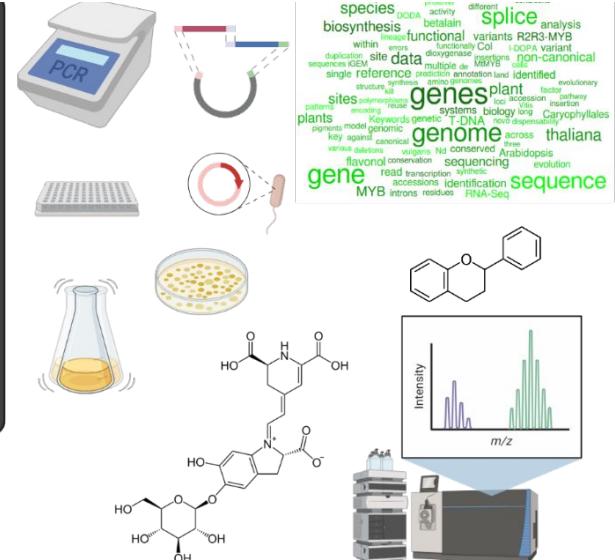
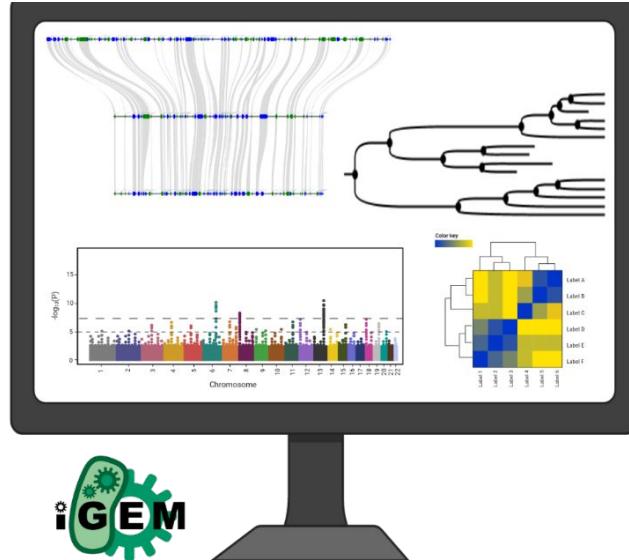
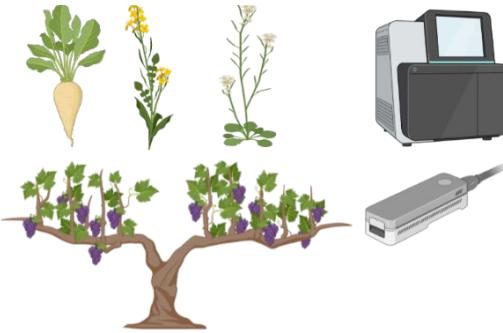




Technische
Universität
Braunschweig



Introduction to (plant) bioinformatics

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

Round of introduction

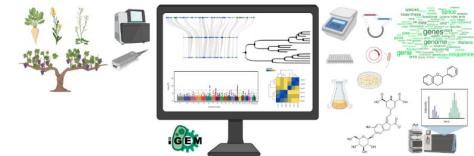
- Name?
- Study program?
- Semester?
- Previous experiences with biochemistry?
- Previous experiences with bioinformatics?
- Expectations?



Boas Pucker

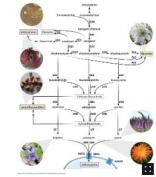
- Biochemistry at HHU Düsseldorf
- (Systems) Biology at Bielefeld University
- Doctoral student (CeBiTec, Bielefeld University)
 - Genomics & Bioinformatics; synthetic biology (iGEM)
- Post doc (Ruhr-University Bochum)
- Post doc (Department of Plant Sciences, Cambridge, UK)
- Plant Biotechnology & Bioinformatics,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

Plant Biotechnology and Bioinformatics (Prof. Pucker)

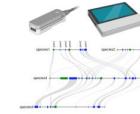


Specialized Metabolism

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.



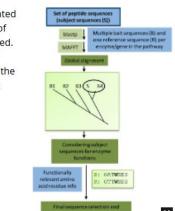
Plant Genomics



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: [Lecture: Grundlagen der Biochemie und Bioinformatik der Pflanzen \(Bio-MB 09\)](#)
 - Skype: (link shared via email)
 - GitHub: <https://github.com/bpucker/teaching>
- Questions: Feel free to ask at any time
- Feedback, comments, or questions: b.pucker@tu-braunschweig.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.

Overview

- 1) Introduction to bioinformatics (today)
- 2) Identification of biosynthesis pathways
- 3) Pathway databases
- 4) Gene expression & co-expression analyses
- 5) Phylogenetic analyses
- 6) Synteny & biosynthetic gene clusters
- 7) Metabolic flux & modeling
- 8) Repetition, questions & quiz

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/>



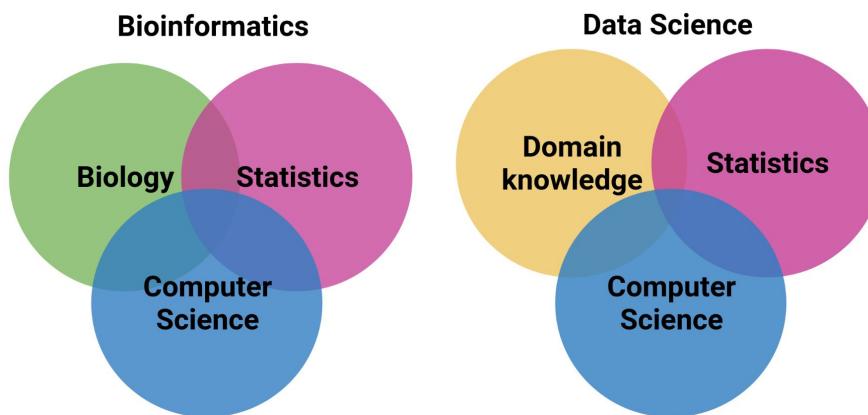
A screenshot of a bioRxiv preprint page. At the top is the bioRxiv logo with the text "THE PREPRINT SERVER FOR BIOLOGY". Below it is a yellow box containing the text "bioRxiv posts many COVID19-related papers. A reminder: they have not been formally peer-reviewed and should not guide health-related behavior or be reported in the press as conclusive." To the right is a "Follow this preprint" button. The main title is "Apiaceae FNS I originated from F3H through tandem gene duplication". Below the title is the author information: Boas Pucker, Massimo Iorizzo, and the DOI: doi: https://doi.org/10.1101/2022.02.16.480750. There is a note that this is a preprint and has not been certified by peer review. Below the author info are social media sharing icons (DOI, ORCID, Mendeley, BioRxiv, Twitter) and a "Preview PDF" button. At the bottom are sections for "Abstract", "Full Text", "Info/History", and "Metrics".

What is bioinformatics?



What is bioinformatics?

- Subdiscipline of biology, statistics, and computer science
- Acquisition, storage, analysis, and dissemination of biological data
- Bioinformatics is a biology-specific data science version

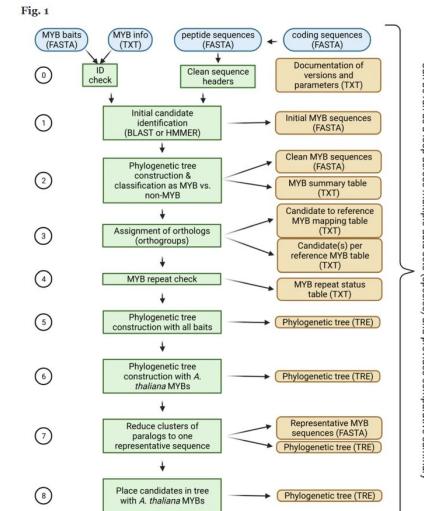
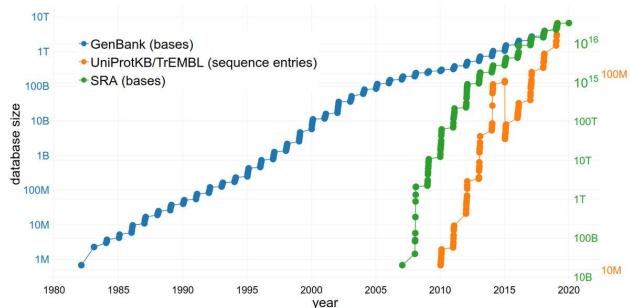
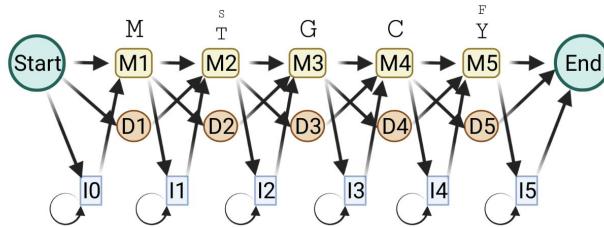


Why do we need bioinformatics?



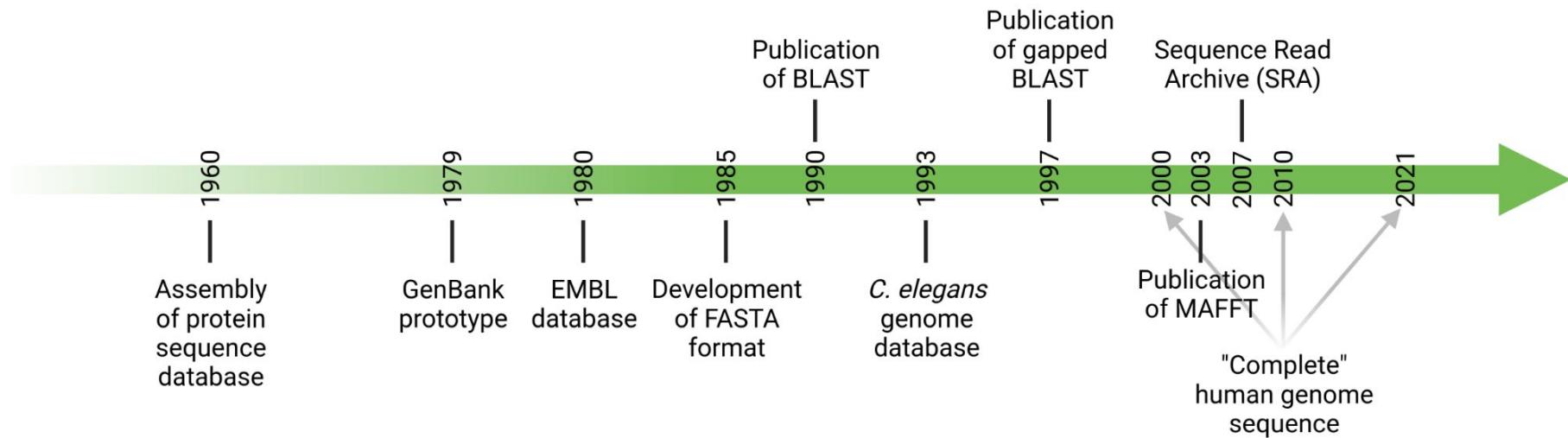
Why do we need bioinformatics?

- Large data sets
- Complex models
- Automatic analyses

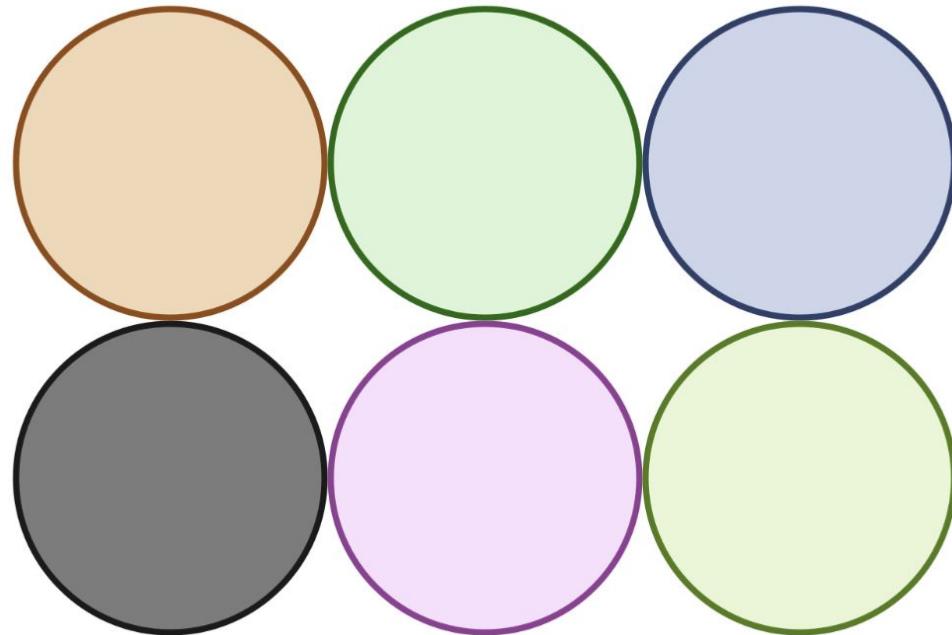


Sielemann & Hafner et al., 2020: 10.7717/peerj.9954
Pucker, 2022: 10.1186/s12864-022-08452-5

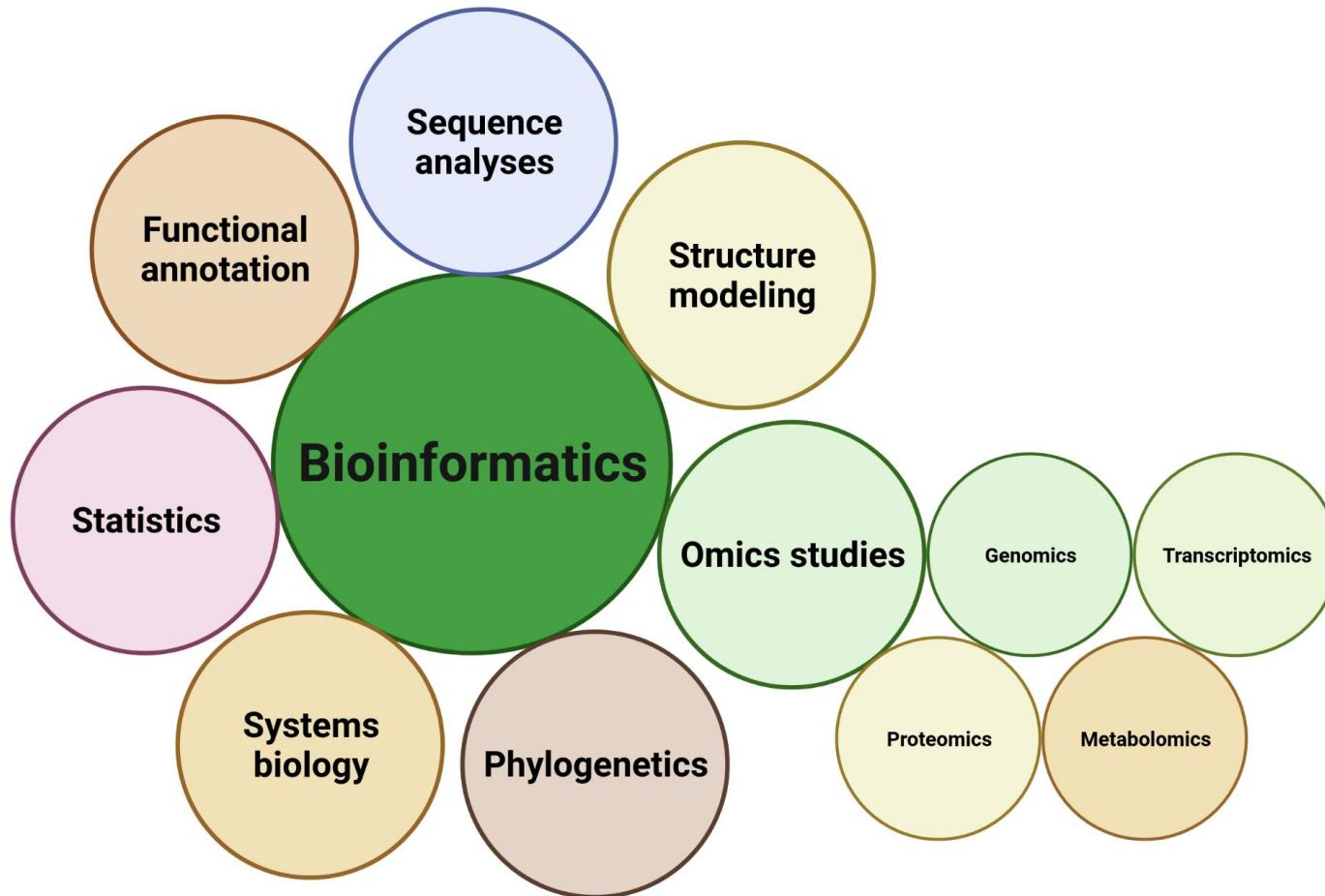
History of bioinformatics



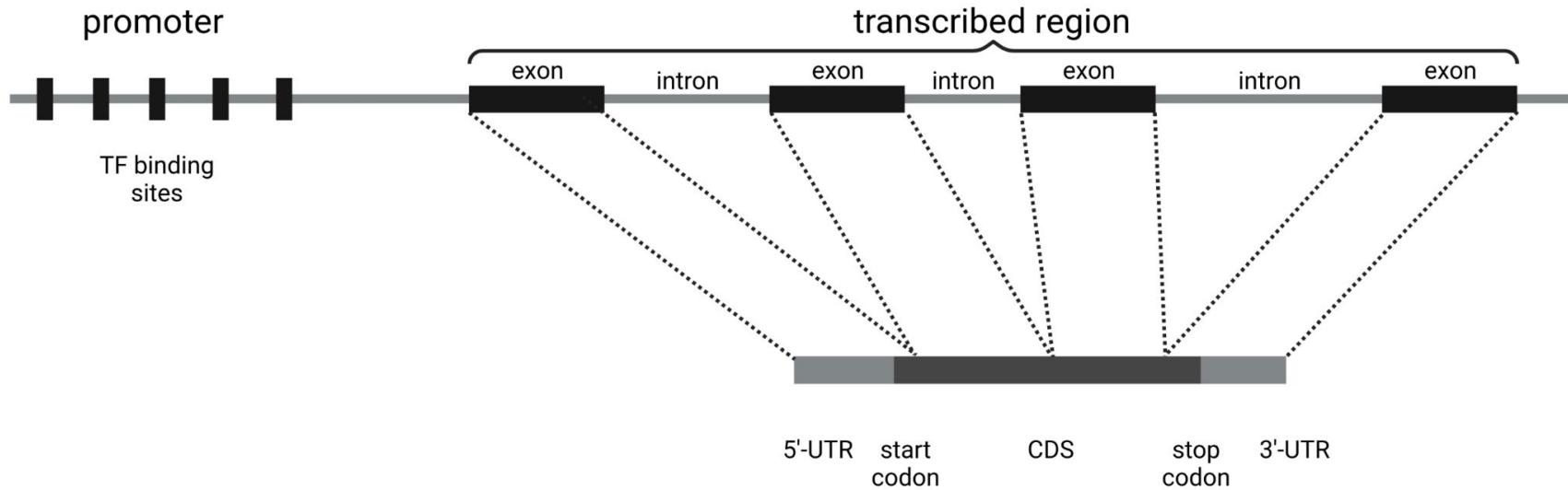
What are the different (sub)fields?



What are the different (sub)fields?



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
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		TPSVQCEIRDAQRNSPATILAIGTANPANE MYQAEYPDFYFRVTKSEHMKELKQMC T S++EIR AQR++GPATILAIGTA P_N +Y0++PD+YFRVTKSEHM ELK+KFKRMC	62			BvCHS AtCHS
Sbjct 2		TSSLEEIRKAQRADGPATILAIGTATPPNCVYQSDFPDYYFRVTKSEHMTTELKEKFKRMC	61			
Query 63		DNSMIKKRYMHVTEELLEENPHLCDFNASSLTDRODILATEVPKLGKEAAVKAIKEWGP D SM1 KRYMH+TEE L+E+N+P+C +SSLRD+I+EV+LKEAAVKAIKEWGP	122			BvCHS AtCHS
Sbjct 62		DKSMILKRYMHLTTEFLKENPNMCTYMGSSLDTRODIVSEEVPRLGKEAAVKAIKEWGP	121			
Query 123		RSKITHVIFCTTSGVDMPGADYQTLKLLGLRPSVKRFMLYQOOGCYAGGTVELAKDIAEN +SKITHVI CTTSVGDMPGADYQTLKLLGLRPSV+RFMLYQOOGC+AGGTVELAKD+AEN	182			BvCHS AtCHS
Sbjct 122		DSKITHVIMCFTTSGVDMPGADYQTLKLLGLRPSVRRFMLYQOOGCFAGGTVELAKDIAEN	181			
Query 183		NRGARVLVVAEITAIIICFRGPTQIHLDSMVQALFfdqagagaviygapDDESIERPIFOLV NRGARVLVVC+EIT ICFRGPT+ HLDMSMVQALFGDGAGA+IVGADPDESIERP+F+v	242			BvCHS AtCHS
Sbjct 182		NRGARVLVVCSEITAICFRGPTETHLDMSMVQALFGDGAGALIVGADPDESIERPLFKMV	241			
Query 243		WAAQTLILPSEGAIIDGLHREVGLTFHLLKDVPGLISKNIEKALVEAQPIGIDDWSNSIFW WAAQTLILPSEGAIIDGLHREVGLTFHLLKDVPGLISKNIEKALVEAQPIGIDDWSNSIFW	302			BvCHS AtCHS
Sbjct 242		WAAQTLILPSEGAIIDGLHREVGLTFHLLKDVPGLISKNIEKALVEAQPIGIDDWSNSIFW	301			
Query 303		VAHPGGRAILDDVESKGLKEDKLKTTFRVLSEYGNMSSACVLFILDEMRKRAMKEGMAT +AHPGG AILD VE+KLGKKE+KL TR+VLSE+GNMSSACVLFILDEMRK+MEG AT	362			BvCHS AtCHS
Sbjct 302		IAHPGGPAILDQVEAKLGLKEEKTATRNVLSEFGNMSSACVLFILDEMRKKSMEGKAT	361			
Query 363		TGEGLEWGVLFGFGPGLTVETVMLHSV 390 TG+GL+WGVLFGFPGPGLTVETV+LHSV	389			BvCHS AtCHS
Sbjct 362		TGDLWGVLFGFPGPGLTVETVVLHSV	389			

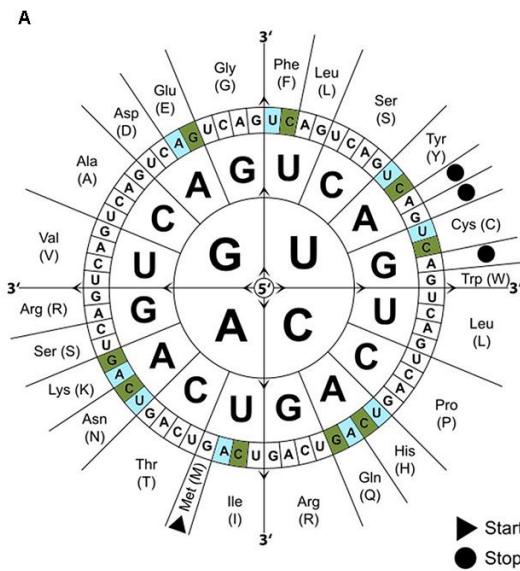
CLUSTAL format alignment by MAFFT FFT-NS-i (v7.487)

```
M--ATPSVQCEIRDAQRNSPATILAIGTANPANE MYQAEYPDFYFRVTKSEHMKELKQK
MVMAGASSLDEIRQAQRADGPAGILAIAGTANPENHVLQAEYPDYYFRITNSEHMTDLKEK
* . . : * : *** : * : * : * : * : * : * : * : * : * : * : * : * : * : *
FKRMCNDNSMIKKRYMHVTEELLEENPHLCDFNASSLTDRODILATEVPKLGKEAAVKAIK
FKRMCNKSTIRKRHMHLTEEFLKLENPHMCAYMAPSLDTRQDIVVVVEVPKLGKEAAVKAIK
***** : * : * ; ; * : * : * : * : * : * : * : * : * : * : * : * : * : *
EWGQPRS KITHVIFCTTSGVDMPGADYQLT KL LGRPSV KR FM LYQOOG CYAGGT VRL A
EWGQPKSKITHVVFCTTSGVDMPGADYQLT KL LGRPSV KR LM LYQOOG CFAGGT VRL A
***** : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : *
DIAENN R G A R V L V V C A E I T I I C F R G P T Q I H L D S M V G Q A L F G D G A G A V I V G A D P D E S I - E R
DIAENN R G A R V L V V C S E I T A V T F R G P S D T H L D S L V G Q A L F S D G A A L I V G S D P D T S V G E K
*:***** :****:**** : **** : * : * : * : * : * : * : * : * : * : * : *
PIFQLVWAQA QT IL P G S E G A I D G H L R E V G L T F H L L K D V P G L I S K N I E K A L V E A F Q P I G I D D
PIFEMVSAQA QT IL P S D G A I D G H L R E V G L T F H L L K D V P G L I S K N I V K S L D E A F K P L G I S D
*** : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : *
WNSIFWVAHPGGRAILDV D E V S K L G L K E D K L K T T R H V L S E Y G N M S S A C V L F I L D E M R K R A M
WNSLFWIAHPGGPAILDQ V E I K L G L K E E K M R A T R H V L S E Y G N M S S A C V L F I L D E M R K S A
***** : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : * : *
KEGMATTGEGLEWGVLFGFGPGLTVETVMLHSVPTAN
KDGVATGEGLEWGVLFGFGPGLTVETVVLHSV - -
*:***** : * : * : * : * : * : * : * : * : * : * : * : * : * : * : *
```



Sequence analysis III

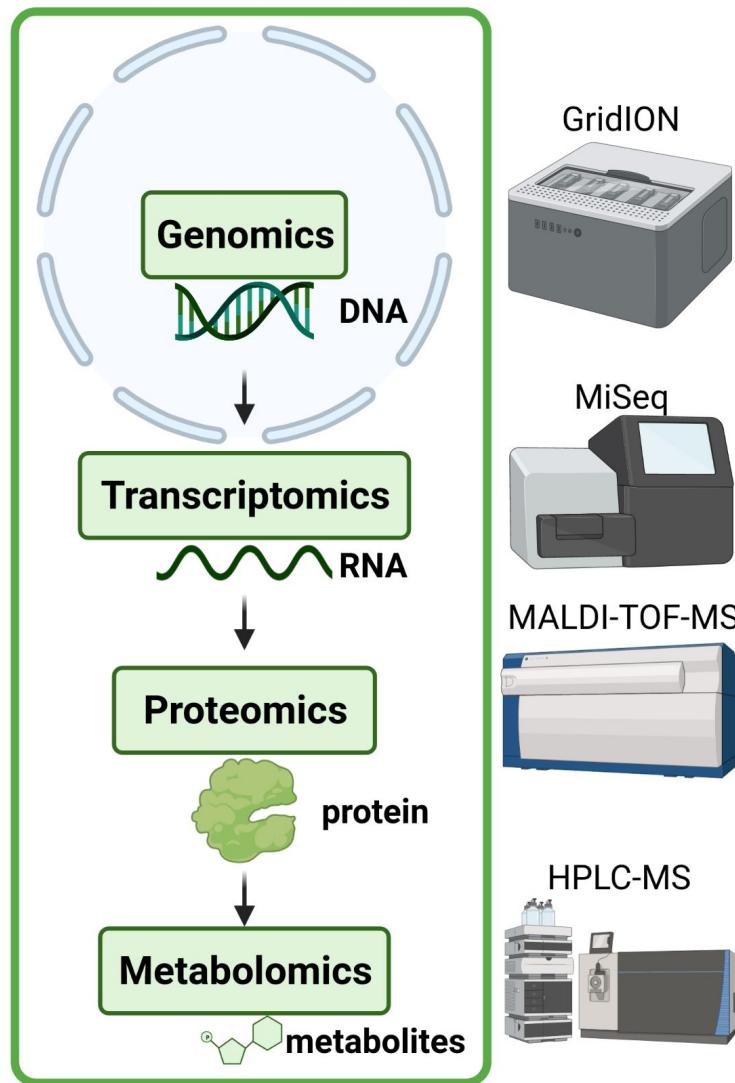
- Codons define amino acids to integrate
- Genetic code is redundant i.e. multiple codons for each amino acid
- Organisms can have a preference for certain codons for a given amino acid
- Optimization of sequences for heterologous expression



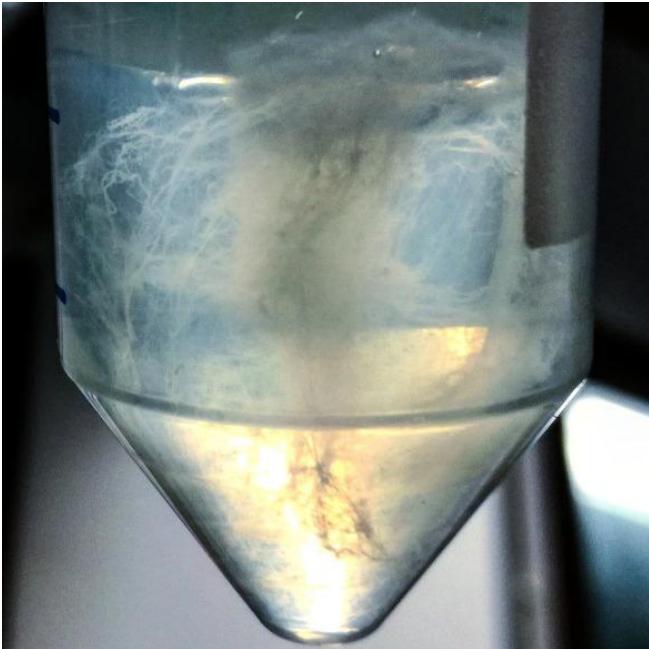
B

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



What is a genome?



Picture by Hanna Schilbert; Pucker, 2020: 10.17504/protocols.io.bcvyiw7w

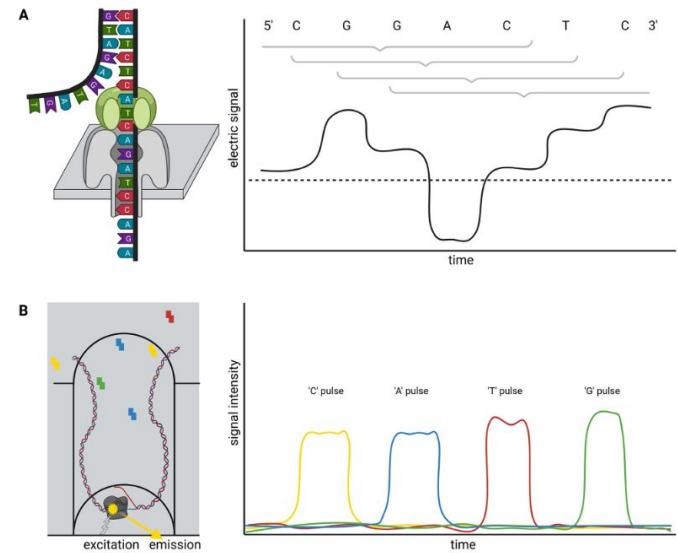


Technische
Universität
Braunschweig

Boas Pucker| Plant Biotechnology & Bioinformatics | MB09-1 | 18

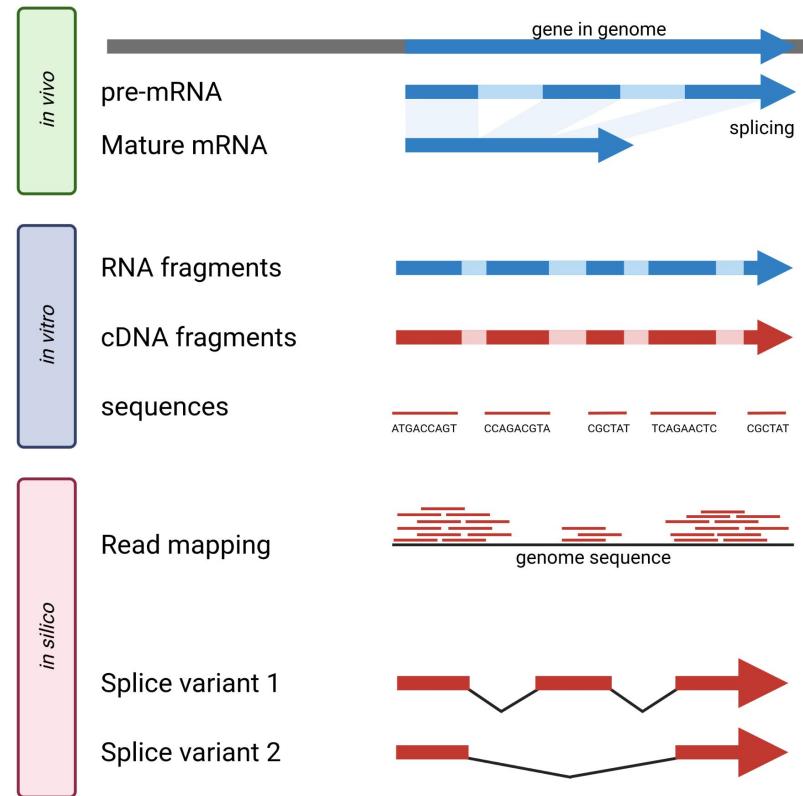
Genomics

- 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, ...
- Comparison of genome sequences
- Handling of large data sets (>1TB) necessary for most projects



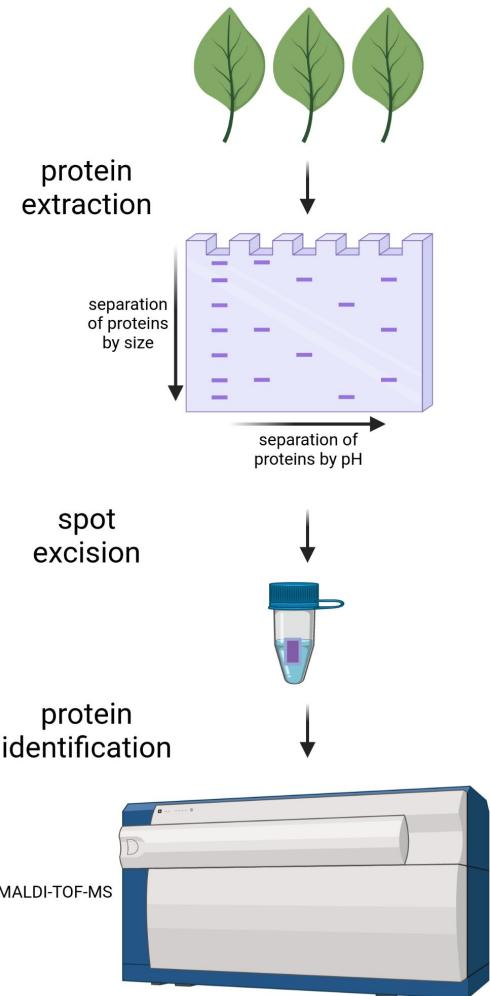
Transcriptomics

- 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,)



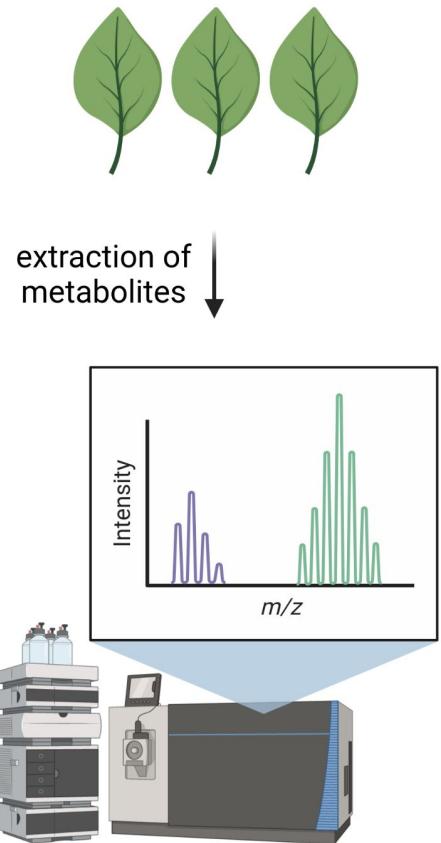
Proteomics

- 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



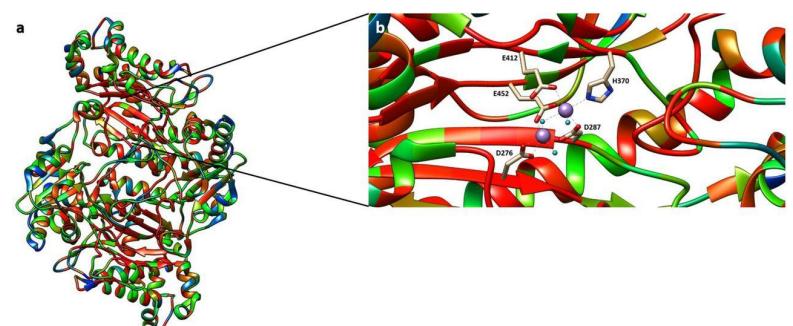
Metabolomics

- 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

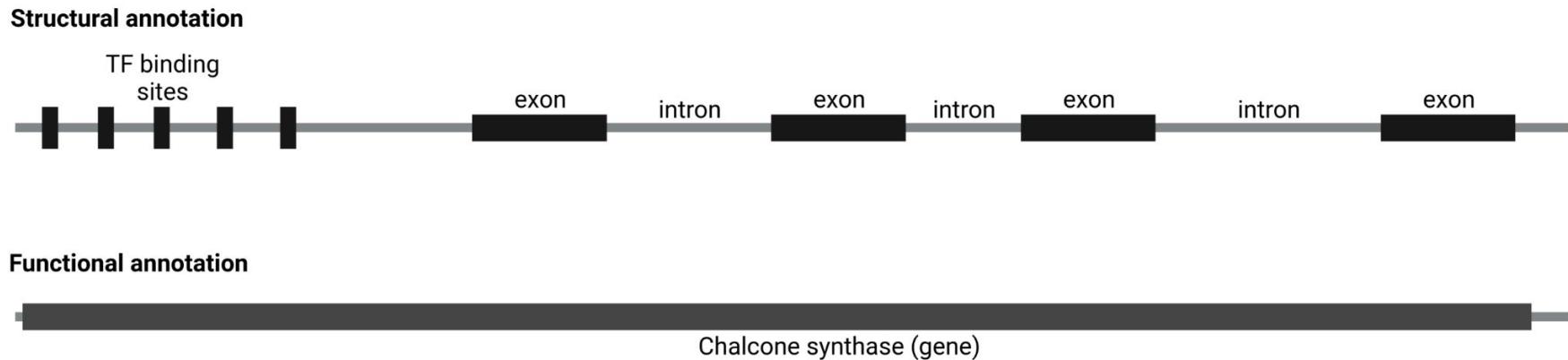
- Modeling of RNA structures
 - Collection of tools: https://molbiol-tools.ca/RNA_analysis.htm
 - BiBiServ: <https://bibiserv.cebitec.uni-bielefeld.de/rna>
 - RNAfold: <http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi>
- Modeling of 3D protein structures
 - Collection of tools:
https://molbiol-tools.ca/Protein_tertiary_structure.htm
 - PredictProtein: <https://predictprotein.org/>
 - AlphaFold2 (10.1021/acs.jcim.1c01114)



Schilbert et al., 2018: 10.1101/423475

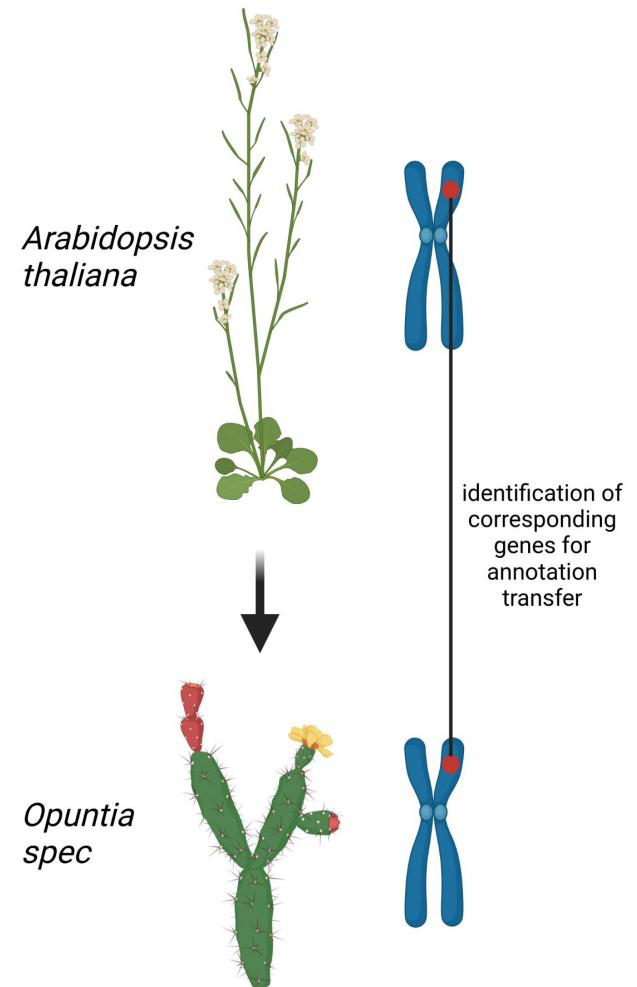
Structural vs. functional annotation

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



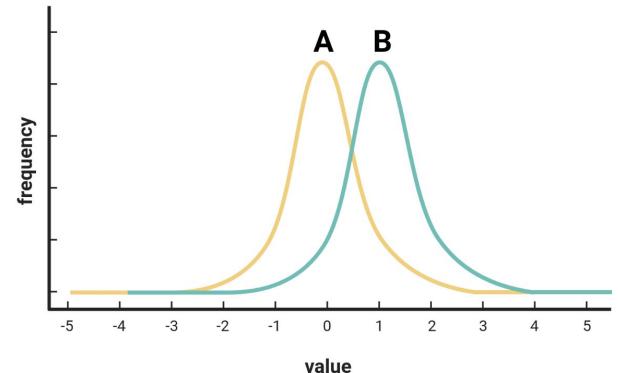
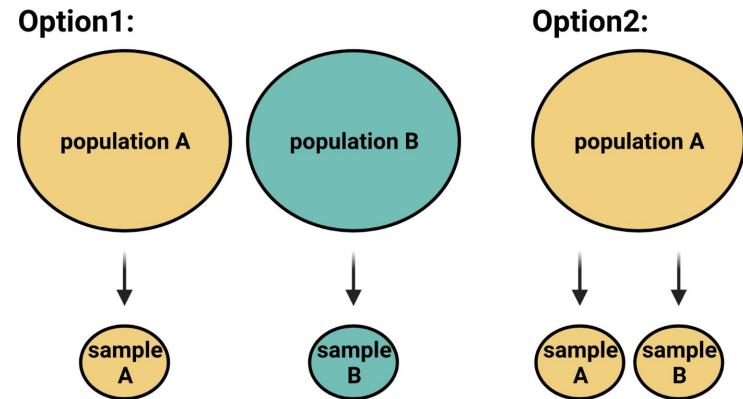
Functional annotation

- 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



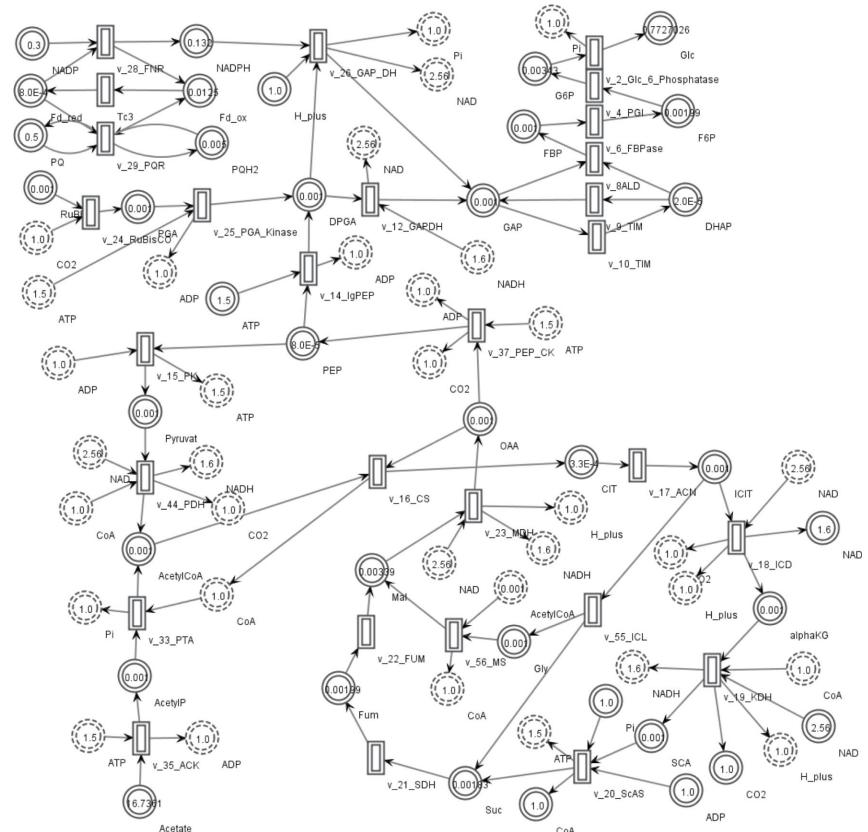
Statistics

- 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

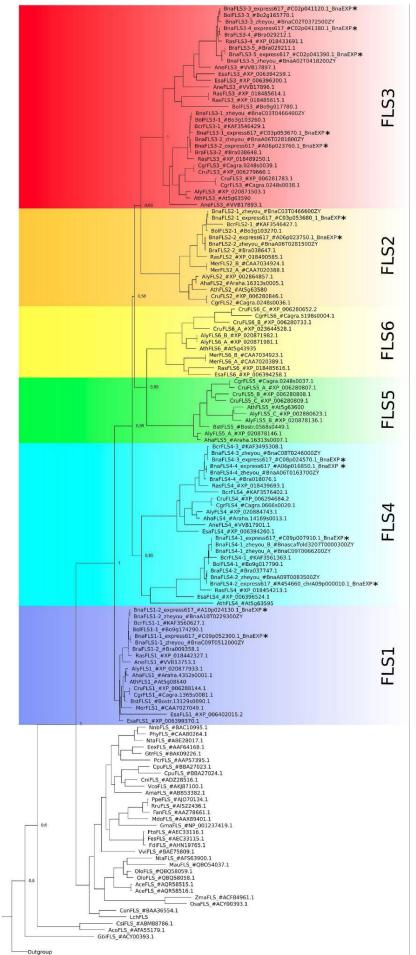
- Description of biological systems with mathematical methods
- Michaelis-Menten kinetics is basis of many metabolic models
- Integration of multiple omics data sets
- Modeling of metabolic flux through pathways
- Example: Modeling of aerobic carbon metabolism in *Chlamydomonas reinhardtii*



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

Phylogenetics

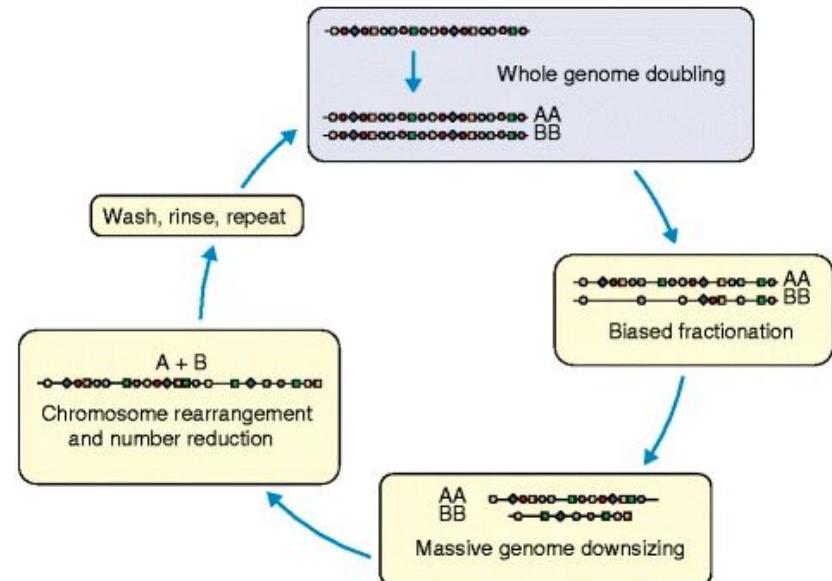
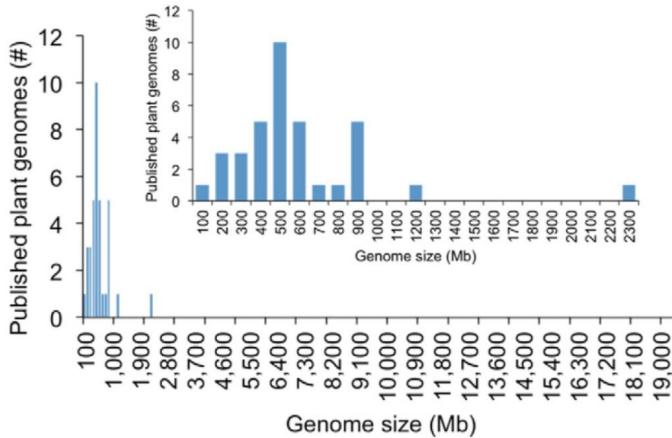
- Analysis of the evolutionary relationships between sequences/species
- Reconstruction of possible models to explain these relationships
- Not limited to sequence similarity, but considered (most likely) ancestral stages



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

Optional
scoreratio

simcut

minsim

minres

minreg

possibilities

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.

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.bot.nat.tu-bs.de/KIPES>

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



KIPES (Knowledge-based Identification of Pathway Enzymes)

Required subject

seotype pep

Optional scenario 0.2
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

Plant Biotechnology and Bioinformatics (Prof. Pucker)
News
Research
BioinfToolServer
Team
Publications
Projects for students
Teaching
IGM
Open Positions
Contact

BioinfToolServer
KIPES
MYB_annotation

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>
- **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>

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

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

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)



How to develop a tool?

1. Identify the problem
2. Precisely describe the problem and break into individual tasks
3. Identify a test data set (positive control)
4. Solve individual tasks
5. Test functionality using positive control
6. Define benchmarking dataset
7. Test functionality based on benchmarking data set
8. Release command line version
9. Add GUI (=Graphical User Interface)?
10. Establish web service?

Tool development - KIPES (example): 1

- Identify the problem:
 - Annotate the flavonoid biosynthesis in a new plant species
- Precisely describe the problem and break into individual tasks:
 - Check and clean input data sets
 - Find genes (CHS, CHI, F3H, FLS, DFR, ...)
 - Provide details about the candidates
 - Collect the CDS and peptide sequences of all players
 - Construct phylogenetic trees of all candidates
 - Analyse gene expression of the candidates

Tool development - KIPES (example): 2

- Identify a test data set (positive control):
 - Flavonoid biosynthesis is well annotated in *Arabidopsis thaliana*
- Solve individual tasks:
 - Check whether user supplied valid sequences
 - Rename sequence names to remove “illegal” characters
 - Identify initial candidates based on BLAST
 - Check these initial candidates by global alignments
 - Screen for presence of functionally conserved amino acids
 - ...

Tool development - KIPES (example): 3

- Test functionality using positive control:
 - Run analysis on *A. thaliana* as technical check
- Define benchmarking dataset:
 - Good representation of actual data sets required (phylogenetic width)
 - Sufficient size to analyze computational requirements / run time
 - Manual validation of results must be possible

Tool development - KIPES (example): 4

- Test functionality based on benchmarking data set:
 - Run analysis of benchmarking data set (repeat with each version)
- Release command line version:
 - Command line version allows quick fixing of bugs
 - Inclusion of additional features possible
 - Assessment of community interest in the tool
 - Tool deployment as part of collaborations
 - Large scale analyses are possible on compute clusters
 - Local data processing has security/safety benefits

Tool development - KIPES (example): 5

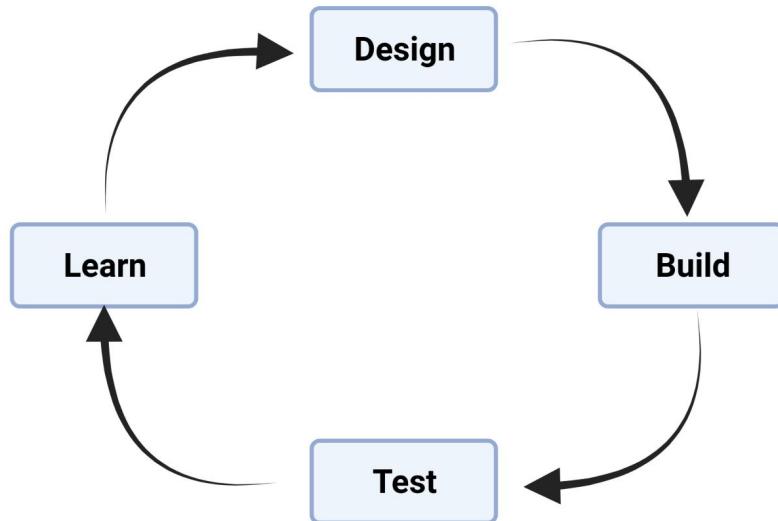
- Add GUI (=Graphical User Interface):
 - Addition of GUI is tedious work
 - Only successful tools justify the additional efforts
 - GUI is only helpful if additional user are expected
 - Still requires installation by (inexperienced) users
- Add web service:
 - Additional work that is only justified by additional benefits (more users/visibility)
 - Requires the availability of computational resources (provided by host)
 - Very convenient to use, but can raise data protection issues

How would you optimize a bioinformatics tool?



Design-Build-Test-Learn Cycle

- Software (any product) can be improved through the DBTL cycle:
 - Design how the software could work
 - Build the software according to the design
 - Test how well the software performs
 - Learn from the testing and start again with an improved design



Version control and repositories

- Version control is important when improving successively
- Different version control systems:
 - Git, CVS, SVN, ...
- 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 branches, and 2 tags. The 'About' section describes the tool as performing 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 like bioinformatics, annotation, python3, genome-wide, myb, r2r3-myb, and a license section for GPL-3.0. The 'Releases' section shows v0.2 (revised preprint) as the latest release. The 'Languages' section shows Python at 100.0%. The repository URL is https://github.com/bpucker/MYB_annotator.

bpucker / MYB_annotator Public

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 2 tags Go to file Code

bpucker bug fixed 22 days ago 150 commits

AthRefMYBs.txt Rename 20210823_MYB_refs.txt to AthRefMYBs.txt 7 months ago

LICENSE Create LICENSE 7 months ago

MYB_annotator.png Add files via upload 2 months ago

MYB_annotator.py bug fixed 22 days ago

MYB_baits.fasta Rename 20210926_MYB_domain_baits_clean.fasta to MYB_baits.fasta 7 months ago

MYB_baits.hmm HMM file of MYB baits for HMMER-based screen 2 months ago

MYB_baits.txt Rename 20210926_MYB_domain_baits_clean.txt to MYB_baits.txt 7 months ago

README.md reference updated (BMC Genomics paper) 24 days ago

col0_example_output.7z Add files via upload 7 months ago

environment.yml updated with all tools 2 months ago

README.md

Automatic annotation of MYBs

DOI: 10.5281/zenodo.6174039

Please get in touch if you need help running the MYB annotator on your own dataset:
Boas Pucker (email)

Background

This tool allows an automatic identification and analysis of MYBs in a CDS or peptide sequence collection. Since the quality of structural annotation has substantially improved during the last years, the focus can now shift towards

About

This tool performs an automatic identification, annotation, and analysis of the MYB gene family in plants. It can be applied to new transcriptome of genome assemblies.

doi.org/10.1101/2021.10.16.464636

bioinformatics annotation python3 genome-wide myb r2r3-myb

Readme License

GPL-3.0 License

6 stars

2 watching

2 forks

Releases 2

v0.2 (revised preprint) Latest on Feb 20 + 1 release

Packages

No packages published

Languages

Python 100.0%

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.

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 citeable
- Uploads are published immediately
- Integration with github possible

The screenshot shows the Zenodo homepage. At the top, there is a search bar, an upload button, and a communities link. On the right, there are buttons for 'Log in' and 'Sign up'. Below the header, there is a section for 'Featured communities' which includes a thumbnail for the 'National COVID Cohort Collaborative (N3C)' and a brief description. A sidebar on the right provides help for uploading and lists Zenodo's priorities related to the COVID-19 outbreak. In the center, there is a 'Recent uploads' section showing three examples of uploaded software packages: 'PyGMT: A Python interface for the Generic Mapping Tools', 'pyspeckit/pyspeckit: Release 0.1.25 - pre-v1.0 to test for paper', and 'Trixi.jl'. Each entry includes the upload date, version, type, access level, and a 'View' button. To the right of these entries is a sidebar titled 'Why use Zenodo?' listing various benefits such as safety, trust, citation, and integration.

zenodo

Search | Upload | Communities

Log in | Sign up

Featured communities

National COVID Cohort Collaborative (N3C)

The National COVID Cohort Collaborative (N3C) is a complementary and synergistic partnership among the Clinical and Translational Science Awards (CTSA) Program hubs, the National Center for Data to Health (CD2H), distributed clinical data networks (PCORnet, OHDSI, ACT/i2b2, TriNetX), and other...

Curated by: CD2H

Need help uploading? Contact us

Browse | New upload

Recent uploads

April 11, 2022 (v0.6.1) Software Open Access

PyGMT: A Python interface for the Generic Mapping Tools

Ueda, Leonardo; Tian, Dongdong; Leong, Wei Ji; Jones, Meghan; Schlitzer, William; Grund, Michael; Toney, Liam; Yao, Jayuan; Magen, Yohai; Materna, Kathryn; Newton, Tyler; Anant, Abhishek; Ziebarth, Malte; Quinn, Jamie; Wessel, Paul

PyGMT is a library for processing geospatial and geophysical data and making publication quality maps and figures. It provides a Pythonic interface for the Generic Mapping Tools (GMT), a command-line program widely used in the Earth Sciences.

Uploaded on April 11, 2022

12 more version(s) exist for this record

View

April 11, 2022 (v0.1.25) Software Open Access

pyspeckit/pyspeckit: Release 0.1.25 - pre-v1.0 to test for paper

Adam Ginsburg; Jordan Mirocha; Vlas Sokolov; Jaime Pineda; Miguel de Val-Borro; Brigitta Sipócz; adamginsburg; Erik Rosolowsky; Allison Youngblood; Igor Petrashevich; Matt Craig; Erik Tollerud; Erwan Pannier; Thomas Robitaille; thogee; Alyssa Bulatik; Luca Beale; Mike Lum; The Gitter Badger

What's Changed Ammonia and doop fitter. by ilgorPetrica in https://github.com/pyspeckit/pyspeckit/pull/345 Updated NHZD model by @pinerd in https://github.com/pyspeckit/pyspeckit/pull/350 Fix for issue 351 by kefflavich in https://github.com/pyspeckit/pyspeckit/pull/352 Fix for lte_molecule..

Uploaded on April 11, 2022

7 more version(s) exist for this record

View

April 11, 2022 (v0.4.30) Software Open Access

Trixi.jl

Schlotte-Lakemper, Michael; Gassner, Gregor J.; Ranocha, Hendrik; Winters, Andrew R.; Chan, Jesse

Adaptive high-order numerical simulations of hyperbolic PDEs in Julia

Uploaded on April 11, 2022

101 more version(s) exist for this record

View

Need help?

Contact us

Zenodo prioritizes all requests related to the COVID-19 outbreak.

We can help with:

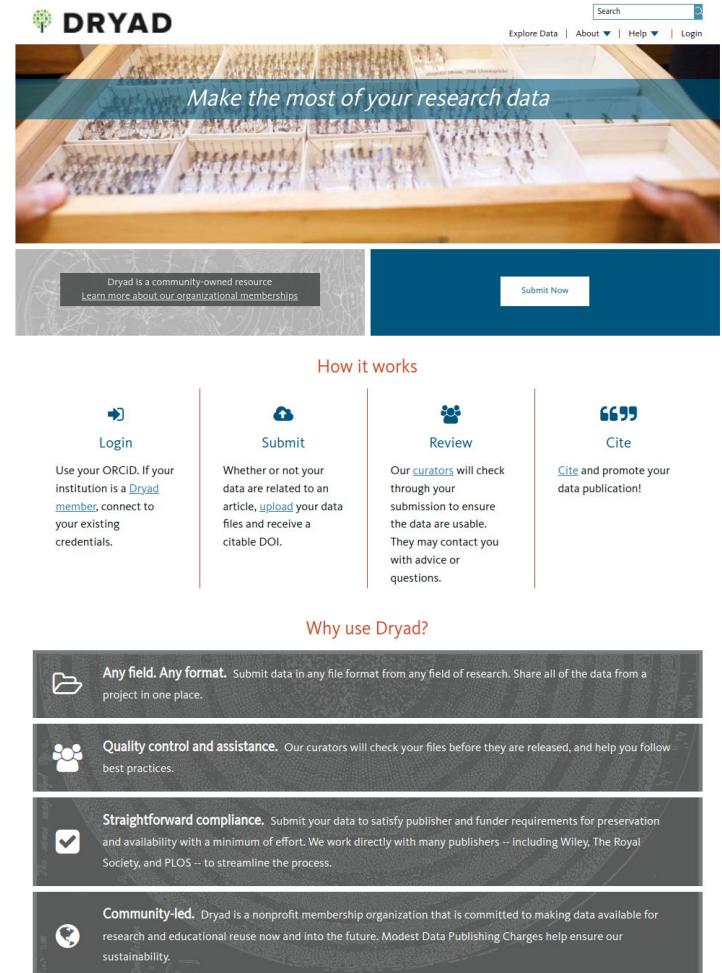
- Uploading your research data, software, preprints, etc.
- One-on-one with Zenodo supporters
- Quota increases beyond our default policy.
- Scripts for automated uploading of larger datasets.

Why use Zenodo?

- **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 re-useable by other
- DOIs make entries citeable



The screenshot shows the Dryad website homepage. At the top right is a search bar and navigation links for '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 a tray of pinned specimens. A sidebar on the left says 'Dryad is a community-owned resource' with a link to 'Learn more about our organizational memberships'. On the right, there's a 'Submit Now' button. The main content area has four sections: 'How it works' (with icons for Login, Submit, Review, and Cite), 'Why use Dryad?' (with icons for Any field, Any format; Quality control and assistance; Straightforward compliance; and Community-led), and a 'Submit' section.

DRYAD

Explore Data | About | Help | Login

Make the most of your research data

Dryad is a community-owned resource
[Learn more about our organizational memberships](#)

Submit Now

How it works

Login

Submit

Review

Cite

Why use Dryad?

Any field. Any format.

Quality control and assistance.

Straightforward compliance.

Community-led.

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.

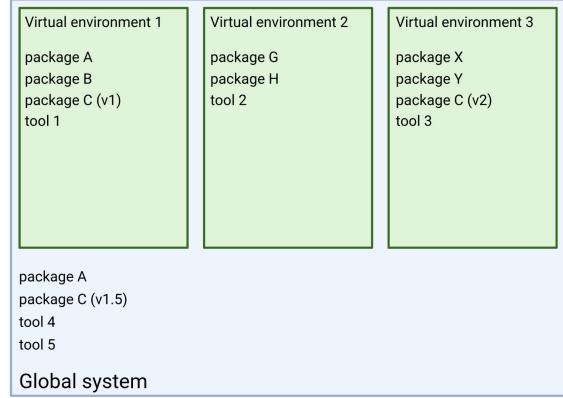
Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment 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 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']
>>>
```

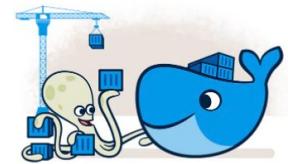
<https://docs.python.org/3/tutorial/venv.html>

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,
...

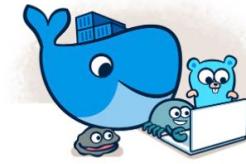
Build

- 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/>



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

- 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 homepage with a blue header featuring the logo and navigation links: HelpDesk, Contact, Intranet, Search, About, Services, Training, de.NBI Cloud, Events, News, Jobs, and ELIXIR-DE. Below the header is a banner for "Big Data Exploitation in Life Science" with a COVID-19 Research link. The main content area includes three circular icons: a blue one for "Service & Tools" (with text about maintaining and developing tools), a pink one for "Training & Education" (with text about training resources), and a light blue one for "de.NBI Cloud" (with text about enabling integrative analyses). Below these are three cards for job openings: "Three-year PhD Position in Genomics/RNA-Sequencing/artificial Neural Network topologies/machine learning" (deadline 2022-03-24), "50 Fellowships for PhDs (w/m/d) and Postdocs" (deadline 2022-04-08), and "Open position as Bioinformatician / Data scientist (f/m)" (deadline 2022-04-11). A "View all News" button is located below the job cards. At the bottom, there's a section for "Women in Data Science 2022 - Perspectives in Industry and Academia with a Focus on Life Sciences" with a calendar icon and a "View all Events" button.

<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>

Time for questions!



Questions

1. Which fields are part of bioinformatics?
2. What is the definition of a genome, transcriptome, proteome, and metabolome?
3. What is the full name of 'BLAST'?
4. Which (sequencing) method can be used to analyze a genome?
5. Which method can be used to analyze a transcriptome?
6. Why do biologists apply statistical methods?
7. Which languages are used for the development of bioinformatics tools?
8. Which steps are involved in software development/improvement?
9. Which platforms enable developers to share their software?
10. Which licence can be used to make software freely available to other researchers?
11. Where can you find (online) computational resources for bioinformatics?