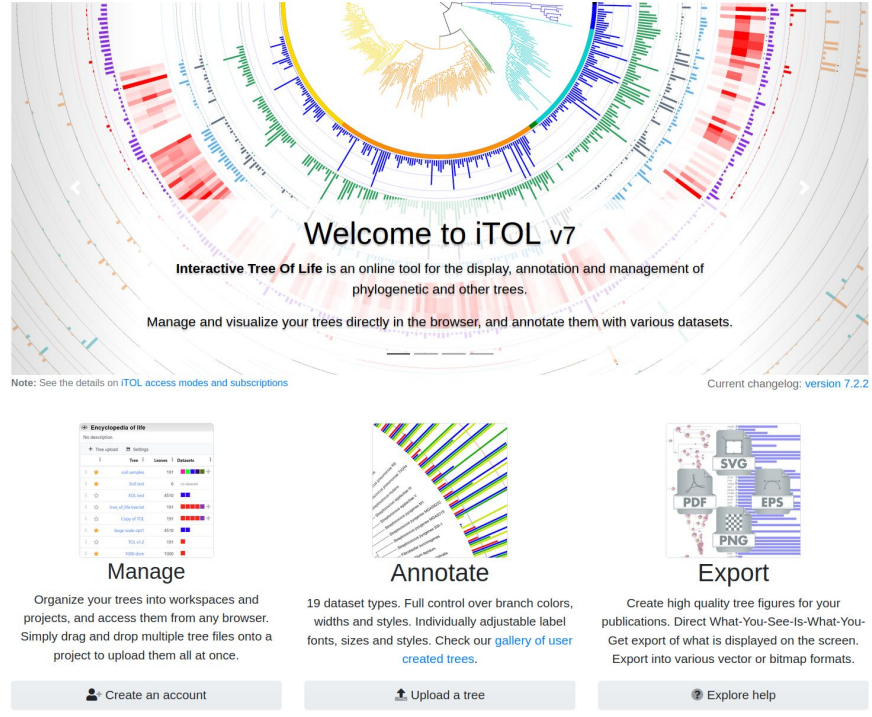



Pucker, 2020: 10.1371/journal.pone.0239275

UNIVERSITÄT BONN iTOL

@PuckerLab

- iTOL is operated by EMBL
- Service is freely available
- Subscription is required to save modified trees (60€/year)
- Export in various file formats possible (SVG, PDF) - conversion possible

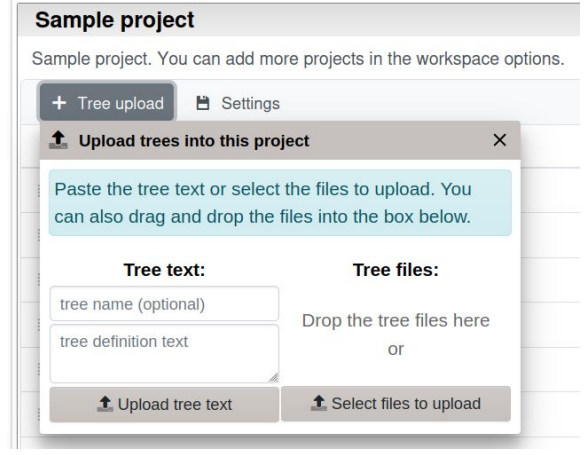





https://itol.embl.de/

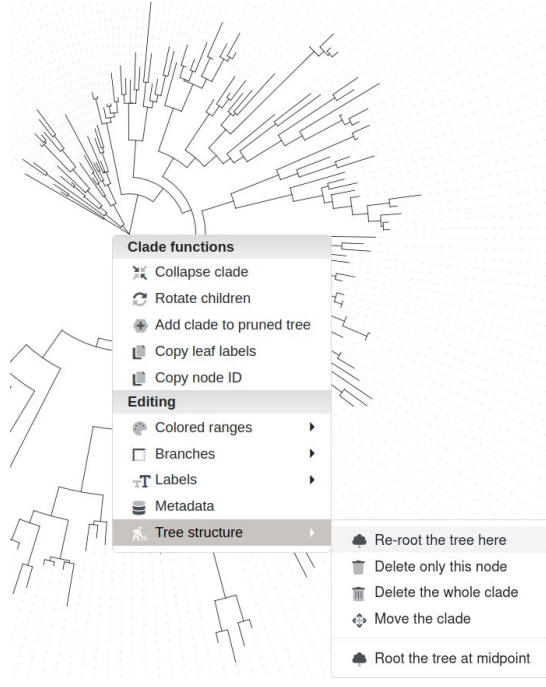
# Importing a tree

- Import of tree file per drag&drop possible
- Tree name is inferred from file
- Tree content can be pasted into form
- Tree name can be set manually



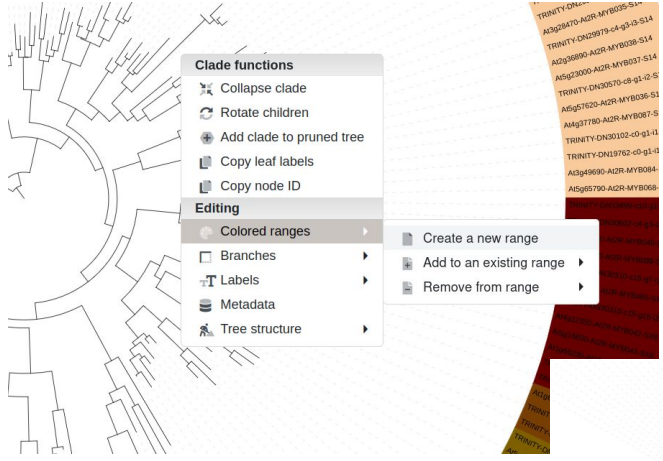
# Rerooting your tree

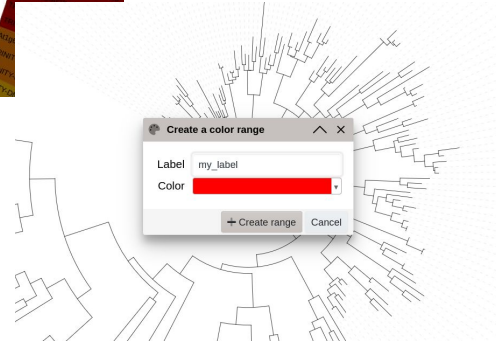
- Trees can be rooted based on an outgroup
- Left click on clade to open menu
- Go to “Tree structure”
- Select “Re-root the tree here”



# Coloring your tree in iTOL

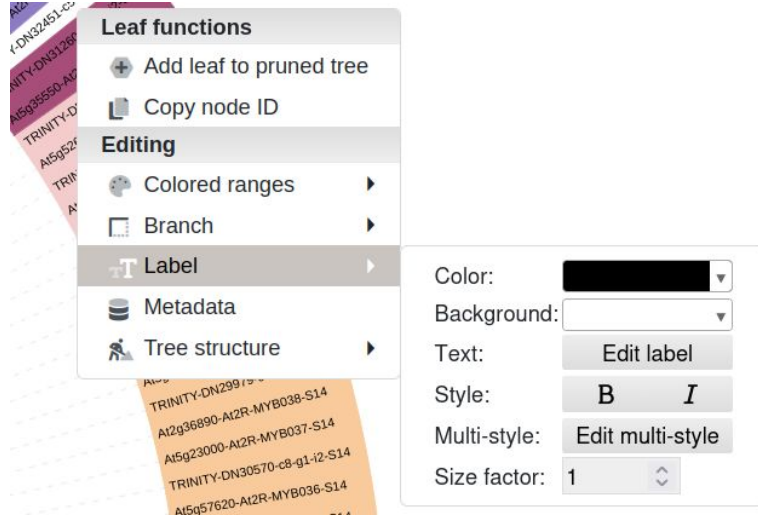
- You can highlight monophyletic clades by color
- Left click on a branch to open “Colored ranges”
- Create a new range by setting label and color



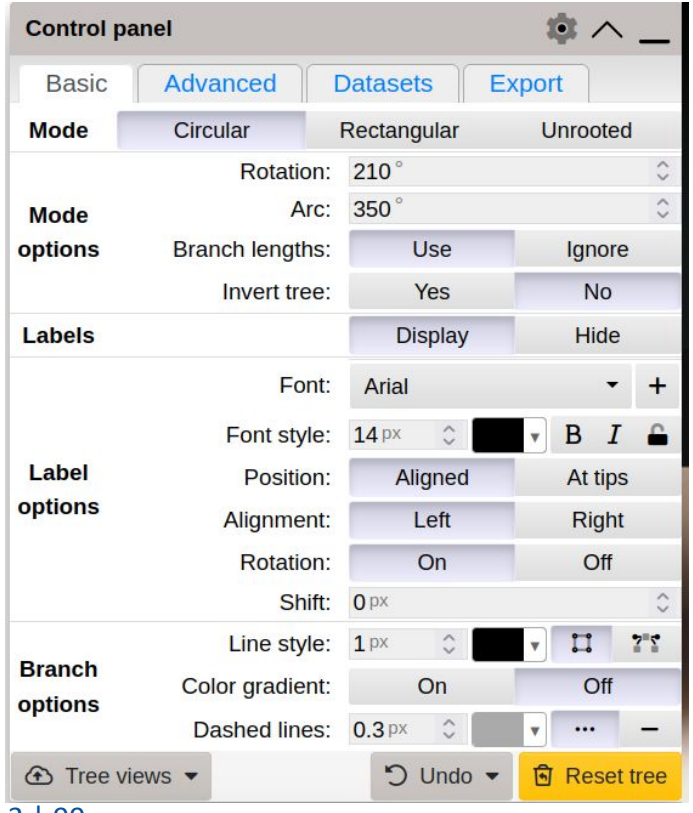
# Renaming sequences

- Individual sequences can be renamed
- Font, size, style of label can be changed
- Only good option for large number of labels (better rename in tree file)



# Basic functions

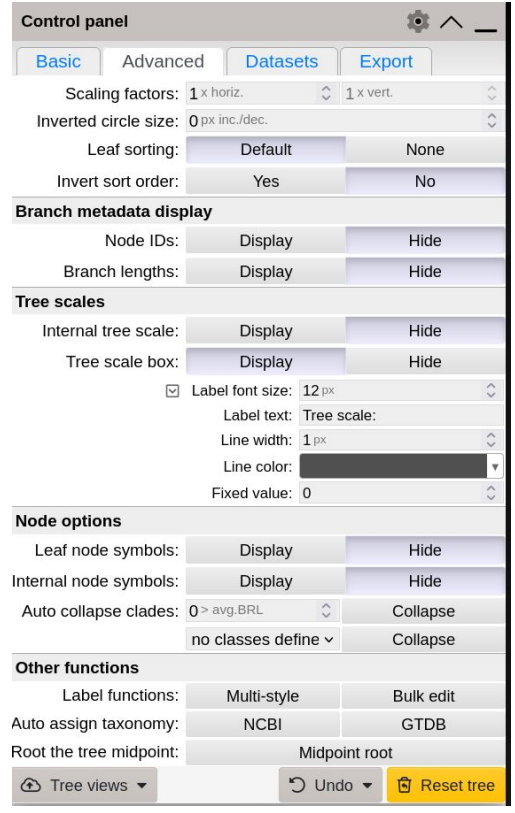
- Mode: circular, rectangular
- Branch lengths: use/ignore
- Labels: display/hide
- Font style: px
- Position: aligned/at tips
- Line style: px



UNIVERSITÄT BONN

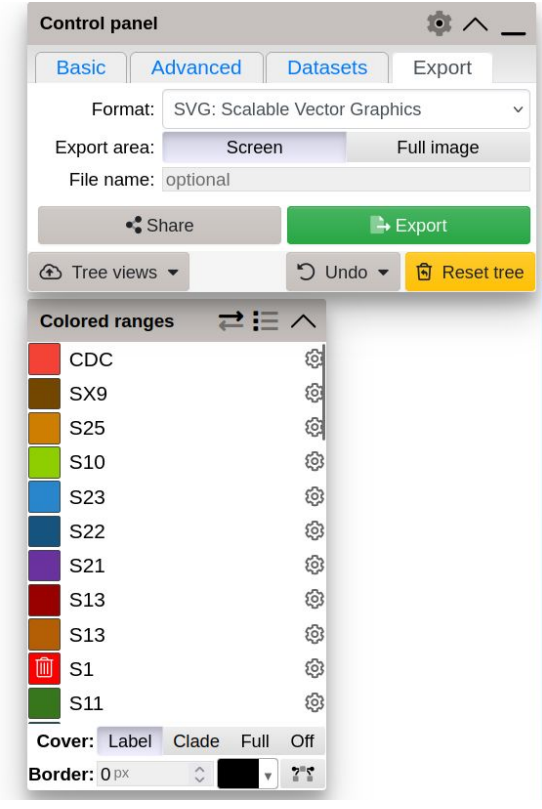
# Advanced functions

- Invert sort order: yes/no
- Node IDs: display/hide
- (Branch lengths: display/hide)
- Tree scale box: display/hide



# Export

- Export as SVG, PDF, and PNG possible
- SVG/PDF can be converted on Ubuntu into any other format
- Convert function can generate almost every file type

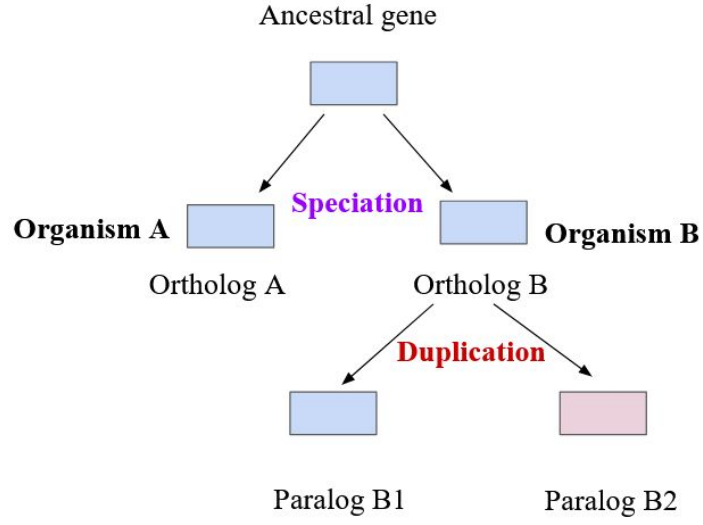


UNIVERSITÄT BONN

# iTOL: exercises

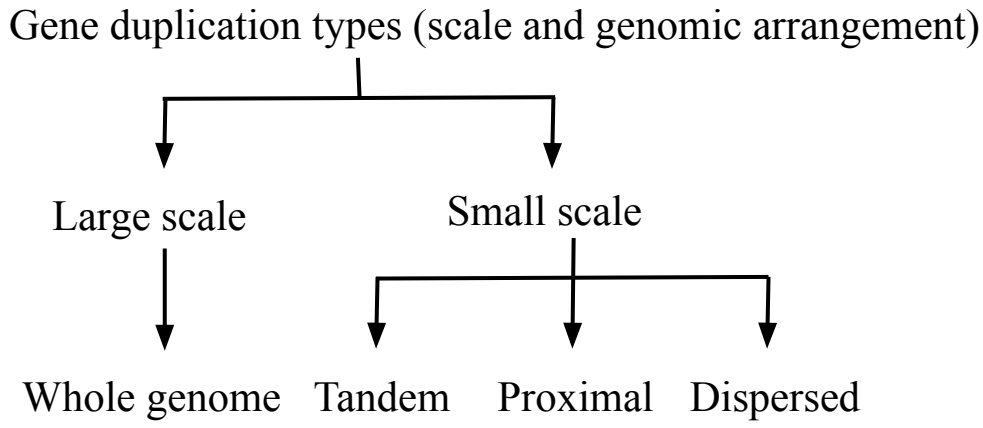

1. Import your tree
2. Root tree
3. Tree circular and rectangle mode
4. Color clades
5. Change some labels
6. Export SVG and convert into JPG

# Comparative genomics

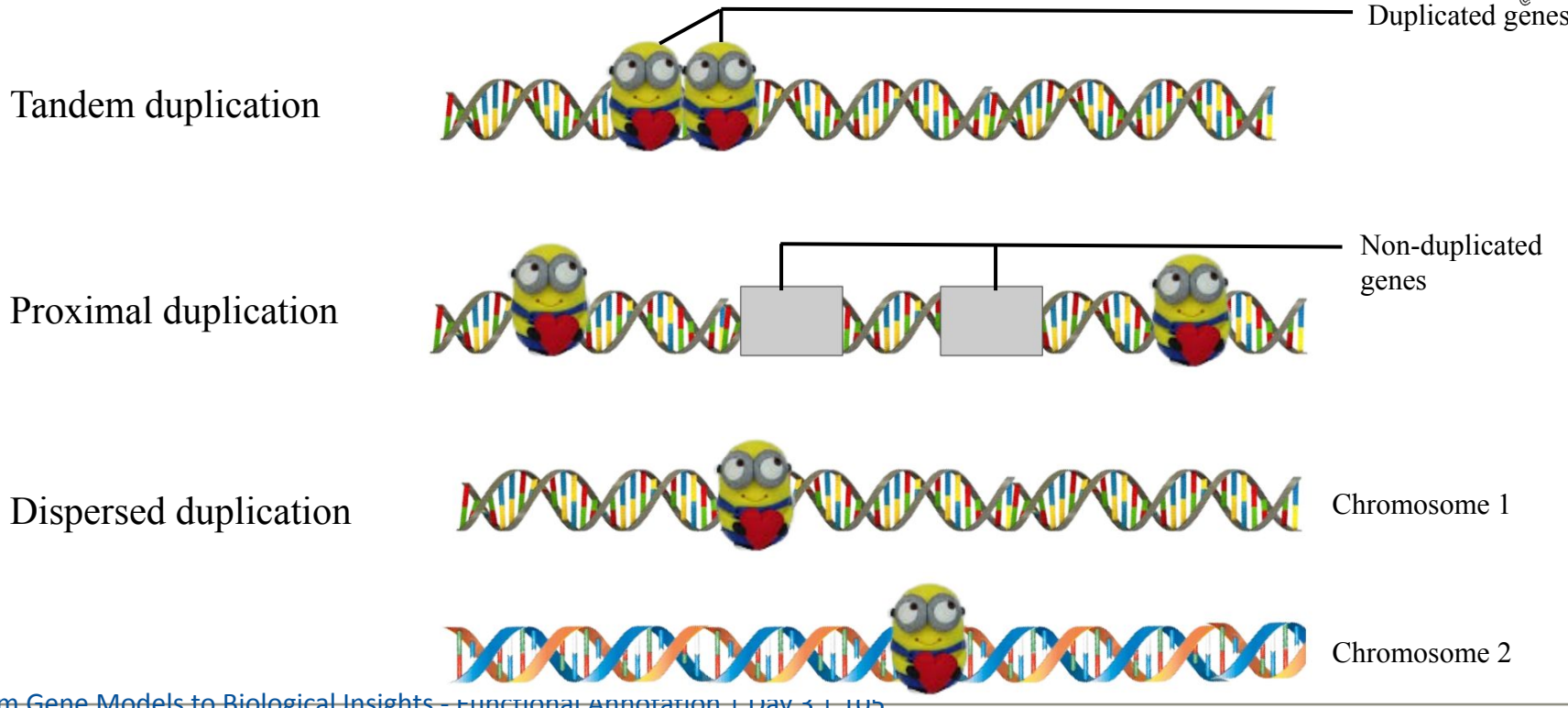



- Gene duplicates can have different fates
- Challenge: Cross species transfer of function

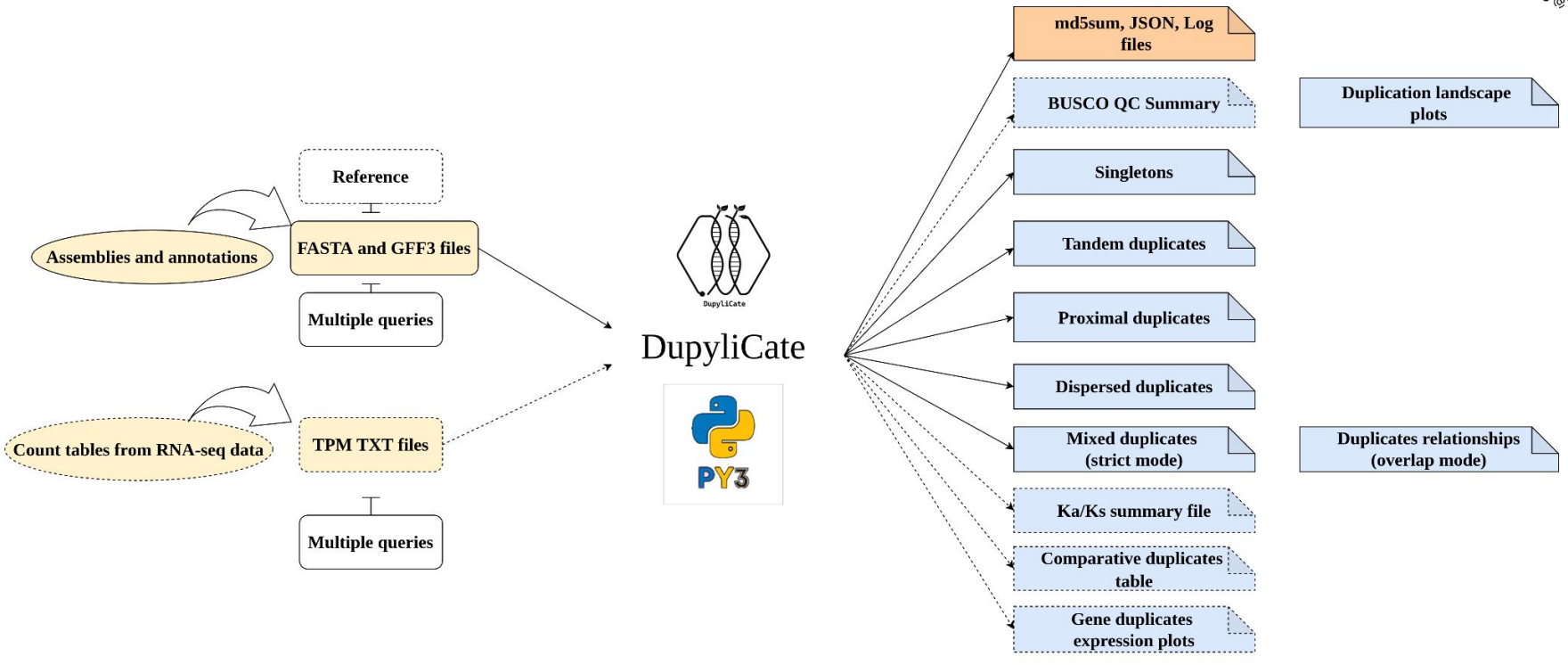
# Gene duplication categories



# Gene duplication categories



# Introducing DupyliCate






Natarajan & Pucker, 2025: 10.1101/2025.10.10.681656

UNIVERSITÄT BONN

# Installing DupyliCate


#Step 1: Check docker installation

docker run hello-world

#Step 2: Pull docker image from docker hub

docker pull shakunthalan/dupylicate:latest

#Step 3: Check the pulled image

docker run --rm -u $(id -u) -v /vol/data:/vol/data shakunthalan/dupylicate:latest

#Step 4: Visit GitHub repo

https://github.com/ShakNat/DupyliCate

UNIVERSITÄT BONN

# Running DupyliCate


## #Run 1 with BUSCO QC and auto scoring

```
docker run --rm -u $(id -u) -v /vol/data:/vol/data shakunthalan/dupylicate:latest \
--pep /vol/data/DupyliCate_input/Athaliana.pep.fasta \
--gff /vol/data/DupyliCate_input/Athaliana.gff3 \
--cores 12 --score auto --mode overlap --clean_up no \
--busco_lineage /vol/data/DupyliCate_input/busco_lineage.txt \
--out /vol/data/user/Dupylicate_Athaliana_auto &
```

# Running DupyliCate


## #Run 2 with BUSCO QC and manual scoring

```
docker run --rm -u $(id -u) -v /vol/data:/vol/data shakunthalan/dupylicate:latest \
--pep /vol/data/DupyliCate_input/Athaliana.pep.fasta \
--gff /vol/data/DupyliCate_input/Athaliana.gff3 \
--cores 12 --score 0.3 --self_simcut 40 --qc yes --mode overlap --clean_up no \
--busco_lineage /vol/data/DupyliCate_input/busco_lineage.txt \
--out /vol/data/user/Dupylicate_Athaliana &
```

UNIVERSITÄT BONN

# Exploring the result files


```
cd /vol/data/user/Dupylicate_Athaliana_auto

cd BUSCO_QC;column -t -s $'\t' BUSCO_QC.tsv | less -S

cd ..

column -t -s $'\t' Summary.tsv | less -S

cd Tandem_duplicates

column -t -s $'\t' Athaliana_tandems.tsv | less -S

cd /vol/data/user/Dupylicate_Athaliana

column -t -s $'\t' Summary.tsv | less -S
```

UNIVERSITÄT BONN

# Exploring the plots


#Execute from cmd prompt in local machine

#copy the BUSCO gene frequency distribution plot

scp -P 30xxx -i path/to/private/key/file -r

ubuntu@134.176.27.78:/vol/data/user/Dupylicate_Athaliana_auto/BUSCO_QC/Athaliana_gene_frequency_distribution.png path/to/destination/folder/in/local/machine

# Exploring the plots


#copy the Duplication frequency plot folder from run 1

scp -P 30xxx -i path/to/private/key/file -r

ubuntu@134.176.27.78:/vol/data/user/Dupylicate_Athaliana_auto/Duplication_frequency_plots

path/to/destination/folder/in/local/machine/Duplication_frequency_plots_auto

#copy the Duplication frequency plot folder from run 2

scp -P 30xxx -i path/to/private/key/file -r

ubuntu@134.176.27.78:/vol/data/user/Dupylicate_Athaliana_auto/Duplication_frequency_plots

path/to/destination/folder/in/local/machine/

UNIVERSITÄT BONN

# Exploring the plots


#copy the Duplication landscape plot

scp -P 30xxx -i path/to/private/key/file -r

ubuntu@134.176.27.78:/vol/data/user/Dupylicate_Athaliana_auto/Duplication_landscape_plots/Athali

ana.png path/to/destination/folder/in/local/machine

Can you guess the relationship between the plot skew and gene duplications?!

More features and use cases -> https://github.com/ShakNat/DupyliCate

UNIVERSITÄT BONN

# Sample self alignment file


| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AT1G01010 | AT1G01010 | 100.000 | 430 | 0 | 0 | 1 | 430 | 1 | 430 | 0.0 | 895 |
| AT1G01010 | AT4G01550 | 32.623 | 469 | 248 | 15 | 1 | 428 | 2 | 443 | 1.46e-56 | 192 |
| AT1G01010 | AT1G02230 | 33.929 | 336 | 158 | 10 | 1 | 330 | 1 | 278 | 6.82e-42 | 151 |
| AT1G01010 | AT1G02250 | 32.092 | 349 | 158 | 14 | 1 | 331 | 1 | 288 | 1.60e-39 | 145 |
| AT1G01010 | AT4G01520 | 39.326 | 178 | 95 | 5 | 1 | 175 | 2 | 169 | 3.16e-33 | 126 |
| AT1G01010 | AT4G01540 | 39.888 | 178 | 94 | 6 | 1 | 175 | 2 | 169 | 1.12e-31 | 125 |
| AT1G01010 | AT3G04420 | 40.854 | 164 | 87 | 2 | 1 | 162 | 1 | 156 | 1.56e-30 | 120 |
| AT1G01010 | AT4G35580 | 40.741 | 162 | 79 | 5 | 5 | 159 | 11 | 162 | 1.06e-28 | 117 |
| AT1G01010 | AT1G33060 | 39.024 | 164 | 81 | 5 | 5 | 160 | 26 | 178 | 3.01e-27 | 114 |
| AT1G01010 | AT3G49530 | 42.308 | 156 | 74 | 6 | 5 | 153 | 15 | 161 | 3.93e-27 | 112 |
| AT1G01010 | AT5G24590 | 40.994 | 161 | 79 | 5 | 5 | 158 | 15 | 166 | 2.19e-26 | 110 |
| AT1G01010 | AT1G02220 | 28.531 | 354 | 202 | 9 | 1 | 344 | 1 | 313 | 1.20e-25 | 107 |
| AT1G01010 | AT3G10490 | 38.182 | 165 | 76 | 6 | 6 | 158 | 30 | 180 | 6.88e-25 | 105 |
| AT1G01010 | AT5G39610 | 34.197 | 193 | 97 | 9 | 6 | 182 | 23 | 201 | 1.61e-23 | 99.4 |
| AT1G01010 | AT3G17730 | 37.821 | 156 | 78 | 5 | 6 | 153 | 9 | 153 | 5.30e-23 | 97.1 |
| AT1G01010 | AT2G17040 | 39.634 | 164 | 88 | 6 | 6 | 167 | 9 | 163 | 7.15e-23 | 97.4 |
| AT1G01010 | AT2G18060 | 37.013 | 154 | 79 | 7 | 6 | 152 | 12 | 154 | 7.27e-23 | 99.0 |

UNIVERSITÄT BONN

# End of day 3


## What you learnt today

- scp transfer between local and remote host
- iTOL tree visualization
- Gene duplication analysis with DupyliCate

# Plant Genome Sequence Assembly and Annotation

## Reads

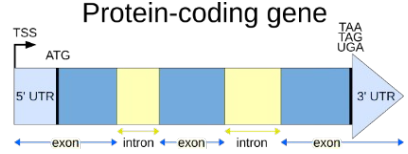
*choosing the right assembler*

### Read quality
Read N50
Coverage depth

## Assembly

### Assembly evaluation
Completeness (BUSCO)
Correctness (Merqury)
Continuity (contig N50, LAI)

## Annotation




*ab initio*
*homology-based*
*evidence-based*

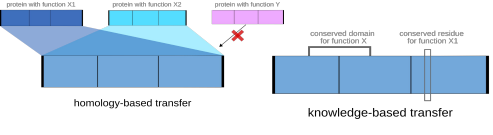
### Annotation evaluation
Number of genes
BUSCO score

# From Gene Models to Biological Insights - Functional Annotation

| Functional annotation | Non protein-coding genes | Phylogenetic analysis | Gene duplications |
|---|---|---|---|

InterProScan5
construct_anno.py

Specialized tools: KIPEs3, MYB_annotator

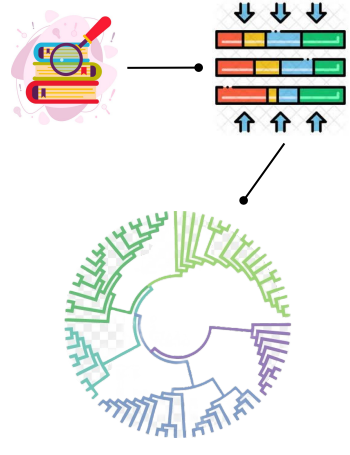



Transposable element

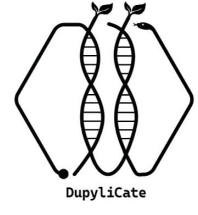

EDTA

ncRNA

Infernal
tRNAscan-SE 2.0



mine, classify and characterize gene duplicates using **DupyliCate**



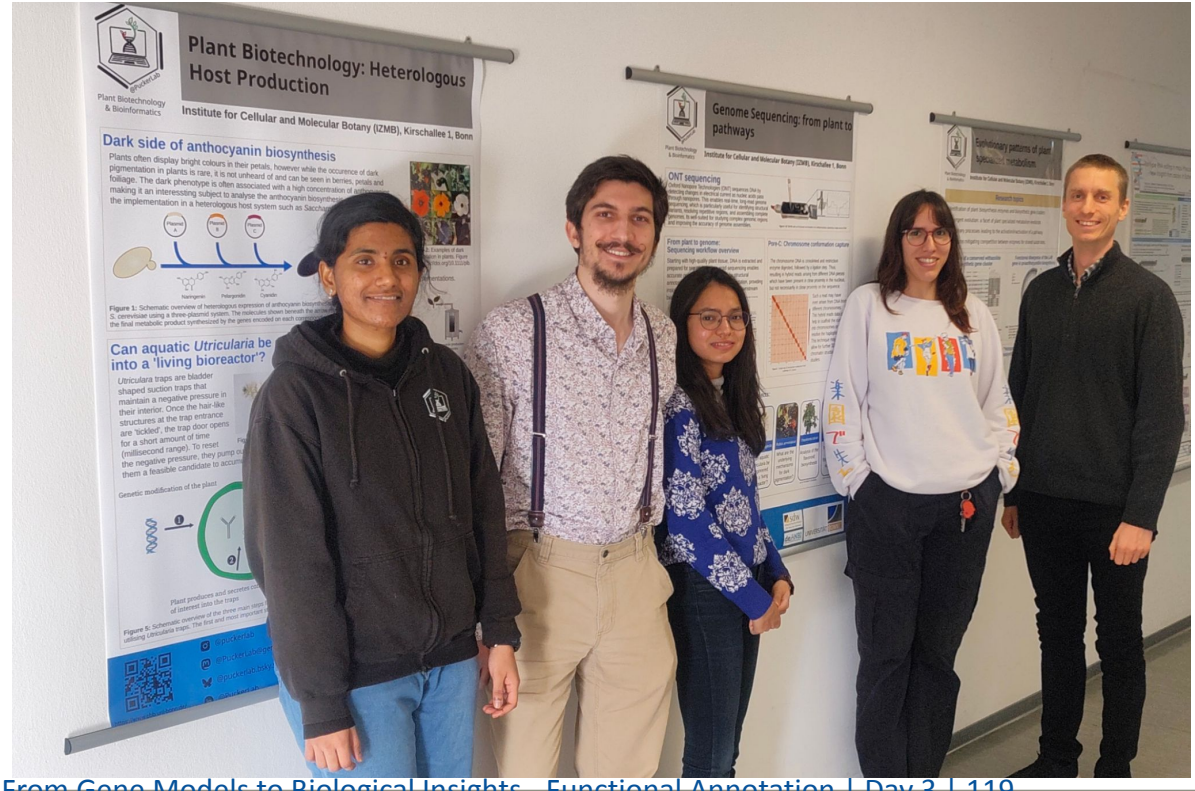
UNIVERSITÄT BONN

# Availability of materials


Please find all course materials here:
https://github.com/bpucker/teaching/tree/master/deNBI2025

UNIVERSITÄT BONN

# Acknowledgements




@PuckerLab

@PuckerLab

Email: pucker@uni-bonn.de
Web: pbb.uni-bonn.de

de.NBI
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE