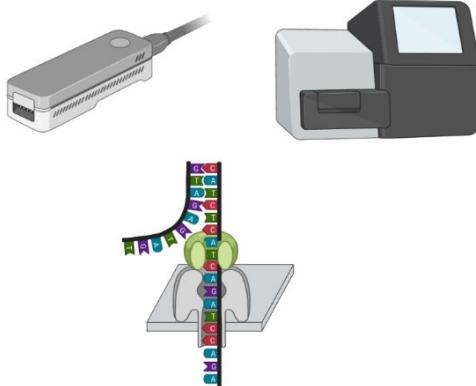
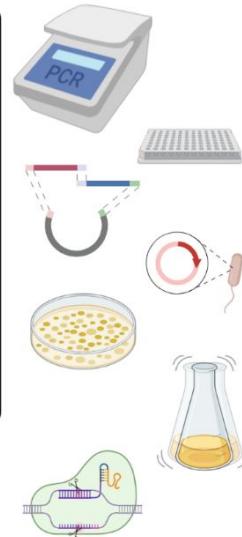
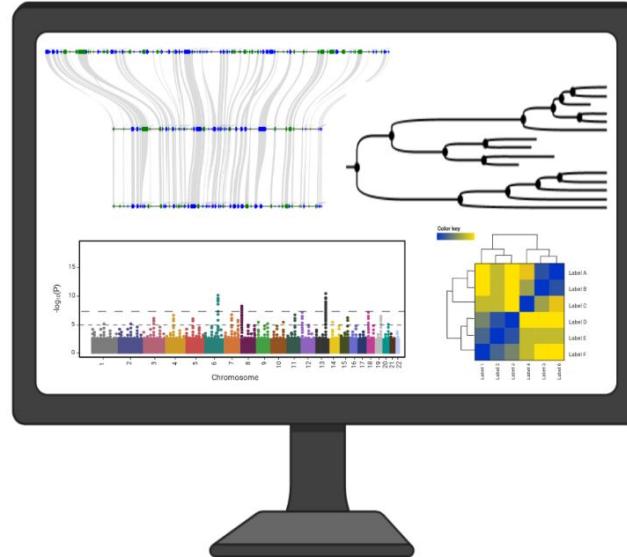




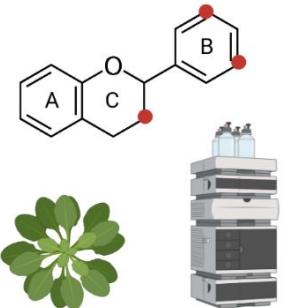
Technische
Universität
Braunschweig



Plant Biotechnology
and Bioinformatics



species biosynthesis proteins analysis different conditions
biosynthesis DODA bellman variants H23-MYB analysis
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 genes plant factors evolutionary
sites annotations pathways loci accession information
plants pigment Koyentra genes systems biology long Canophylales
key against canonical for conserved Arabidopsis
flavonoid conservation sequencing evolution
read transcription synthesis accessions identification sequence
gene MYB introns residues RNA-Seq



Genomics #2 - Big Data Analytics in Life Sciences

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

Outline - Genomics #2

- Structural & functional annotation
- Read mapping, variant calling, variant annotation
- Genome-Wide Association Studies (GWAS) / Mapping-By-Sequencing (MBS)
- Comparative genomics / synteny analysis
- Phylogenetic analyses

Genome sequence annotation



Finding genes in a genome sequence

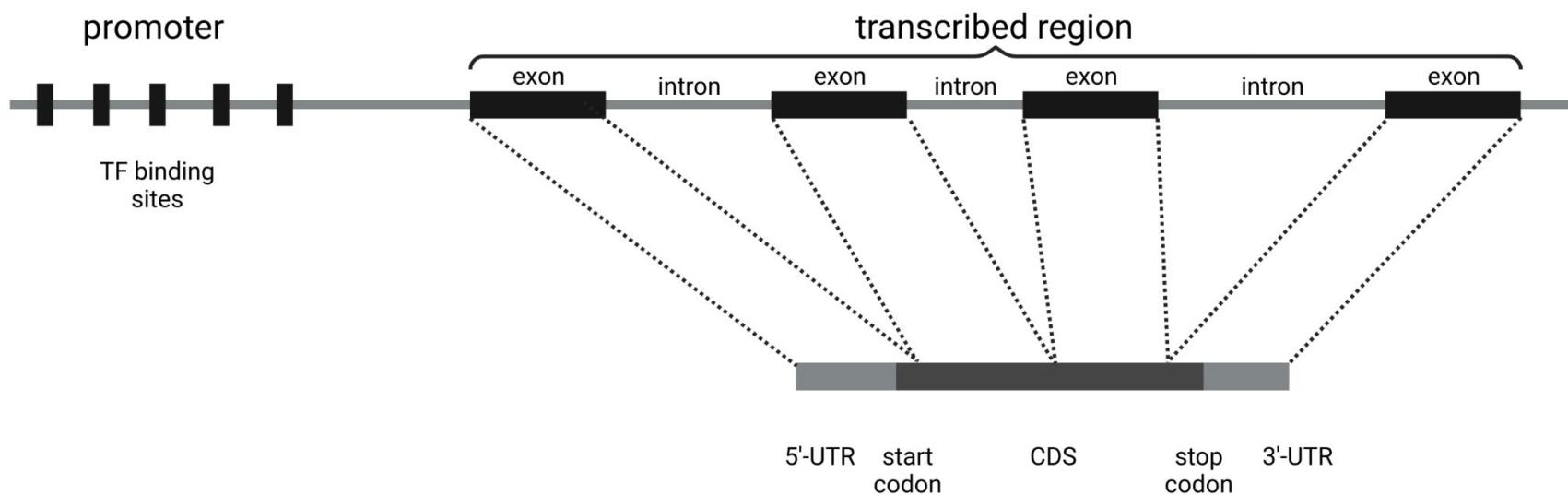
Plant gene structure

CDS = (Protein) Coding Sequence

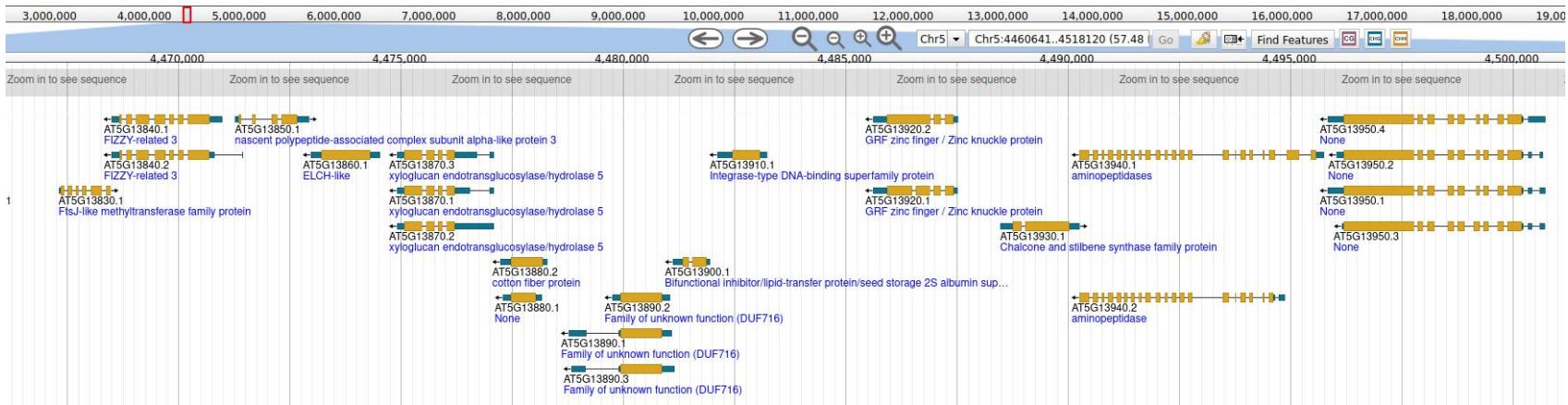
ORF = Open Reading Frame

UTR = UnTranslated Region

TF = Transcription Factor

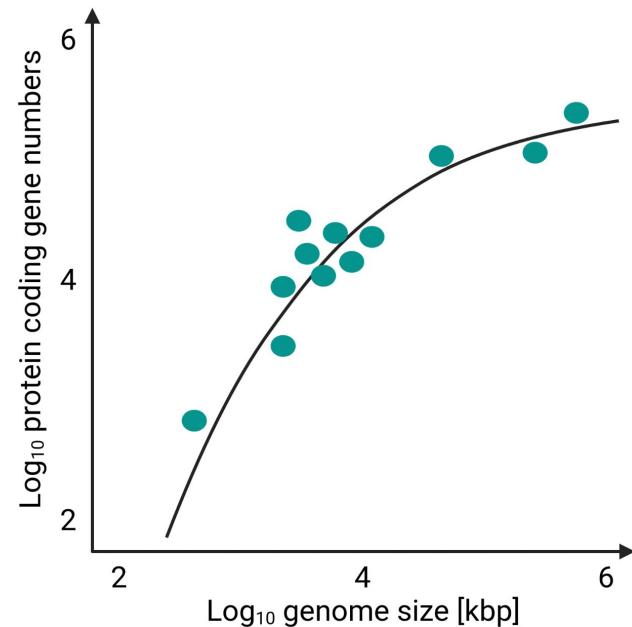


Finding genes in a genome sequence



Gene numbers

- Average number of genes in plants: 27200
- Gene number is not proportional to genome size

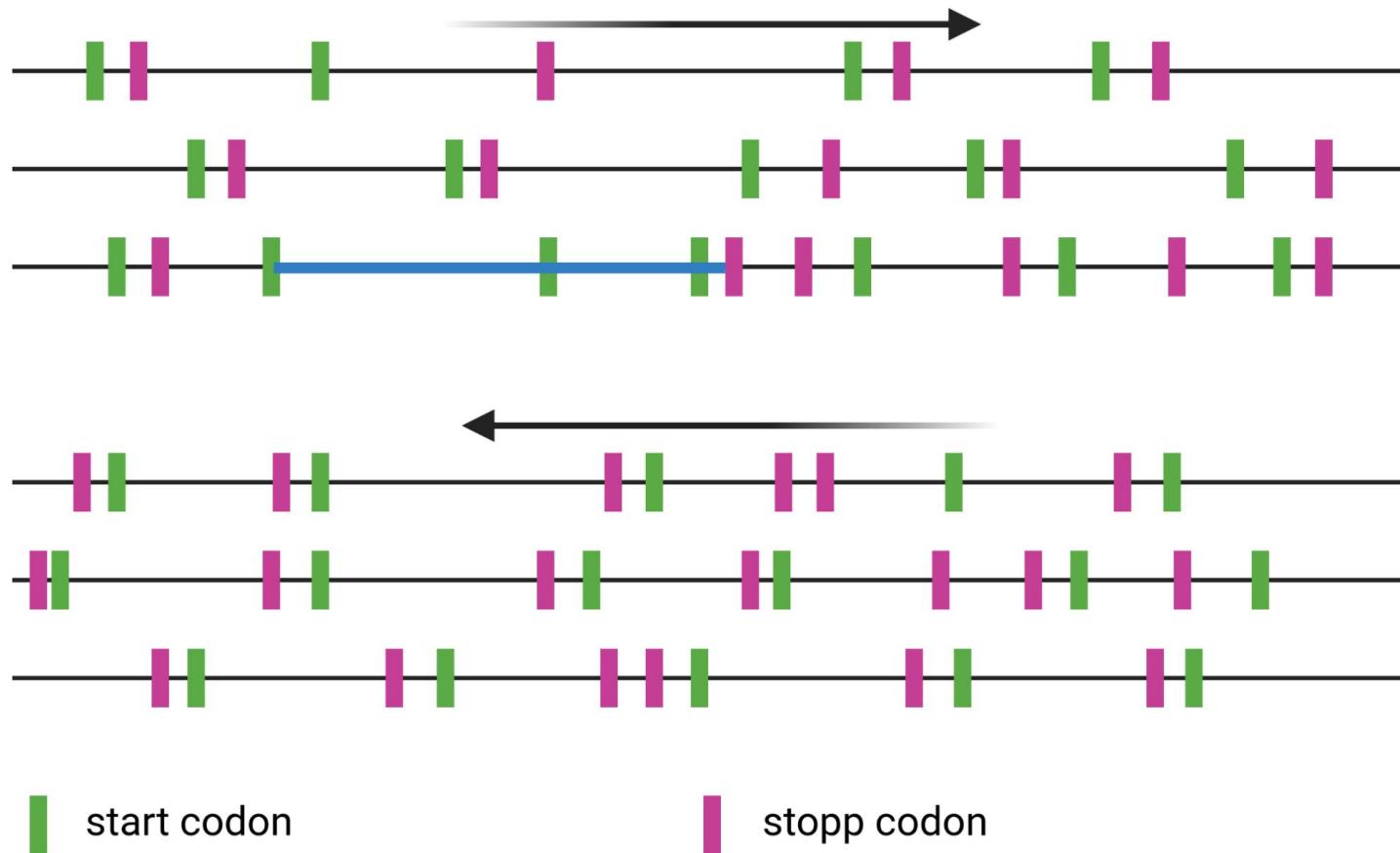


Pucker & Brockington, 2019: 10.1186/s12864-018-5360-z
Michael, 2014: 10.1093/bfgp/elu005

Repeat masking

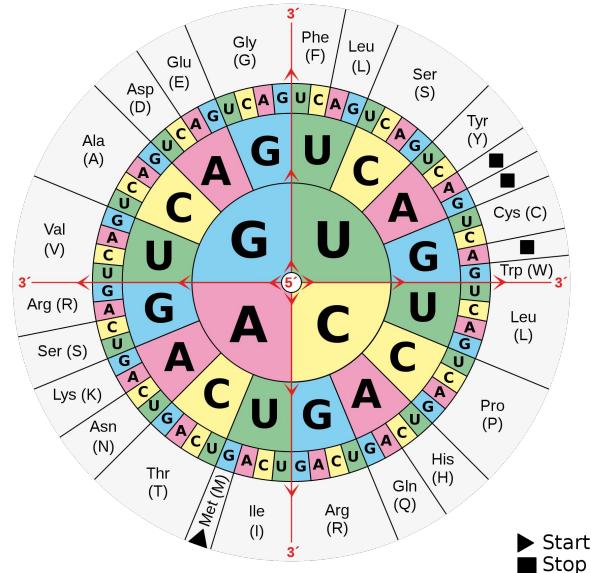
- Simple sequence repeats (SSR)
- Transposable elements (TE)
- Centromeric repeats (CEN)
- Telomeric repeats (TEL)

Finding ORFs



Codon usage

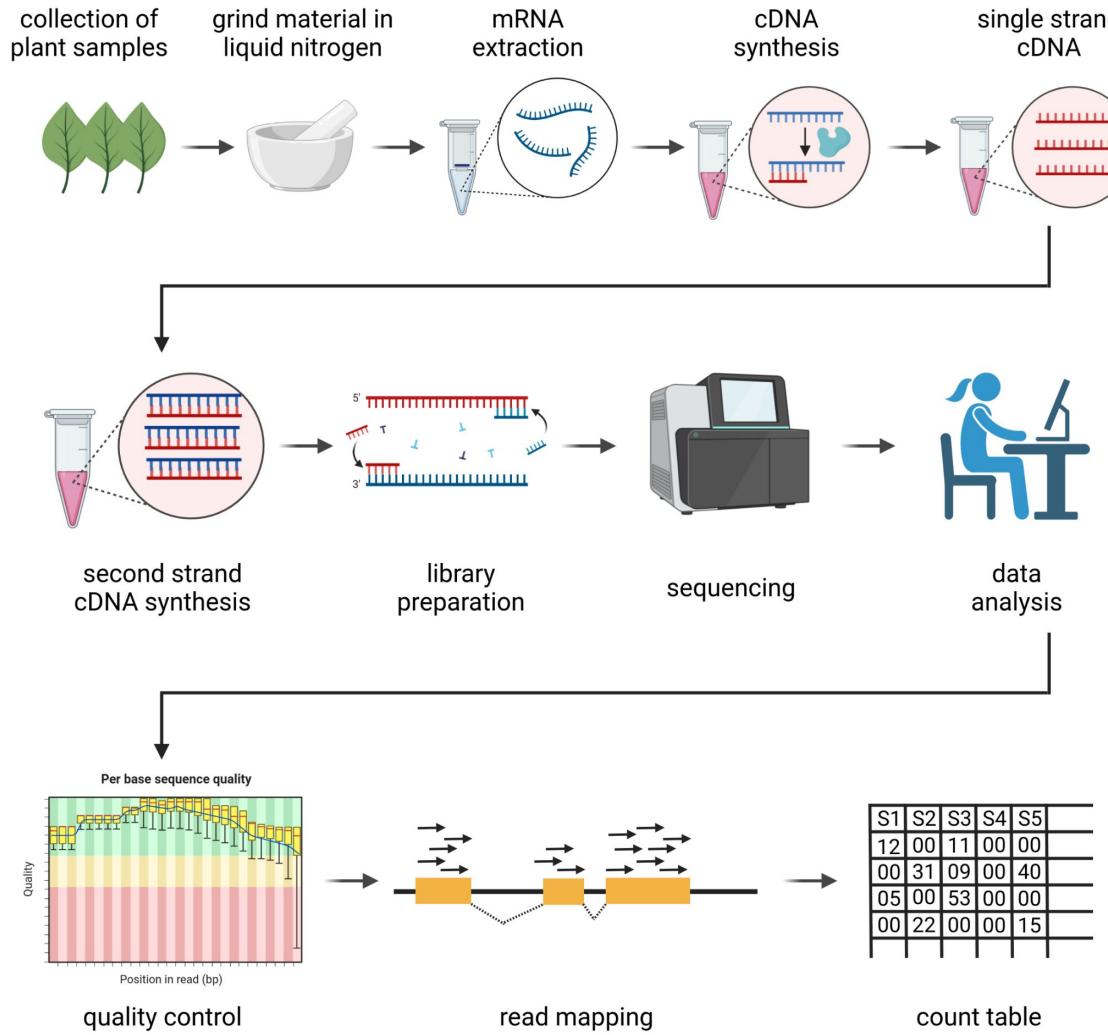
- Protein coding sequences have specific properties
 - Codon usage is different between species i.e. different species prefer different codons for certain amino acids
 - Rare codons slow down translation (useful between domains)
 - Usage of dicodons (=hexamers)
 - Codon usage can give additional CDS/ORF support



Parameters for *ab initio* gene prediction

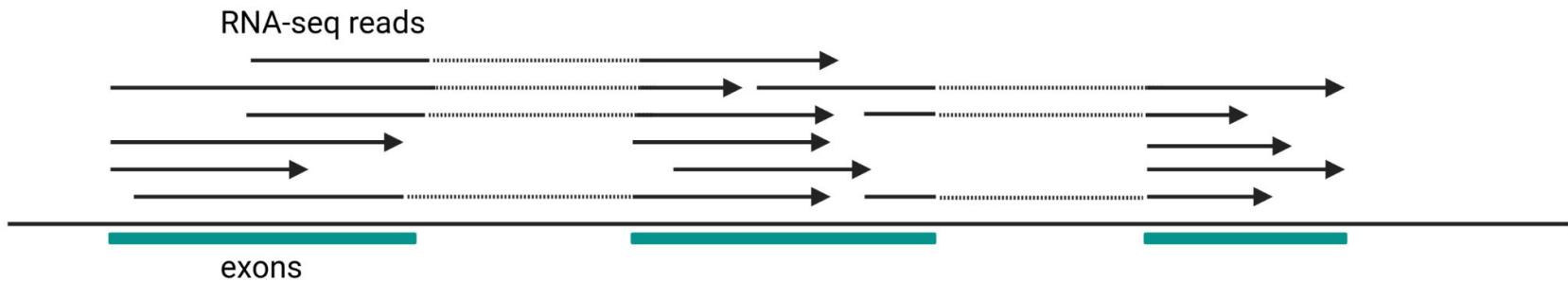
- Features: utr, exon, intron, intergenic region
- Composition of different features (codons)
- feature lengths
- Number of features per gene
- Possible splice sites

RNA-seq



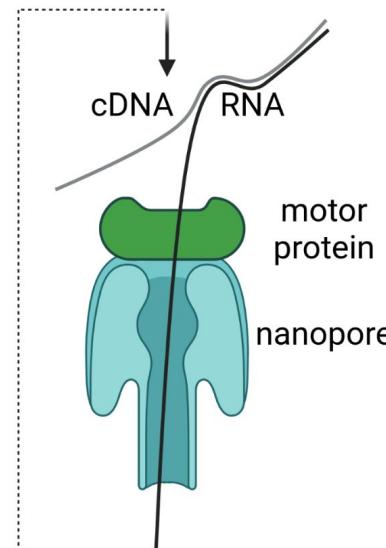
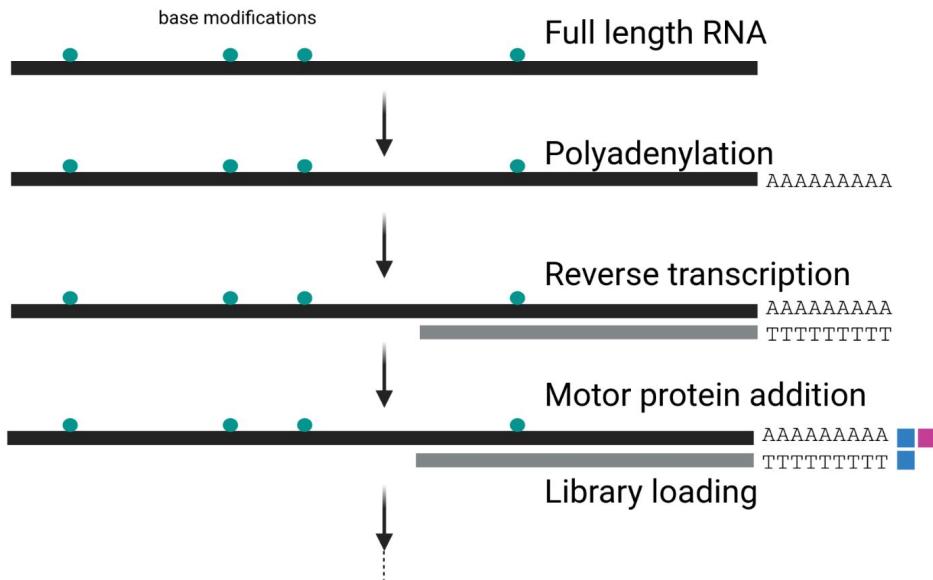
RNA-seq hints

- Aligned RNA-seq reads indicate exon positions
- Splitting of reads indicates intron positions
- CDS can be identified as ORF within the covered regions



Full length transcript sequences

- Capturing full length transcripts for RNA-seq (Illumina)
- Full length cDNA sequencing (PacBio, ONT)
- Direct RNA sequencing or cDNA sequencing (ONT)



Proteins as hints (exonerate)

- Exonerate can align coding sequences/peptide sequences to DNA
 - Intron positions are identified and intron information is given
 - Exonerate can handle different types of splice sites

1	: ATGGTGTGGCTGGTCTCTCTTTGGATGAGATCAGACAGGCTCAGAGAGCTGA :	56
4488762	: ATGGTGTGGCTGGTCTCTCTTTGGATGAGATCAGACAGGCTCAGAGAGCTGA :	4488817
57	: TGGACCTGCAGGCATCTGGCTATTGGACTGCTAACCTGAGAACCATGTGCTTC :	112
4488818	: TGGACCTGCAGGCATCTGGCTATTGGACTGCTAACCTGAGAACCATGTGCTTC :	4488873
113	: AGGGCAGAGTATCTGACTACTACTTCCGATCACCAACAGTGAAACATGACCGAC :	168
4488874	: AGGGCAGAGTATCTGACTACTACTTCCGATCACCAACAGTGAAACATGACCGAC :	4488929
169	: CTCAGGAGAAGTTCAGCGCATGT >>> Target Intron 1 >>> GC :	195
		++
		86 bp
		++
4488930	: CTCAGGAGAAGTTCAGCGCATGTgt.....agGC :	4489042
196	: GACAAGTCGACAATTGGAAACGTCACATGCTGACGGAGGAATTCTCAAGGA :	251
4489043	: GACAAGTCGACAATTGGAAACGTCACATGCTGACGGAGGAATTCTCAAGGA :	4489098
252	: AAACCCACACATGTGCTTACATGGCTCTCTGGACACCCAGACAGGACATCG :	307
4489099	: AAACCCACACATGTGCTTACATGGCTCTCTGGACACCCAGACAGGACATCG :	4489154
308	: TGGTGGTCGAAGTCCCTAACGCTAGGCAAAGAACGGCAGTGAAGGCCATCAAGGAG :	363
4489155	: TGGTGGTCGAAGTCCCTAACGCTAGGCAAAGAACGGCAGTGAAGGCCATCAAGGAG :	4489210



Annotation quality assessment

- BUSCO (Benchmarking Universal Single-Copy Orthologs)
 - Example: C:98%[S:85%,D13%],F:1.2%,M0.8%,n:1500
 - C = complete BUSCO genes
 - S = single copy BUSCO genes
 - D = duplicates BUSCO genes
 - F = fragmented BUSCO genes
 - M = missing BUSCO genes
 - n = number of BUSCO query sequences
- NL Rome: assessment of high continuity genome sequences

Transposable elements (TEs)

- Transposons shape plant genomes (genome obesity)
- Systematics:
 - Class I (retrotransposons)
 - LTR: Copia, Gypsy, Bel-Pao, Retrovirus, ERV
 - DIRS: DIRS, Ngaro, VIPER
 - PLE: Penelope
 - LNE: R2, RTE, Jockey, L1, I
 - SINE: tRNA, 7SL, 5S
 - Class II (DNA transposons) - Subclass 1
 - TIR: Tc1-Mariner, hAT, Mutator, Merlin, Transib, P, PiggyBac, PIF-Harbinger, CACTA
 - Crypton
 - Class II (DNA transposons) - Subclass 2
 - Helitron
 - Maverick

Annotation of TEs

- Benchmarking study of TE annotation tools: <https://github.com/oushujun/EDTA>
- RepeatMasker: <https://www.repeatmasker.org/>
 - Screens genomic sequence for TEs
 - Soft/hard masking of genomic sequence
 - Dfam and Repbase are important databases
 - Dfam: open collection of TE sequences (<https://www.dfam.org/home>)
 - Repbase: subscription-based collection (<https://www.girinst.org/>)
 - Different search engines can be used
- RepeatModeler2: <http://www.repeatmasker.org/RepeatModeler>
 - Pipeline for discovery of TEs
- Extensive *de novo* TE Annotator (EDTA): <https://github.com/oushujun/EDTA>
 - Complex pipeline for TE annotation

Ou et al., 2019: 10.1186/s13059-019-1905-y
Flynn et al., 2020: 10.1073/pnas.1921046117
Tarailo-Graovac & Chen, 2009: 10.1002/0471250953.bi0410s25

Generic Feature Format (GFF)

- SequenceID
- Source
- Feature type
- Start
- End
- Score
- Strand
- Phase
- Attributes (IDs)

ctg000000_np2	AUGUSTUS	gene	3003	3986	.>	->	.>	ID=g1;
ctg000000_np2	AUGUSTUS	mRNA	3003	3986	0.17	->	.>	ID=g1.t1;Parent=g1;
ctg000000_np2	AUGUSTUS	stop_codon	3003	3005	.>	->	0>	0> ID=g1.t1.stop1;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	CDS	3003	3666	0.43	->	1>	ID=g1.t1.CDS1;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	exon	3003	3666	.>	->	.>	ID=g1.t1.exon1;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	intron	3667	3798	0.39	->	.>	ID=g1.t1.intron1;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	CDS	3799	3986	0.33	->	0>	ID=g1.t1.CDS2;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	exon	3799	3986	.>	->	.>	ID=g1.t1.exon2;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	start_codon	3984	3986	.>	->	0>	0> ID=g1.t1.start1;Parent=g1.t1;
ctg000000_np2	AUGUSTUS	gene	5688	7376	.>	->	.>	ID=g2;
ctg000000_np2	AUGUSTUS	mRNA	5688	7376	0.87	->	.>	ID=g2.t1;Parent=g2;
ctg000000_np2	AUGUSTUS	stop_codon	5688	5690	.>	->	0>	0> ID=g2.t1.stop1;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	CDS	5688	6149	0.96	->	0>	ID=g2.t1.CDS1;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	exon	5688	6149	.>	->	.>	ID=g2.t1.exon1;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	intron	6150	7208	0.94	->	.>	ID=g2.t1.intron1;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	CDS	7209	7376	0.87	->	0>	ID=g2.t1.CDS2;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	exon	7209	7376	.>	->	.>	ID=g2.t1.exon2;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	start_codon	7374	7376	.>	->	0>	0> ID=g2.t1.start1;Parent=g2.t1;
ctg000000_np2	AUGUSTUS	gene	7686	8051	.>	->	.>	ID=g3;
ctg000000_np2	AUGUSTUS	mRNA	7686	8051	0.18	->	.>	ID=g3.t1;Parent=g3;
ctg000000_np2	AUGUSTUS	stop_codon	7686	7688	.>	->	0>	0> ID=g3.t1.stop1;Parent=g3.t1;
ctg000000_np2	AUGUSTUS	CDS	7686	8051	0.18	->	0>	ID=g3.t1.CDS1;Parent=g3.t1;
ctg000000_np2	AUGUSTUS	exon	7686	8051	.>	->	.>	ID=g3.t1.exon1;Parent=g3.t1;
ctg000000_np2	AUGUSTUS	start_codon	8049	8051	.>	->	0>	0> ID=g3.t1.start1;Parent=g3.t1;
ctg000000_np2	AUGUSTUS	gene	8251	8559	.>	->	.>	ID=g4;
ctg000000_np2	AUGUSTUS	mRNA	8251	8559	0.37	->	.>	ID=g4.t1;Parent=g4;
ctg000000_np2	AUGUSTUS	stop_codon	8251	8253	.>	->	0>	0> ID=g4.t1.stop1;Parent=g4.t1;
ctg000000_np2	AUGUSTUS	CDS	8251	8559	0.37	->	0>	ID=g4.t1.CDS1;Parent=g4.t1;
ctg000000_np2	AUGUSTUS	exon	8251	8559	.>	->	.>	ID=g4.t1.exon1;Parent=g4.t1;
ctg000000_np2	AUGUSTUS	start_codon	8557	8559	.>	->	0>	0> ID=g4.t1.start1;Parent=g4.t1;
ctg000000_np2	AUGUSTUS	gene	10459	11261	.>	->	.>	ID=g5;
ctg000000_np2	AUGUSTUS	mRNA	10459	11261	0.53	->	.>	ID=g5.t1;Parent=g5;
ctg000000_np2	AUGUSTUS	stop_codon	10459	10461	.>	->	0>	0> ID=g5.t1.stop1;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	CDS	10459	10809	0.7	->	0>	ID=g5.t1.CDS1;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	exon	10459	10809	.>	->	.>	ID=g5.t1.exon1;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	intron	10810	11162	0.53	->	.>	0> ID=g5.t1.intron1;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	CDS	11163	11261	0.8	->	0>	ID=g5.t1.CDS2;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	exon	11163	11261	.>	->	.>	ID=g5.t1.exon2;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	start_codon	11259	11261	.>	->	0>	0> ID=g5.t1.start1;Parent=g5.t1;
ctg000000_np2	AUGUSTUS	gene	12226	12531	.>	+>	.>	ID=g6;
ctg000000_np2	AUGUSTUS	mRNA	12226	12531	0.82	+>	.>	ID=g6.t1;Parent=g6;
ctg000000_np2	AUGUSTUS	start_codon	12226	12228	.>	+>	.>	0> ID=g6.t1.start1;Parent=g6.t1;
ctg000000_np2	AUGUSTUS	CDS	12226	12531	0.82	+>	.>	ID=g6.t1.CDS1;Parent=g6.t1;



Sharing structural annotation

- Species specific databases:
 - TAIR: *Arabidopsis thaliana*
 - BananGenomeHub: *Musa acuminata*
- EMBL/EBI: European Nucleotide Archive of the European Bioinformatics Institute (<https://www.ebi.ac.uk/ena/browser/home>)
- PLAZA (<https://bioinformatics.psb.ugent.be/plaza/>)
- Phytozome (<https://phytozome-next.jgi.doe.gov/>)

Functional annotation

- What is the function of a gene?
- Knockout experiments for all genes are time consuming and expensive
- Annotation transfer: orthologs are assumed to have the same function
- Tools:
 - BLAST: <https://blast.ncbi.nlm.nih.gov/Blast.cgi>
 - Pfam: <https://pfam.xfam.org/>
 - InterProScan5: <http://www.ebi.ac.uk/interpro/search/sequence/>
 - Shoot: <https://github.com/davidemms/SHOOT>
 - KEGG: <https://www.genome.jp/kegg/>
 - GO: <http://geneontology.org/>
 - MetaCyc: <https://metacyc.org/>
 - KIPES: <https://github.com/bpucker/KIPES>
 - BRENDA: <https://www.brenda-enzymes.org/>
 - Mercator: <https://plabipd.de/portal/mercator4>

Altschul et al., 1990: 10.1016/S0022-2836(05)80360-2
Mistry et al., 2021: 10.1093/nar/gkaa913
Jones et al., 2014: 10.1093/bioinformatics/btu031
Karp et al., 2002: 10.1093/nar/30.1.59
Emms & Kelly, 2022: 10.1186/s13059-022-02652-8
Kanehisa & Goto, 2000: 10.1093/nar/28.1.27
Ashburner et al., 2000: 10.1038/75556
Pucker et al., 2020: 10.3390/plants9091103
Schomburg et al., 2002: 10.1093/nar/30.1.47
Schwacke et al., 2019: 10.1016/j.molp.2019.01.003

BLAST: Basic Local Alignment Search Tool

- Probably the most famous website of the NCBI
- Comparison of sequences against a large database
- Similar sequences are likely to have similar functions (ideally orthologs)
- Numerous variants of the initial BLASTn were developed

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. [Learn more](#)

NEWS
BLAST+ 2.13.0 is here!
Starting with this release, we are including the blastn_vdb and tblasn_vdb executables in the BLAST+ distribution.

Thu, 17 Mar 2022 12:00:00 EST [More BLAST news...](#)

Web BLAST

Nucleotide BLAST
Nucleotide ▶ nucleotide

blastx
translated nucleotide ▶ protein

tblastn
protein ▶ translated nucleotide

Protein BLAST
protein ▶ protein

BLAST Genomes

Enter organism common name, scientific name, or tax id **Search**

Human Mouse Rat Microbes

Standalone and API BLAST

[Download BLAST](#)
Get BLAST databases and executables

[Use BLAST API](#)
Call BLAST from your application

[Use BLAST in the cloud](#)
Start an instance at a cloud provider

Specialized searches

SmartBLAST
Find proteins highly similar to your query

Primer-BLAST
Design primers specific to your PCR template

Global Align
Compare two sequences across their entire span (Needleman-Wunsch)

CD-search
Find conserved domains in your sequence

IgBLAST
Search immunoglobulins and T cell receptor sequences

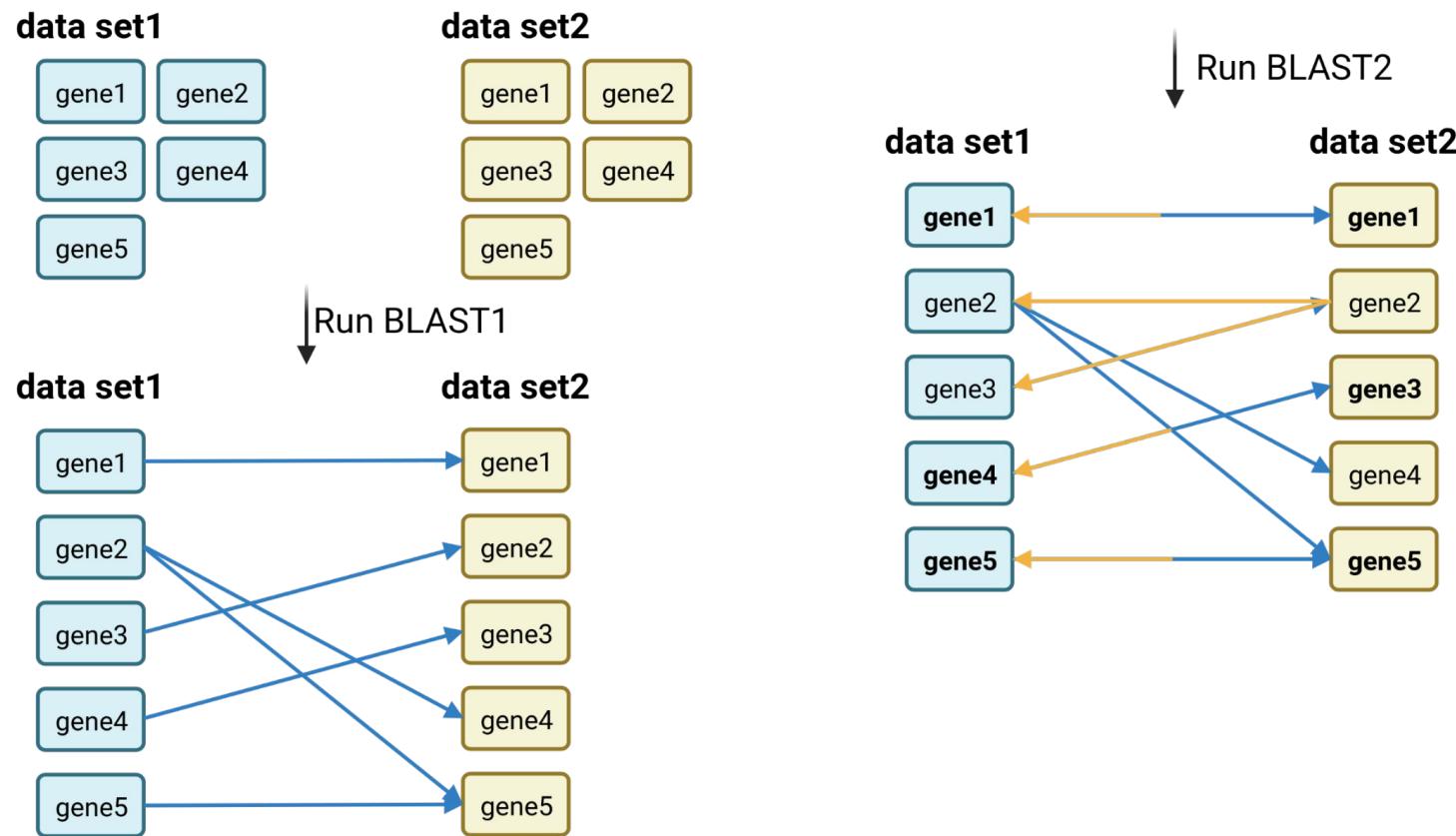
VecScreen
Search sequences for vector contamination

CDART
Find sequences with similar conserved domain architecture

Multiple Alignment
Align sequences using domain and protein constraints

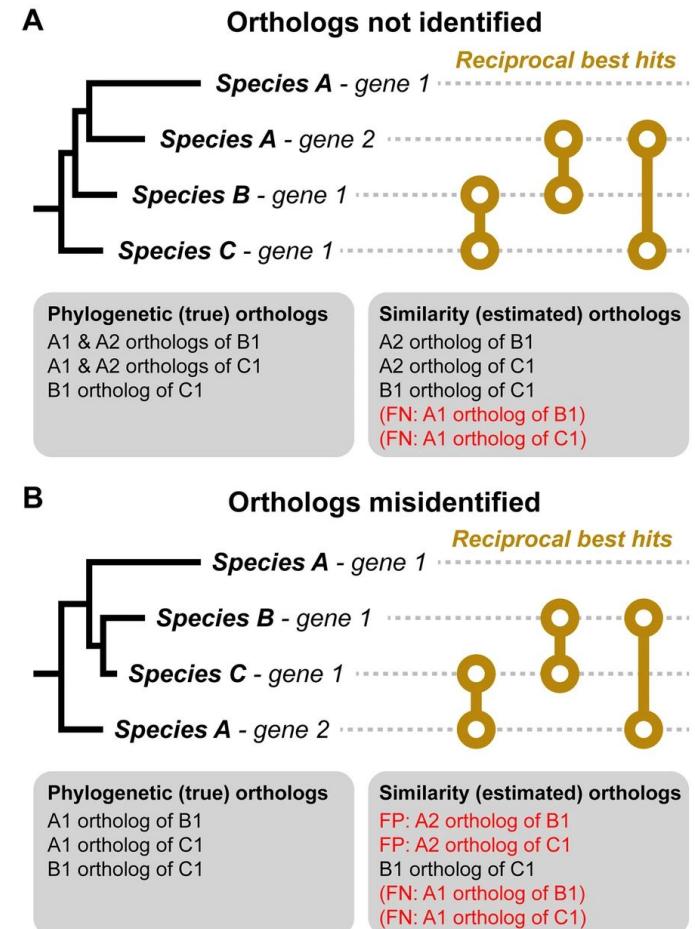
MOLE-BLAST
Establish taxonomy for uncultured or environmental sequences

Reciprocal Best BLAST Hits (RBHs)



OrthoFinder2

- Identification of orthologs often not possible
- Orthogroups are better reflection of reality
- Assumption: orthologs have the same function



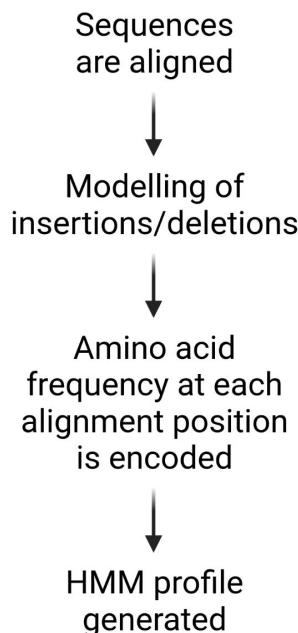
Pfam: Protein family database

HOME | SEARCH | BROWSE | FTP | HELP | ABOUT

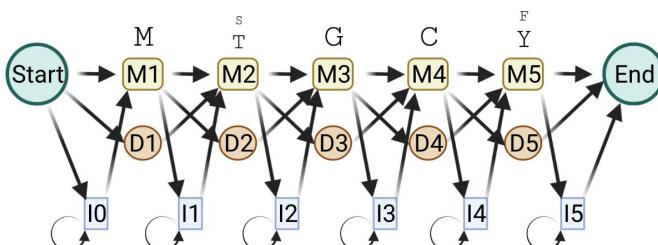
Pfam 35.0 (November 2021, 19632 entries)

The Pfam database is a large collection of protein families, each represented by **multiple sequence alignments** and **hidden Markov models (HMMs)**. [More...](#)

- Assignment of protein functions based on Hidden Markov Models (HMMs)
- Sequences are screened based on HMM profile



seq1	MTGC - Y	2 = deletion
seq2	MSGC - F	5 = insertion
seq3	MTGC - Y	
seq4	M - GCAY	
	1 2 3 4 5 6	



QUICK LINKS YOU CAN FIND DATA IN PFAM IN VARIOUS WAYS...

- [SEQUENCE SEARCH](#) Analyze your protein sequence for Pfam matches
[VIEW A PFAM ENTRY](#) View Pfam annotation and alignments
[VIEW A CLAN](#) See groups of related entries
[VIEW A SEQUENCE](#) Look at the domain organisation of a protein sequence
[VIEW A STRUCTURE](#) Find the domains on a PDB structure
[KEYWORD SEARCH](#) Query Pfam by keywords

JUMP TO **Go** **Example**

Enter any type of accession or ID to jump to the page for a Pfam entry or clan, UniProt sequence, PDB structure, etc.

Or view the [help](#) pages for more information

Recent Pfam blog posts

[Pfam 35.0 is released](#) (posted 19 November 2021)

Pfam 35.0 contains a total of 19,632 families and clans. Since the last release, we have built 460 new families, killed 7 families and created 12 new clans. UniProt Reference Proteomes has increased by 7% since Pfam 34.0, and now contains 61 million sequences. Of the sequences that are in UniProt Reference Proteomes, 75.2% have [...]

[AlphaFolding the Protein Universe](#) (posted 22 July 2021)

Hot on the tail of our inclusion of the Baker group's trRosetta structural models we are excited to announce the inclusion of models from AlphaFold 2.0 generated by DeepMind and stored in the AlphaFold Database (AlphaFold DB). AlphaFold 2.0's performance in the CASP14 competition was spectacular, producing near experimental quality structure models. The new AlphaFold [...]

[Google Research Team bring Deep Learning to Pfam](#) (posted 24 March 2021)

We are delighted to announce the first fruits of a collaboration between the Pfam team and a Google Research team led by Dr Lucy Colwell, with Maxwell Bileschi and David Belanger. In 2019, Colwell's team published a preprint describing a new deep learning method that was trained on Pfam data, and which improves upon the [...]

Citing Pfam

If you find Pfam useful, please consider [citing](#) the reference that describes this work:

[Pfam: The protein families database in 2021](#): J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G.A. Salazar, E.L.L. Sonnhammer, S.C.E. Tosatto, L. Paladini, S. Raj, L.J. Richardson, R.D. Finn, A. Bateman

[Nucleic Acids Research](#) (2020) doi: 10.1093/nar/gkaa913



Pfam is part of the ELIXIR infrastructure

Pfam is an ELIXIR service [Read more](#)

Comments or questions on the site? Send a mail to pfam-help@ebi.ac.uk.
European Molecular Biology Laboratory

<http://pfam.xfam.org/>

Mistry et al., 2020: 10.1093/nar/gkaa913

InterProScan5

- Screen of protein sequences against collection of protein signatures (KEGG, Pfam, PANTHER)
- Allows the assignment of functional annotation terms
- Available as web service, but also as stand alone tool
- Search based on sequence similarity via DIAMOND

InterProScan 5 Sequence Search

This form allows you to scan your sequence for matches against the InterPro collection of protein signature databases.

This web form is for debugging purposes only and is not supported. To submit jobs to InterProScan 5, please visit the InterPro Sequence Search or the InterProScan 5 Web services.

Please Note

This web form is for internal debugging purposes and will be retired in April 2016. If you have any questions/concerns please contact us via the feedback link above.

STEP 1 - Enter your input sequence

Enter or paste a PROTEIN sequence in any supported format:
uniprot:KPY_M_HUMAN

Or, upload a file: [Choose File] [No file chosen]

Use a example sequence | Clear sequence | See more example inputs

STEP 2 - Select the applications to run:

TIGRFAM BFLD Phobus SignalP SignalP_EUK
 SignalP_GRAM_POSITIVE SignalP_GRAM_NEGATIVE SUPERFAMILY PANTHER GeneID
 Hmmp ProSiteProfiles ProSitePatterns Cds SMART
 CDD PRINTS Pfam ModDBlite
 TMHMM

STEP 3 - Submit your job

Be notified by email (Tick this box if you want to be notified by email when the results are available)

Results for job iprscan5-I20220320-102557-0980-87120812-p2m

Tool Output		Submission Details									
Download in XML format		Download in TSV format		Download in GFF3 format		Download in SVG format		Download HTML tarball file		Download in JSON format	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	Gene3D	G3DSA:3.40.47.10	-	1	241	4.8E-101	T	20-03-2022	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	SUPERFAMILY	SSF53901	Thiolase-like	241	393	2.98E-51	T	20-03-20	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	SUPERFAMILY	SSF53901	Thiolase-like	10	237	2.38E-78	T	20-03-20	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	Pfam	PF02797 Chalcone and stilbene synthases, C-terminal domain		244	394	1.5E-71			
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	PANTHER	PTHR11877:SF81 BNAA02G30320D	PROTEIN	6	394	0.0	T	20-03-2022	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	Gene3D	G3DSA:3.40.47.10	-	242	395	9.7E-62	T	20-03-2022	IPR01603
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	PANTHER	PTHR11877	HYDROXYMETHYLGLUTARYL-COA SYNTHASE	6	394	0.0	T		
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	CDD	cd00831 CHS_like	21	390	0.0	T	20-03-2022	-	-
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	PIRSF	PIRSF000451 PKS_III	7	394	0.0	T	20-03-2022	IPR011141	
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	Pfam	PF00195 Chalcone and stilbene synthases, N-terminal domain		10	233	2.9E-119			
sp P13114 CHSY_ARATH	ba46b0f06bfbf1c91d161f0f91310f43	395	ProSitePatterns	PS00441 Chalcone and stilbene synthases active site.		161	177	-			

http://www.ebi.ac.uk/Tools/services/web/toolform.ebi?tool=iprscan5&sequence=uniprot:KPY_M_HUMAN
Jones et al., 2014: 10.1093/bioinformatics/btu031

Shoot

- Initial search based on sequence similarity
- Phylogenetic relationships of sequences are considered based on a tree
- Universal tool, but computationally more intensive than a simple sequence similarity analysis

SHOOT.bio - the phylogenetic search engine

SHOOT is a phylogenetic alternative to BLAST. Instead of returning a list of similar sequences to a query sequence it returns a maximum likelihood phylogenetic tree with your query sequence embedded in it.

Try it out: <https://shoot.bio/>

Preprint: <https://www.biorxiv.org/content/10.1101/2021.09.01.458564>

Using the SHOOT command line tool

SHOOT allows you to search a protein sequence against a database of gene trees. It returns your gene grafted into the correct position within its corresponding gene tree.

Preparing a SHOOT phylogenetic database

0. Install dependencies:

- Python libraries: ete3, sklearn, biopython
- DIAMOND
- MAFFT
- EPA-ng & gappa (<https://github.com/lczech/gappa>)
- Alternatively, IQ-TREE can be used instead of the combination EPA-ng + gappa

1. Run an OrthoFinder analysis on your chosen species, using the multiple sequence alignment option for tree inference, "-M msa".

- Paper: Emms, D.M., Kelly, S. OrthoFinder: phylogenetic orthology inference for comparative genomics. *Genome Biol* 20, 238 (2019). <https://doi.org/10.1186/s13059-019-1832-y>
- GitHub: <https://github.com/davidemms/OrthoFinder>
- Tutorials: <https://davidemms.github.io/>

2. Run `python create_shoot_db.py RESULTS_DIRECTORY`, replacing "RESULTS_DIRECTORY" with the path to the OrthoFinder results directory from step 1.

3. Resolve polytomies (only necessary if using EPA-ng): `python bifurcating_trees.py RESULTS_DIRECTORY`

The OrthoFinder RESULTS_DIRECTORY is now a SHOOT database.

Running SHOOT

```
python shoot INPUT_FASTA SHOOT_DB
```

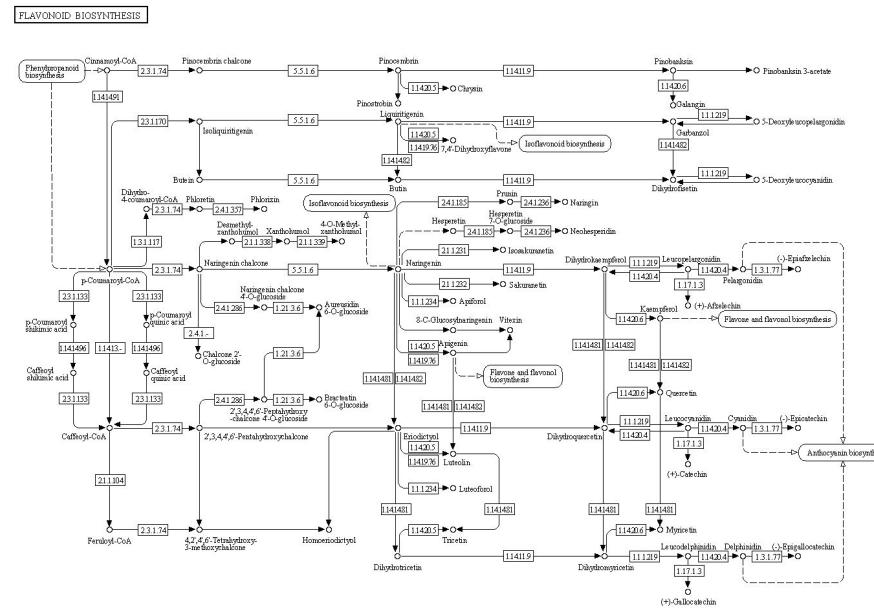
where INPUT_FASTA is a fasta file containing the amino acid sequence for the search and SHOOT_DB is the SHOOT database directory created using the steps above.

<https://github.com/davidemms/SHOOT>



KEGG: Kyoto Encyclopedia of Genes and Genomes

- Maps of pathways showing the individual reactions with catalyzing enzymes
 - Information about genomes and genes
 - Chemical details about enzymes, substrates, and products
 - KEGG is financed through a subscription model (for FTP download), but website is freely accessible



map00941 Flavonoid biosynthesis

Gene Ontology (GO)

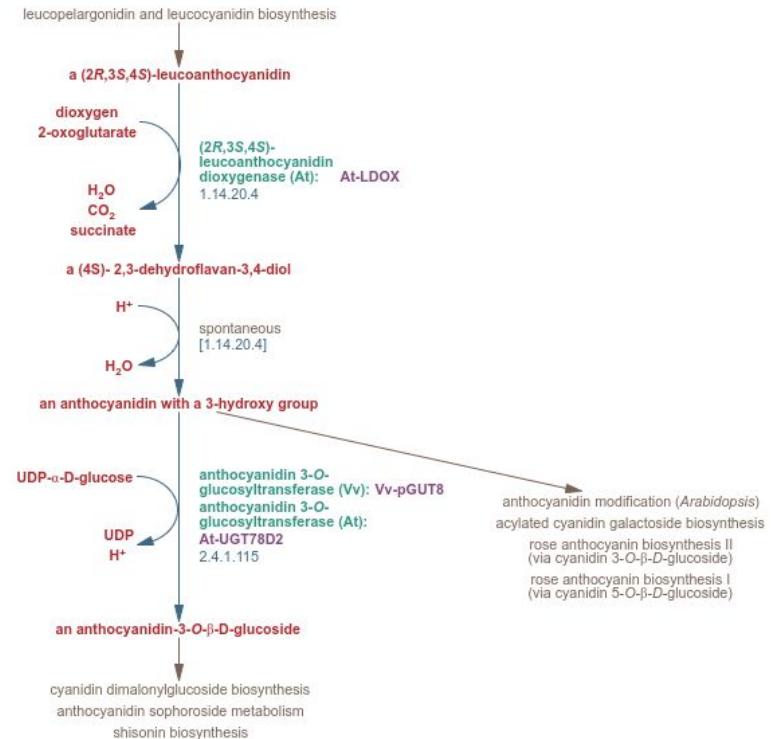
- Defined statements about the function of a gene (controlled vocabulary)
- Hierarchical structure
 - Example: ‘metabolic process’ > ‘biosynthetic process’ > ... > ‘chalcone synthase’
- Supported by the Alliance of Genome Resources
- Connected with various other databases e.g. TAIR, FlyBase, Reactome, UniProt
- Machine readable to allow automatic processing
- Tools: Blast2GO and AmiGO
 - analyze the function of a sequence (web service and standalone)

■ GO:0008150 biological_process
■ GO:0008152 metabolic process
■ GO:0009058 biosynthetic process
■ GO:0071704 organic substance metabolic process
■ GO:0009812 flavonoid metabolic process
■ GO:1901576 organic substance biosynthetic process
▼ GO:0009813 flavonoid biosynthetic process
■ GO:0009718 anthocyanin-containing compound biosynthetic process
■ GO:0051551 aurone biosynthetic process
■ GO:0033485 cyanidin 3-O-glucoside biosynthetic process
■ GO:0033486 delphinidin 3-O-glucoside biosynthetic process
■ GO:0051553 flavone biosynthetic process
■ GO:0009716 flavonoid phytoalexin biosynthetic process
■ GO:0051557 leucoanthocyanidin biosynthetic process
■ GO:0009964 negative regulation of flavonoid biosynthetic process
■ GO:0033487 pelargonidin 3-O-glucoside biosynthetic process
■ GO:0009963 positive regulation of flavonoid biosynthetic process
■ GO:0009962 regulation of flavonoid biosynthetic process

<http://amigo.geneontology.org/amigo/term/GO:0009813>
The Gene Ontology Consortium, 2007: 10.1093/nar/gkm883

MetaCyc: Metabolite Encyclopedia

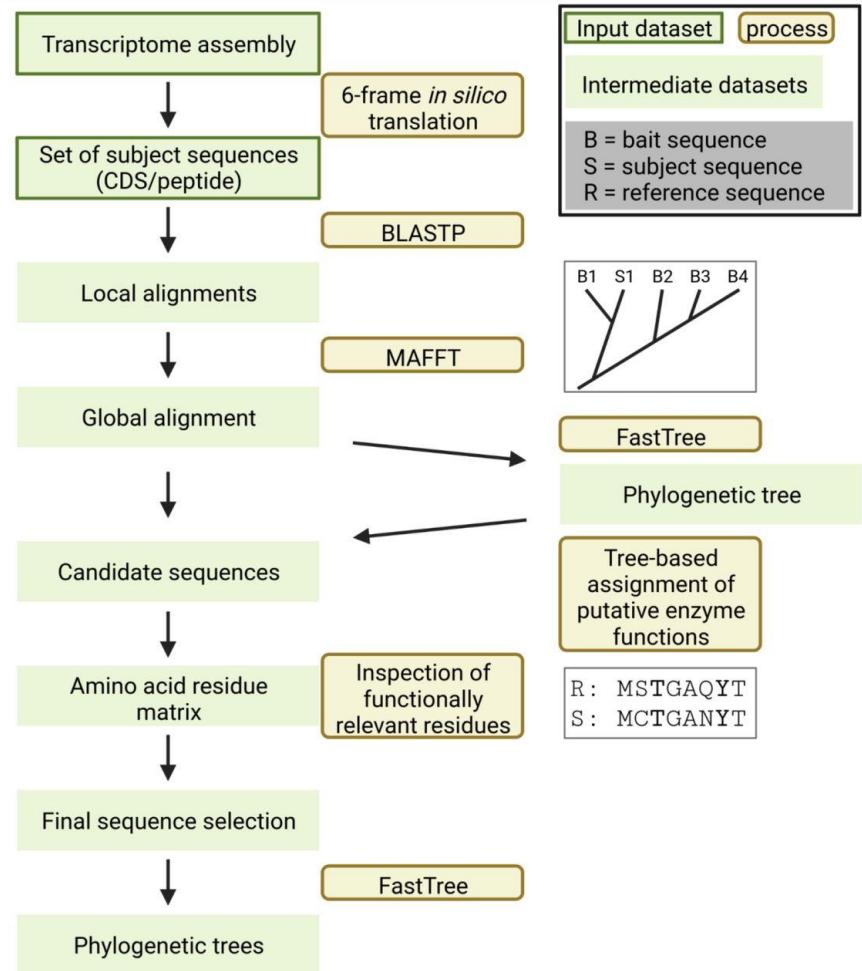
- Integrates genomic data with functional annotation
- Visualization of pathway databases
- Shows intermediates and enzymes of biosynthesis pathways
- MetaFlux: flux-balance analysis



<https://metacyc.org/>
Karp et al., 2015: <https://doi.org/10.48550/arXiv.1510.03964>
Figure: <https://biocyc.org/META/NEW-IMAGE?type=PATHWAY&object=PWY-5125>

KIPEs

- KIPEs = Knowledge-based Identification of Pathway Enzymes
- Identification of genes involved in well-studied biosynthesis pathways
- Prediction about functionality of enzymes
- Requires existing knowledge from other species



<https://github.com/bpucker/KIPEs>

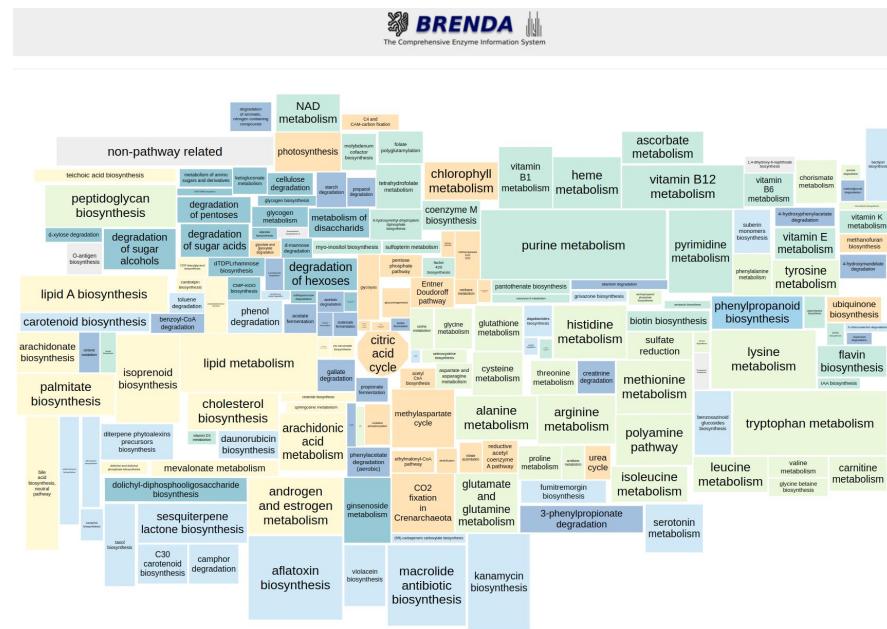
Pucker et al., 2020: 10.3390/plants9091103

Rempel et al., 2023: 10.1371/journal.pone.0294342



BRENDA: BRAunschweig ENzyme DAtabase

- Enzyme database hosted at TU Braunschweig (BRICS)
 - Text and structure-based queries
 - Visualization of pathways
 - Manual curation of datasets
 - Many details about enzyme properties (substrates, kinetics, mutants, ...)



Mercator - functional annotation of protein sequences

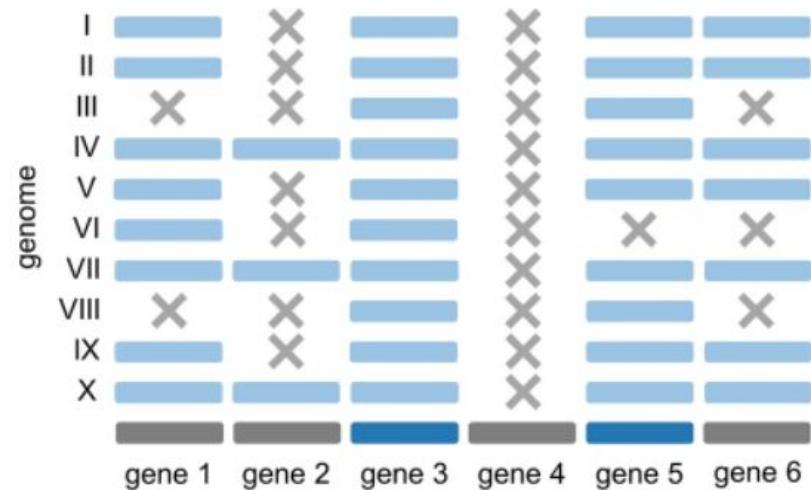
- Online tool for the annotation of all protein sequences in a submitted FASTA file
- FASTA format:
 - Header line starts with ‘>’ followed by the sequence name
 - Header is followed by unrestricted number of sequence lines

```
>TRINITY_DN100013_c0_g1_i1
MIGPMPGMEGKLMPLPGASPVGLEVVLVASTPILSDTSLCTPSFYHFLLLGPITSISNLIVRPFLLSSITVIYGRLCFFAFDFYVY
>TRINITY_DN100016_c0_g1_i1
MPKKSSNIQQNNQNRRIKRKGRRPKKKYVQQIDSSDEDILSVRHASTRPRIISIRRNEIPMRPEIHIPPTSIPSFNNPNNIHQSSSDRQMPP
>TRINITY_DN100019_c0_g1_i1
MPPKAADKKPAAKAPATASKAPEKKDAGKKTAAATGDKKRTKARKETYSSYIYKVLKQVHPDTGISNRAMSILNSFVNDFERVATEASKLA
>TRINITY_DN10001_c0_g1_i2
MAKVGNPVIDETDGSVNEPESSEKNIEVSSSTQAPESTNTTELLVNEKKAFSLATPAVRRVAREHNIDINNIKGTGKNGRITKEDILNYV
>TRINITY_DN100025_c0_g1_i1
MVENQDGCFKPGWKEFVRNSDLEGGDFLVNLVDKISYQVVIFDGTCACPDKLCPFSIMNPIFIQHLRNKIFLSKKEEIKLKGNRKVHSVNEN
>TRINITY_DN10002_c0_g1_i1
MSDELNQVFQRCREQKRPVVFVAFVTAGYPDSEETVDILLGLEAGGADIIIELGIPFTDPMVDGKTIQDANNVALENKIDISKCLSYVSESRAK
```

<https://plabipd.de/portal/mercator-sequence-annotation>
Lohse et al., 2014: 10.1111/pce.12231
Haak et al., 2018: 10.3389/fmolb.2018.00062

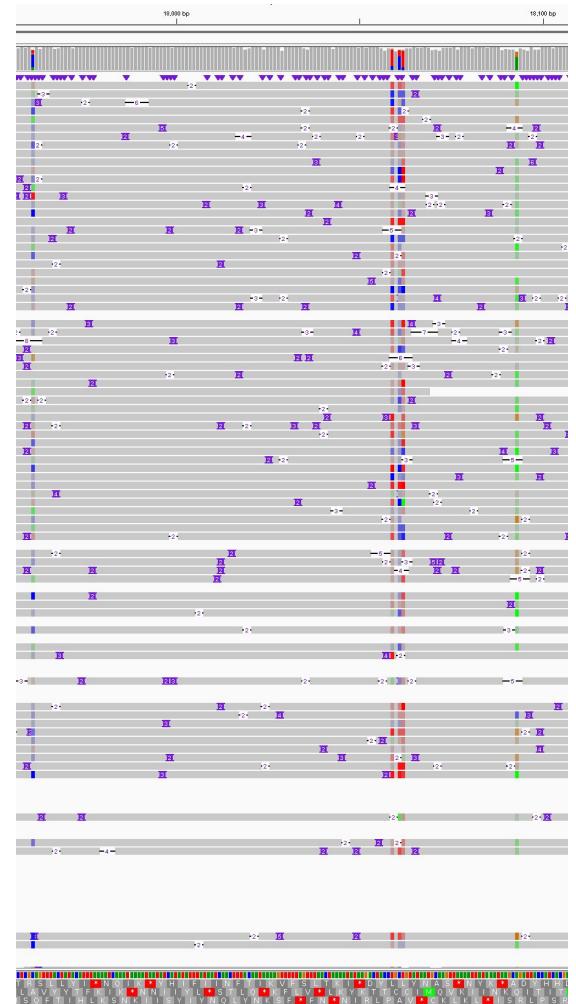
Intraspecific variation

- Different accessions of the same species have genomic differences
- Genotype determines phenotype (traits)
- Specific traits are relevant for breeding or biotechnological/agronomical applications



Differences between samples and reference

- Differences are identified against a reference genome sequence (classic approach)
- Reads of a sample are aligned against the reference (mapping)
- Differences are identified by tools (variant calling)



Read mapping

- Alignment of reads to a genome sequence
- Mapping needs to be fast at the cost of accuracy
- Manual inspection of read mappings via Integrated Genomics Viewer (IGV)

SAM/BAM

- SAM = Sequence Alignment/Map format
- BAM = Binary Alignment/Map format (binary version of SAM)
- Another way to store read information: contains information from FASTA and FASTQ file (reads mapped to reference)

Variant calling

- Identification of sequence differences between reads and reference sequence
- Differences are listed in a specific file type: Variant Caller Format (VCF)
- ONT long reads are well suited for the identification of large structural variants

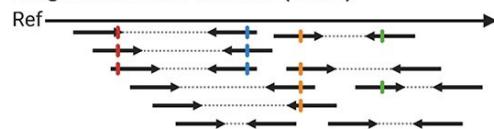
Types of sequence variants

- SNVs vs. SNPs
 - Single Nucleotide Variants = no information about their effect
 - Single Nucleotide Polymorphisms = no detrimental impact
- MNVs
 - Multiple Nucleotide Variants
- InDels
 - Insertions: additional bases in sample compared to reference
 - Deletions: loss of bases in sample compared to reference
- Inversions
 - Orientation of sequence differs between sample and reference
- Tandem duplications

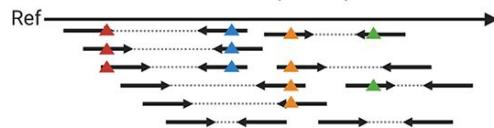
Read mapping vs. de novo genome sequence assembly

A NGS variant calling

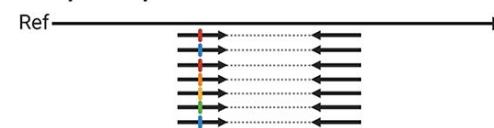
Single nucleotide variants (SNVs)



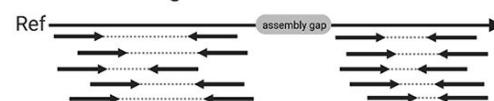
Small insertion/deletions (InDels)



Collapsed repeats



Inaccessible regions

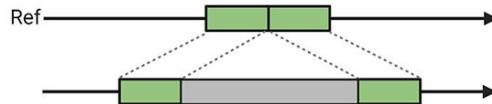


B long read variant calling

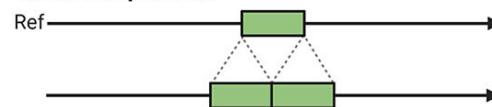
Deletion



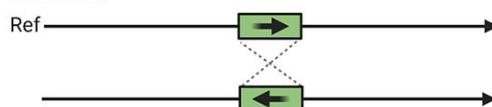
Insertion



Tandem duplication

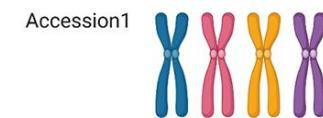


Inversion

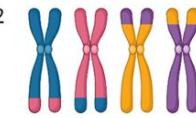


C de novo assembly

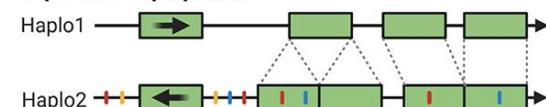
Chromosomal rearrangements



Accession2:



Separated haplophases

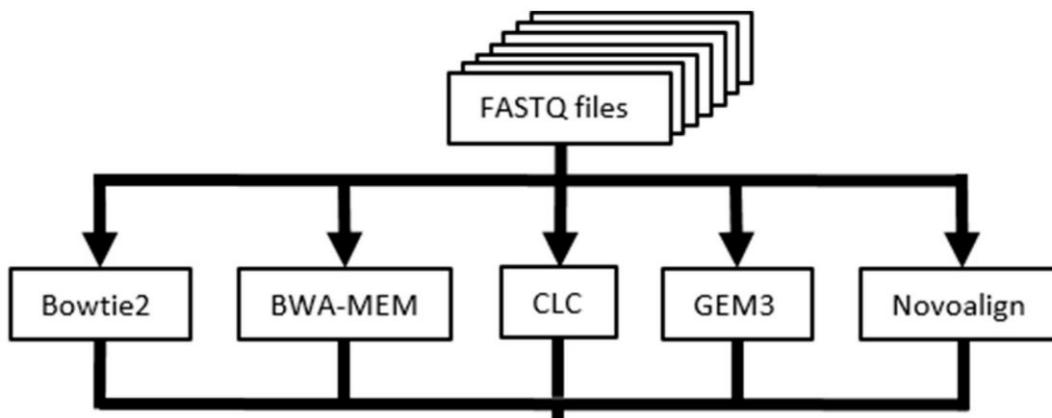


Challenges of read mappings

- Speed / computational costs
 - BLAST would be too slow
- Specific assignment of reads / repeats
 - (especially challenging in polyploid species)
- Splitting around InDels or across introns

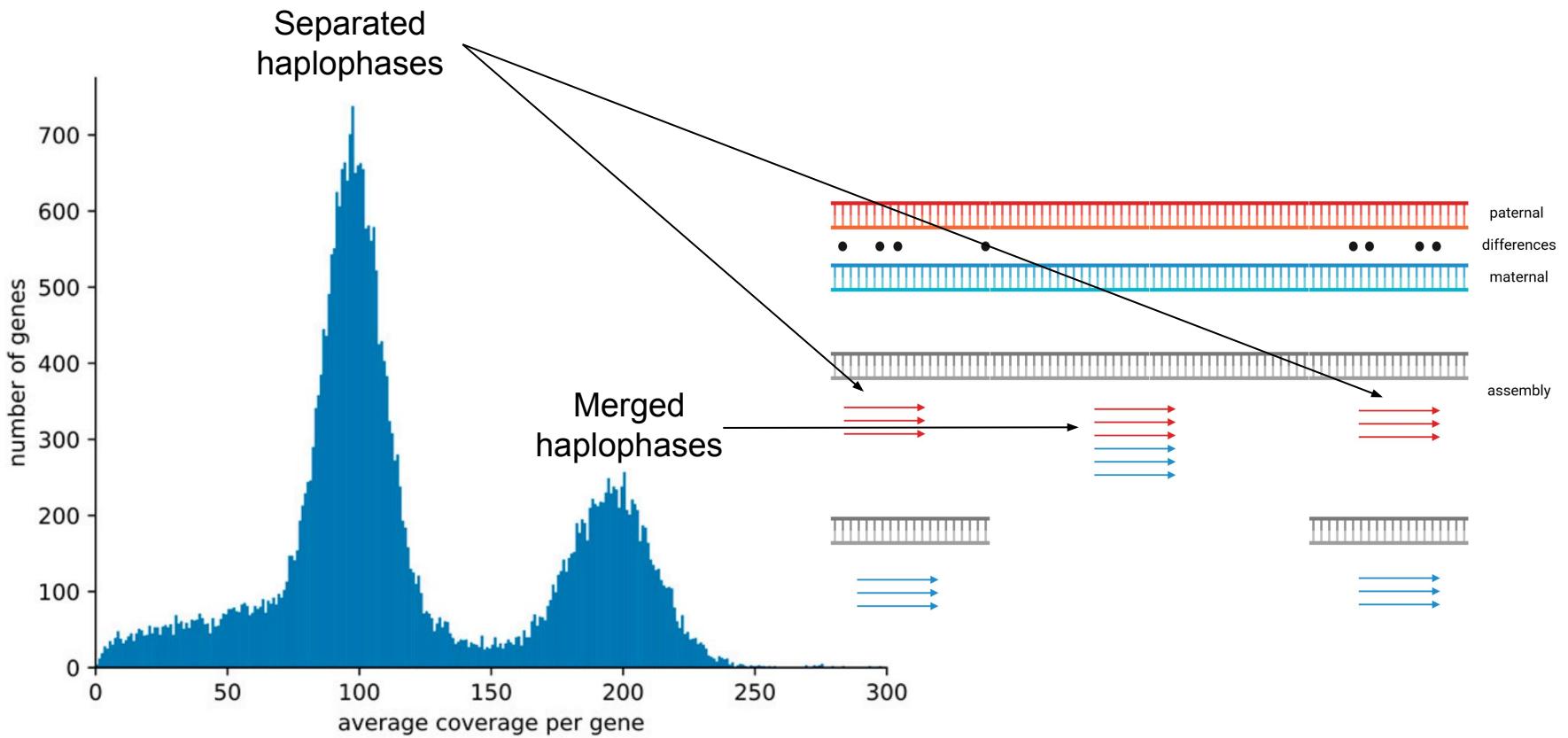
Short read mappings

- Large number of tools for mappings
- DNA seq read mappers: BWA MEM, bowtie2
- RNA-seq read mappers (split read): STAR, HiSAT2

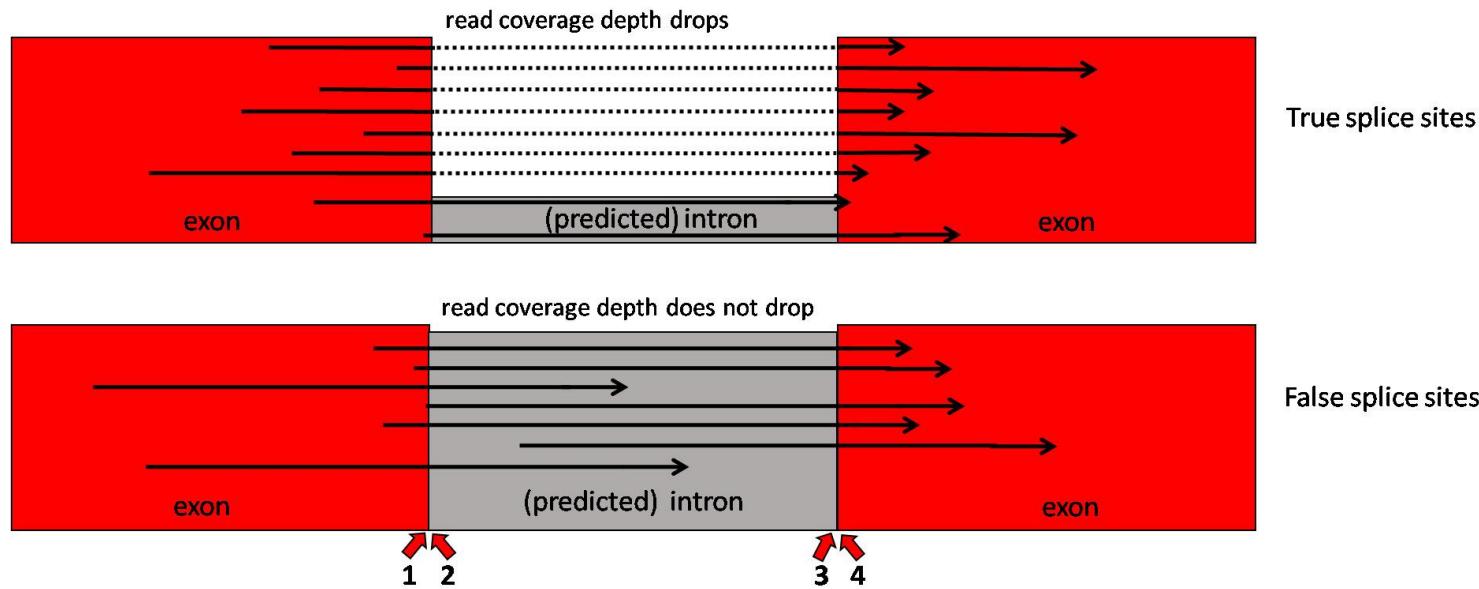


Li, 2013: 10.48550/arXiv.1303.3997
Langmead & Salzberg, 2012: 10.1038/nmeth.1923
Dobin et al., 2013: 10.1093/bioinformatics/bts635
Kim et al., 2015: 10.1038/nmeth.3317
Schilbert et al., 2020: 10.3390/plants9040439

Coverage analysis: haplotype phasing / ploidy



Coverage analysis: splice sites

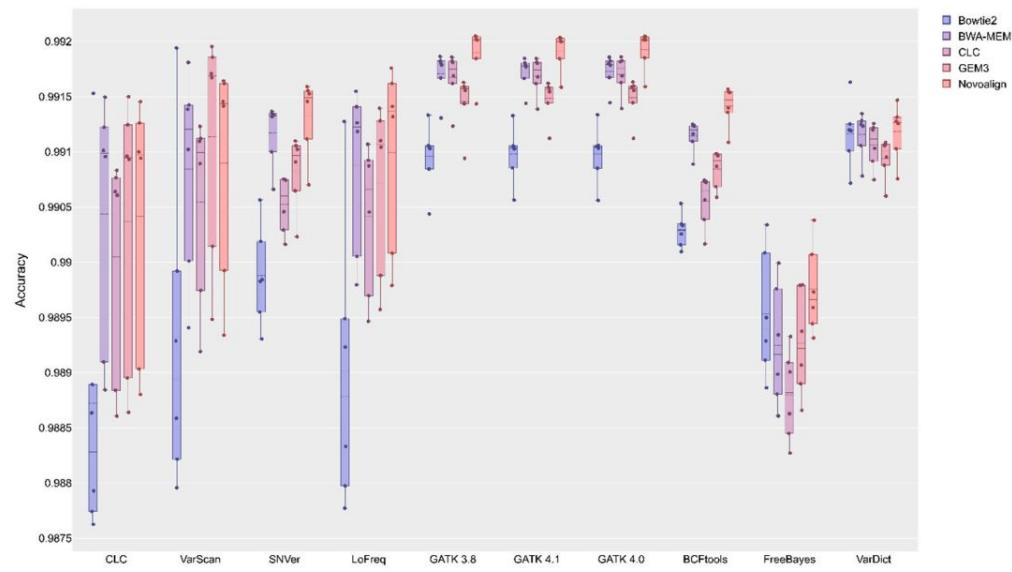
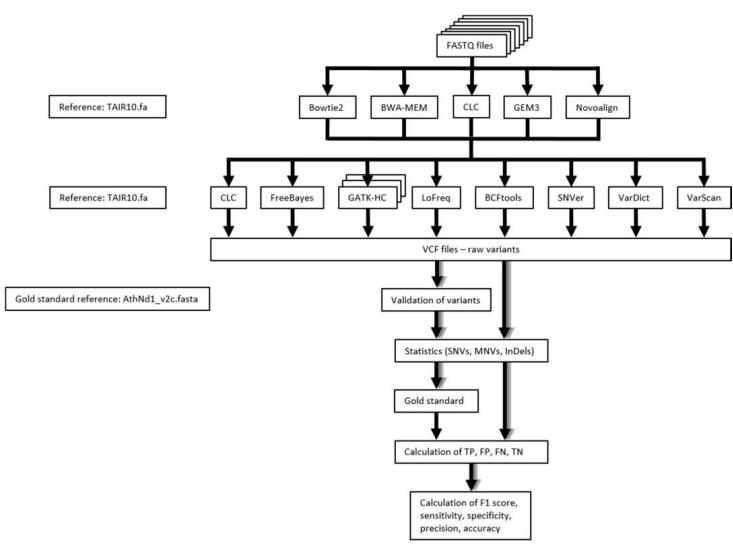


Short read variant callers

- Millions of differences between samples and reference
- Automatic detection with tools required
- Tools:
 - Genome Analysis Tool Kit (GATK): <https://gatk.broadinstitute.org/hc/en-us>
 - Samtools: <http://www.htslib.org/>
 - VarDict: <https://anaconda.org/bioconda/vardict>
 - VarScan: <https://github.com/dkoboldt/varsan>
 - SNVer: <http://snver.sourceforge.net/>
 - FreeBayes: <https://github.com/freebayes/freebayes>

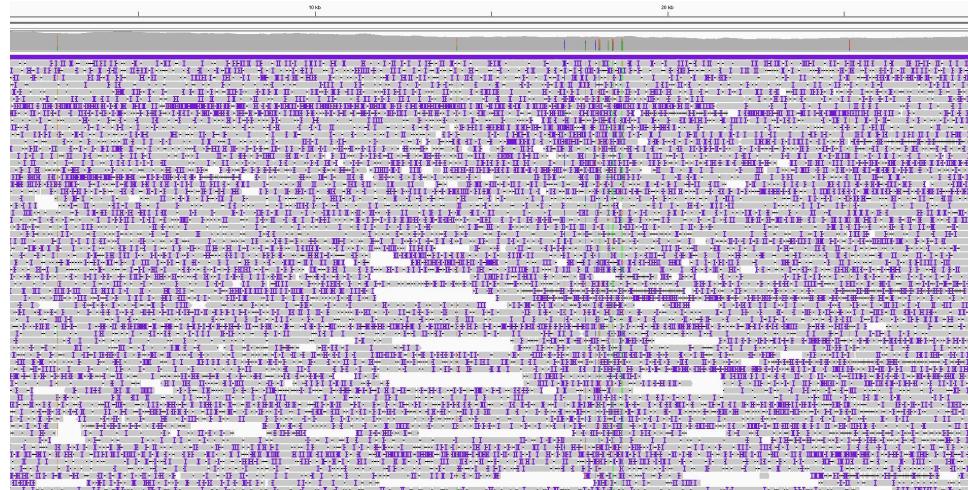
McKenna et al., 2010: 10.1101/gr.107524.110
Li et al., 2009: 10.1093/bioinformatics/btp352
Lai et al., 2016: 10.1093/nar/gkw227
Koboldt et al., 2009: 10.1093/bioinformatics/btp373
Wei et al., 2011: 10.1093/nar/gkr599
Garrison & Marth, 2012: 10.48550/arXiv.1207.3907

Benchmarking of variant callers



Long read mapping

- Noisy alignments due to high rate of sequencing errors
- InDels are particularly abundant (purple ticks)
- Rapid development of new tools
- Benchmarking studies are missing and would require frequent updates



Long read mapping tools

- Collection of long read tools: <https://long-read-tools.org/>
- Over 600 tools available
- Examples:
 - BWA-SW: <http://bio-bwa.sourceforge.net/>
 - Minimap2: <https://github.com/lh3/minimap2>
 - GraphMap: <https://github.com/isovic/graphmap>
 - NGMLR: <https://github.com/phlres/ngmlr>
 - Winnowmap2: <https://github.com/marbl/Winnowmap>

Amarasinghe et al., 2021: 10.1093/gigascience/giab003
Li & Durbin, 2010: 10.1093/bioinformatics/btp698
Li, 2018: 10.1093/bioinformatics/bty191
Sovic, 2016: 10.1038/ncomms11307
Sedlazeck et al., 2018: 10.1038/s41592-018-0001-7
Jain et al., 2022: 10.1038/s41592-022-01457-8

Long read variant calling

- Rapid development of new tools and lack of benchmarking studies
- Tools:
 - SVIM2: <https://github.com/eldariont/svim>
 - Longshot: <https://github.com/pjedge/longshot>
 - NanoCaller: <https://github.com/WGLab/NanoCaller>
 - Sniffles2: <https://github.com/fritzsedlazeck/Sniffles>

Heller & Vingron, 2019: 10.1093/bioinformatics/btz041
Edge & Basal, 2019: 10.1038/s41467-019-12493-y
Ahsan et al., 2021: 10.1186/s13059-021-02472-2
Smolka et al., 2022: 10.1101/2022.04.04.487055

Variant Call Format (VCF)

- Chromosome: name of sequence in the reference
 - Position: position on the specified sequence
 - ID (not relevant in plant biology): variants in humans have IDs
 - Reference allele: nucleotide(s) in the reference sequence at the specified position
 - Alternative allele: nucleotide(s) in the sample at the same position
 - Quality: tool specific value that can be used for filtering
 - Filter status: specifies filter name if variant was filtered out
 - Info: collection of information
 - Format: explains the fields in the following sample columns
 - DATA_SET1, DATA_SET2, DATA_SET3, ...: one column per sample

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	DATA_SET
R105	10929	.	A	G	58.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=-1.242;DP=8;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=44.81;MRankSum=-2.100;QD=7.33;ReadPosRankSum=-0.619;SOR=0.307	GT:AD:DP:G0:PL	0/1;1,6,2,8;66,66,0,230
R105	10944	.	A	G	58.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.068;DP=8;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=44.81;MRankSum=-2.100;QD=7.33;ReadPosRankSum=1.465;QD=0.307	GT:AD:DP:G0:PL	0/1;1,6,2,8;66,66,0,230
R105	10955	.	G	T	61.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.068;DP=7;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=44.16;MRankSum=1.981;QD=0.81;ReadPosRankSum=1.465;QD=0.446	GT:AD:DP:G0:PL	0/1;1,5,2,7;69,69,0,204
R105	10962	.	A	G	61.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.068;DP=8;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=44.91;MRankSum=1.981;QD=0.81;ReadPosRankSum=1.465;QD=0.446	GT:AD:DP:G0:PL	0/1;1,5,2,7;69,69,0,204
R105	13234	.	T	C	32.64	MQ_filter;QD_filter	AC=1;AF=0.500;AN=2;BaseRankSum=1.062;DP=35;ExcessHet=3.0193;FS=2.78;MLEAC=1;MLEAF=0;MQ=37.70;MRankSum=-1.106;QD=0.31;ReadPosRankSum=-0.289;SOR=1.911	GT:AD:DP:G0:PL	0/1;2,23,3;25:40;40,0,706
R105	13260	.	T	G	72.64	MQ_filter	AC=1;AF=0.500;AN=2;BaseRankSum=0.172;DP=36;ExcessHet=3.0193;FS=6.110;MLEAC=1;MLEAF=0;MQ=38.68;MRankSum=1.349;QD=0.25;ReadPosRankSum=0.243;SOR=3.442	GT:AD:DP:G0:PL	0/1;23,6,29;80,80,0,695
R105	13395	.	T	A	183.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.979;DP=19;ExcessHet=3.0193;FS=0.929;MLEAC=1;MLEAF=0;MQ=43.94;MRankSum=1.631;QD=0.67;ReadPosRankSum=1.186;S0.3126	GT:AD:DP:G0:PL	0/1;13,16;19;99;191,0,413
R105	13538	.	T	C	61.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.359;DP=27;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=40.56;MRankSum=1.847;QD=0.47;ReadPosRankSum=1.881;SOR=0.760	GT:AD:DP:G0:PL	0/1;21,4,25;69,69,0,660
R105	14511	.	C	T	39.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=0.563;DP=13;ExcessHet=3.0193;FS=0.522;MLEAC=1;MLEAF=0;MQ=40.71;MRankSum=2.771;QD=0.35;ReadPosRankSum=2.217;SOR=1.546	GT:AD:DP:G0:PL	0/1;9,4,13;47,47,0,352
R105	14542	.	G	A	51.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.825;DP=15;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=42.36;MRankSum=2.843;QD=0.34;ReadPosRankSum=0.919;SOR=1.022	GT:AD:DP:G0:PL	0/1;10,15;15;59;59,0,390
R105	15681	.	A	G	116.64	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=1.897;DP=43;ExcessHet=3.0193;FS=0.187;MLEAC=1;MLEAF=0;MQ=48.84;MRankSum=2.068;QD=0.92;ReadPosRankSum=0.363;SOR=1.721	GT:AD:DP:G0:PL	0/1;34,6,48;99,12,1,1161
R105	15712	.	C	T	70.64	MQ_filter	AC=1;AF=0.500;AN=2;BaseRankSum=1.307;DP=35;ExcessHet=3.0193;FS=0.262;MLEAC=1;MLEAF=0;MQ=39.58;MRankSum=2.129;QD=0.68;ReadPosRankSum=-0.994;SOR=1.238	GT:AD:DP:G0:PL	0/1;30,34;38;78,78,0,1247
R105	15719	.	C	T	64.64	MQ_filter;QD_filter	AC=1;AF=0.500;AN=2;BaseRankSum=0.166;DP=37;ExcessHet=3.0193;FS=0.209;MLEAC=1;MLEAF=0;MQ=62;MRankSum=-2.020;QD=0.180;ReadPosRankSum=1.360;SOR=1.251	GT:AD:DP:G0:PL	0/1;32,43,36;78,72,0,1310
R105	16037	.	C	A	50.64	MQ_filter	AC=1;AF=0.500;AN=2;BaseRankSum=0.000;DP=7;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=0.308;MRankSum=0.303;QD=7.23;ReadPosRankSum=-0.712;SOR=0.802	GT:AD:DP:G0:PL	0/1;5,2,7;58,58,0,171
R105	16582	.	C	T	58.60	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=-0.712;DP=8;ExcessHet=3.0193;FS=0.680;MLEAC=1;MLEAF=0;MQ=38.22;MRankSum=-0.319;QD=0.73;ReadPosRankSum=0.366;SOR=2.236	GT:AD:DP:G0:PL	0/1;6,2,8;66,66,0,246
R105	17718	.	A	G	79.64	MQ_filter	AC=1;AF=0.500;AN=2;BaseRankSum=0.728;DP=18;ExcessHet=3.0193;FS=7.656;MLEAC=1;MLEAF=0;MQ=40.50;MRankSum=-29.01;ReadPosRankSum=-0.858;QD=0.42;SOR=0.099;SOR=0.041	GT:AD:DP:G0:PL	0/1;13,5,18;87,87,0,392
R105	37909	.	A	G	51.64	MQ_filter	AC=1;AF=0.500;AN=2;BaseRankSum=-0.431;DP=9;ExcessHet=3.0193;FS=17.92;MLEAC=1;MLEAF=0;MQ=37.28;MRankSum=0.549;QD=0.266	GT:AD:DP:G0:PL	0/1;7,2;9;59;59,0,254
R105	45724	.	C	CTTATA	61.60	PASS	AC=1;AF=0.500;AN=2;BaseRankSum=0.218;DP=7;ExcessHet=3.0193;FS=0.000;MLEAC=1;MLEAF=0;MQ=0.343;MRankSum=0.967;QD=0.88;ReadPosRankSum=0.210;SOR=0.446	GT:AD:DP:G0:PL	0/1;5,2,7;69,69,0,204

genomic Variant Call Format (gVCF)

- Generally similar to VCF file
- One entry for each position in the reference
- Not just variants, but also non-variants are listed



Variant Annotation - SnpEff

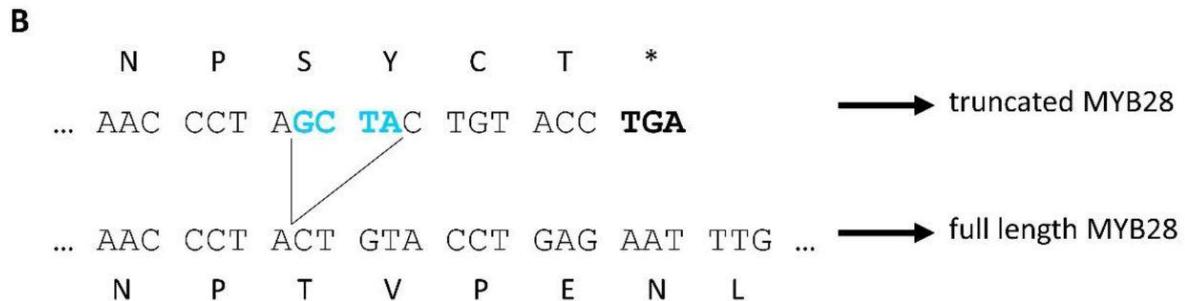
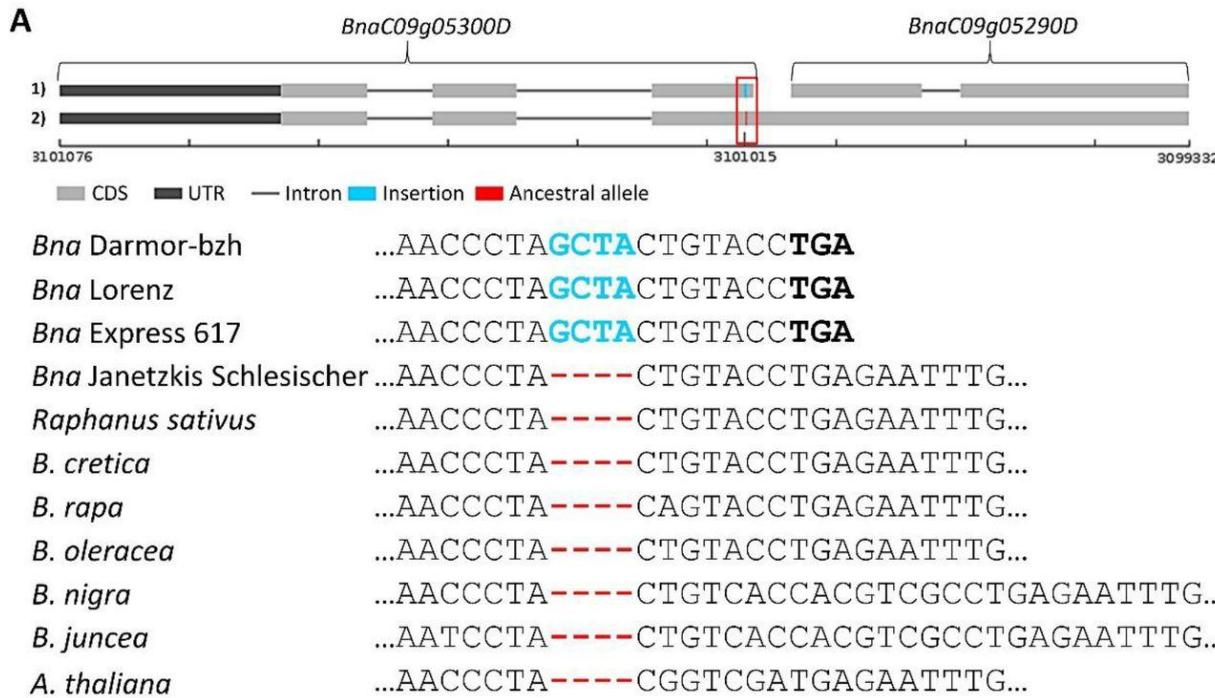
- High throughput annotation of SNVs and larger variants
- Reference sequence (FASTA) and structural annotation (GFF) required
- Freely available software: <http://pcingola.github.io/SnpEff/>
- Addition of one annotation field to VCF file

```
chrA01 179964 . T A . PASS GSL_High,GSL_Low;ANN=A|stop_gained|HIGH|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.863T>A|p.Leu288*|863/1834|863/1834|288/610| GT:AD:DP:GQ:PL
chrA01 179967 . T C . PASS GSL_High,GSL_Low;ANN=C|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.866T>C|p.Phe289Ser|866/1834|866/1834|289/610| GT:AD:DP:GQ:PL
chrA01 179979 . C T . PASS GSL_Low;ANN=T|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.878C>T|p.Thr293Met|878/1834|878/1834|293/610| GT:AD:DP:GQ:PL
chrA01 179988 . A T . PASS GSL_Low;ANN=T|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.887A>T|p.Tyr296Phe|887/1834|887/1834|296/610| GT:AD:DP:GQ:PL
chrA01 180003 . G A . PASS GSL_Low;ANN=A|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.982G>A|p.Gly301Asp|902/1834|902/1834|301/610| GT:AD:DP:GQ:PL
chrA01 180024 . A G . PASS GSL_High,GSL_Low;ANN=G|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.923A>G|p.His308Arg|923/1834|923/1834|308/610| GT:AD:DP:GQ:PL
chrA01 180039 . T C . PASS GSL_High,GSL_Low;ANN=C|missense_variant|Moderate|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.938T>C|p.Ile313Thr|938/1834|938/1834|313/610| GT:AD:DP:GQ:PL
chrA01 180048 . G A . PASS GSL_High,GSL_Low;ANN=A|stop_retained_variant|Low|Gene_BnaA01g00340D|Gene_BnaA01g00340D|transcript|BnaA01g00340D|Coding|5/9|c.947G>A|p.Ter316Ter|947/1834|947/1834|316/610| GT:AD:DP:GQ:PL
```

SnpEff predictions

Type	Meaning	Example
SNP	Single Nucleotide Polymorphism	Reference = 'A', Sample = 'C'
Ins	Insertion	Reference = 'A', Sample = 'AGT'
Del	Deletion	Reference = 'AC', Sample = 'C'
MNP	Multiple Nucleotide Polymorphism	Reference = 'ATA', Sample = 'GTC'
MIXED	MNP and InDel	Reference = 'ATA', Sample = 'GTCAGT'

Example: Glucosinolate biosynthesis in rapeseed



Schilbert et al., 2022: 10.1101/2022.06.01.494149

Variant Annotation - NAVIP

- NAVIP = Neighborhood-Aware Variant Impact Predictor
- Variants can have interacting effects e.g. compensate each other
- Considering multiple variants while predicting effects
- Tool freely available: <https://github.com/bpucker/NAVIP>

wild type:	GAT TCA AGA AGA ATG
peptide:	D S R R M
	A>T A>C
variant 1:	GAT TCA TGA AGA ATG (STOP)
peptide:	D S * R M
variant 2:	GAT TCA AGC AGA ATG (amino acid substitution)
peptide:	D S S R M
combined:	GAT TCA TGC AGA ATG (amino acid substitution)
peptide:	D S C R M

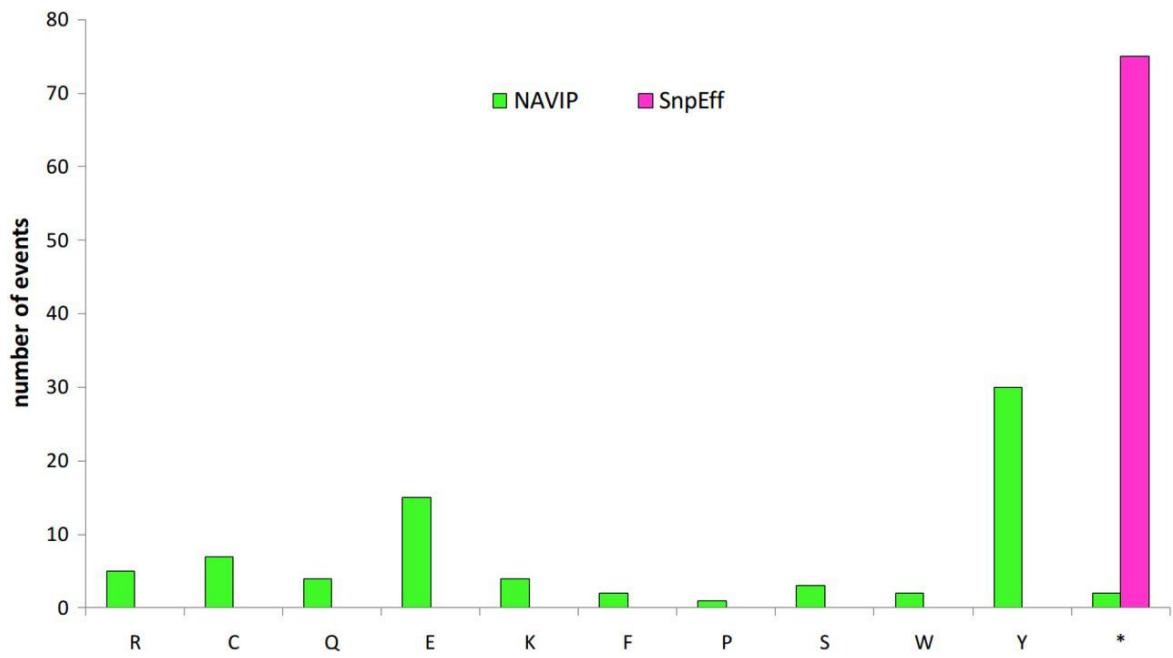
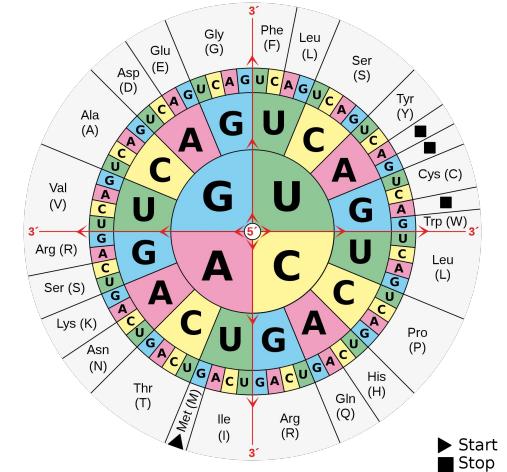
Connected SNVs

wild type:
peptide:
 A>T
 GAT TCA **AGA** AGA ATG
 D S R R M

variant 1:
peptide:
 GAT TCA **TGA** AGA ATG (STOP)
 D S * R M

variant 2:
peptide:
 GAT TCA **AGC** AGA ATG (amino acid substitution)
 D S **S** R M

combined:
peptide:
 GAT TCA **TGC** AGA ATG (amino acid substitution)
 D S **C** R M



Baasner et al., 2019: 10.1101/596718



Compensating InDels

wild type:
peptide:

T>TA C>CGC
GTG TAT CTG CGC ATT
V Y L R I

variant 1:
peptide:

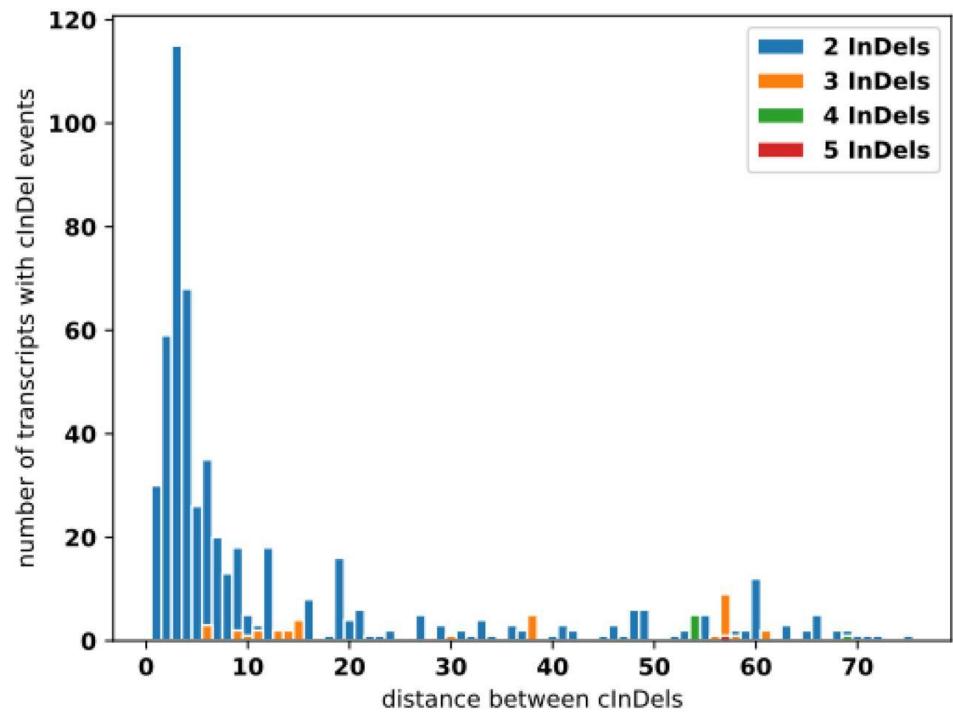
GTG TAT ACT GCG CAT T (frameshift)
V Y T A H

variant 2:
peptide:

GTG TAT CTG CGC GCA TT (frameshift)
V Y L R A

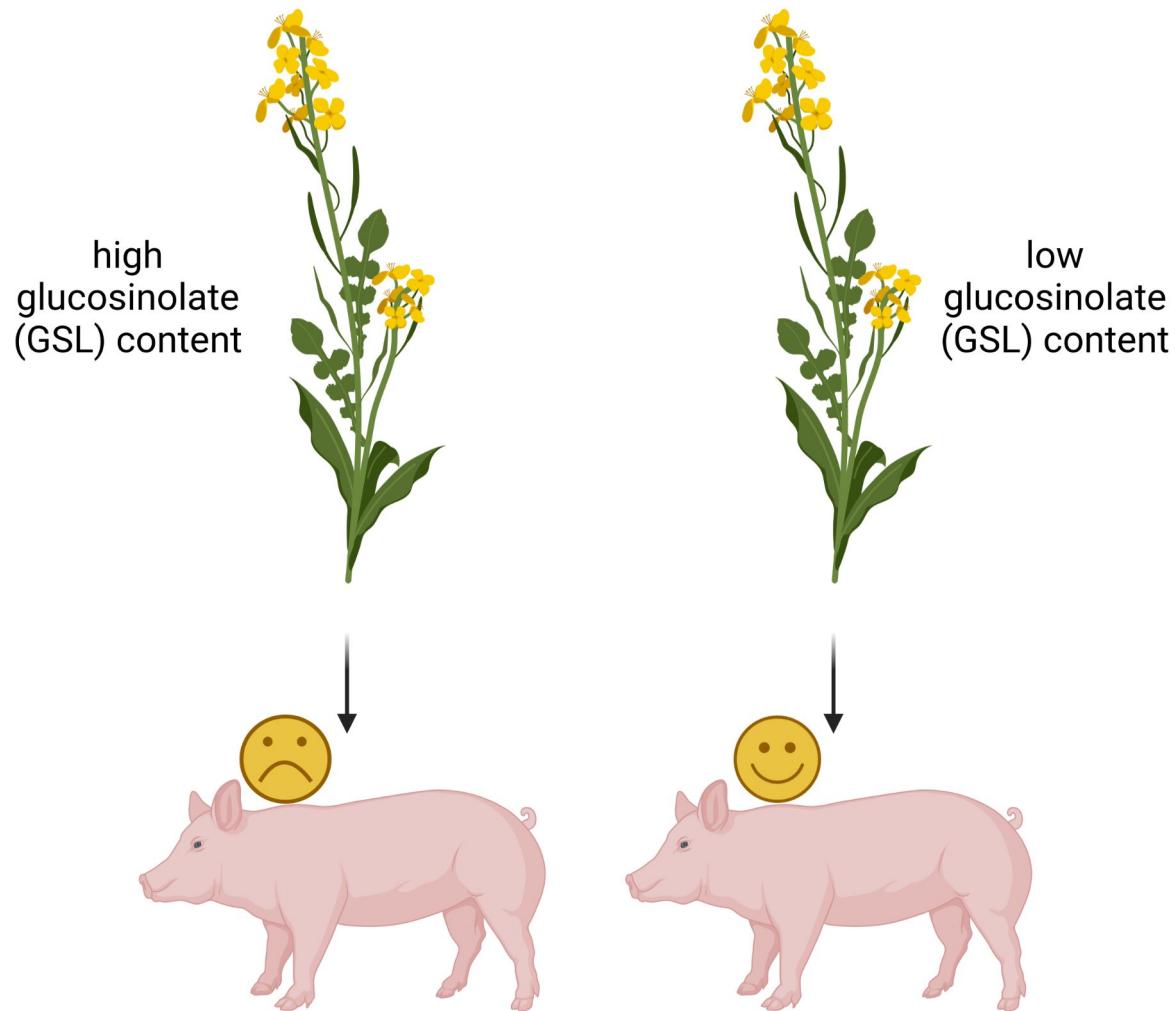
combined:
peptide:

GTG TAT ACT GCG CGC ATT
V Y L A R I

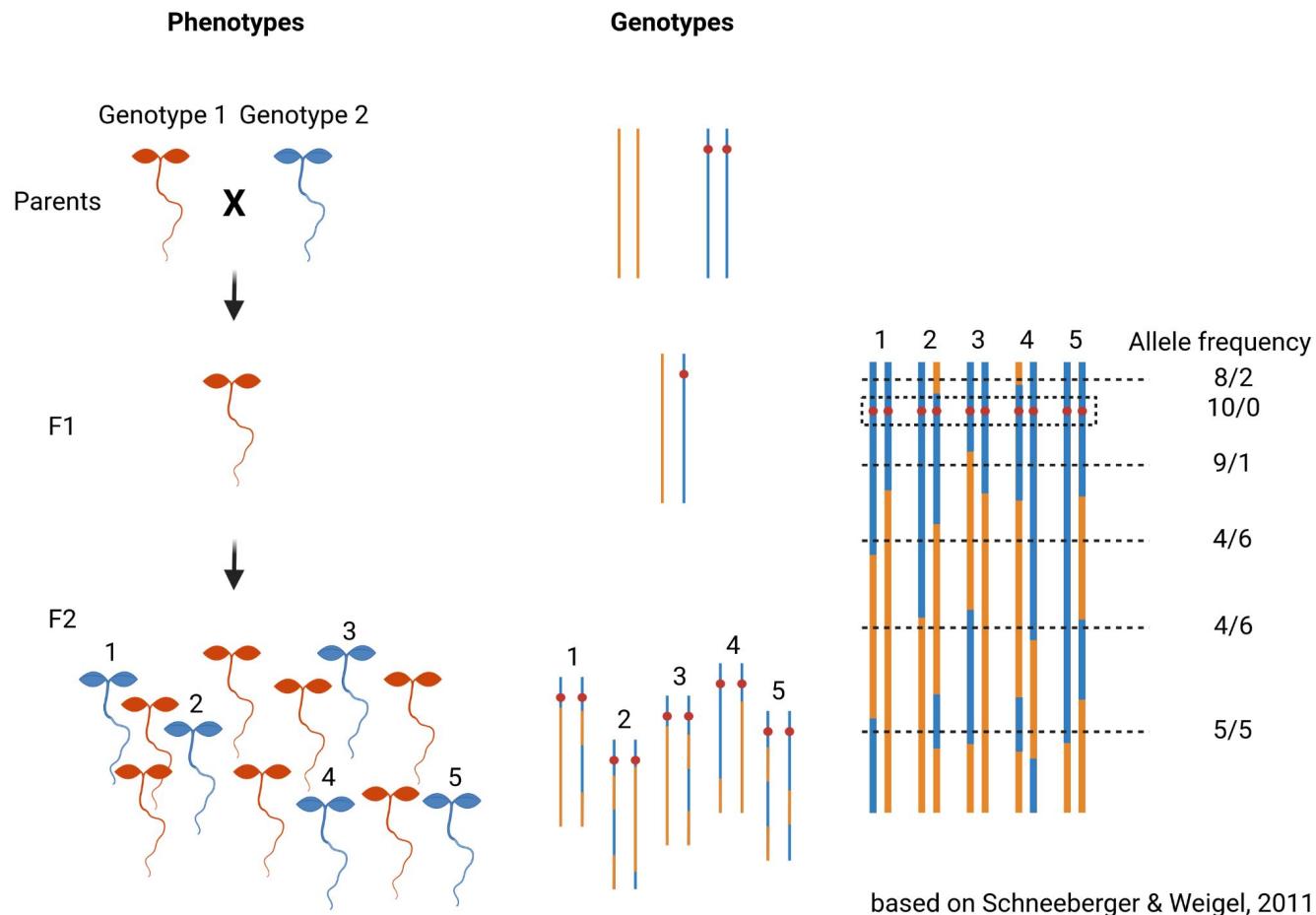


Baasner et al., 2019: 10.1101/596718

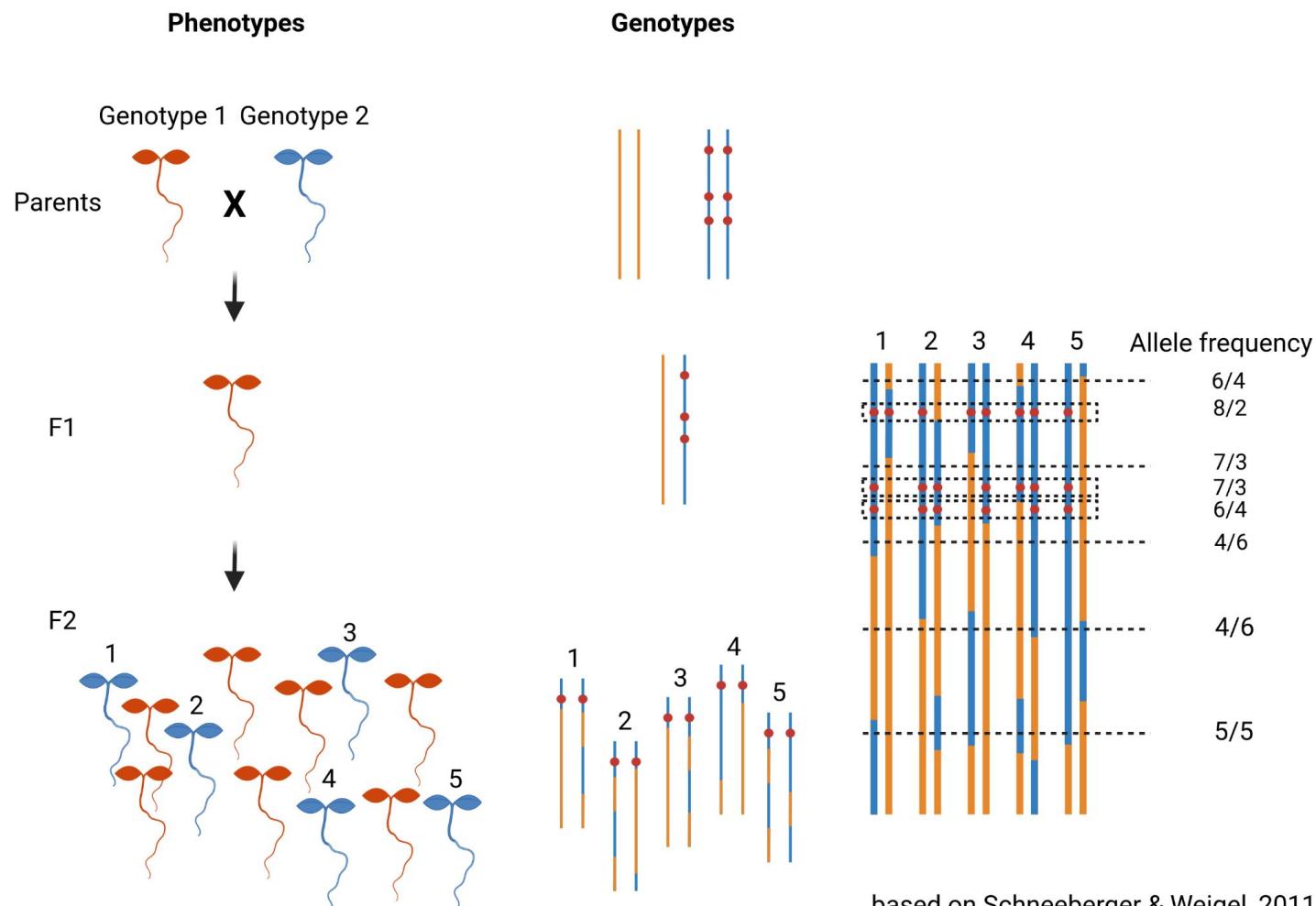
Question: What determines differences between plants?



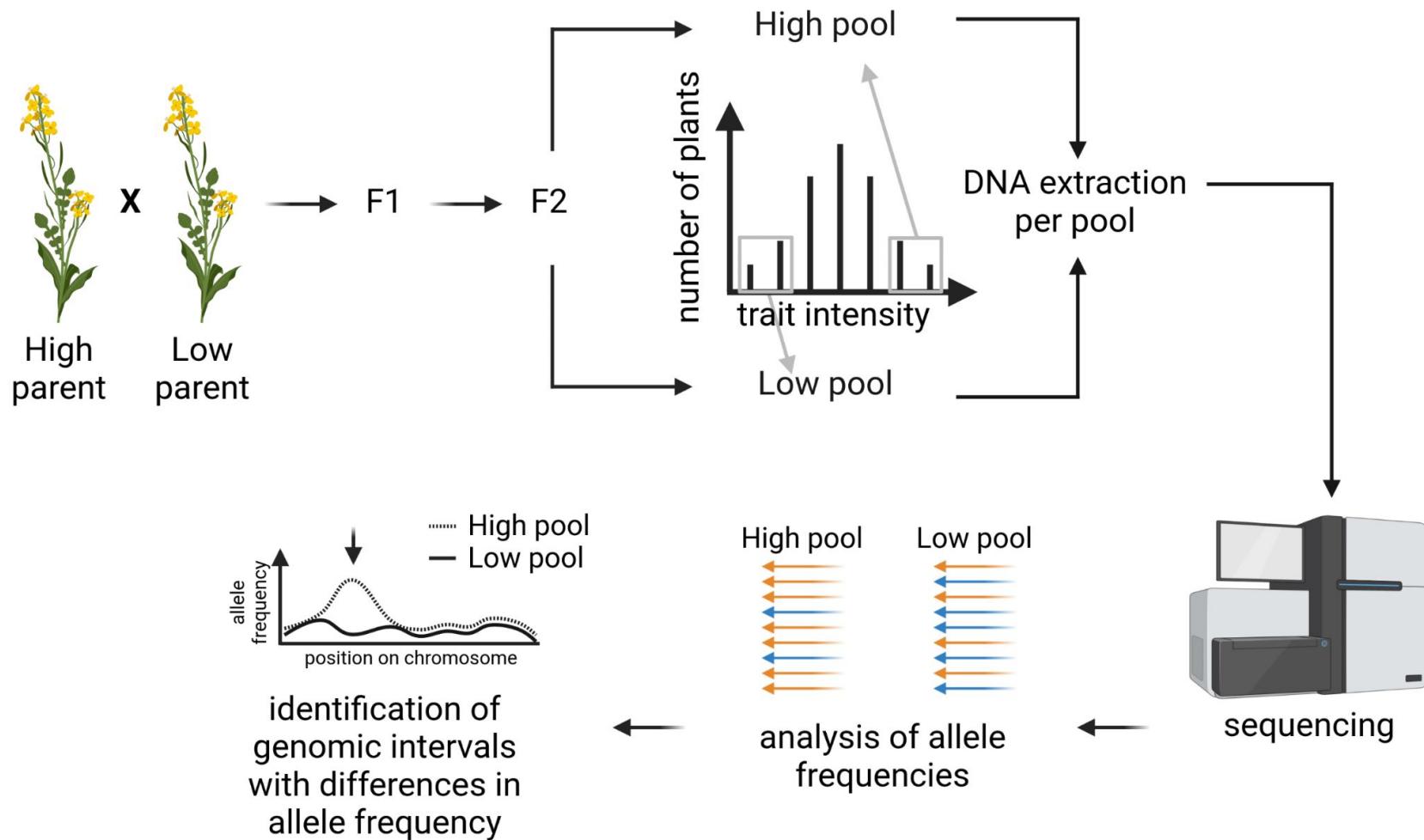
Mapping-by-sequencing (1)



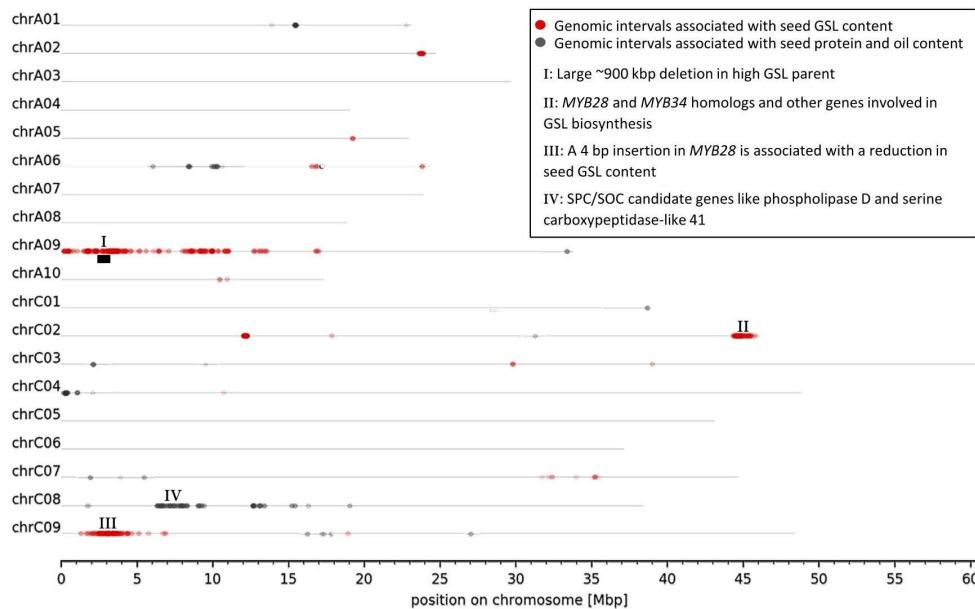
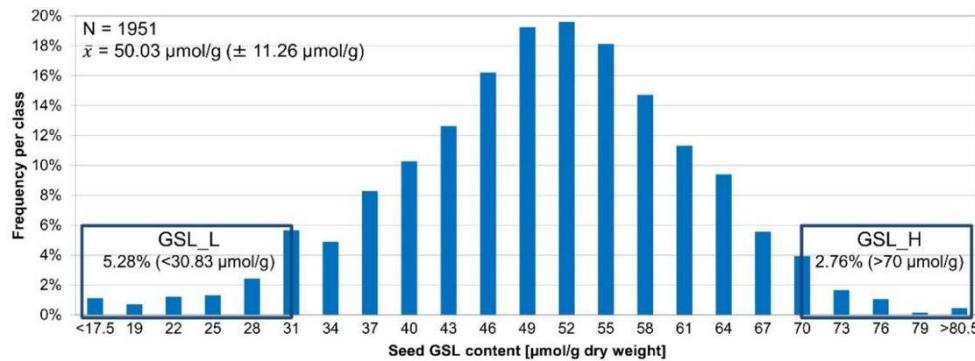
Mapping-by-sequencing (2)



Mapping-by-sequencing (3)



Mapping-by-sequencing (4)

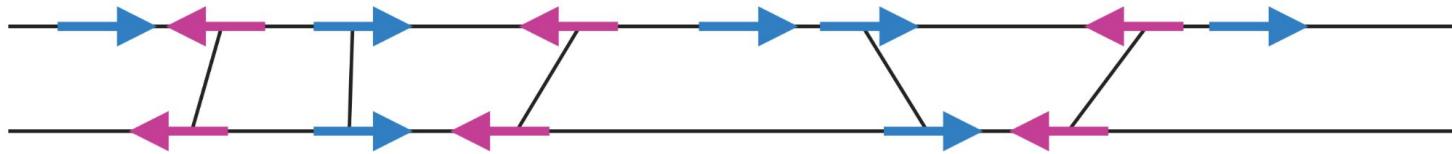


Variant annotation

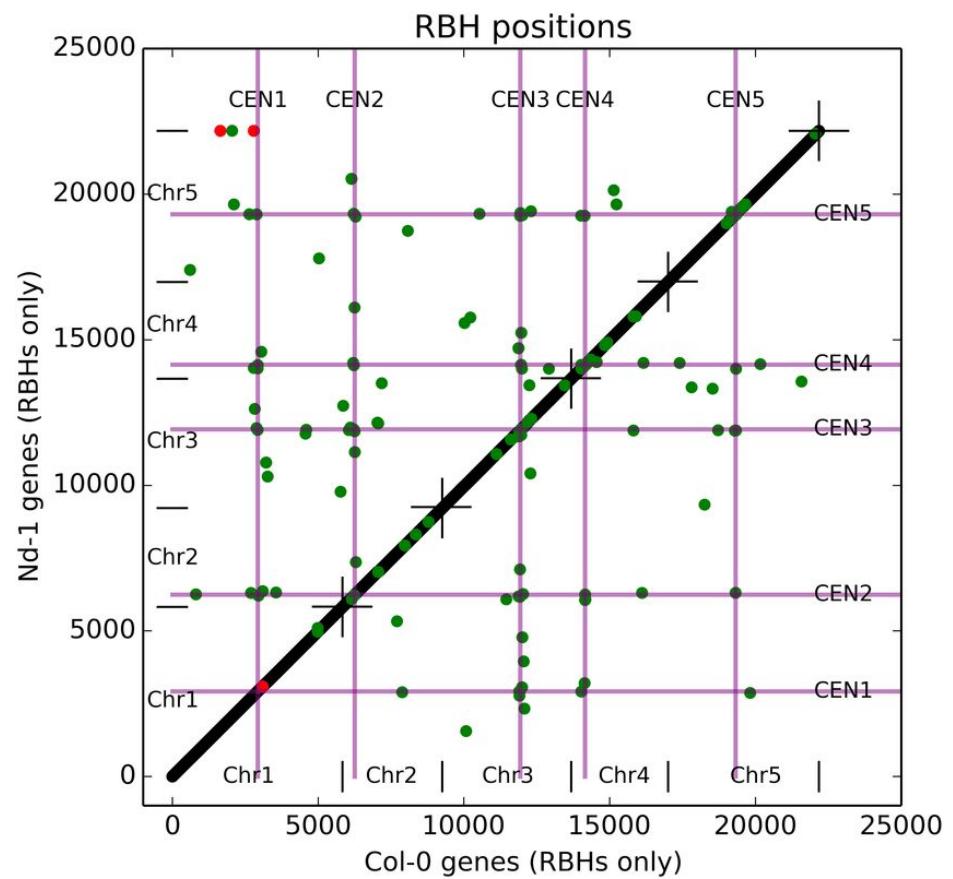
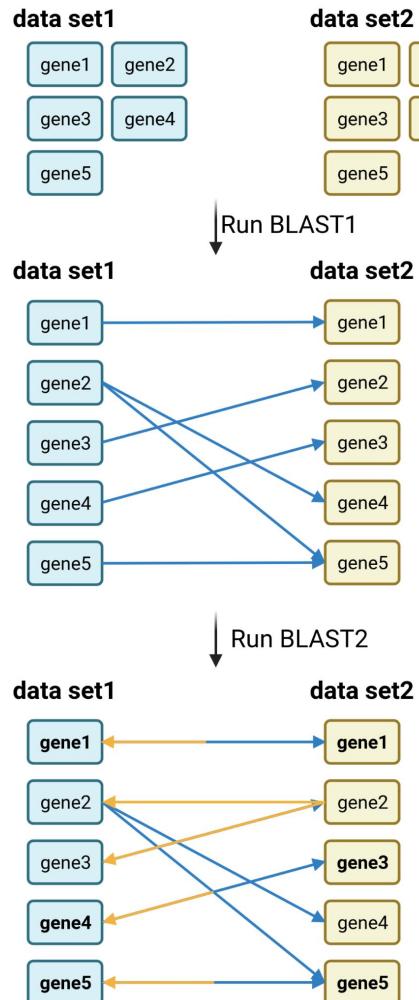
- Variants can have functional consequences
- Variants in coding sequences can change amino acids
- Variants outside CDS can have regulatory consequences

Synteny

Synteny = Same order of genes in different genomes

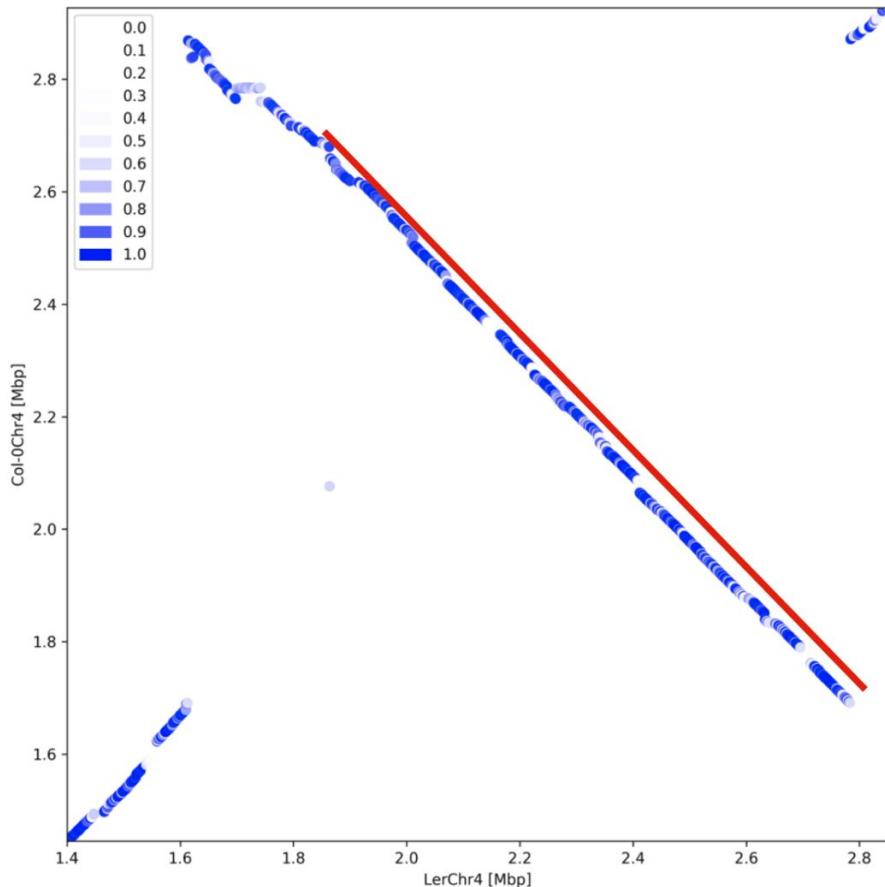
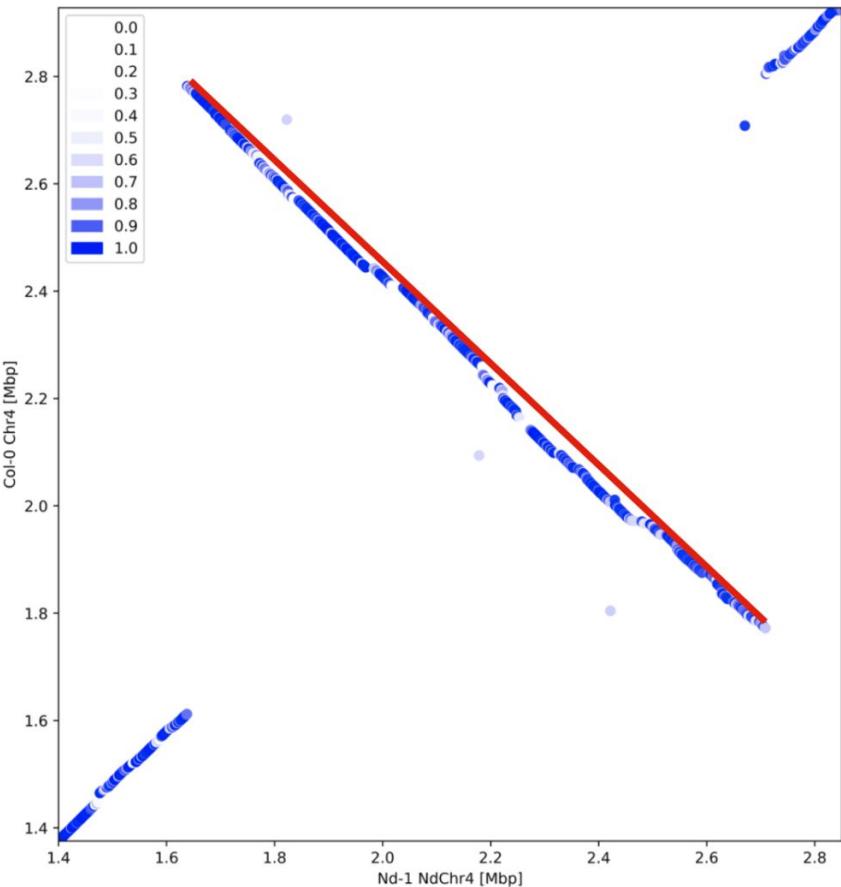


Reciprocal Best BLAST hits (RBHs)



Pucker et al., 2016: 10.1371/journal.pone.0164321

Dot plots



JCVI/MCscan

- 1 Extraction of mRNA sequences based on genomic positions of genes

species1 ——————

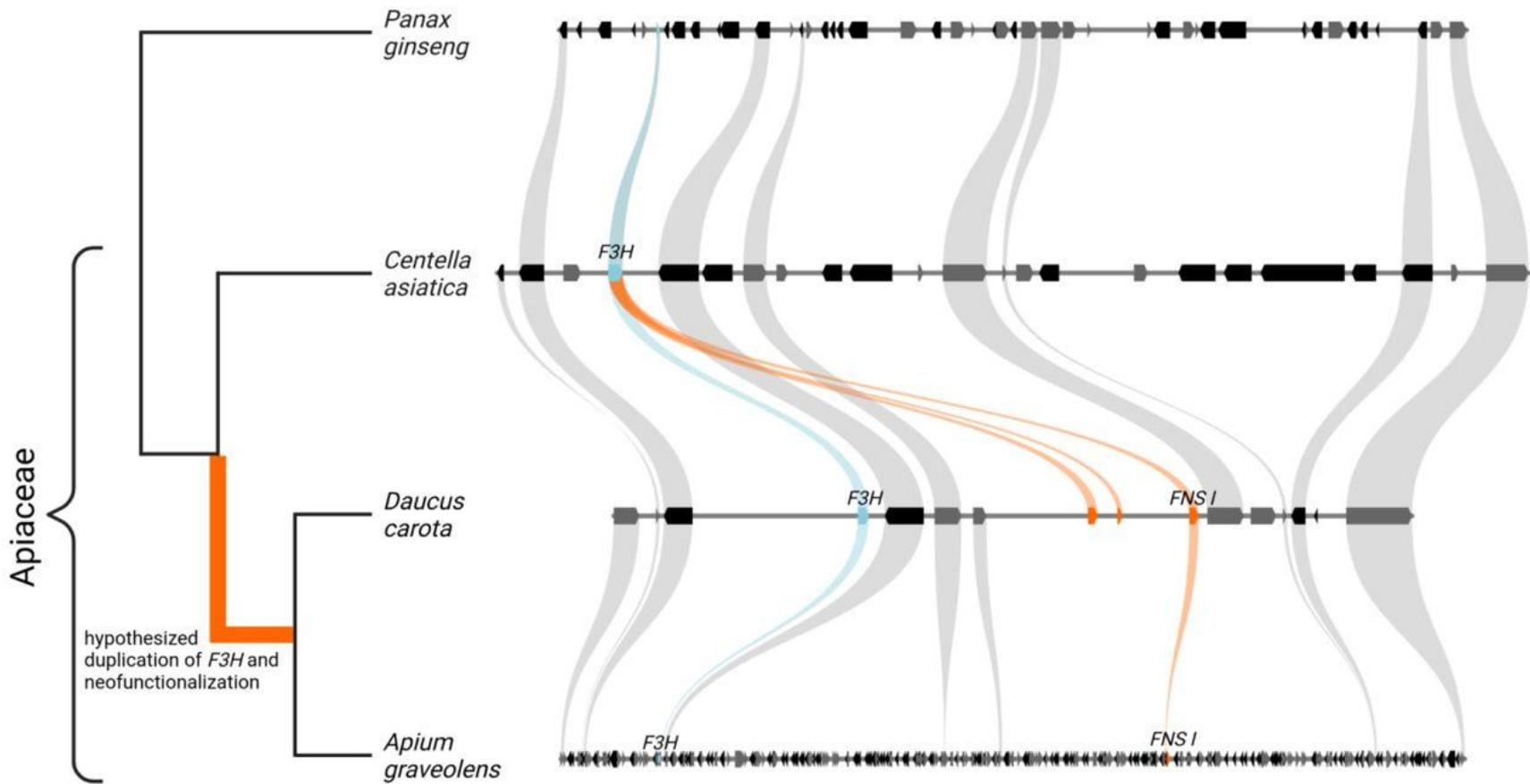
- 2 Comparison of concatenated mRNA sequences via BLAST

species2 ——————

- 3 Identification of syntenic blocks



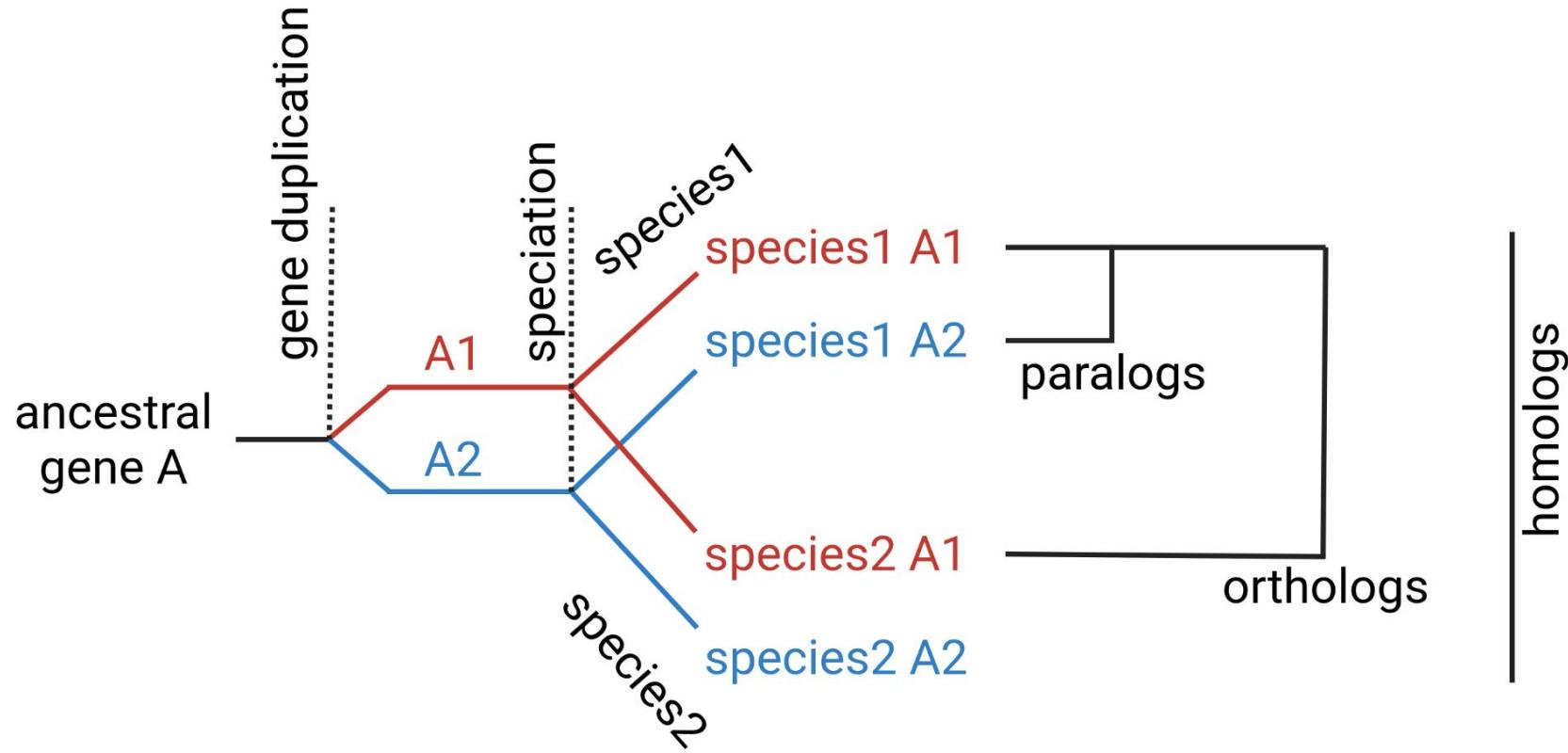
Evolution of FNS I in the Apiaceae



Pucker & Iorizzo, 2022: 10.1101/2022.02.16.480750

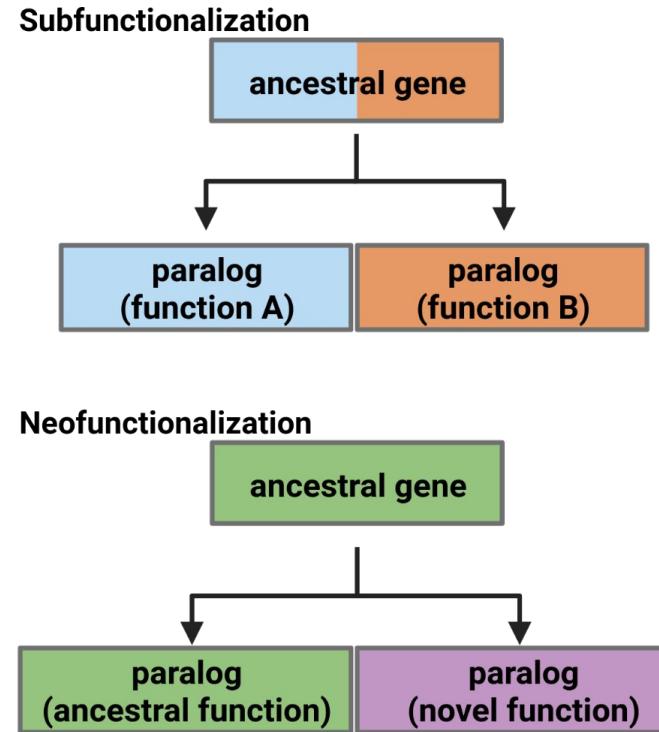


Paralogs & orthologs

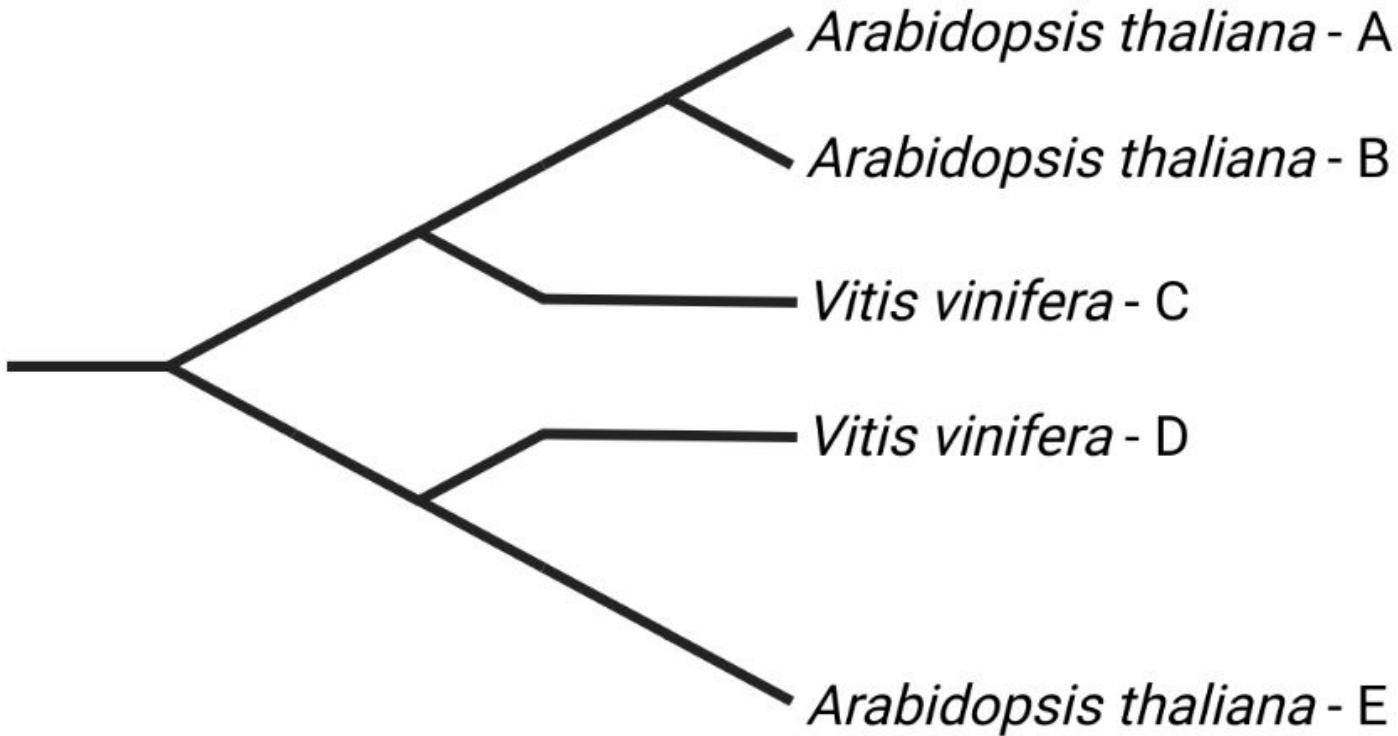


Sub/Neofunctionalization

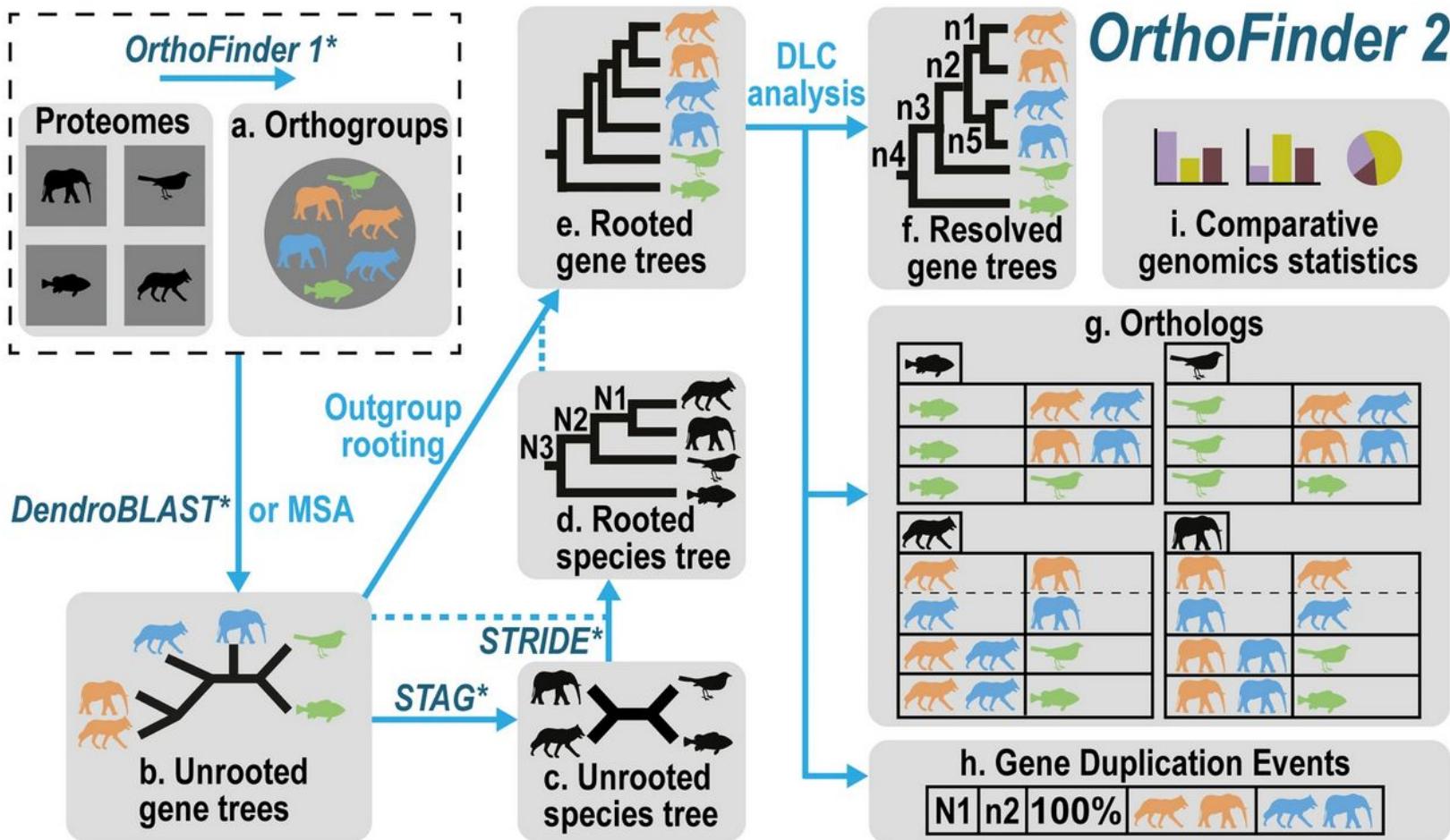
- Sub/neofunctionalization can be achieved through changes in gene expression following the duplication
- Gene copies can be restricted to specific tissues/conditions



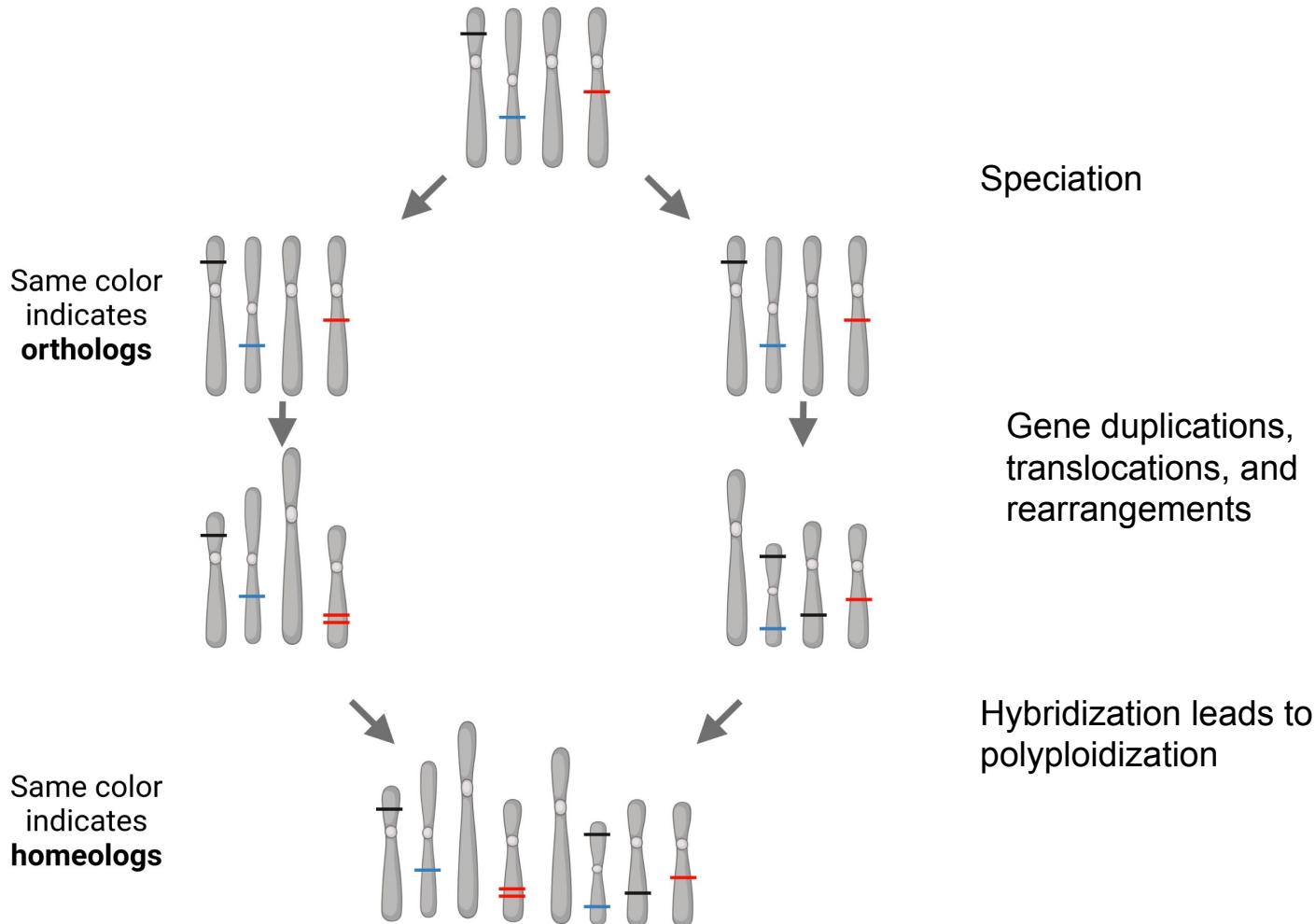
Which are orthologs/paralogs?



OrthoFinder2



Homeologs



Glover et al., 2016: 10.1016/j.tplants.2016.02.005

Summary

- Structural & functional annotation
- Read mapping, variant calling, variant annotation
- Genome-Wide Association Studies (GWAS) / Mapping-By-Sequencing (MBS)
- Comparative genomics / synteny analysis
- Phylogenetic analyses

Time for questions!



Practical part: outline day3

- Gene prediction (BRAKER3)
- Functional annotation (Mercator) & biosynthesis pathway annotation (KIPES3)
- Long read mapping (minimap2)
- Variant calling and annotation (SVIM, SnpEff, NAVIP)
- Synteny analysis (MCscan)
- Alignment construction (MAFFT)
- Phylogenetic tree construction (FastTree2, IQ-TREE2)
- Phylogenetic tree vissualization (iTOL)

Exercises: 3-1

- Install HISAT2 and map some RNA-seq reads to the assembly
- Run gene prediction with BRAKER3
- Run BUSCO on structural annotation
- Perform a functional annotation with Mercator and KIPES3



HISAT2 RNA-seq read mapping

- Split read alignment (RNA-seq vs. genome sequence)
- Availability: <http://daehwankimlab.github.io/hisat2/>
- Usage:

```
hisat2-build ref_genome.fa ref_index
```

```
hisat2 -x ref_index -1 fw_reads.fastq -2 rv_reads.fastq -S output.sam
```

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

```
samtools sort -o sorted_output.bam output.bam
```

```
samtools index sorted_output.bam
```

Gene prediction with BRAKER3 (structural annotation)

- Predicting gene models of protein encoding genes
- Availability: <https://github.com/Gaius-Augustus/BRAKER>
- Usage:

```
braker.pl \
--species=plantX \
--genome=assembly.fasta \
--bam=file1.bam,file2.bam \
--output_dir /vol/data/braker3_output/ \
--threads 10
```

Mercator

- Quick functional annotation of large numbers of polypeptide sequences
- Upload of FASTA file on website: https://www.plabipd.de/mercator_main.html

Mercator4 v6.0

Job submission Result tree viewer Result Heatmap viewer

Sequence type Protein DNA ⓘ

Include Prot-scriber annotations (Beta version) ⓘ

Include Swissprot annotations ⓘ

Upload FASTA file No file selected.

Use demo FASTA file

Job name ⓘ

Email address ⓘ

KIPEs3

- Annotation of genes in a biosynthesis pathway (e.g. flavonoid biosynthesis)
- Availability: <https://github.com/bpucker/KIPEs>
- Usage:

```
python3 KIPEs3.py \
--baits /vol/data/baits/ \
--positions /vol/data/flavbio_residues/ \
--subject /vol/data/your_peps.fasta \
--out /vol/data/kipes_results/
```

Exercises: 3-2

- Map digi1 and digi4 reads with minimap2 to assembly
- Identify sequence variants with SVIM2 based on the mapping
- Predict variant effects with SnpEff and NAVIP
- Perform a synteny analysis with between a digi1 and digi4 assembly with MCscan
- Generate Circos plots comparing digi1 and digit4 assemblies

Long read mapping with minimap2

- Alignment of long reads to genome sequence
- Availability: <https://github.com/lh3/minimap2>
- Usage (ONT reads):

```
minimap2 -t 10 -ax map-ont ref.fasta ont_reads.fastq > alignment.sam
```

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

```
samtools sort -o sorted_alignment.bam alignment.bam
```

```
samtools index sorted_alignment.bam
```

Variant calling with SVIM2

- Detect sequence differences between reads and reference sequence (assembly)
- Availability: <https://github.com/eldariont/svim>
- Usage:

```
svim alignment \
/vol/data/svim_output/ \
--reference reference.fa \
--reads sorted_alignment.bam \
--min_sv_size 50 \
--max_sv_size 100000 \
--min_read_support 5
```

Variant impact prediction with SnpEff

- Predicting the functional consequences of sequence variants
- Availability: <https://pcingola.github.io/SnpEff/>
- Building a database:

```
nano snpEff.config  
my_species.genome : My species  
mkdir -p /vol/data/snpeff/data/my_species  
cp /path/to/genome.fasta /vol/data/snpeff/data/my_species/sequences.fa  
cp /path/to/annotation.gff /vol/data/snpeff/data/my_species/genes.gff  
java -jar snpEff.jar build -gff3 -v my_species  
java -jar snpEff.jar databases | grep my_species
```

- Running variant annotation:

```
java -jar snpEff.jar -v my_species variants.vcf > variants.annotated.vcf
```

Variant impact prediction with NAVIP

- Predicting the functional consequences of sequence variants
- Availability: <https://github.com/bpucker/NAVIP>
- Usage:

```
runnavip.sh \
-i input.vcf \
-g ref.gff \
-f ref.fasta \
-o /vol/data/navip_output/
```

Synteny analysis with MCscan

- Compare genomic region across species
- Availability: [https://github.com/tanghaibao/jcvi/wiki/MCscan-\(Python-version\)](https://github.com/tanghaibao/jcvi/wiki/MCscan-(Python-version))
- Usage:

```
# Convert GFF to BED format
python -m jcvi.formats.gff bed --type=mRNA /path/to/genome1.gff > genome1.bed
python -m jcvi.formats.gff bed --type=mRNA /path/to/genome2.gff > genome2.bed

# Index the CDS files
samtools faidx /path/to/genome1.cds
samtools faidx /path/to/genome2.cds

# Create BLAST databases
makeblastdb -in /path/to/genome1.cds -dbtype nucl
makeblastdb -in /path/to/genome2.cds -dbtype nucl

# Perform reciprocal BLAST
blastn -query /path/to/genome1.cds -db genome2.cds -out genome1_vs_genome2.blast -evalue 1e-5 -outfmt 6 -num_threads 4
blastn -query /path/to/genome2.cds -db genome1.cds -out genome2_vs_genome1.blast -evalue 1e-5 -outfmt 6 -num_threads 4

# Run JCVI MCScanX
python -m jcvi.compara.catalog ortholog --dbtype nucl --blastp genome1 genome2

# Generate a dot plot
python -m jcvi.graphics.dotplot dotplot genome1.genome2.anchors
```

Circos plot generation

- Comparison of genome sequences
- Inclusion of genome-wide distribution of various elements possible
- Availability: <https://circos.ca/>



GUIDE IMAGES SOFTWARE DOCUMENTATION PRESENTATIONS NEWS CITATIONS SUPPORT

DATA VISUALIZATION CIRCULAR APPROACH FEATURES GENOMIC DATA GENERAL DATA TABULAR VISUALIZATION PUBLISHED IMAGES

Circos > introduction

WHAT IS CIRCOS?

CIRCULAR VISUALIZATION

Circos is a software package for [visualizing data and information](#). It visualizes data in a [circular layout](#) — this makes Circos ideal for exploring relationships between objects or positions. There are [other reasons](#) why a circular layout is advantageous, not the least being the fact that it is attractive.

Circos is ideal for creating publication-quality infographics and illustrations with a high [data-to-ink ratio](#), richly layered data and pleasant symmetries. You have fine control each element in the figure to tailor its focus points and detail to your audience.

Four circular Circos plots are shown side-by-side, each displaying a complex network of interactions between genomic elements. The plots use various colors (green, blue, red, yellow) to distinguish different data series or chromosomes. The inner and outer rings represent different layers of data, such as protein-protein interactions or regulatory elements.

Exercises: 3-3

- Collect sequences (NCBI)
- Construct alignment (MAFFT)
- Phylogenetic tree construction (FastTree2, IQ-TREE2)
- Visualization (iTOL)

NCBI as sequence source

- NCBI offers huge collection of nucleotide and polypeptide sequences
- Links to references (publications) available
- Selection of specific plant taxa possible
- Download in FASTA format possible

The screenshot shows the NCBI search interface with the query 'chalcone synthase'. The results page displays 20 items out of 20609. Each result entry includes a checkbox, the protein name, species, accession number, GI number, BioProject, Nucleotide, PubMed, Taxonomy, GenPept, Identical Proteins, FASTA, and Graphics links. A sidebar on the right lists taxonomic filters and related data.

See Gene information for chalcone synthase synthase
chalcone synthase in *Arabidopsis thaliana* 1 Gene record
synthase in *Hibiscus syriacus* (3) All 3 Gene records

See the results of this search (7036 items) in our new [Identical Protein Groups](#) database.

Items: 1 to 20 of 20609

<< First < Prev Page 1 of 1031 Next > Last >>

[chalcone synthase \[Vitis vinifera\]](#)
1. 393 aa protein
Accession: NP_001267879.1 GI: 526117870
BioProject Nucleotide PubMed Taxonomy
GenPept Identical Proteins FASTA Graphics

[chalcone synthase \[Vitis vinifera\]](#)
2. 454 aa protein
Accession: NP_001268064.1 GI: 526117591
BioProject Nucleotide PubMed Taxonomy
GenPept Identical Protein FASTA Graphics

[chalcone synthase \[Rhizobium favelukesii\]](#)
3. 350 aa protein
Accession: CDM6559.1 GI: 584451655
BioProject Nucleotide Taxonomy
GenPept Identical Proteins FASTA Graphics

[chalcone synthase \[Medicago sativa\]](#)
4. 391 aa protein
Accession: WFF64131.1 GI: 2475025180
Nucleotide Taxonomy
GenPept Identical Proteins FASTA Graphics

Taxonomic Groups [List]
eukaryota (1252)
green plants (9842)
vascular plants (9757)
seed plants (9703)
more... (54)
more... (85)
fungi (710)
ascomycete fungi (694)
more... (16)
animals (11)
more... (15)
bacteria (9988)
actinobacteria (8245)
high G+C Gram-positive bacteria (8240)
more... (5)
firmicutes (816)
proteobacteria (541)
α-proteobacteria (391)
γ-proteobacteria (142)
more... (8)
CFB group bacteria (85)
cyanobacteria (49)
more... (251)
unclassified sequences (5)
other sequences (5)
archaea (1)

Find related data
Database: Select

chalcone synthase [Vitis vinifera]

NCBI Reference Sequence: NP_001267879.1

[GenPept](#) [Identical Proteins](#) [Graphics](#)

>NP_001267879.1 chalcone synthase [Vitis vinifera]
MVSVAEIRKAQRAEGPATVLAIGTATPANCYQADYPDYYFRITNSEHMTTELKEFKRMCESINKRYM
HLTEELILKENPNVCAYMAPSLARQDMVVVEVPKLGKEAAAKAIKEWQPKSKITHLVFCTTSGVDMMPGA
DYQLTLLGLKPSVKRLMMYQQGCFAGGTVLRLAKDLAENNAGSRVLVVCSEITAVTFRGPSDTHLDLSV
GQALFGDGAAAVIIGADPTKIERPLFELVSAAQTILPDSEGAIDGHLREVGLTFHLLKDPGLISKNIE
KSLVEAFTPIGISDWNSLFWIAHPGGPAILDQVELKLGKEEKLRATRHLSEYGNMSSACVLFIDEMR
KKSIEEGKGSTGEGLEWGVLFGFGLTVETVVLHSVAPAAH

Alignment construction with MAFFT

- Generation of multiple sequence alignment
- Thousands of sequences can be processed efficiently
- Input and output of MAFFT is a FASTA file by default
- Availability: <https://mafft.cbrc.jp/alignment/software/>
- Usage:

```
mafft sequences.fasta > sequences.fasta.aln
```

Phylogenetic tree construction (FastTree2, IQ-TREE2)

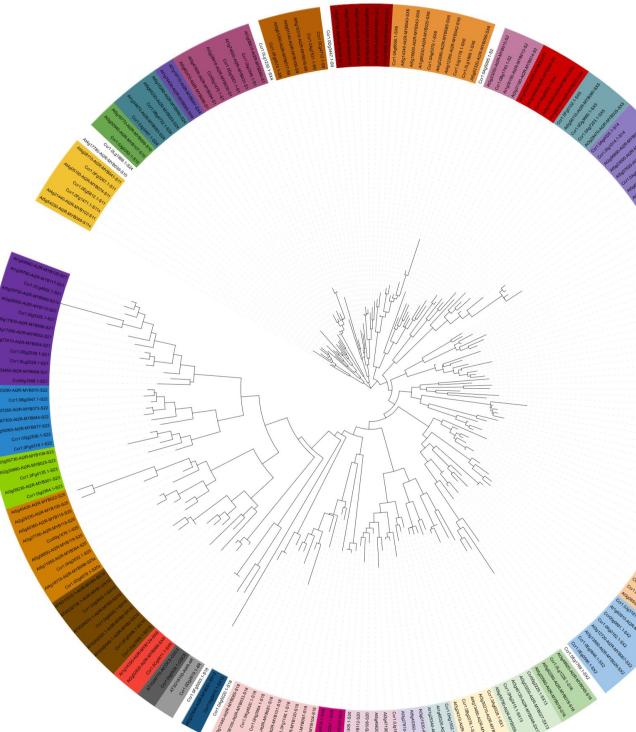
- Construction of a phylogenetic tree from alignment
- Availability:
 - FastTree2: <http://www.microbesonline.org/fasttree/>
 - IQ-TRE2: <http://www.iqtree.org/>
- Usage:

```
FastTree -wag alignment.fasta > tree.nwk
```

```
iqtree2 -s alignment.fasta -m MFP -bb 1000 -alrt 1000
```

Visualize tree (iTOL)

- Visualization of phylogenetic trees
- Modification, highlighting and labeling of tree
- Availability: <https://itol.embl.de/>



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

Time for questions!

