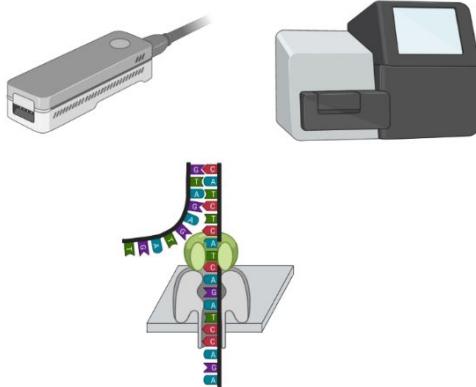
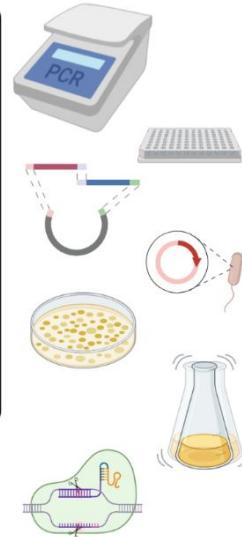
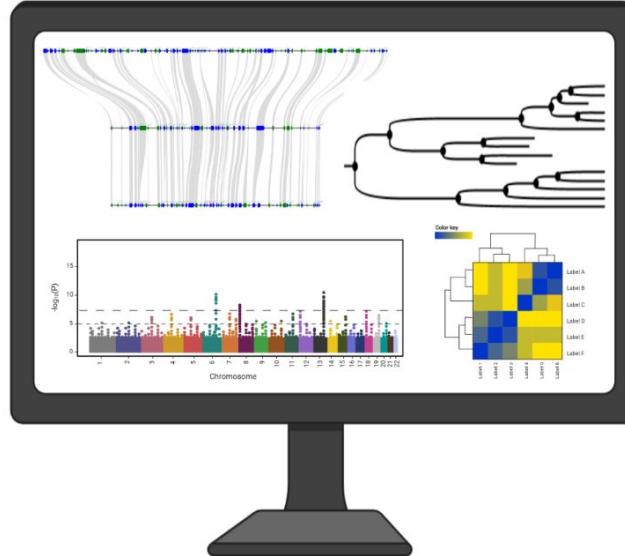




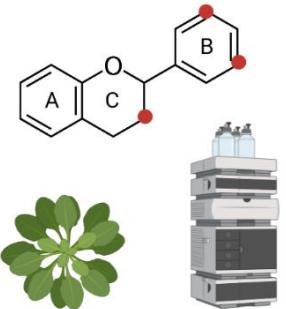
Technische  
Universität  
Braunschweig



Plant Biotechnology  
and Bioinformatics



species biosynthesis proteins analysis different conditions  
biosynthesis DODA bellman variants H23-MYB  
within genes site data functionally Col loco variant  
dissolve sequence KEGG multiple divergent variants non-canonical  
single reference structures amino acid annotation level identified  
synthesis synthesis pathway evolutionary  
sites annotations protein accessions pathway inference  
plants pigment Koyentra genes systems biology long Canophylales  
key against canonical for conserved Arabidopsis  
flavonoid conservation sequencing evolution  
accessions identification sequence MYB introns residues RNA-Seq



# Introduction - Big Data Analytics in Life Sciences

Prof. Dr. Boas Pucker (Plant Biotechnology and Bioinformatics)

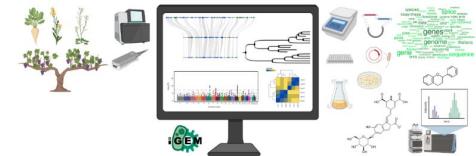
# Round of introduction

- Name?
- Affiliation?
- Research interests?
- Previous experiences with bioinformatics?
- Expectations?

# Boas Pucker

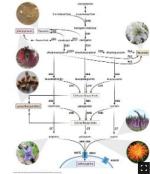
- Biochemistry at HHU Düsseldorf
- (Systems) Biology at Bielefeld University
- Doctoral student (CeBiTec, Bielefeld University)
  - Genomics & Bioinformatics; synthetic biology (iGEM)
- Post doc (Ruhr-University Bochum)
- Post doc (Department of Plant Sciences, Cambridge, UK)
- Plant Biotechnology & Bioinformatics, TU Braunschweig (since 2021)
  - Specialized plant metabolites, applied bioinformatics

Plant Biotechnology and Bioinformatics (Prof. Pucker)

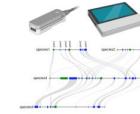


## Specialized Metabolism

Plants produce a plethora of specialized metabolites. These allow plants not only to cope with numerous environmental challenges, but they can also have beneficial effects on humans. Many plant species have been successfully used in traditional medicine. Investigations of specialized plant metabolism can reveal the biosynthesis pathways and enable heterologous production of drug candidates. Our fundamental research on biosynthetic pathways paves the way for translation into industrial applications.



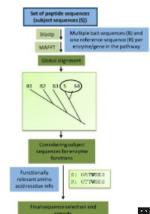
## Plant Genomics



Plant genomes harbour genes, which are the blue prints for enzymes involved in various biosynthetic pathways. Knowledge about plant genomes unravels the mysteries of the specialized biosynthetic pathways in plants. Rapid developments in long read sequencing technologies enable us to study even large and complex plant genomes.

## Bioinformatics

Specific biological questions can be answered through sophisticated tools. Automatic analysis of large data sets and the integration of genomic, transcriptomic, and metabolic data are often required. Applications include the discovery of biosynthetic pathways and their regulators, the detection of biosynthetic gene clusters, and the identification of tolerance/resistance mechanisms. Bioinformatic tools are an effective way to generate hypotheses and to guide molecular biology experiments.



<https://www.tu-braunschweig.de/en/ifp/pbb>

# Availability of slides

- All materials are freely available (CC BY) - after the lectures:
  - StudIP: Big Data Analytics for Life Scientists (SoSe 2024)
  - GitHub: <https://github.com/bpucker/teaching>
- Questions: Feel free to ask at any time
- Feedback, comments, or questions: b.pucker[a]tu-bs.de



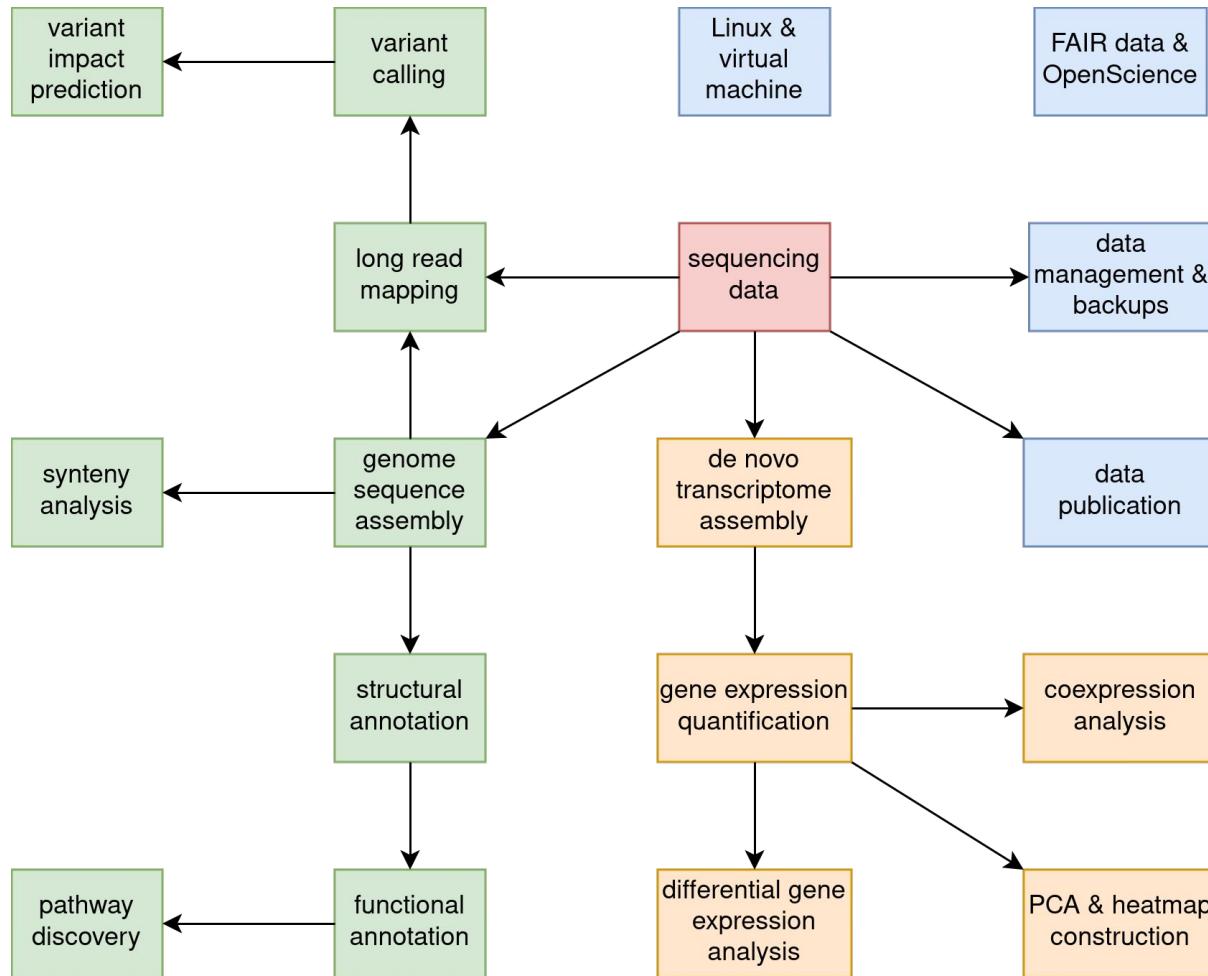
My figures and content can be re-used in accordance with CC BY 4.0, but this might not apply to all images/logos. Some figure were constructed using bioRender.com.

# DOI - finding the literature

- Suggestions for detailed literature will be included on slides
- DOI = Digital Object Identifier
- Unique and short way to point to a publication
- How to resolve a DOI? <https://dx.doi.org/>

A screenshot of a bioRxiv preprint page. At the top is the bioRxiv logo and the text "THE PREPRINT SERVER FOR BIOLOGY". Below is a yellow box containing the text "bioRxiv posts many COVID19-related papers. A reminder: they have not been formally peer-reviewed and should not guide health-related behavior or be reported in the press as conclusive." A "New Results" section shows a preprint titled "Apiaceae *FNS I* originated from *F3H* through tandem gene duplication" by Boas Pucker and Massimo Iorizzo. The DOI is listed as doi: https://doi.org/10.1101/2022.02.16.480750. Below the title are download links for PDF, HTML, and other formats, along with metrics like 0 citations and 46 reads. A "Abstract" section provides a brief summary of the research.

# Big Data Analytics in Life Sciences

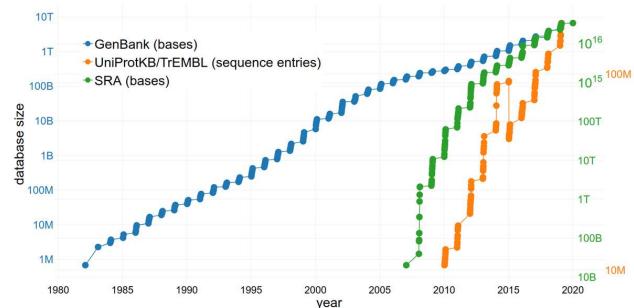


# Overview

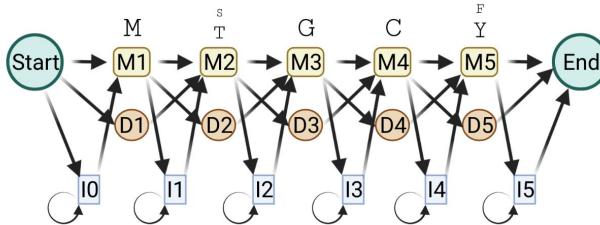
- Systems Biology & Omics (Genomics, Transcriptomics, Proteomics, Metabolomics)
- Bioinformatics tools and languages
- Data Life Cycle, potential for data upcycling, open science principles
- Data management: files, folders, and backups
- Introduction to Linux (Ubuntu)
- Introduction to the de.NBI cloud

# Why do we need bioinformatics?

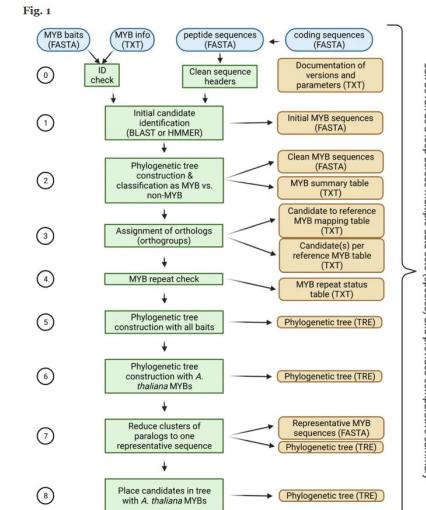
- Large data sets



- Complex models

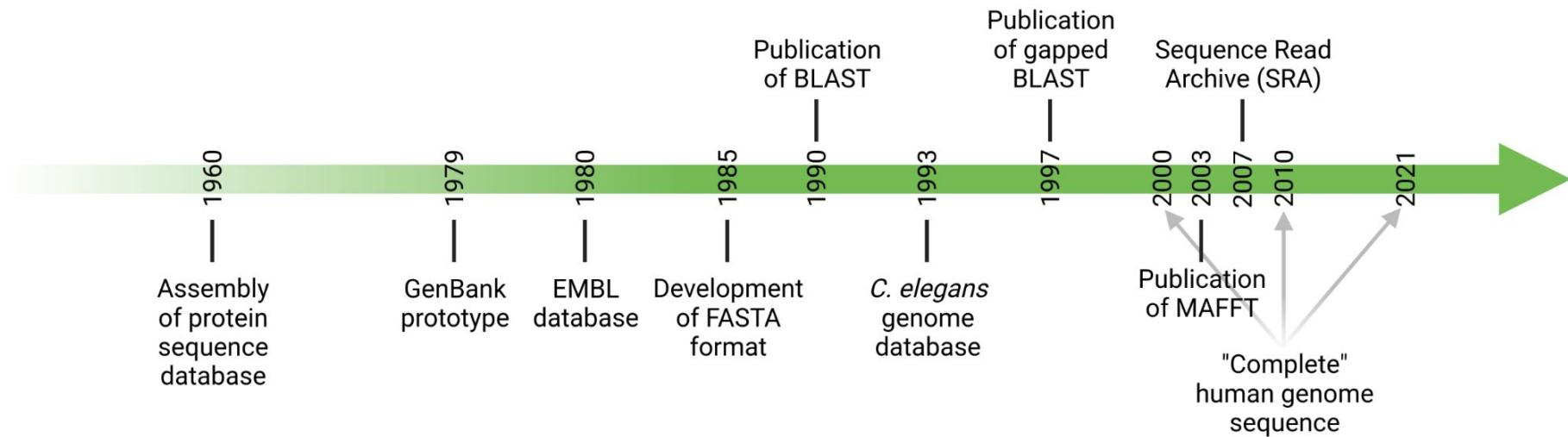


- Automatic analyses



Sielemann & Hafner et al., 2020: 10.7717/peerj.9954  
Pucker, 2022: 10.1186/s12864-022-08452-5

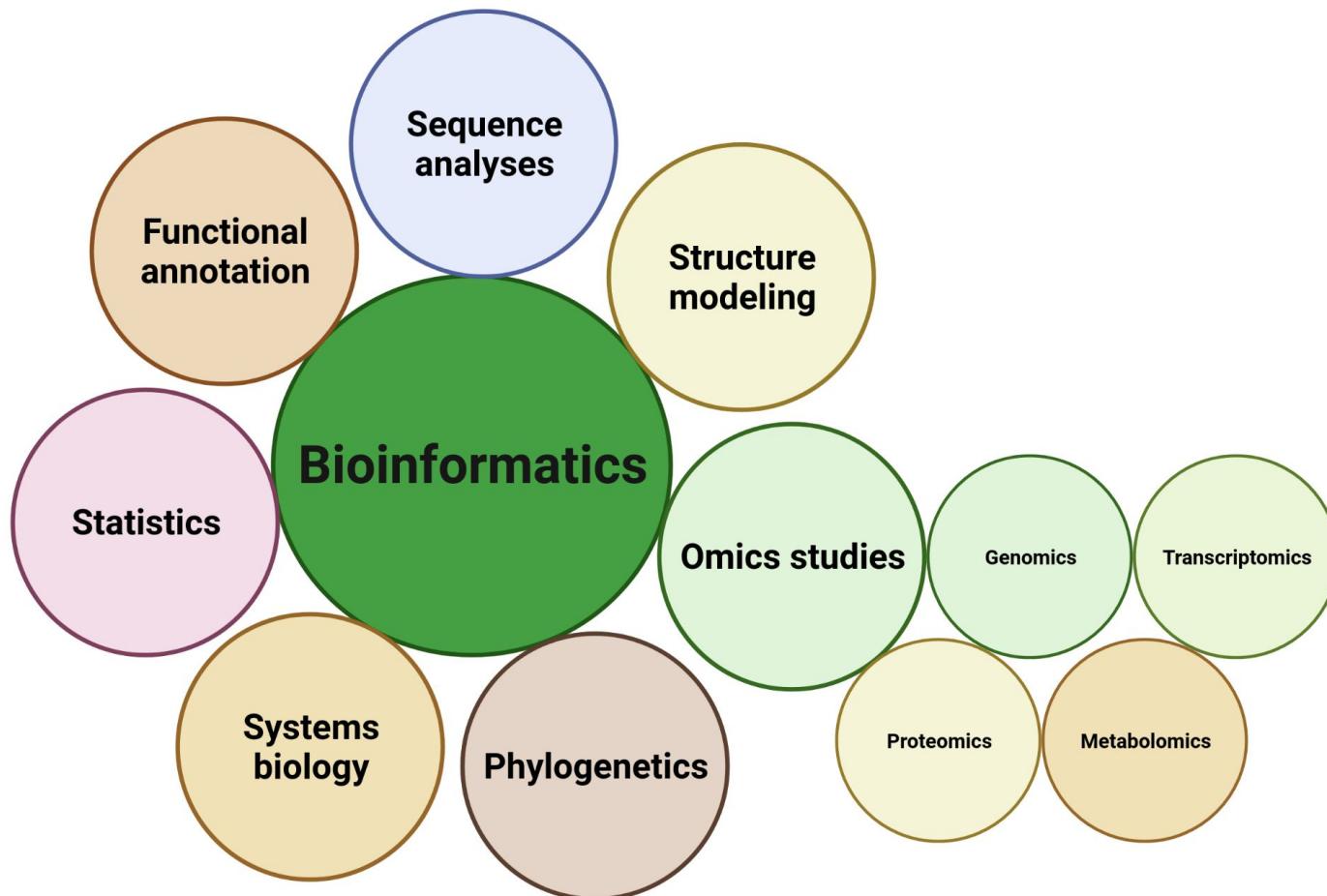
# History of bioinformatics



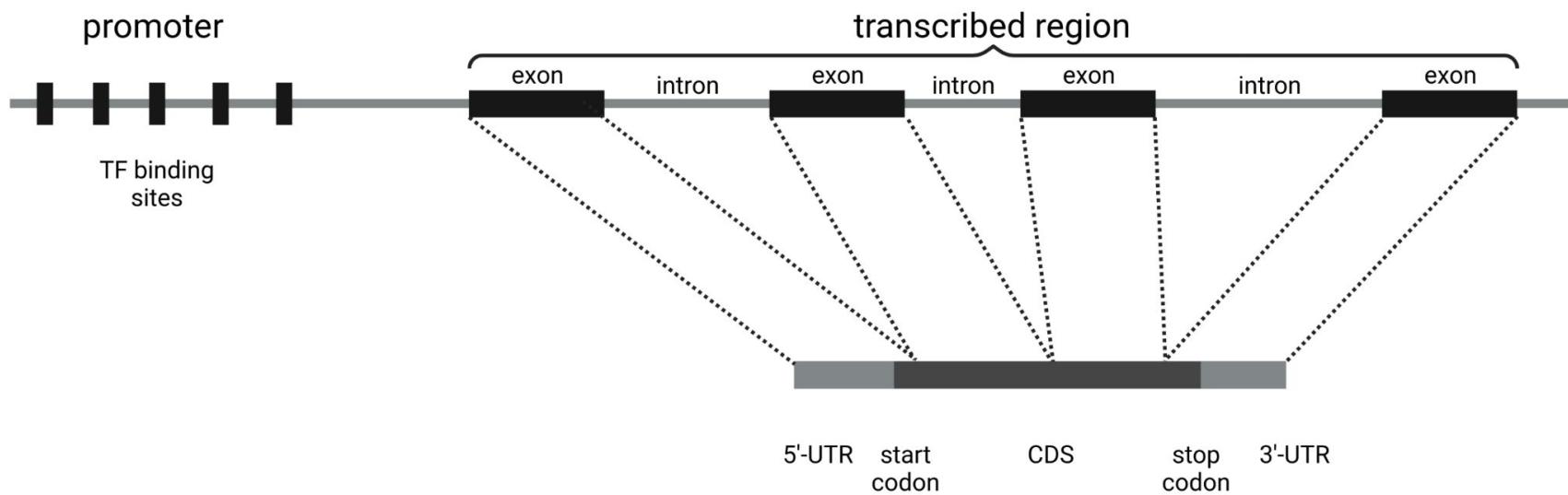
# Artificial Intelligence

- Experience with ChatGPT (OpenAI)?
- Do you see potential and risk?
- AI does not only work for text, but also for code, images, movies, games, gene prediction, pattern recognition, ....

# What are the different (sub)fields of bioinformatics?



# Sequence analysis



# Sequence analysis II

- Comparison of sequences to find similarities
  - Analysis of sequence composition
  - Important methods:
    - BLAST = Basic Local Alignment Search Tool
    - MAFFT = Multiple Alignment using Fast Fourier Transform

#### chalcone synthase [*Chenopodium quinoa*]

Sequence ID: XP\_021762431.1 Length: 392 Number of Matches: 1

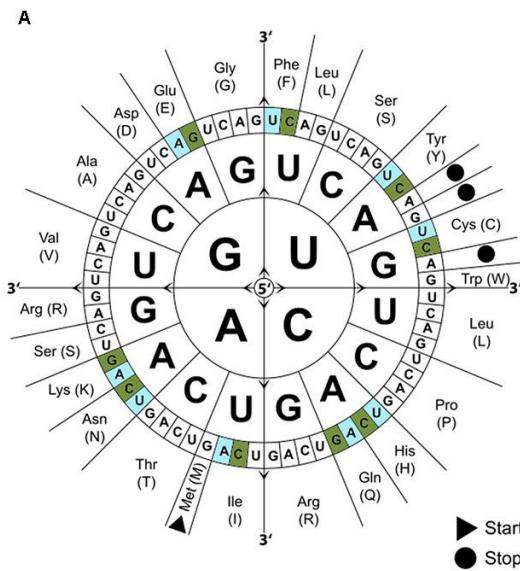
Range 1: 2 to 389	GenPept	Graphics	▼ Next Match	▲ Previous Match	
Score	Expect	Method	Identities	Positives	Gaps
666 bits(1719)	0.0	Compositional matrix adjust.	323/388(83%)	362/388(93%)	0/388(0%)
Query 3	TPSVOEIRDAORSNGPATILAITANGANPANEQYAAPDFYFRVTESEHMKELOKFKRMC T S++EIR AQR++GPATILAIGTA P_N +Y0++PD+YFRVTKSEHM ELK+KFKRM				62
Sbjct 2	TSSLEEIRKAQRADGPATILAIGTATPPNCVYQSDFPDYYFRVTKESEHMTTELKEFKRMC				61
Query 63	DNSM1KKRYMHVTEELLEENPHLCDFNASSLDTRQDILATEVPKLGEAAVKA1KEWGP D SM1 KRYMH+TEE L+ENP+C + SSLDTRQD+ +EVLP+KEEAAVKA1KEWGP				122
Sbjct 62	DKSM1KKRYMHVLTTEENPHLCDFNASSLDTRQDILATEVPKLGEAAVKA1KEWGP DKSM1KKRYMHVLTTEENPHLCDFNASSLDTRQDILATEVPKLGEAAVKA1KEWGP				121
Query 123	RSKITHVIFCTTSVGDPAGDYOLTLKLLGLRPSVSKRFMLYQQGCGYAGGTVLRLAKDIAEN +SKITHVI CTTSVGDPAGDYOLTLKLLGLRPSV+RFMLYQQGCG+AGGTVLRLAKD+AEN				182
Sbjct 122	KSKitHVIMCTTSVGDPAGDYOLTLKLLGLRPSVSRFFMLYQQGCGFAGGTVLRLAKDIAEN KSKitHVIMCTTSVGDPAGDYOLTLKLLGLRPSVSRFFMLYQQGCGFAGGTVLRLAKDIAEN				181
Query 183	NRGARVLVVCSEAIIITCFRPGTOIHLDSMVQGALFfdgagavivgapdPDESIERPIFOLV NRGARVLVCC-EIT ICFRPGT+ HLDSMVQGALFCDGAGA+IVGADPDESIERP+F+V				242
Sbjct 182	NRGARVLVVCSEAIIITCFRPGTETLHDSMVQGALFGDGAGALIVGADPDESIERPLFKMV NRGARVLVVCSEAIIITCFRPGTETLHDSMVQGALFGDGAGALIVGADPDESIERPLFKMV				241
Query 243	WAAQTLIPGSEGAIJDGHLLREVGLTFHLLKDVPGLISKNIEKALVEAFOPIGDDWSNISFW WAAQTLIPGSEGAIJDGHLLREVGLTFHLLKDVPGLISKNIEKALVEAFOPIGDDWSNISFW				302
Sbjct 242	WAAQTLIPGSEGAIJDGHLLREVGLTFHLLKDVPGLISKNIEKALVEAFOPIGDDWSNISFW WAAQTLIPGSEGAIJDGHLLREVGLTFHLLKDVPGLISKNIEKALVEAFOPIGDDWSNISFW				301
Query 303	VAHPGGRAILDDVESKGLKEDKLKTTRHVLSEYGNMSSACVLFILDEMRKRKAMEGAT +AHPGG AILD VE+KLGLKE+KL TR+VSE+LMGNMSSACVLFILDEMRKR+MKEG AT				362
Sbjct 302	IAHPPGPAILDQVEAKLGLKEEKLTATRNVLSEFGNMSSACVLFILDEMRKKSMEKGAT IAHPPGPAILDQVEAKLGLKEEKLTATRNVLSEFGNMSSACVLFILDEMRKKSMEKGAT				361
Query 363	TGEGLEWGVLFGFPGPLTVETVMLHSPV 390 TG+GL+WGVLFGFPGPLTVETV+LHSVP				
Sbjct 362	TGDGLDWGVLFGFPGPLTVETVVLHSPV 389				

CLUSTAL format alignment by MAFFT EFT-NS-i (v7.487)



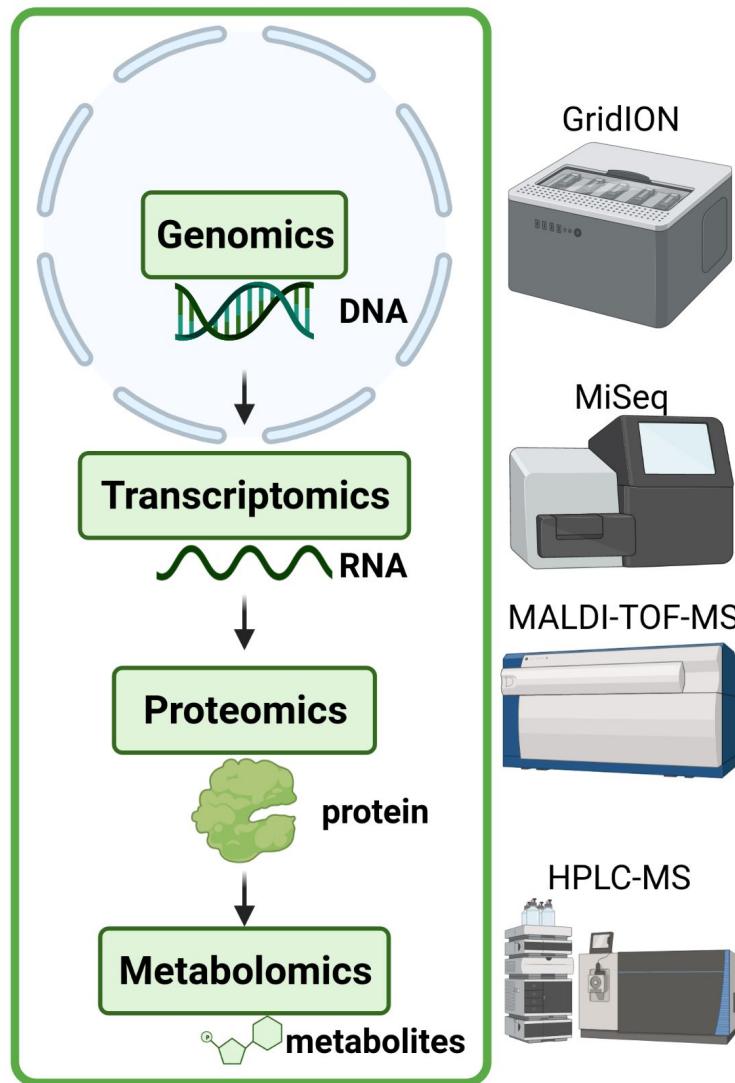
# Sequence analysis III

- Codons define amino acids to integrate
- Genetic code is redundant i.e. multiple codons for each amino acid
- Organisms can have a preference for certain codons for a given amino acid
- Optimization of sequences for heterologous expression



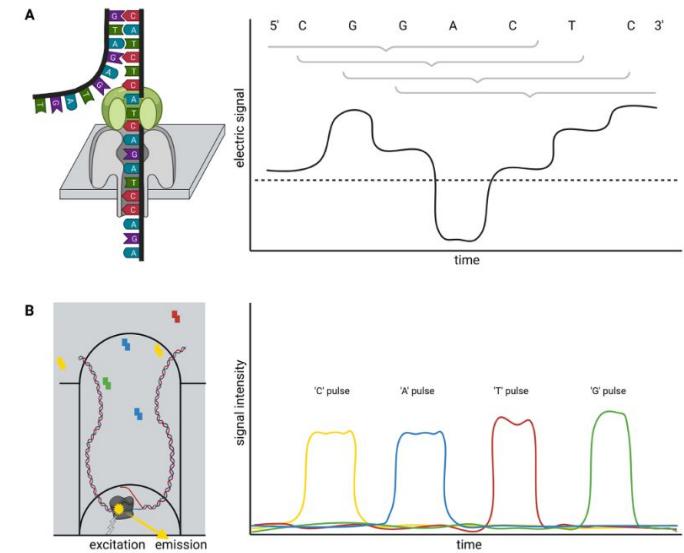
Amino acid	Codon	<i>P. patens</i>	<i>A. thaliana</i>	<i>S. cerevisiae</i>	<i>S. pombe</i>	<i>H. Sapiens</i>
Cys	UGC	+	+	-	+	+
	UGU	-	-	+	-	-
Glu	GAA					
	GAG	++	+	-	+	+
Phe	UUC	+++	++	++	++	++
	UUU	-	-	-	-	-
His	CAC	+++	++	++	++	++
	CAU	-	-	-	-	-
Ile	AUA	-	-	/	-	-
	AUC	++	/	-	-	+
	AUU	-	-	-	-	-
Lys	AAA	-	-	-	-	-
	AAG	+++	++	++	++	++
Asn	AAC	+++	++	++	++	++
	AAU	-	-	-	-	-
Gln	CAA	-	-	-	+	-
	CAG	++	+	+	+	+
Tyr	UAC	+++	++	++	++	++
	UAU	-	-	-	-	-

# Omics levels



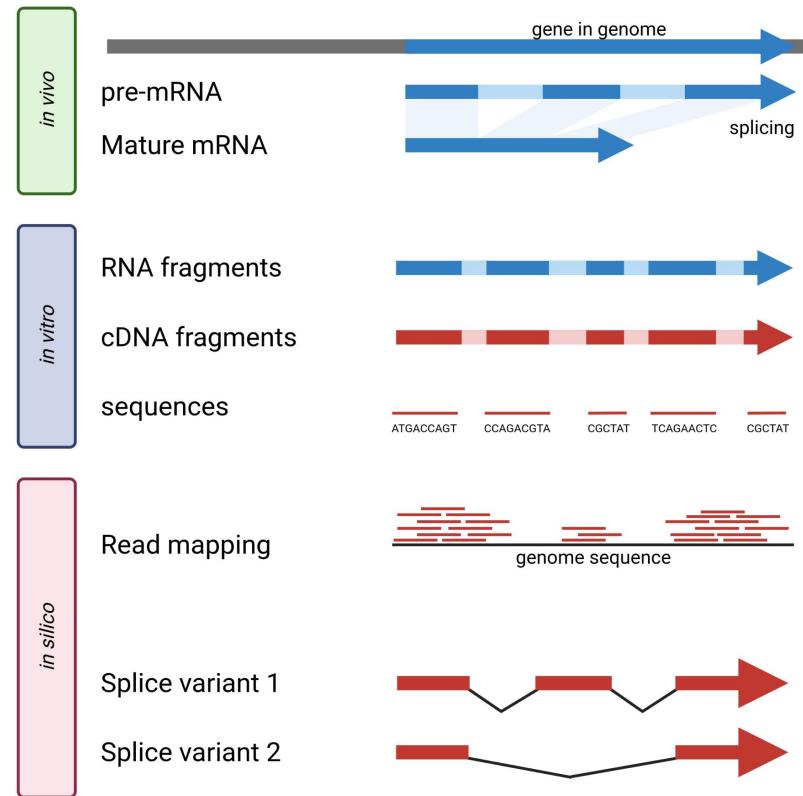
# Genomics

- Genome = sum all of genetic information in a cell
- Plants have DNA in nucleus (nucleome), chloroplasts (plastome), and mitochondria (chondromes)
- Sequencing plant genomes: Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PacBio)
- Genome sequence assembly: HiCanu, FALCON, Shasta, NextDenovo2 ...
- Comparison of genome sequences
- Handling of large data sets (>1TB) necessary for most projects



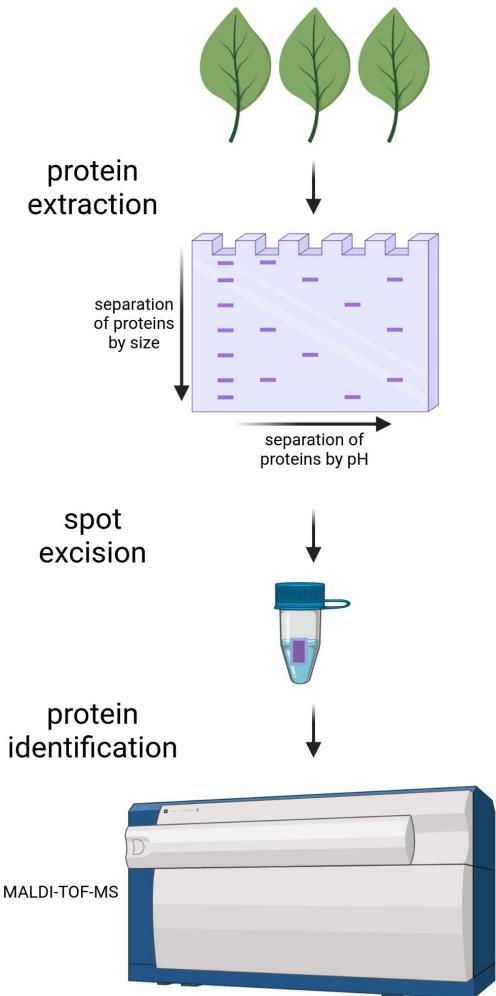
# Transcriptomics

- Transcriptome = sum of transcripts in a cell/tissue under defined conditions and at a defined time point
- Systematic investigation of gene expression:
  - Microarrays: method of choice for many years
  - RNA-Seq: current method of choice to study gene expression systematically
- Comparison of different samples (tissues, conditions, genotypes, ....)



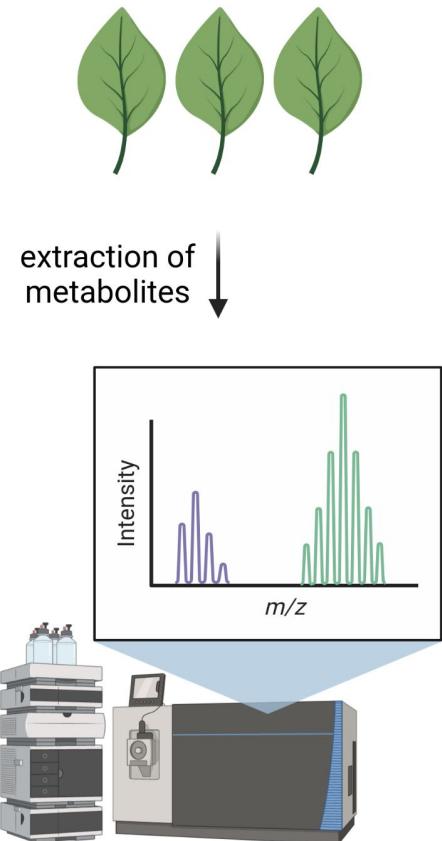
# Proteomics

- Proteome = sum of all proteins in a cell/tissue under defined conditions at a defined time point
- Methods for systematic investigation:
  - High Performance Liquid Chromatography (HPLC)-Mass Spectrometry (MS)
  - Separation of proteins via 2D gels and investigation of spots by MS
- Heterogenous properties of proteins make systematic analysis challenging



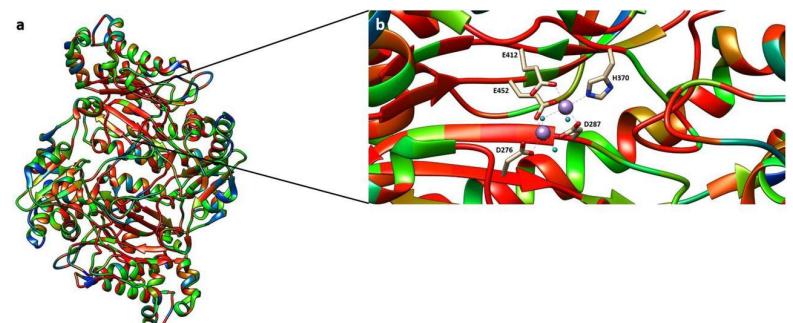
# Metabolomics

- Metabolom = sum all of metabolites in a cell/tissue under defined conditions at a defined time point
- Methods for systematic metabolite analysis:
  - HPLC-MS/MS
  - GC-MS/MS
- Chemical diversity of metabolites poses a challenge for the analysis
- Identification of metabolites remains a challenge



# Structure modeling

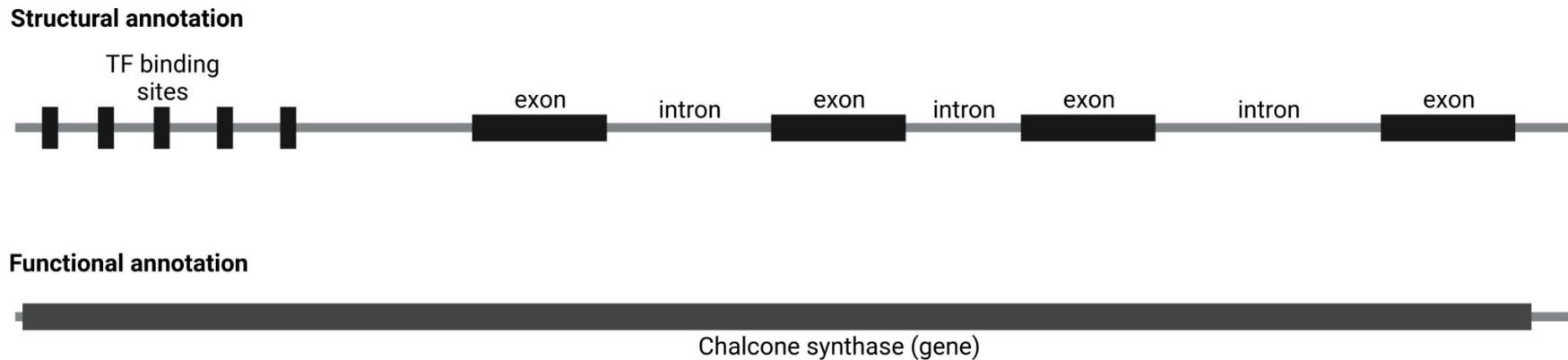
- Modeling of RNA structures
  - Collection of tools: [https://molbiol-tools.ca/RNA\\_analysis.htm](https://molbiol-tools.ca/RNA_analysis.htm)
  - BiBiServ: <https://bibiserv.cebitec.uni-bielefeld.de/rna>
  - RNAfold: <http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi>
- Modeling of 3D protein structures
  - Collection of tools:  
[https://molbiol-tools.ca/Protein\\_tertiary\\_structure.htm](https://molbiol-tools.ca/Protein_tertiary_structure.htm)
  - PredictProtein: <https://predictprotein.org/>
  - AlphaFold2 (10.1021/acs.jcim.1c01114)
  - AlphaFold3 (10.1038/s41586-024-07487-w)



Schilbert et al., 2018: 10.1101/423475

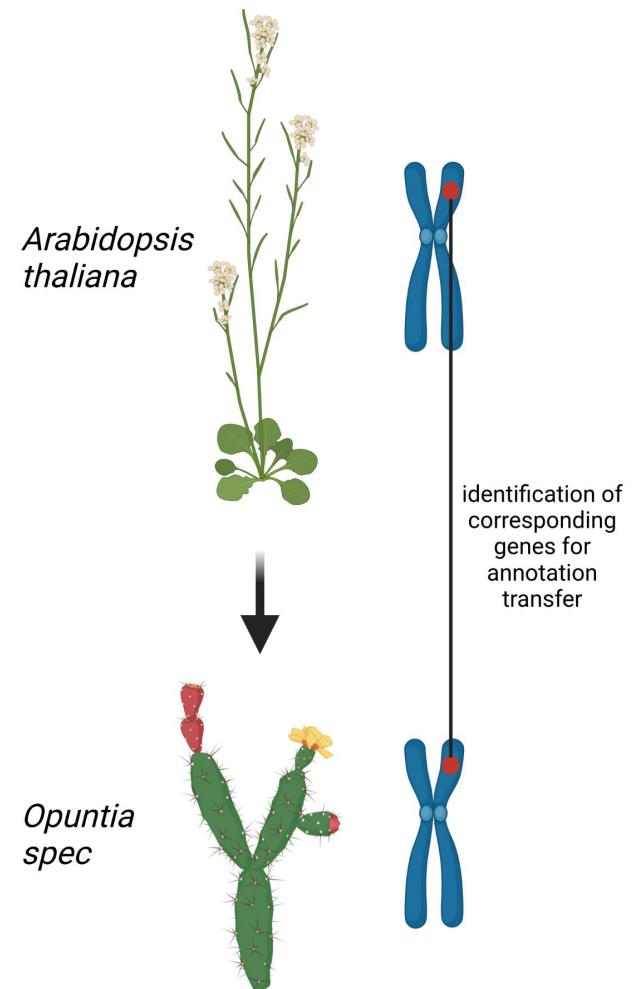
# Structural vs. functional annotation

- Structural annotation = position of features (e.g. exons) in a genome sequence
- Functional annotation = biochemical function of a feature (e.g. gene)



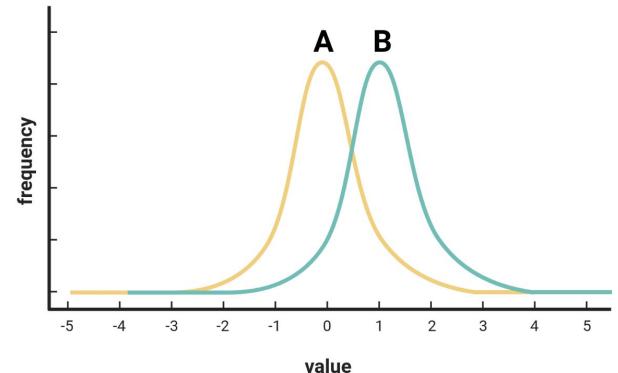
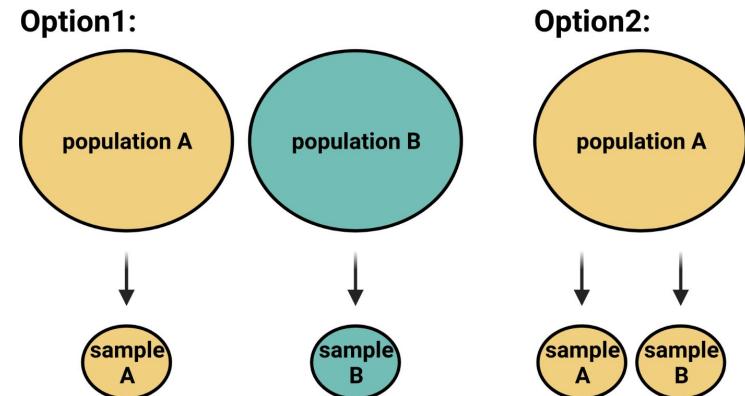
# Functional annotation

- Assumption: similar sequences (genes) have similar functions
- Sequence similarity indicates origin from a shared common ancestor
- Ancestor is likely to have inherited the same function to offspring
- Transfer of functional annotation based on sequence similarity
- *Arabidopsis thaliana* gene functions are well studied and usually serve as reference



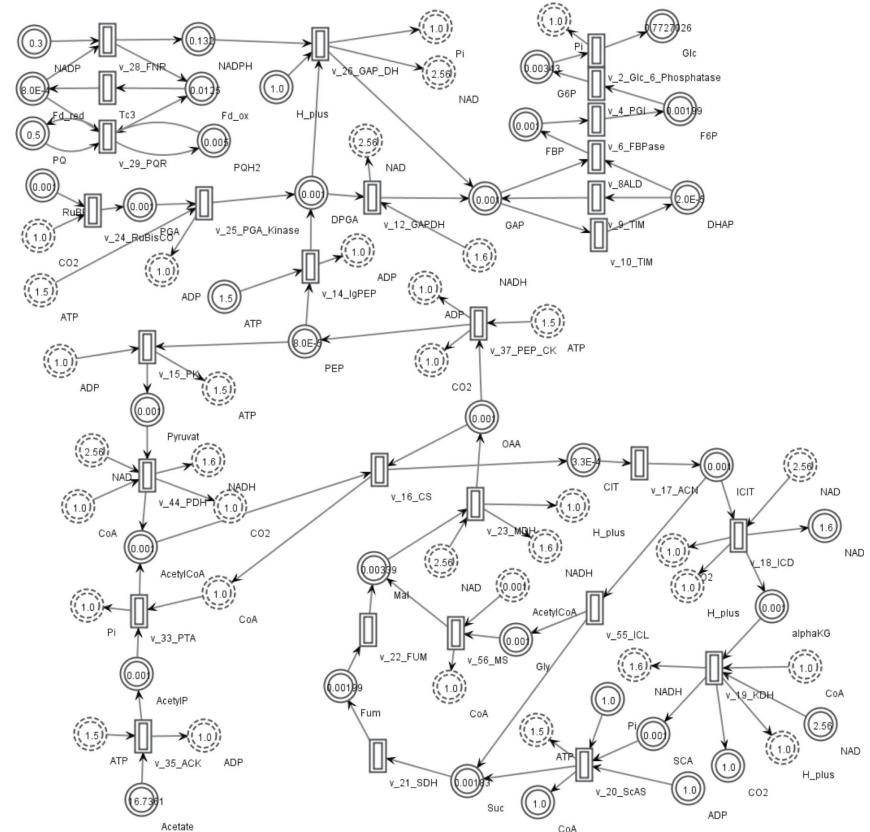
# Statistics

- Statistical tests to assess data sets
- Important for analysis of omics data sets
- Concept of a statistical test:
  - $H_0$  = difference of two samples can be explained by random effects
  - $H_1$  = certain factor(s) determine the difference of groups



# Systems biology

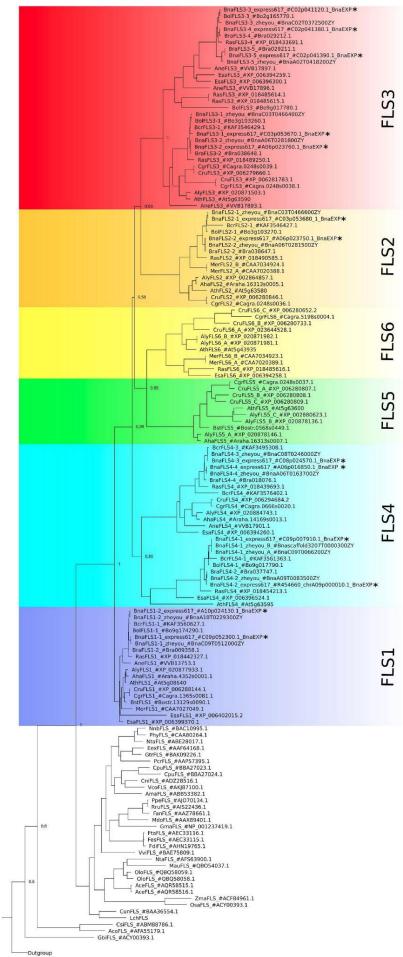
- Description of biological systems with mathematical methods
- Michaelis-Menten kinetics is basis of many metabolic models
- Integration of multiple omics data sets
- Modeling of metabolic flux through pathways
- Example: Modeling of aerobic carbon metabolism in *Chlamydomonas reinhardtii*



Brinkrolf et al., 2018: 10.1515/jib-2018-0018

# Phylogenetics

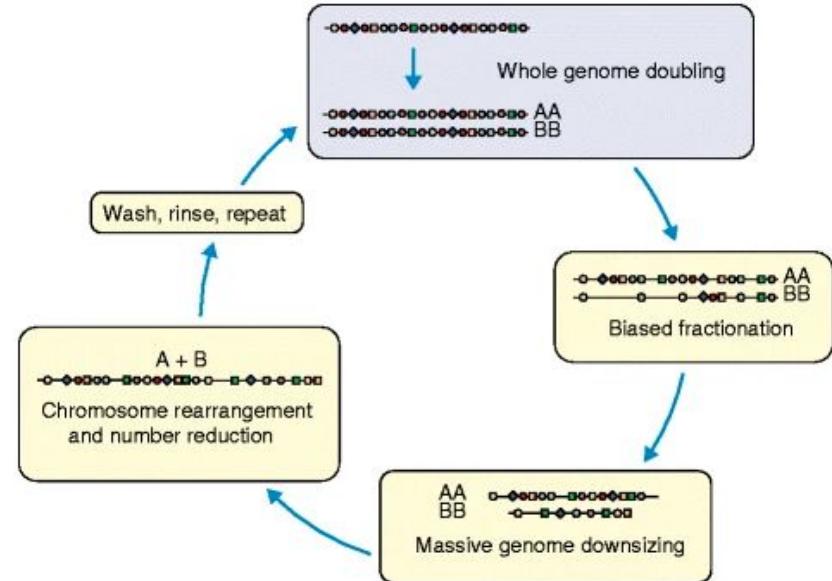
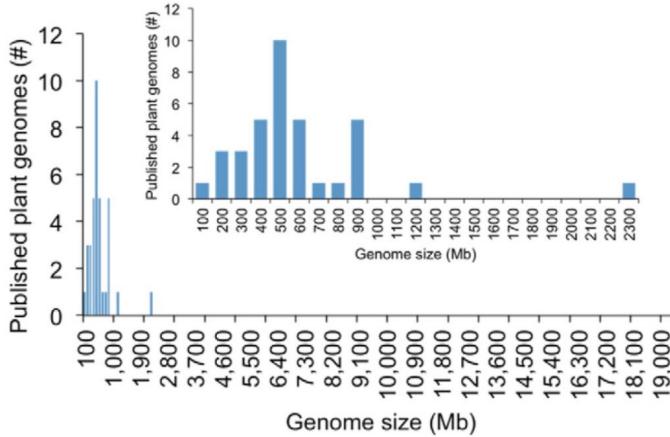
- Analysis of the evolutionary relationships between sequences/species
  - Reconstruction of possible models to explain these relationships
  - Not limited to sequence similarity, but considered (most likely) ancestral stages



Schilbert et al., 2021: 10.3389/fpls.2021.733762

# Phylogenomics

- Genomes evolve over time
- Whole genome duplication is crucial source for new sequences in plants
- Polyploidy = multiple copies of all chromosomes
- Transposable Elements (TEs) can increase genome size
- Huge differences in chromosome numbers and genome sizes



Wendel et al., 2016: 10.1186/s13059-016-0908-1  
Michael & Jackson, 2013: 10.3835/plantgenome2013.03.0001in



# Do you know any bioinformatics tools?

Required subject

Choose File No file chosen

seqtype

pep

Optional scoreratio

0.3

simcut

0.4

minsim

0.4

minres

0.0

minreg

0.0

possibilities

3

Privacy

I would like to be notified when my job status changes. Please contact me via the following email address: \_\_\_\_\_

I want my browser to keep track of my current jobs and parameter settings. This requires the use of cookies.

<https://pbb-tools.de/>

# Types of bioinformatics tools

- Different types of tools:
  - Command line: require a basic understanding of Linux, but allow the processing of large data sets
    - Example: long read-based t-DNA analysis (loreta);  
<https://doi.org/10.1186/s12864-021-07877-8>
  - Graphical user interface (GUI): easier to use, but usually restricted to smaller data sets
    - Example: Integrated Genome Viewer (IGV),  
<https://igv.org/>
  - Web services: easy to use, but usually restricted to small data sets
    - Example: Knowledge-Based Identification of Pathway Enzymes (KIPES); <http://pbb-tools.de/KIPES>

```
samtools view -S -b sample.sam > sample.bam
```



KIPES (Knowledge-based Identification of Pathway Enzymes)

Required  
Choose File: No file chosen

swtype: pep

Optional  
scoreratio: 0.2

simcut: 0.4

minsim: 0.4

minres: 0.0

minreq: 0.0

possibilities: 3

Privacy  
 I would like to be notified when my job status changes. Please contact me via the following email address: \_\_\_\_\_  
 I want my browser to keep track of my current jobs and parameter settings. This requires the use of cookies.

# How to run command line tools

- **Trimmomatic:** Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*, btu170.

<http://www.usadellab.org/cms/?page=trimmomatic>

```
java -jar <path to trimmomatic jar> SE [-threads <threads>] [-phred33 | -phred64] [-trimlog <logFile>] <input> <output> <step 1> ...
```

- **Samtools:** Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, 1000 Genome Project Data Processing Subgroup, The Sequence Alignment/Map format and SAMtools, *Bioinformatics*, Volume 25, Issue 16, 15 August 2009, Pages 2078–2079,

<https://doi.org/10.1093/bioinformatics/btp352>

<http://www.htslib.org/doc/samtools-view.html>

```
samtools view -S -b sample.sam > sample.bam
```

# Where to find the right tool?

## Genetics

Resources for the study of genes and heredity in living organisms. They provide analytical approaches for studying gene structure, genetic variation, the regulation of gene expression and the relationships between organism genotype and phenotype.

Genetics	▶ Gene expression
Gene regulation	▶ Gene expression profiling
Transcription factors and regulatory sites	▶ Functional, regulatory and non coding RNA
Genetic variation	▶ DNA polymorphism
Genotyping	▶ Genotype and phenotype
Population genetics	▶ Gene functional annotation
Gene and protein families	▶ Gene prediction
Enrichment analysis	▶ Genome visualisation

## Proteins

Resources for protein sequence analysis, protein visualization, and the prediction, modeling and analysis of protein structure. They provide analytical approaches to interpret and explore protein sequence and structure data, and to understand protein function and mechanism of action.

Protein sequence analysis	▶ Protein structure analysis
Protein folds and structural domains	▶ Gene and protein families
Protein structural models and surfaces	▶ Protein sites, features and motifs
Transcription factors and regulatory sites	▶ Protein interactions
Protein modelling	▶ Protein fold recognition
Protein structure prediction	▶ Protein secondary structure prediction
Protein binding site prediction	▶ Protein property calculation
Protein subcellular localisation prediction	▶ Protein folding, stability and design

## Nucleic acids

Resources for the processing and analysis of nucleic acid sequence, structural and other data. They provide analytical methods to interpret and explore nucleic acid sequence and structural data in context of gene structure, genetic variation, and the regulation of gene expression.

DNA	▶ RNA
Genetic mapping	▶ Read mapping
Functional, regulatory and non-coding RNA	▶ Methylation analysis
DNA polymorphism	▶ SNP detection
Nucleic acid sequence analysis	▶ Nucleic acid sites, features and motifs
Nucleic acid feature detection	▶ Transcription factor binding site prediction
Transcription factors and regulatory sites	▶ Transcriptional regulatory element prediction
Nucleic acid structure analysis	▶ RNA secondary structure prediction

<https://bio.tools/>

# Do you know any programming languages?

**string integer float**  
**float list dict**  
**sorted key value**  
**def function print**  
**else elif**



# Script and programming languages

- Perl: one of the first script languages used in bioinformatics (sequences)
  - Examples: AUGUSTUS and helper scripts
  - <https://www.perlfoundation.org/>
- Python: established script language for bioinformatics (sequences)
  - Examples: BUSCO, KIPEs, MYB\_annotator
  - <https://www.python.org/psf/>
- Julia: script language that is gaining attention (structures)
  - Examples: biojulia
  - <https://julialang.org/>
- R: established script language for bioinformatics (numbers/statistics)
  - Examples: DESeq2, ggplot2,
  - <https://www.r-project.org/about.html>
- Java: powerful for the analysis of large data sets
  - Integrated Genomics Viewer (IGV), SnpEff, samtools, Trimmomatic, ...
  - <https://www.java.com/de/> (commercial)



# Repositories

- OpenAccess and OpenMethods make computational tools freely accessible
- Online repositories that allow sharing of code:
  - GitHub, Bitbucket, GitLab, SourceForge, ...

The screenshot shows a GitHub repository page for 'bpucker / MYB\_annotator'. The repository has 150 commits, 2 forks, and 6 stars. It contains files like AthRefMYBs.txt, LICENSE, MYB\_annotator.png, MYB\_annotator.py, MYB\_baits.fasta, MYB\_baits.hmm, MYB\_baits.txt, README.md, col0\_example\_output.7z, and environment.yml. The 'About' section describes it as a tool for automatic identification, annotation, and analysis of the MYB gene family in plants. It includes a DOI link (doi.org/10.1101/2021.10.16.464636), tags (bioinformatics, annotation, python3, genome-wide, myb, r2r3-myb), and links to Readme, License, and Releases (v0.2 (revised preprint)). The 'Packages' section shows no packages published, and the 'Languages' section shows Python at 100.0%.

[https://github.com/bpucker/MYB\\_annotator](https://github.com/bpucker/MYB_annotator)

# Licences

- MIT: leanest licence (everything is possible)
- Apache: similar to MIT, but lengthy
- GPL (General Public License): ensures that derived work remains open
- BSD (Berkeley Software Distribution): similar to MIT, but more cases specified

## MIT License

Copyright (c) [year] [fullname]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Conda

- Bioinformatic tools often have dependencies
- Dependency = module/tool that needs to be installed already
- Conda enables installation without administrator privileges



*Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN, and more.*

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

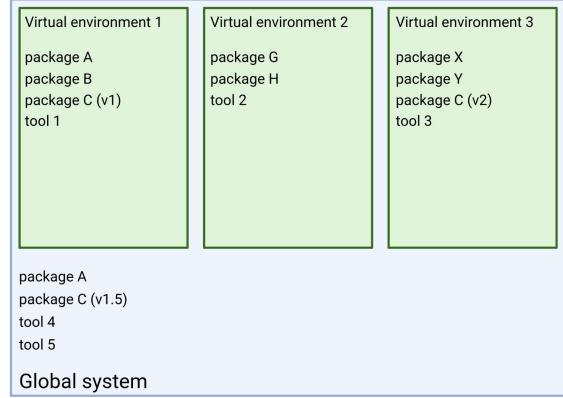
Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

In its default configuration, conda can install and manage the thousand packages at [repo.anaconda.com](http://repo.anaconda.com) that are built, reviewed and maintained by Anaconda®.



# Virtual environment

- Virtual environment allows controlled installation of tools without interfering with system
- Tool and all dependencies are installed in a box
- Different versions of packages/modules are required for different tools
- Different virtual environments can be used for different tools



## 12.2. Creating Virtual Environments

The module used to create and manage virtual environments is called `venv`. `venv` will usually install the most recent version of Python that you have available. If you have multiple versions of Python on your system, you can select a specific Python version by running `python3` or whichever version you want.

To create a virtual environment, decide upon a directory where you want to place it, and run the `venv` module as a script with the directory path:

```
python3 -m venv tutorial-env
```

This will create the `tutorial-env` directory if it doesn't exist, and also create directories inside it containing a copy of the Python interpreter and various supporting files.

A common directory location for a virtual environment is `./venv`. This name keeps the directory typically hidden in your shell and thus out of the way while giving it a name that explains why the directory exists. It also prevents clashing with `.env` environment variable definition files that some tooling supports.

Once you've created a virtual environment, you may activate it.

On Windows, run:

```
tutorial-env\Scripts\activate.bat
```

On Unix or Mac OS, run:

```
source tutorial-env/bin/activate
```

(This script is written for the bash shell. If you use the `csh` or `fish` shells, there are alternate `activate.csh` and `activate.fish` scripts you should use instead.)

Activating the virtual environment will change your shell's prompt to show what virtual environment you're using, and modify the environment so that running `python` will get you that particular version and installation of Python. For example:

```
$ source ./envs/tutorial-env/bin/activate
(tutorial-env) $ python
Python 3.5.1 (default, May  6 2016, 10:59:36)
...
>>> import sys
>>> sys.path
['', '/usr/local/lib/python35.zip',
 './envs/tutorial-env/lib/python3.5/site-packages']
>>>
```

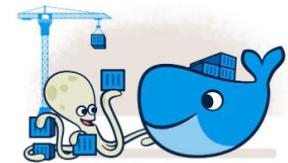
<https://docs.python.org/3/tutorial/venv.html>

# Docker

- Solution to prepare an environment for running tools
- Build: developer of tool generates an image of the required environment
- Share: image is shared with users of the tool
- Run: user can mount the image to run a tool in a defined environment
- Alternatives: podman, OpenVZ, VirtualBox, Kubernetes, LXC, ZeroVM,  
...

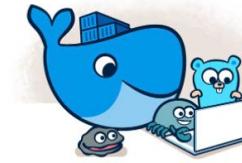
## Build

- Get a head start on your coding by leveraging Docker Images to efficiently develop your own unique applications on Windows and Mac. Create your multi-container application using Docker Compose.
- Integrate with your favorite tools throughout your development pipeline
  - Docker works with all development tools you use including VS Code, CircleCI and GitHub.
- Package applications as portable container images to run in any environment consistently from on-premises Kubernetes to AWS ECS, Azure ACI, Google GKE and more.



## Share

- Leverage Docker Trusted Content, Including Docker Official Images and Images from Docker Verified Publishers from the Docker Hub repository.
- Innovate by collaborating with team members and other developers and by easily publishing Images to Docker Hub.
- Personalize developer access to Images with roles based access control and get insights into activity history with Docker Hub Audit Logs.



## Run

- Deliver multiple applications hassle free and have them run the same way on all your environments including design, testing, staging and production – desktop or cloud-native.
- Deploy your applications in separate containers independently and in different languages. Reduce the risk of conflict between languages, libraries or frameworks.
- Speed development with the simplicity of Docker Compose CLI and with one command, launch your applications locally and on the cloud with AWS ECS and Azure ACI.



# Galaxy

- Graphical user interface (web-based platform) for bioinformatic workflows
- Open source enables local installation (e.g. on compute cluster)
- Supported by de.NBI, ELIXIR, and many others

Galaxy is an open source, web-based platform for data intensive biomedical research. If you are new to Galaxy start here or consult our help resources. You can install your own Galaxy by following the tutorial and choose from thousands of tools from the Tool Shed.



Donate to the James P. Taylor Foundation for Open Science

[Learn More](#)

Want to learn the best practices for the analysis of SARS-CoV-2 data using Galaxy? Visit the Galaxy SARS-CoV-2 portal at <https://galaxyproject.org/projects/covid19/>



The Galaxy Team is a part of the Center for Comparative Genomics and Bioinformatics at Penn State, the Department of Biology at Johns Hopkins University and the Computational Biology Program at Oregon Health & Science University.

This instance of Galaxy is utilizing infrastructure generously provided by the Texas Advanced Computing Center, with support from the National Science Foundation.

The development and maintenance of this site are supported by NIH NHGRI award U24 HG006620 and NSF award 1929694. Additional support is provided by NIH awards AI134384 and HG010263, as well as NSF award 1931533.

This is a free, public, internet accessible resource. Data transfer and data storage are not encrypted. If there are restrictions on the way your research data can be stored and used, please consult your local institutional review board or the project PI before uploading it to any public site, including this Galaxy server. If you have protected data, large data storage requirements, or short deadlines you are encouraged to setup your own local Galaxy instance or run Galaxy on the cloud.

<https://usegalaxy.org/>  
Afgan et al., 2018: 10.1093/nar/gky379

# CyVerse

- Workspace for data-driven science
- Cloud storage to exchange files
- Cloud computing to run analyses on data sets
- Training courses in data science



**Analyze & Share**  
Do all your research in one place with our easy to use bioinformatics tools, image analyses, cloud services, and resources for reproducibility, automation, storing, and sharing.

[Learn More](#)



**Build Your Skills**  
Access our educational webinars, hands-on workshops, self-paced tutorials, and more to teach yourself and your students how to do open, collaborative science.

[Learn More](#)



**Bring Your Own**  
Connect your own storage or compute power, build off our APIs, or even install your own version of CyVerse. We can help you find out what is right for your project and goals.

[Learn More](#)

<https://cyverse.org/>

- German Network for Bioinformatics Infrastructure Service, Training, Cooperations & Cloud Computing
- Centrally funded support for bioinformatic analyses in Germany
- Almost 100 tools provided and supported by de.NBI
- Training events for next generation scientists
- de.NBI cloud enables researchers in Germany to conduct bioinformatics research

The screenshot shows the de.NBI website homepage. At the top, there's a navigation bar with links for HelpDesk, Contact, Intranet, and Search. The main header features the de.NBI logo and the text "Big Data Exploitation in Life Science". Below this, there's a section for "Our COVID-19 Research". The page is divided into three main sections: "Service & Tools", "Training & Education", and "de.NBI Cloud". Under "Service & Tools", it says "de.NBI maintains and develops almost 100 software tools and four internationally recognized databases." Under "Training & Education", it lists "30 Fellowships for PhDs (w/m/d) and Postdocs" and "Open position as Bioinformatician / Data scientist (f/m/d)". Under "de.NBI Cloud", it says "The de.NBI cloud is an excellent solution to enable integrative analyses for the entire life sciences community in Germany." At the bottom, there's a section for "Events" with a link to "View all Events".

<https://www.denbi.de/>

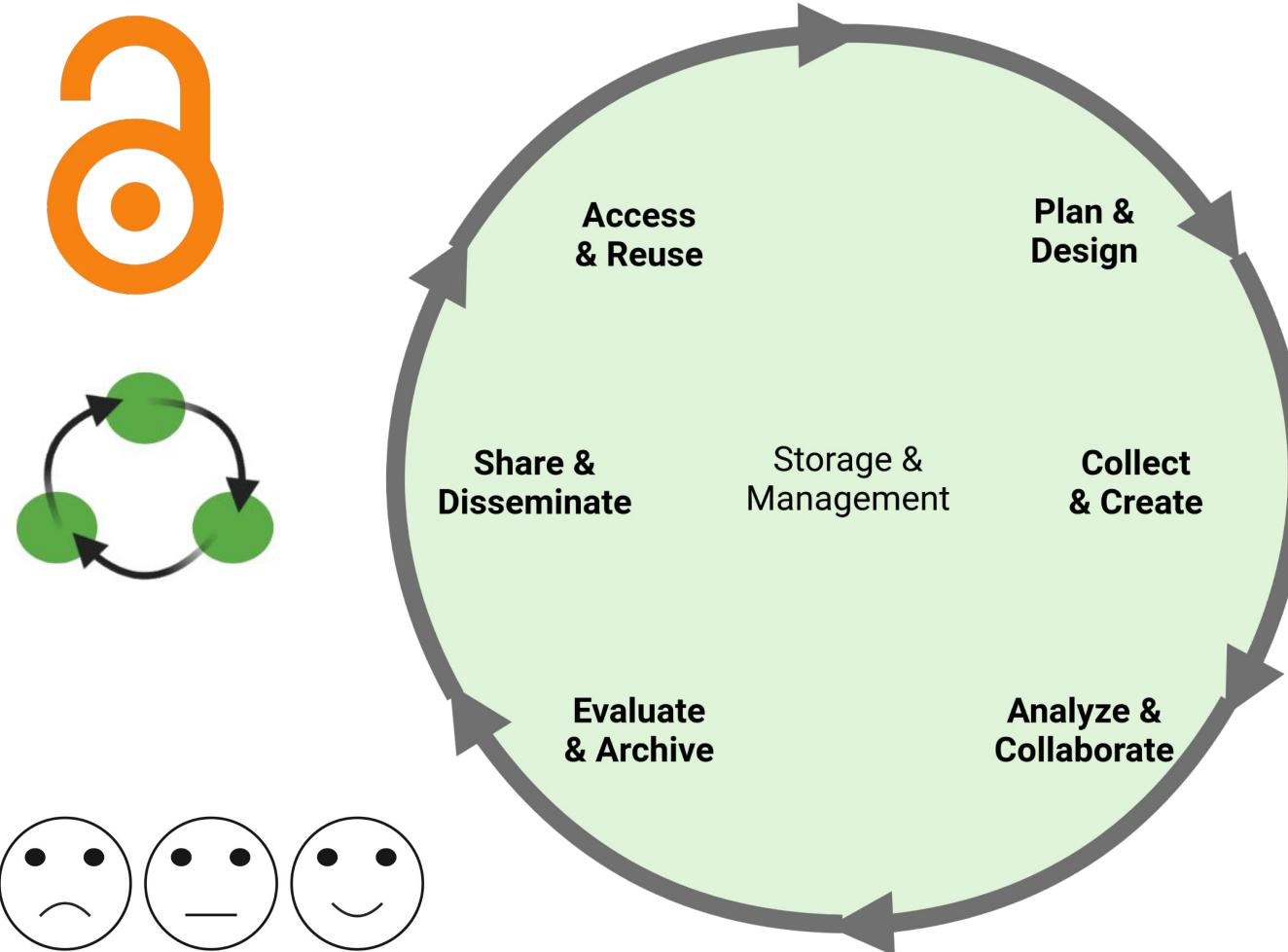
# ELIXIR

- Intergovernmental organisation of life scientists and computer scientists in Europe
- Provides resources for European life scientists
- 23 ELIXIR nodes in different countries
- EMBL-EBI heads ELIXIR
- de.NBI is German ELIXIR node



<https://elixir-europe.org/about-us/who-we-are>

# Data life cycle



PLAN

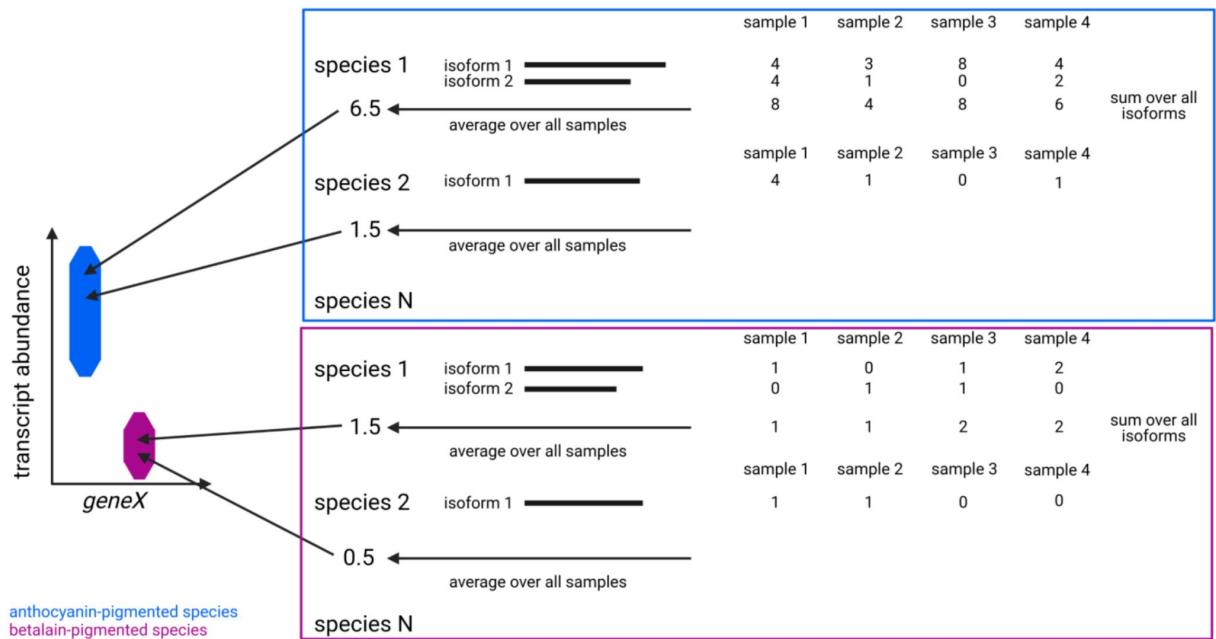


[https://commons.wikimedia.org/wiki/File:RDM\\_02\\_Collect-and-Capture\\_frame01.svg](https://commons.wikimedia.org/wiki/File:RDM_02_Collect-and-Capture_frame01.svg)  
<https://www.picpedia.org/keyboard/a/analyze.html>

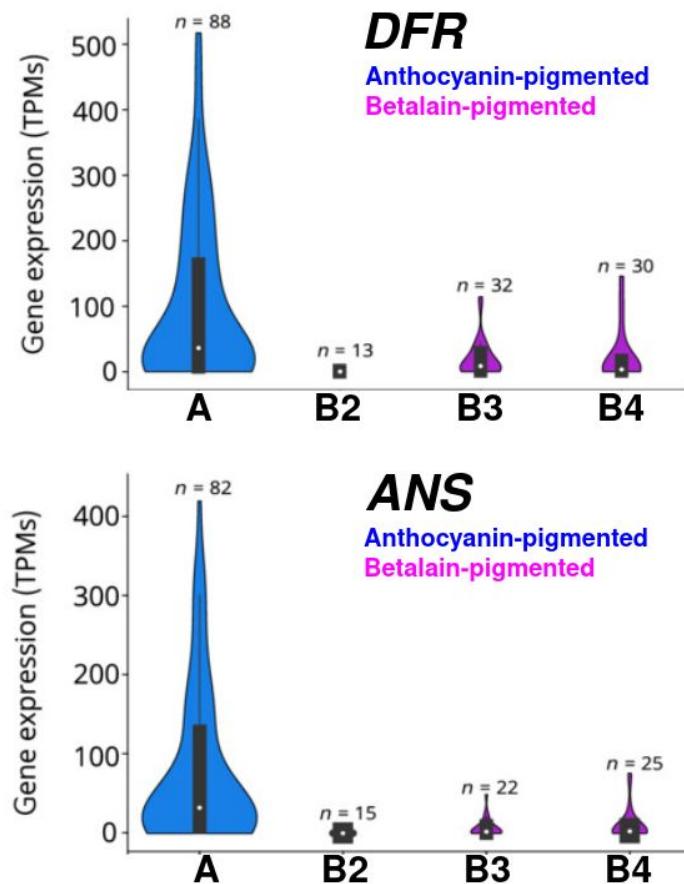
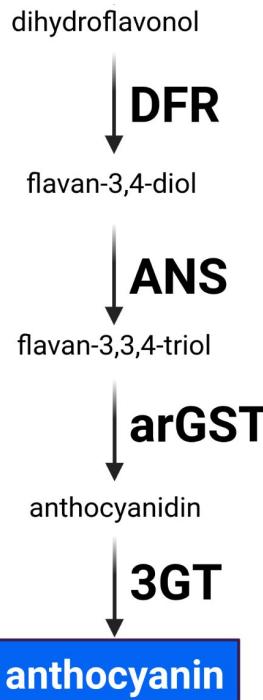


# Data Upcycling: Cross-species transcriptomics

- Genome sequences of multiple species
- Transcriptome assemblies of >300 species
- >4500 RNA-seq data sets



# Low transcription of anthocyanin biosynthesis genes



# FAIR data

- Findable:
  - Globally unique and persistent identifier
  - Metadata must be available in connection to the identifier
- Accessible:
  - Retrieval based on the identifier
  - Protocol is open, free, and universally applicable
  - Authentication is possible where needed
  - Metadata are available even if data are restricted
- Interoperable:
  - Metadata use a formal, accessible, broadly accessible language
  - Vocabulary need to follow FAIR standards
- Re-usable:
  - Clear and accessible data usage license
  - Meet domain-relevant community standards

# OpenData / OpenScience

- Everyone can access and re-use these data sets
- Facts cannot be owned by someone
- Huge economic potential through re-use; advantages for society
- Possible restrictions: name author, share-alike
- Related initiative: open source, open content, open access, open education

# OpenProtocols

- Enables others to reproduce experiments
- Protocols are precisely described and freely available to everyone
- DOIs can be assigned to protocols
- Protocols.io is a platform to support the exchange of protocols
- Example: gDNA extraction protocol



<https://www.protocols.io/welcome>

This screenshot shows a detailed view of a protocol page on protocols.io. The page title is 'Plant DNA extraction and preparation for ONT sequencing'. It includes author information (Boas Pucker), date (Mar 21, 2020), and a DOI (doi.org/10.17504/protocols.io.bcvyiw7w). The page has sections for 'Steps', 'Warnings', 'Materials', 'Metadata', and 'Metrics'. The 'Abstract' section describes the protocol as a CTAB-based protocol for extracting genomic DNA from a wide range of plant species. It mentions successful application for genome sequencing of Arabidopsis thaliana, Beta vulgaris (sugar beet), Brassica napus (rapeseed/canola), Dioclea dumetorum (yams), Hedyotis umbelligerum moses, Vitis vinifera (grapevine), and Trifoliate yam (Dioclea dumetorum) using long read sequencing. References at the bottom cite Stadelj et al. (2020) and Pucker et al. (2019).

doi:10.17504/protocols.io.bcvyiw7w

# How to submit a protocol?

- Protocol submission:
  - Create a new protocol (can remain private; 5 in free version)
  - Assign a DOI to make it citable
  - Make protocol public
- PDF submission and conversion for \$50 (service fee)

The screenshot shows the protocols.io website interface. At the top, there's a file upload dialog with options like 'New folder', '+ New protocol', and 'Upload files'. Below the dialog, a banner reads 'A perfect fit, for any need' and 'From chemistry to computational workflows. Create a protocol that fits any need.' There are six categories of protocols displayed in a grid:

- Generic protocol
- Biology protocol
- Computational Workflow
- Chemistry method
- Collection of methods
- Document

Below the categories, there's a 'NEW STEP' button and a note: 'If you have an existing protocol you can [insert steps from a text file](#)'. On the right side, there's a sidebar with various experimental parameters and units, such as Amount, Concentration, Temperature, Duration, Protocol, Document, Equipment, Reagent, Citation, Note, Safety Information, Expected Result, Ge Coordinates, Configuration, Smart Component, Shaker, Gels, pH, Cost, Pressure, Thickness, and Relative Humidity.

# Creative Commons Licenses

- CC0: creative commons (no restrictions)
- CC BY: no restrictions, but name authors
- CC BY-SA: name authors and share results under same license
- CC BY-NC: name authors; only non-commercial use
- CC BY-NC-SA: name authors; only non-commercial use; share under same license

# License stacking

- What happens if we combine different data sets?
- CC0 + CC BY
- CC0 + CC NC-SA
- CC BY + CC BY-NC
- ...

# Software licenses

- MIT: leanest licence (everything is possible)
- Apache: similar to MIT, but lengthy
- GPL (General Public License): ensures that derived work remains open
- BSD (Berkeley Software Distribution): similar to MIT, but more cases specified

License	Commercial use	Distribution	Modification	Patent use	Private use	Disclose source	Licensing and copyright notice	Network use in distribution	Same license	State changes	Liability	Trademark use	Warranty
BSD Zero Clause License	●	●	●	●	●						●	●	●
BSD 2-Clause License	●	●	●	●	●						●	●	●
BSD 2-Clause Clear License	●	●	●	●	●						●	●	●
BSD 2-Clause "New" or "Revised" License	●	●	●	●	●						●	●	●
BSD 3-Clause "New" or "Revised" License	●	●	●	●	●						●	●	●
Boost Software License 1.0	●	●	●	●	●						●	●	●
Creative Commons Attribution 4.0 International	●	●	●	●	●						●	●	●
Creative Commons Attribution-NonCommercial 4.0 International	●	●	●	●	●						●	●	●
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International	●	●	●	●	●						●	●	●
Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International	●	●	●	●	●						●	●	●
CC0 License	●	●	●	●	●						●	●	●
CERN Open Hardware License Version 2.1	●	●	●	●	●						●	●	●
CERN Open Hardware License Version 2.1 - Strong	●	●	●	●	●						●	●	●
CERN Open Hardware License Version 2.1 - Weakly Reciprocal	●	●	●	●	●						●	●	●
Educational Software License v0.9	●	●	●	●	●						●	●	●
EPIC Public License 1.0	●	●	●	●	●						●	●	●
European Union Public License 1.1	●	●	●	●	●						●	●	●
European Union Public License 1.2	●	●	●	●	●						●	●	●
GNU General Public License v2.0	●	●	●	●	●						●	●	●
GNU General Public License v3.0	●	●	●	●	●						●	●	●
GNU Lesser General Public License v3.0	●	●	●	●	●						●	●	●
MIT No Attribution	●	●	●	●	●						●	●	●
MIT License	●	●	●	●	●						●	●	●
Mozilla Public License 2.0	●	●	●	●	●						●	●	●
Open Source Initiative OSI License	●	●	●	●	●						●	●	●
Microsoft Public License	●	●	●	●	●						●	●	●
Mozilla Public License	●	●	●	●	●						●	●	●
Mulan Permissive License - Version 2	●	●	●	●	●						●	●	●
University of Illinois Open Source License	●	●	●	●	●						●	●	●
Apache Commons Open Office License v1.0	●	●	●	●	●						●	●	●
BSL Open Font License 1.1	●	●	●	●	●						●	●	●
Open Software License 3.0	●	●	●	●	●						●	●	●
PostgreSQL License	●	●	●	●	●						●	●	●
The Unlicense	●	●	●	●	●						●	●	●
Unlicense	●	●	●	●	●						●	●	●
Do What The F*ck You Want To Public License	●	●	●	●	●						●	●	●
40s License	●	●	●	●	●						●	●	●

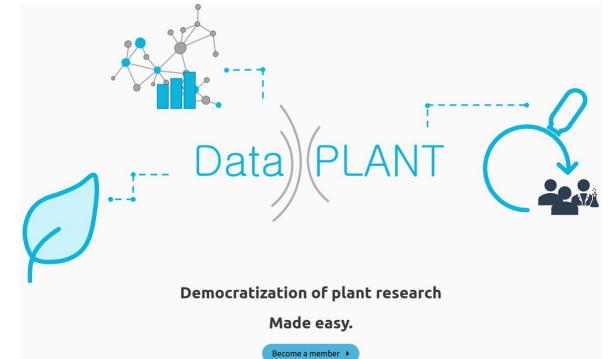
<https://choosealicense.com/appendix/>

# Which license would you select and why?

- A script to analyze a gene family?
- A table with species observed in a particular forest?
- FASTQ files of a RNA-seq project?
- Genome sequence and the corresponding annotation?
- $K_M$  value and  $V_{max}$  value of an enzyme?
- A protocol for efficient transformation of a plant species?

# nfdi4plants

- NFDI = Nationale Forschungsdaten Infrastruktur e.V.
- Provide sustainable, annotated data management platform
- Pave way to pure data publications with research context
- Omics and imaging at petabyte size
- ARC = annotated research context
- Add user-oriented services to existing IT infrastructure (make submissions easy)

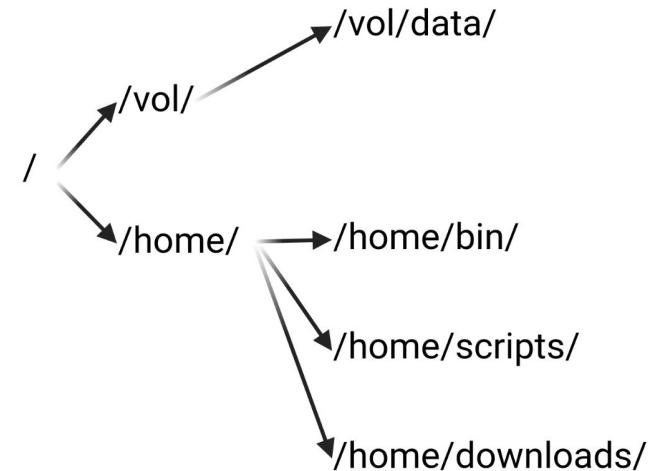


# How to structure your data?

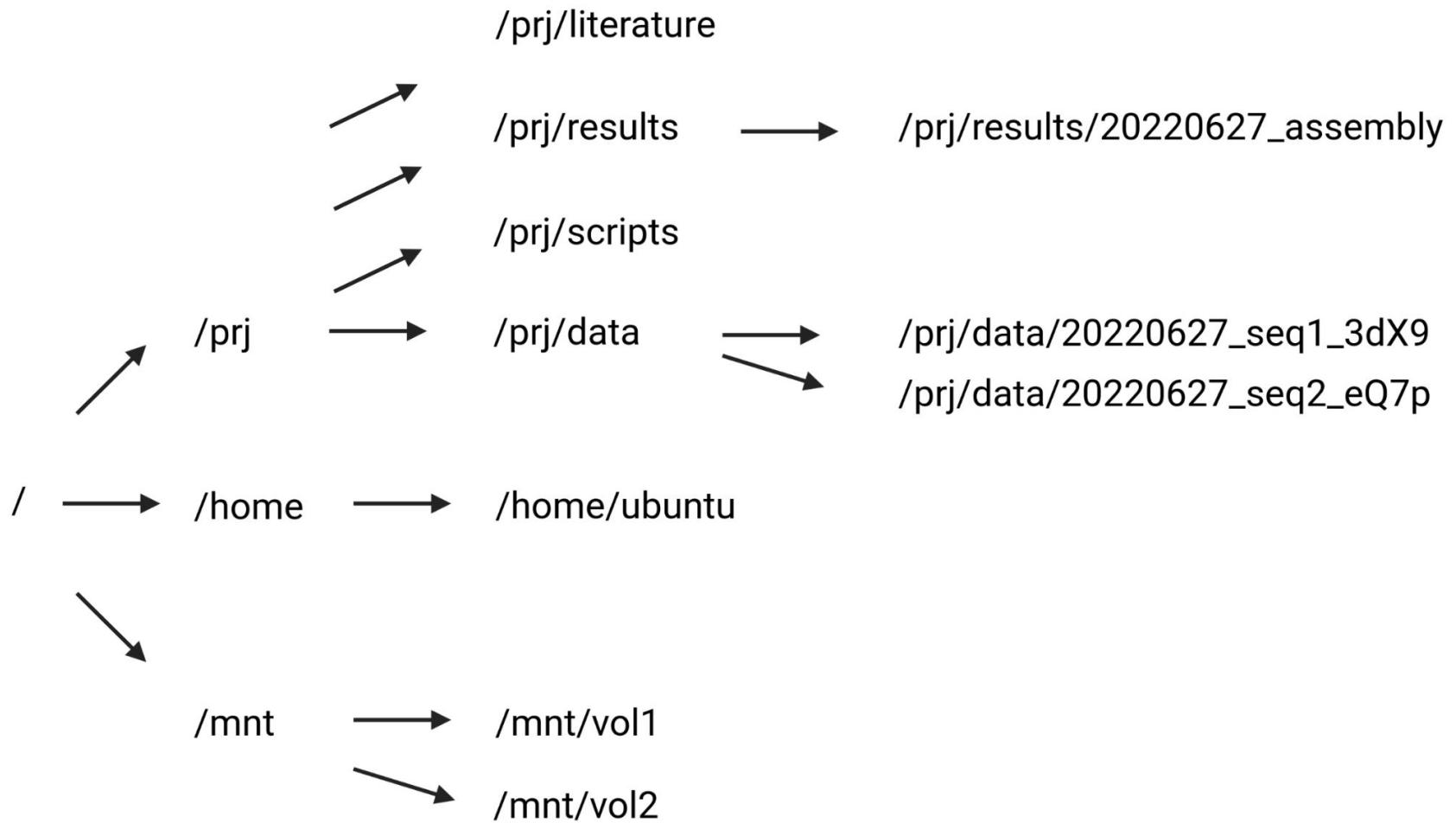
- Document every step (e.g. in a README)
  - Origin of data sets; versions of tools, parameters of analyses
- Keep raw data sets separated from scripts and results
- Sort data by project
- Structure analysis related data sets/results by date

# Linux file system structure

- Linux (Ubuntu) is operation system of choice for bioinformatics
- Hierarchical structure of directory ('/' is basis)
- Separation of tool installation and data sets



# Example: Linux file system



# What do you know about file names?



# Naming files

- File names should be informative and concise
- Never use ‘new’ or ‘final’; use version numbers instead
- Use date as file name prefix (e.g. 2022-06-27 or 20220627)
- Never use spaces in file or folder names (underscore or minus as replacement)
- Include your initials as suffix in collaborative projects

# Documentation

- Document as much as possible
- Others must be able to repeat your experiments/analyses
- Document dates of data acquisition and processing
- Document all steps of data processing
- Document versions and parameters of applied tools

# Backup strategy

- At least 3 copies
- Storage at different sites
- Update backups regularly (e.g. daily)
- Storage options:
  - Internal SSD or hard drive
  - External hard drive
  - Cloud (NextCloud, Dropbox, gdrive, AWS, ...)
  - Institutional tape storage



# Zenodo

- Data are permanently stored in CERN's data centre
- OpenAIRE allows everyone to engage in open science
  - socio-technical infrastructure for scholarly communication
- DOI makes entries citable
- Uploads are published immediately
- Integration with GitHub possible

The screenshot shows the Zenodo homepage with the following sections:

- Featured communities:** National COVID Cohort Collaborative (N3C) is highlighted, showing its description and a "View" button.
- Recent uploads:** PyGMT: A Python interface for the Generic Mapping Tools (April 11, 2022) and pyspeckit/pyspeckit: Release 0.1.25 - pre-v1.0 to test for paper (March 27, 2022) are listed with their details and "View" buttons.
- Why use Zenodo?** A list of reasons includes:
  - **Safe** – your research is stored safely for the future in CERN's Data Centre for as long as CERN exists.
  - **Trusted** – built and operated by CERN and OpenAIRE to ensure that everyone can join in Open Science.
  - **Citable** – every upload is assigned a Digital Object Identifier (DOI), to make them citable and trackable.
  - **No waiting time** – Uploads are made available online as soon as you hit publish, and your DOI is registered within seconds.
  - **Open or closed** – Share e.g. anonymized clinical trial data with only medical professionals via our restricted access mode.
  - **Versioning** – Easily update your dataset with our versioning feature.
  - **GitHub integration** – Easily preserve your GitHub repository in Zenodo.
  - **Usage statistics** – All uploads display standards compliant usage statistics

# Dryad

- International open-access repository
- Suitable for large data sets, but also an option for software
- CC0 licence makes everything completely reusable by other
- DOIs make entries citable



The Dryad website homepage features a top navigation bar with links for Search, Explore Data, About, Help, and Login. Below the header is a banner with the text "Make the most of your research data" and a background image of a person handling scientific specimens. A call-to-action button labeled "Submit Now" is visible on the right.

**How it works**

- Login:** Use your ORCID. If your institution is a Dryad member, connect to your existing credentials.
- Submit:** Whether or not your data are related to an article, upload your data files and receive a citable DOI.
- Review:** Our curators will check through your submission to ensure the data are usable. They may contact you with advice or questions.
- Cite:** Cite and promote your data publication!

**Why use Dryad?**

- Any field. Any format.** Submit data in any file format from any field of research. Share all of the data from a project in one place.
- Quality control and assistance.** Our curators will check your files before they are released, and help you follow best practices.
- Straightforward compliance.** Submit your data to satisfy publisher and funder requirements for preservation and availability with a minimum of effort. We work directly with many publishers -- including Wiley, The Royal Society, and PLOS -- to streamline the process.
- Community-led.** Dryad is a nonprofit membership organization that is committed to making data available for research and educational reuse now and into the future. Modest Data Publishing Charges help ensure our sustainability.

# Linux/Ubuntu

- Freely available
- Regular updates
- Graphical user interface possible
- Mouse-wheel-copy-option
- Installation of tools via command line



# Summary

- Systems Biology & Omics (Genomics, Transcriptomics, Proteomics, Metabolomics)
- Bioinformatics tools and languages
- Data Life Cycle, potential for data upcycling, open science principles
- Data management: files, folders, and backups
- Introduction to Linux (Ubuntu)
- Introduction to the de.NBI cloud

# Time for questions!



# Practical part: outline day1

- How to get help from an AI (and the limitations)
- Jupyter Notebook
- Working in a virtual machine
- Installing computational tools
- Finding usage of computational tools
- Transferring files (filezilla, scp)

# Getting AI support

- Available AIs: ChatGPT, BingChat, Bard/Gemini, (TUBS AI tool box = ChatGPT3.5)
- AI use requires precise description of problem
- Specify all important details to obtain best results
- Check plausibility of returned output
- AI cannot be responsible for output & plagiarism is possible

# ChatGPT3.5



8

You

How to generate a plot in python that visualizes a list of values?



ChatGPT

To generate a plot in Python that visualizes a list of values, the most commonly used library is `matplotlib`. Here's a step-by-step guide on how to do it:

## 1. Install Matplotlib (if you haven't already):

```
bash
```

Copy code

```
pip install matplotlib
```

## 2. Generate a Simple Plot:

Here's a basic example to plot a list of values.

```
python
```

Copy code

```
import matplotlib.pyplot as plt

# Sample data
values = [1, 2, 3, 4, 5]

# Create a plot
plt.plot(values)

# Show the plot
plt.show()
```

## 3. Customize the Plot:

You can customize various aspects of the plot like adding titles, labels, changing colors, line styles, etc.

```
python
```

Copy code

```
import matplotlib.pyplot as plt

# Sample data
values = [1, 2, 3, 4, 5]
labels = ['A', 'B', 'C', 'D', 'E']

# Create a plot
plt.plot(labels, values, marker='o', linestyle='--', color='g', label='Series 1')

# Add titles and labels
plt.title('Sample Plot')
plt.xlabel('Categories')
plt.ylabel('Values')

# Show legend
plt.legend()

# Show the plot
plt.show()
```



# ChatGPT limitations

- Account required
- Refuses to report scientific publications (used to make them up)
- Refuses to report sequences
- Quality depends on input data and knowledge is outdated
- ChatGPT3.5 performs above average human performance in many cases
- ChatGPT4.0 is available now

# Bing Chat

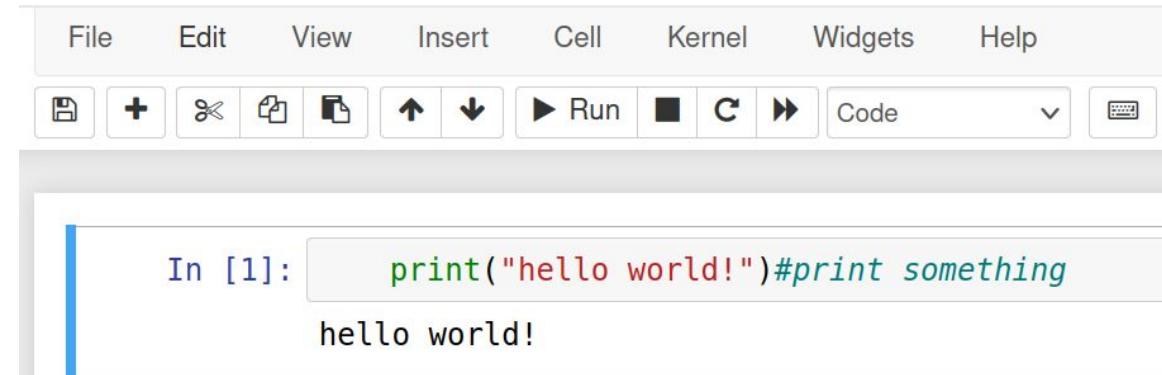
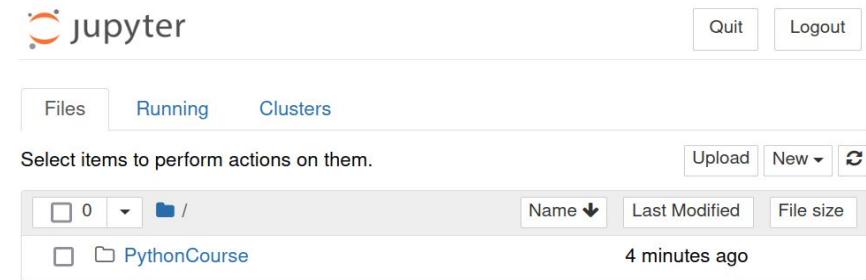
- Provides summary of search results
- Requires Microsoft Edge as web browser
- Account required

# Installing Jupyter Notebook

- Linux:
  - \$ sudo apt update
  - \$ sudo apt install python3-pip python3-dev
  - \$ mkdir python\_course
  - \$ cd python\_course
  - \$ virtualenv python\_course
  - \$ source python\_course/bin/activate
  - \$ pip install jupyter
  - \$ jupyter notebook
- Windows:
  - 1) Install Anaconda  
(<https://www.geeksforgeeks.org/how-to-install-anaconda-on-windows/>)
  - 2) Install Python3 through Anaconda
  - 3) Install Jupyter through Anaconda  
(<https://www.geeksforgeeks.org/how-to-install-jupyter-notebook-in-windows/>)
- Mac:
  - See instructions above
- ChromeOS:
  - JupyterLab (online): <https://jupyter.org/>

# Starting Jupyter Notebook

- Start Jupyter Notebook
- Create a new file:
  - ‘New’
  - Python3 ipykernel
  - Change name
- Python code is written and executed in Jupyter Notebook



# Adding elements to a Jupyter Notebook

- Code: you know this already
- Header: structure your work/documentation
- Markdown: comments with specific formatting

A screenshot of a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Run, along with a cell type selector and a Markdown dropdown. The main area shows two cells. The first cell, labeled 'In [1]:', contains the Python code `print("hello world!")#print something`. The output of this cell is 'hello world!'. The second cell, which is a header cell, contains the text '# My first Python script'.

A screenshot of a Jupyter Notebook interface. The top menu bar and toolbar are identical to the previous screenshot. The main area shows three cells. The first cell is a header cell containing '# My first Python script'. The second cell is a comment cell containing the text 'My \*\*comments\*\* to remember \_everything\_.' The third cell, labeled 'In [1]:', contains the Python code `print("hello world!")#print something`. The output of this cell is 'hello world!'. The entire content of the second cell is highlighted in blue.

# de.NBI cloud

- Virtual machine (VM) for data analysis
- Accounts are required for access (ORCID or TUBS SSO for login)
- Addition to project required for access
- User create pair of private and public keys for authentication



de.NBI ELIXIR-DE Services Training de.NBI Cloud News



## Compute Power for Your Project

In life sciences today, the handling, analysis and storage of enormous amounts of data is a challenging issue. For example, new sequencing and imaging technologies result in the generation of large scale genomic and image data. Hence, an appropriate IT infrastructure is crucial to perform analyses with such large datasets and to ensure secure data access and storage. In addition, it is difficult to directly compare result data that have been processed at different sites, due to a lack in standardization of workflows. The de.NBI cloud is an excellent solution to enable integrative analyses for the entire life sciences community in Germany and the efficient use of data in research and application.

To a large extent, de.NBI will close the gap of the missing computational resources for researchers in Germany. A federated de.NBI Cloud concept and infrastructure leads to the reduction in overall infrastructure and operational costs.

[Click here to enter the de.NBI Cloud Portal](#)

Cloud Access Portal  
de.NBI Cloud Flyer  
de.NBI Cloud Poster  
Cloud Training Courses

### Get access to the Cloud

- 1.) Register for an [ELIXIR Account](#) and apply for membership in the de.NBI virtual organisation.
- 2.) Log in to our [de.NBI Cloud portal](#) to manage your projects and project members.

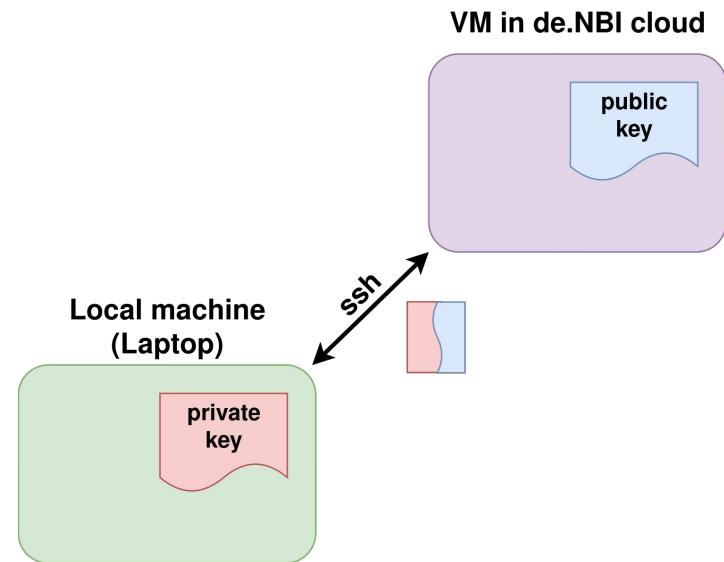
# Working in a virtual machine

- Generate key pair (private key file download starts automatically)

- Starting a VM through de.NBI cloud web interface

- Connection to VM: ssh

```
ssh -i /path/to/key ubuntu@123.456.78.666 -p 12345
```



# Linux introduction

Connection to virtual machine (VM):

```
$ ssh -i /path/to/private_key ubuntu@123.133.7.45 -p 1234  
(base) ubuntu@agilezuse-10552:~$
```

'\$' is used to indicate that following text needs to go into terminal

# indicates comment (should not be transferred into terminal)

Frequent issues:

- 1) Path to private key file not correct
- 2) Private key file too public

# Transferring files

- Filezilla: graphical user interface for file transfer protocols
  - <https://filezilla-project.org/>
- Scp (secure copy): command line file transfer method
- Wget: command line file transfer method
  - [https://ftp.gnu.org/old-gnu/Manuals/wget-1.8.1/html\\_mono/wget.html](https://ftp.gnu.org/old-gnu/Manuals/wget-1.8.1/html_mono/wget.html)
- Rsync: sophisticated file transfer method that avoids redundant transfers
  - <https://wiki.ubuntuusers.de/rsync/>

# Permissions

- Files can have different permissions:
  - Read (r)
  - Write (w)
  - Execute (x)
- Users have full permissions to edit their own files
- Downloaded files are usually not executable without adjustment
- `chmod XXX <FILE_NAME>...` can be run to change file permissions

# Basic linux commands

- Moving through the folder structure:
  - \$ cd /full/path/to/folder #change into specific folder
  - \$ cd subfolder #change into subfolder
  - \$ cd .. #change into parent directory
- Checking content of a folder:
  - \$ ls #shows content of current folder
  - \$ ls -lh #shows more details
  - -l triggers display of additional details
  - -h human readable
  - -a show also hidden files

# Basic linux commands 2

- **Move file:** `mv /path/to/source/file.txt /path/to/dest/file.txt`
- **Copy file:** `cp source_file.txt destination_file.txt`
- **Generate new folder:** `mkdir <folder_name>`

# Exercises: 1-1

- Register for a de.NBI cloud account: <https://www.denbi.de/cloud>
- Form a group of four persons to share one virtual machine & start VM
- Use your new AI skills to find the command to mount a drive as /vol/data
- Generate a members subfolder on this new drive
- Generate your personal folder within the members folder
- Generate a data folder for raw input data
- Generate a bin folder for all your tools

# Mounting disk

- Find the name(path) of the disk: `lsblk`
- Generate a folder that serves as mounting point: `sudo mkdir /vol/data`
- Mount the drive: `sudo mount /dev/vdx /vol/data`
- Change permissions: `chmod 777 /vol/data`
- Check available space: `df /vol/data -h`

# How to install bioinformatics tools

- Install tools from package management (apt-get install)

Installing mafft package on Ubuntu is as easy as running the following command on terminal:

```
sudo apt-get update  
sudo apt-get install mafft
```

- Download a precompiled binary

## Releases

The kallisto GitHub repository is [here](#).

Version	Date	Mac	Linux	Windows	Rock64	Source
Release notes: v0.46.1	October 04, 2019					
Release notes: v0.46.0	June 12, 2019					
Release notes: v0.45.0	November 17, 2018					
Release notes: v0.44.0	January 29, 2018					

- Compile from source

## COMPILING FROM SOURCE

Download the latest [release from](#) and uncompress it

```
# Get latest STAR source from releases  
wget https://github.com/alexdobin/STAR/archive/2.7.10b.tar.gz  
tar -xzf 2.7.10b.tar.gz  
cd STAR-2.7.10b
```

```
# Alternatively, get STAR source using git  
git clone https://github.com/alexdobin/STAR.git
```

## Compile under Linux

```
# Compile  
cd STAR/source  
make STAR
```

For processors that do not support AVX extensions, specify the target SIMD architecture, e.g.

```
make STAR CXXFLAGS SIMD=sse
```

- Install via conda

## Installers



conda install [?](#)

To install this package run one of the following:

```
conda install -c bioconda shasta
```

<https://howtoinstall.co/en/mafft>  
<http://pederlelab.github.io/kallisto/download>  
<https://github.com/alexdobin/STAR>  
<https://anaconda.org/bioconda/shasta>



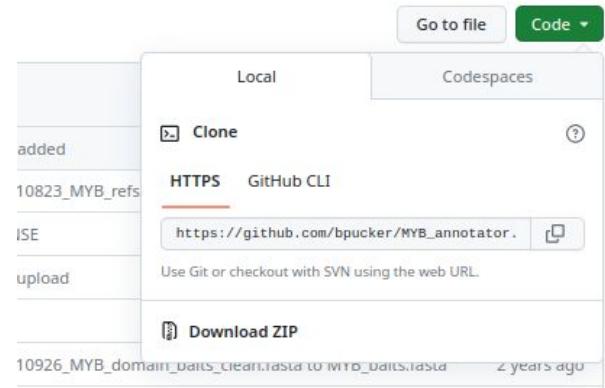
# Retrieving tools from GitHub

- Find GitHub repository corresponding to tool of interest
- Read the installation instructions
- Download single file via “Raw”, “Save as”
- Download everything via “Code” as ZIP archive
- Use git clone to get the content

The screenshot shows the GitHub repository page for `bpucker/MYB_annotator`. The repository has 166 commits and 2 branches. It includes files like `ATHMYBs.fasta`, `MYB_baits.fasta`, and `README.md`. The repository is licensed under the MIT License. It has 13 stars and 4 forks. The `Code` tab is selected, showing the repository's structure. The `About` section describes the tool as performing automatic identification, annotation, and analysis of the MYB gene family in plants. The `Releases` section shows one release (v0.2) with a link to the latest commit. The `Packages` section indicates no packages have been published. The `Languages` section shows Python as the primary language.

# Cloning a GitHub repository

1. Go to GitHub repository website
2. Click “code”
3. Copy URL (HTTPS)
4. Open a terminal
5. Change working directory to the location where you like to have the tool
6. Clone the repository: `git clone <URL>`



# Virtual environments

- Python tool (venv) for dependency management and project isolation
- Conda keeps packages separated and works without administrator rights
- Increase reproducibility of research
- Configure virtual environment:
  - `python3 -m pip install --user --upgrade pip`
  - `python3 -m pip --version`
  - `python3 -m pip install --user virtualenv`
  - `python3 -m venv env`
  - `source env/bin/activate`
  - [here you can do things inside the virtual environment]
  - `deactivate` #to leave the virtual environment again

# Running Python scripts

- Run script or open argument infos:
  - \$ python <SCRIPT\_NAME>
- Run script with arguments:
  - \$ python <SCRIPT\_NAME> --argument1\_name <ARGUMENT1>  
--argument2\_name <ARGUMENT2>
- Scripts show help message if started with insufficient arguments

```
python3 ./KIPES3.py --baits ./flavonoid_baits/ --positions  
.flavonoid_residues/ --out ./ --subject ./croton_red.fasta --seqtype pep --scoreratio 0.3 --simcut  
40.0 --minsim 0.4 --minres 0.0 --minreg 0.0 --possibilities 3 --cpus 1
```

# Running other tools

- Show help message:
  - \$ <NAME\_OF\_TOOL>
  - \$ <NAME\_OF\_TOOL> -h
  - \$ <NAME\_OF\_TOOL> -- help
- Providing arguments is different for each tool:
- Most tools show help message if provided with insufficient/wrong arguments

# Exercises: 1-2

- Install the assembler Shasta
- Install the gene prediction tool BRAKER3
- Install the long read mapper minimap2
- Install the variant caller SVIM2
- Install BUSCO5
- Install KIPES3

# Time for questions!

