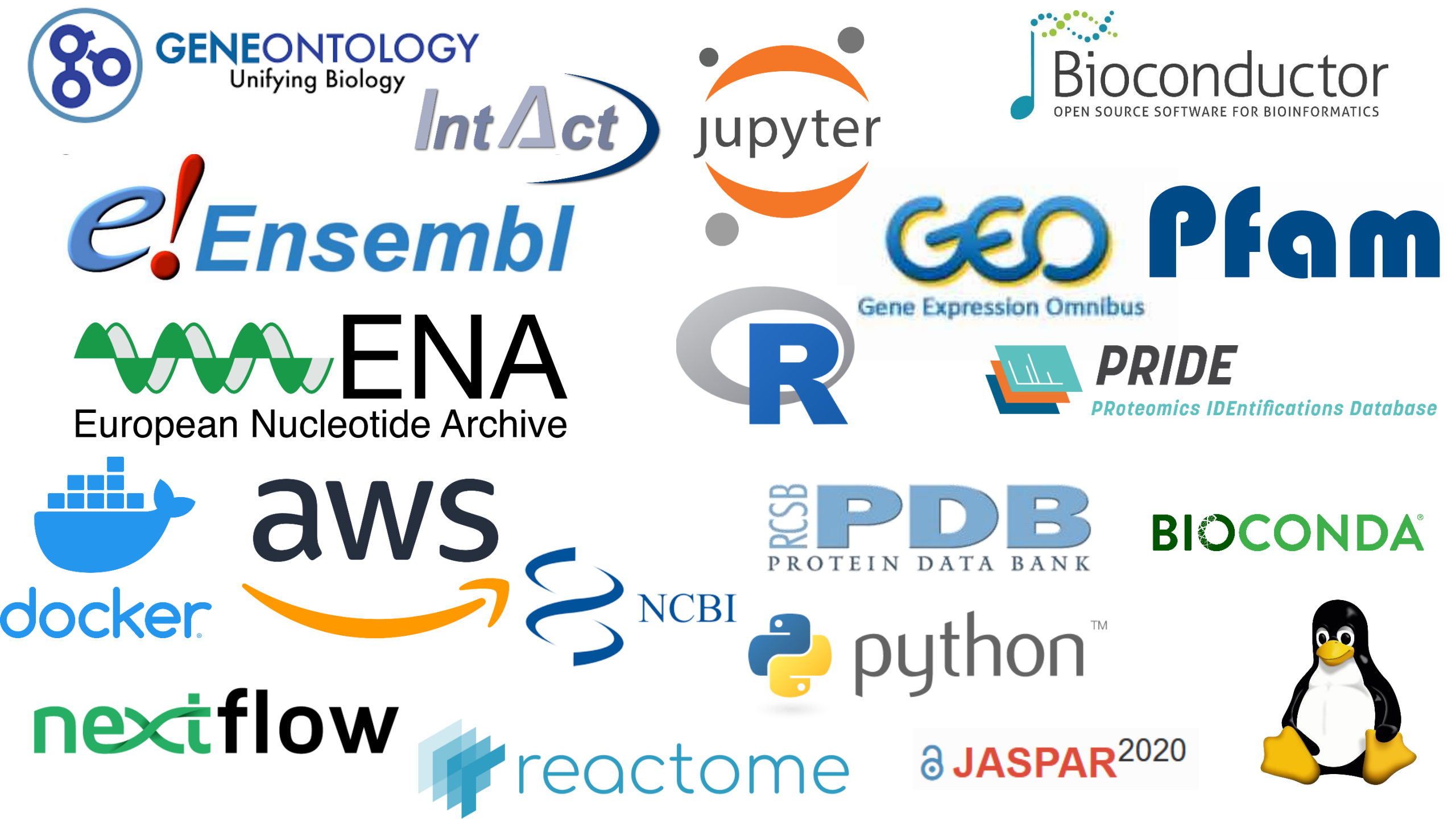
Introduction to Biological Big Data

Simon Andrews
simon.andrews@babraham.ac.uk
@simon_andrews

v2023-10-12

Babraham Bioinformatics

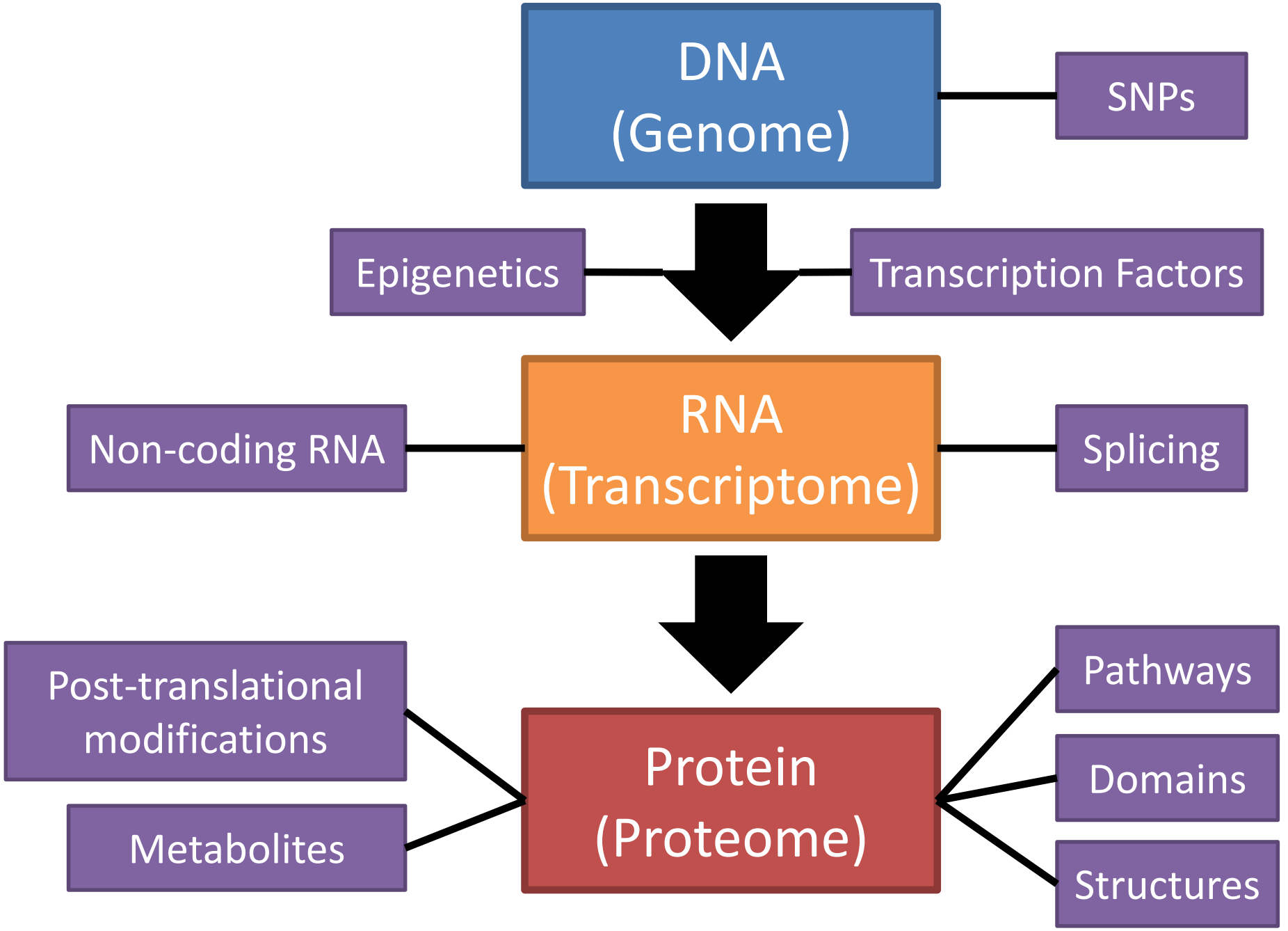

GENEONTOLOGY
Unifying Biology
IntAct
jupyter
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Ensembl
GEO
Gene Expression Omnibus
Pfam
ENA
European Nucleotide Archive
R
PRIDE
PRoteomics IDEntifications Database
docker
aws
RCSB PDB
PROTEIN DATA BANK
BIOCONDA
NCBI
python
nextflow
reactome
JASPAR2020

Course Structure

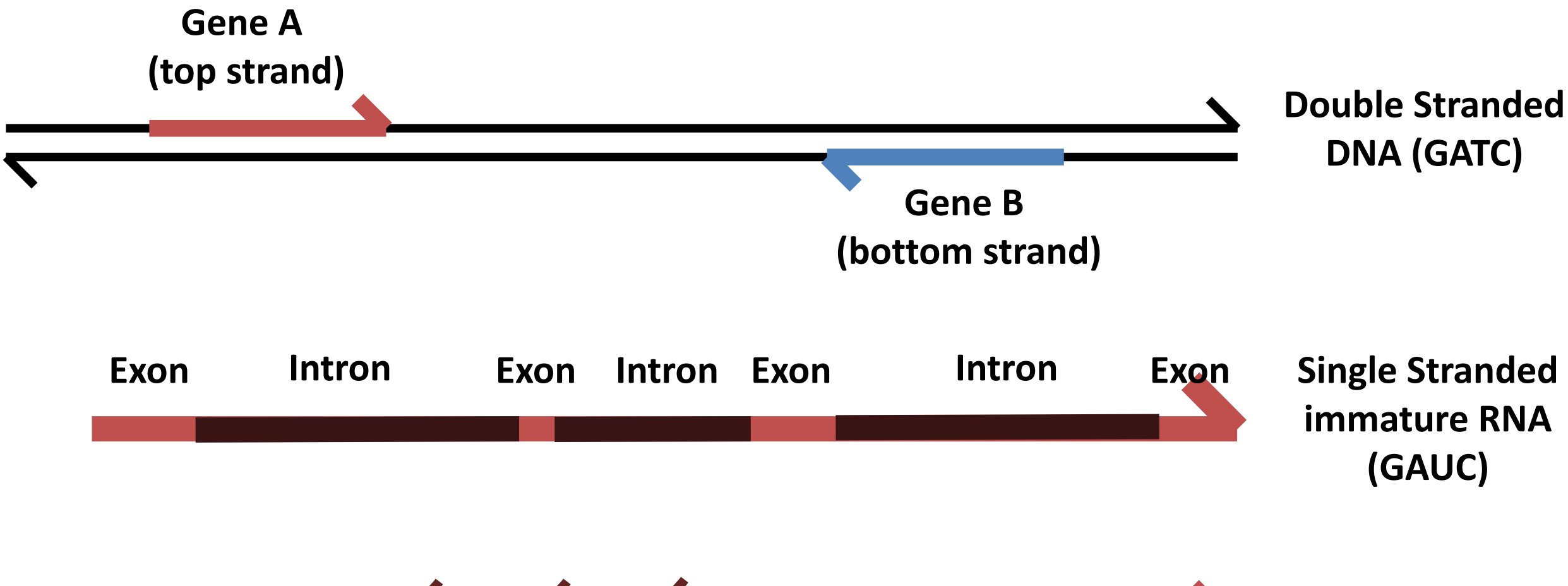
- Central Dogma Data Sources
  - Genomes and Annotations
  - Protein Domains and Structures
  - Reactions, Pathways and Interactions
- Experimental Techniques, Datatypes and Repositories
  - Sequencing and Variants
  - Proteomics and Metabolites
  - Flow and Imaging
- Practical Computation for Bioinformatics
  - Analysis approaches
  - Computing platforms for big data
  - Selecting bioinformatics software
  - Languages, Frameworks and pipelines

Central Dogma Data Resources

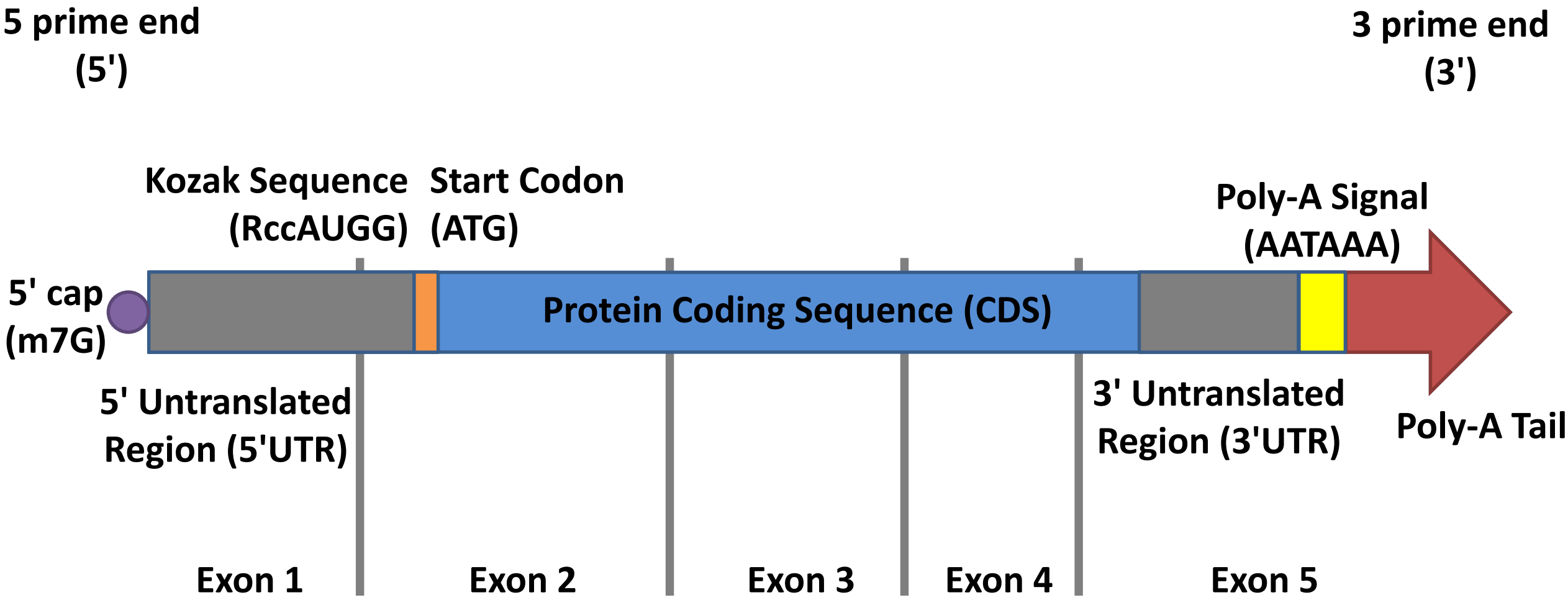
Babraham Bioinformatics


DNA
(Genome)
SNPs
Epigenetics
Transcription Factors
RNA
(Transcriptome)
Non-coding RNA
Splicing
Post-translational
modifications
Metabolites
Protein
(Proteome)
Pathways
Domains
Structures

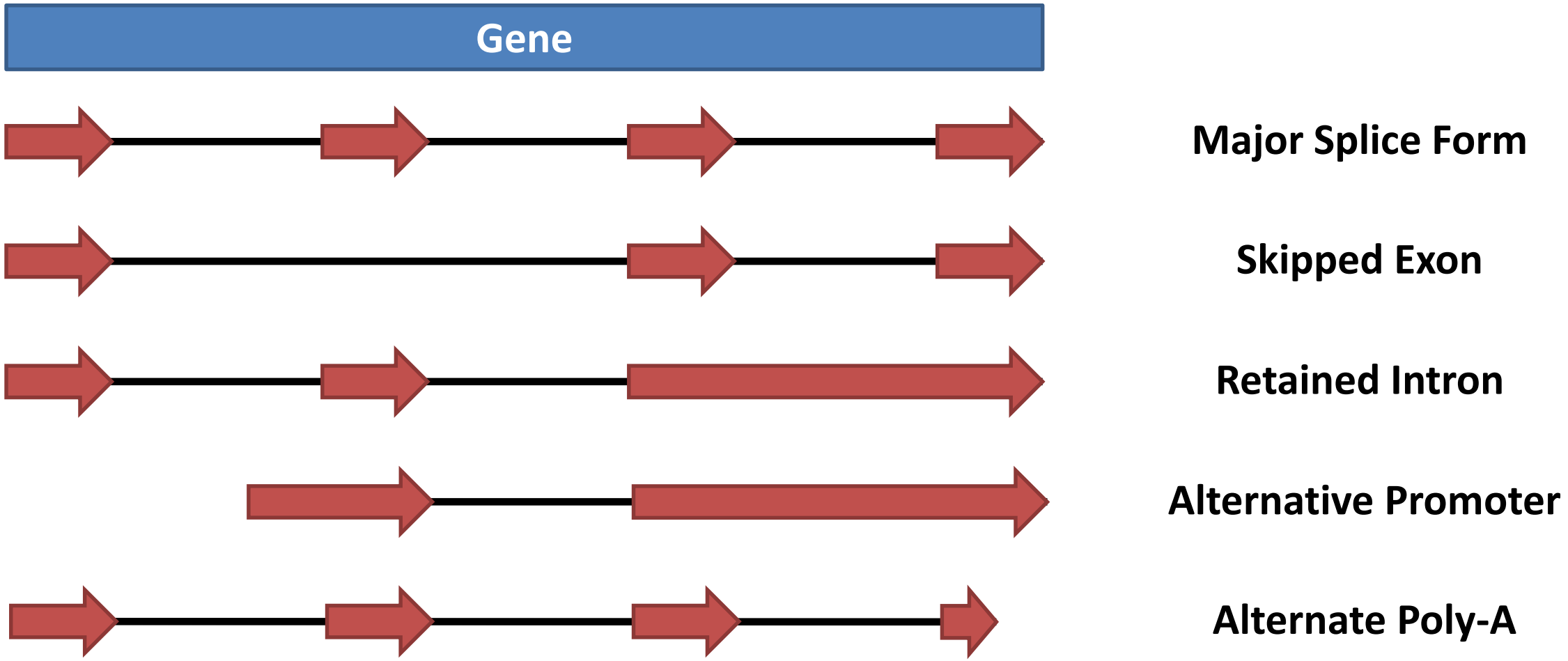
Annotation Structure



Mature Transcript Structure



Alternative Splicing



Genomes and Annotations

- Genome Assemblies
  - Underlying sequence of the organism's chromosomes
  - Often starts as scaffolds / contigs
  - Eventually assembled into chromosomes (still with holes)
    - Only one chromosome sequence per chromosome
  - Represents an 'average' individual (unless backcrossed)
  - Variations (natural or clinical) are stored separately)
  - Assembly is refined and improved over time, new releases get new names

Genome Assembly Nomenclature

- Chromosome / Scaffold sequences
  - Originally deposited with ENA / NCBI as sequence records

- Genome Assembly
  - Given an official name by a supervising group (sometimes two!)
  - Fixed coordinates at that point

Current Human Genome

- Assembly Name: GRCh38
- Current Patch: GRCh38.p13
- Managed by: Genome Reference Consortium
- Assembly type: Chromosomal
- Chromosome: Chr1 = CM000663.2 = NC_000001.11
- Genome: GCF_000001405.39 (Assembly Refseq)
  GCA_000001405.28 (Assembly Genbank)

Genome Annotation Sets

- Built on top of a specific assembly
- Combination of prediction tools and real data
- Main annotation is Genes, Transcripts, Coding Sequences
- Many other tracks often added

- Different sites will have different annotations
- Annotations updated more frequently than assemblies

Genome Annotation Details

Genome-Annotation-Data

```
##Genome-Annotation-Data-START##
Annotation Provider::NCBI
Annotation Status::Updated annotation
Annotation Name::Homo sapiens Updated Annotation Release 109.20210226
Annotation Version::109.20210226
Annotation Pipeline::NCBI eukaryotic genome annotation pipeline
Annotation Software Version::8.6
Annotation Method::Best-placed RefSeq; propagated RefSeq model
Features Annotated::Gene; mRNA; CDS; ncRNA
##Genome-Annotation-Data-END##
```

General stats

Total No of Genes	60649	Total No of Transcripts	237012
Protein-coding genes	19955	Protein-coding transcripts	86757
Long non-coding RNA genes	17944	- full length protein-coding	61015
Small non-coding RNA genes	7567	- partial length protein-coding	25742
Pseudogenes	14773	Nonsense mediated decay transcripts	18881
- processed pseudogenes	10667	Long non-coding RNA loci transcripts	48752
- unprocessed pseudogenes	3565		
- unitary pseudogenes	241		
- polymorphic pseudogenes	49		
- pseudogenes	15	Total No of distinct translations	63968
		Genes that have more than one distinct translations	13689
Immunoglobulin/T-cell receptor gene segments			
- protein coding segments	409		
- pseudogenes	236		

Assembly	GRCh38.p14 (Genome Reference Consortium Human Build 38), INSDC Assembly GCA_000001405.29, Dec 2013
Base Pairs	3,099,750,718
Golden Path Length	3,099,750,718
Assembly provider	Genome Reference Consortium
Annotation provider	Ensembl
Annotation method	Full genebuild
Genebuild started	Jan 2014
Genebuild released	Jul 2014
Genebuild last updated/patched	Mar 2023
Database version	110.38
Gencode version	GENCODE 44

Gene counts (Primary assembly)

Coding genes	19,831 (excl 650 readthrough)
Non coding genes	25,959
Small non coding genes	4,864
Long non coding genes	18,874 (excl 319 readthrough)
Misc non coding genes	2,221
Pseudogenes	15,239 (excl 1 readthrough)
Gene transcripts	252,894

Viewing Annotated Genomes

- Mostly web based
  - Species specific sites
  - Generic multi-species sites
- Often adds more information
  - Regulation, conservation, repeats
  - Experimental datasets
  - Upload your own

Species specific genome viewer sites

Arabidopsis https://www.arabidopsis.org

Drosophila https://flybase.org/

Nematode worms https://wormbase.org

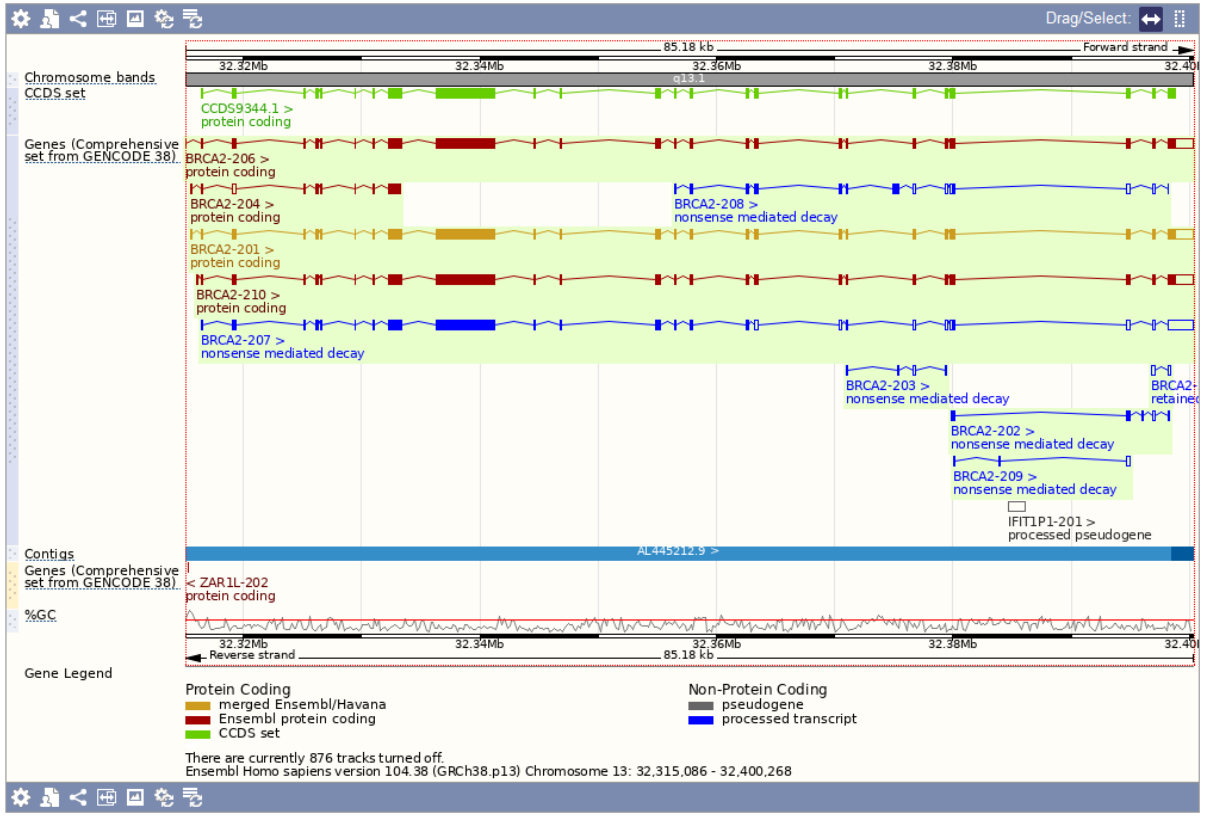
Generic genome viewer sites

Ensembl https://www.ensembl.org

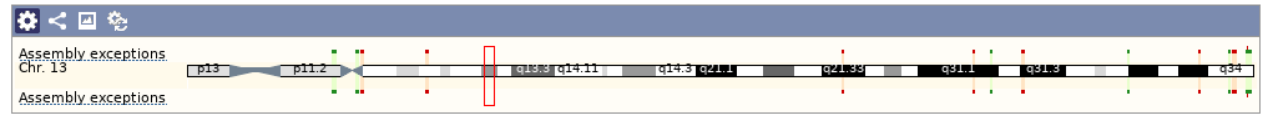
UCSC Browser https://genome.ucsc.edu

WashU Browser https://epigenomegateway.wustl.edu/

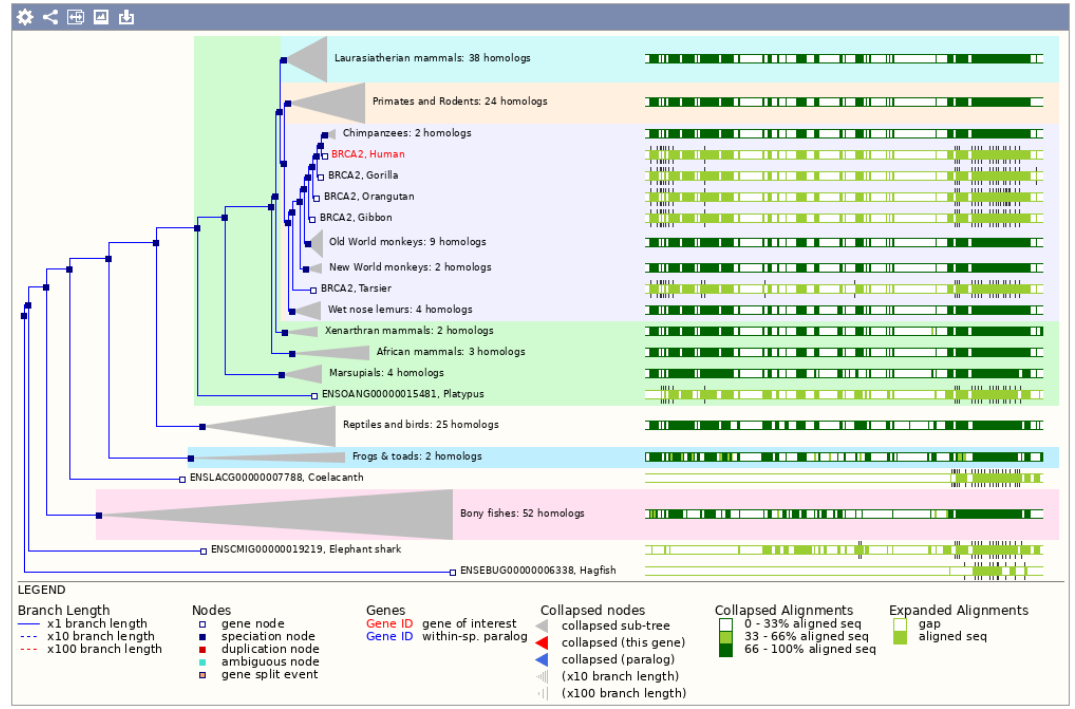
Name	Transcript ID	bp	Protein	Biotype	CCDS	UniProt Match
BRCA2-201	ENST00000380152.8	11954	3418aa	Protein coding	CCDS9344	P51587
BRCA2-210	ENST00000680887.1	11880	3418aa	Protein coding	CCDS9344	-
BRCA2-206	ENST00000544455.6	11854	3418aa	Protein coding	CCDS9344	P51587
BRCA2-204	ENST00000530893.6	2011	481aa	Protein coding	-	A0A590UJI7
BRCA2-207	ENST00000614259.2	11763	2649aa	Nonsense mediated decay	-	-



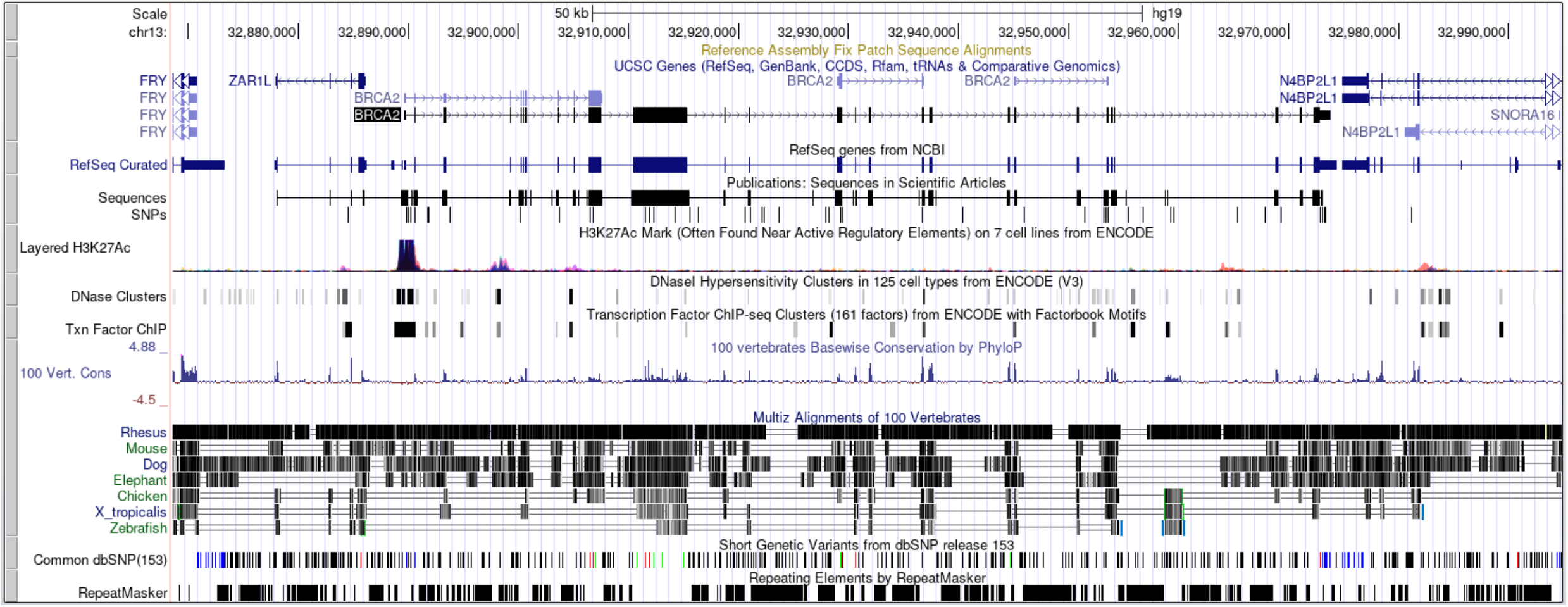
Chromosome 13: 32,315,086-32,400,268



Region in detail



Track Based Displays



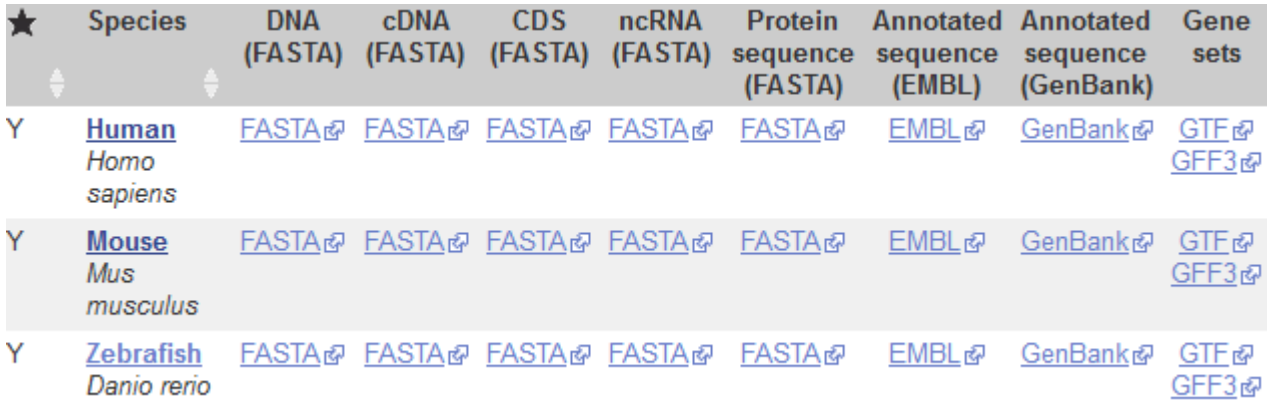
Large Scale Queries

- Large scale querying and export of genomic data

- Annotations, Sequences, Variants etc.
  - Select data type (eg genes)
  - Select genome species
  - Select genes / regions / identifiers
  - Select attributes to export
  - Generate report

Genome File Formats

- Genome Assemblies
  - Chr sequence, FastA format
  - A small header plus DNA bases
  - Also used for RNA / protein



★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	Annotated sequence (EMBL)	Annotated sequence (GenBank)	Gene sets
Y	**Human** *Homo sapiens*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3
Y	**Mouse** *Mus musculus*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3
Y	**Zebrafish** *Danio rerio*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3

- Gene Annotations
  - GFF or GTF format (both very similar)
  - Hierarchical format linking exons to transcripts to genes

https://www.ensembl.org/info/data/ftp/index.html

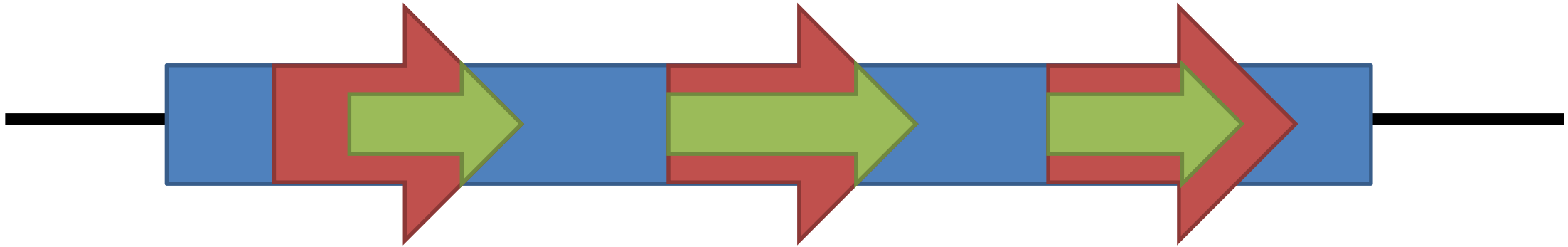
FastA Format Data

```
>I dna:chromosome chromosome:R64-1-1:I:1:230218:1 REF
CCACACCACACCCACACACCCACACACCACACCACACACCACACCCACACACACA
CATCCTAACACTACCCTAACACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTT
ACCCTCCATTACCCTGCCTCCACTCGTTACCCTGTCCCATTCAACCATACCACTCCGAAC
CACCATCCATCCCTCTACTTACTACCACTCACCCACCGTTACCCTCCAATTACCCATATC
>II dna:chromosome chromosome:R64-1-1:II:1:813184:1 REF
AAATAGCCCTCATGTACGTCTCCTCCAAGCCCTGTTGTCTCTTACCCGGATGTTCAACCA
AAAGCTACTTACTACCTTTATTTTATGTTTACTTTTTATAGGTTGTCTTTTTATCCCACT
TCTTCGCACTTGTCTCTCGCTACTGCCGTGCAACAAACACTAAATCAAAACAATGAAATA
CTACTACATCAAAACGCATTTTCCCTAGAAAAAAAATTTTCTTACAATATACTATACTAC
```

IUPAC Ambiguity Codes

IUPAC Code	Meaning
A	A
C	C
G	G
T/U	T
M	A or C
R	A or G
W	A or T
S	C or G
Y	C or T
K	G or T
V	A or C or G
H	A or C or T
D	A or G or T
B	C or G or T
N	G or A or T or C

Annotation Descriptions



Gene

Exon (combined into transcript)

Coding Exon

GFF (Strictly GFF.2)

- Comprehensive annotation format
- Tab delimited
- Flexible – able to accommodate multi-features

GFF File Fields

1. Chromosome
2. Source
3. Feature Type
4. Start
5. End
6. Score
7. Strand (+/-)
8. Frame (1,2,3)
9. Group/Attributes

```
1 hav gene       11869 14409 . + . ID=gene:ENSG223972;Name=DDX11L1;description=DEAD/H-box 1;gene_id=ENSG223972
1 hav transcript 11869 14409 . + . ID=transcript:ENST456328;Parent=gene:ENSG223972;Name=DDX11L1-002;
1 hav exon       11869 12227 . + . Parent=transcript:ENST456328;exon_id=ENSE2234944;rank=1
1 hav exon       12613 12721 . + . Parent=transcript:ENST456328;exon_id=ENSE3582793;rank=2
1 hav exon       13221 14409 . + . Parent=transcript:ENST456328;exon_id=ENSE2312635;rank=3
```

Positions are 1-indexed, fully open

GTF

- Targeted at gene structure definition
- Variant of GFF with stricter rules about attributes
  - Attributes must use `gene_id` and `transcript_id`
  - Commas mandatory and single space delimited

```
1 havana gene       11869 14409 . + . gene_id "ENSG223972"; gene_name "DDX11L1";
1 havana transcript 11869 14409 . + . gene_id "ENSG223972"; transcript_id "ENST456328"; transcript_name "DDX11L1-202";
1 havana exon       11869 12227 . + . gene_id "ENSG223972"; transcript_id "ENST456328"; exon_number "1"; exon_id "ENSE2234944";
1 havana exon       12613 12721 . + . gene_id "ENSG223972"; transcript_id "ENST456328"; exon_number "2"; exon_id "ENSE3582793";
1 havana exon       13221 14409 . + . gene_id "ENSG223972"; transcript_id "ENST456328“; exon_number "3"; exon_id "ENSE2312635";
```

Genome Exploration Exercise

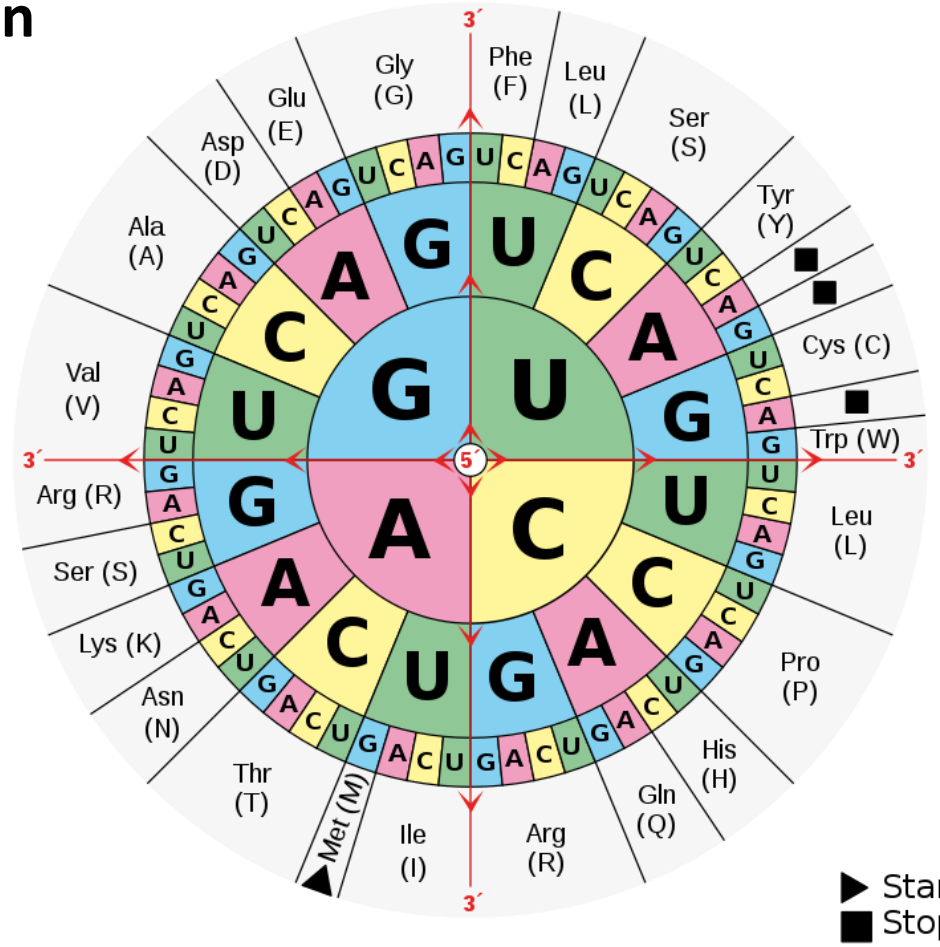
mRNA Translation into Protein

Start Codon **Stop Codon**

```
GACACC ATG AGC ACT GAA ... CTG TGA
UTR    Met Ser Thr Glu     Arg Stp
```

- Most species use the same code
- Some have minor differences

https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi



Codon Usage

Codon Statistics Database

Genes analyzed: **Nuclear genes** | Ribosomal proteins | Mitochondrial genes

Species: *Homo sapiens*
Taxonomy ID: 9606
Assembly: GCF_000001405.39 | GRCh38.p13

Genetic code: 1
Number of genes: 19850
Number of codons: 11577026

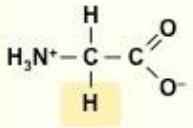
Amino Acid	Codon	Count	RSCU	Preferred
Ala	GCA	187108	0.921	Unpreferred
Ala	GCC	323249	1.590	Preferred
Ala	GCG	89097	0.438	Preferred
Ala	GCT	213559	1.051	Unpreferred
Arg	AGA	142934	1.303	Unpreferred
Arg	AGG	140481	1.281	Unpreferred
Arg	CGA	70319	0.641	Unpreferred
Arg	CGC	119972	1.094	Preferred
Arg	CGG	132275	1.206	Preferred
Arg	CGT	52129	0.475	Unpreferred
Asn	AAC	212987	1.037	Preferred
Asn	AAT	197831	0.963	Unpreferred
Asp	GAC	287974	1.059	Preferred
Asp	GAT	255933	0.941	Unpreferred

Often internal

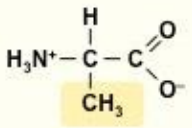
Often surface

Often Binding or catalytic sites

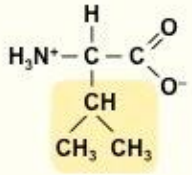
NON-POLAR



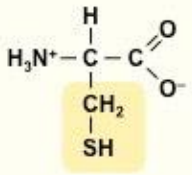
Glycine
(Gly / G)



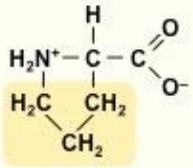
Alanine
(Ala / A)



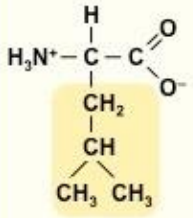
Valine
(Val / V)



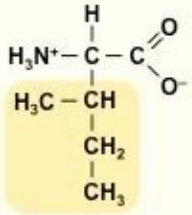
Cysteine
(Cys / C)



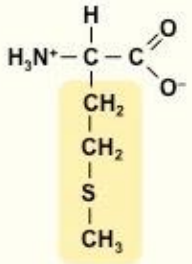
Proline
(Pro / P)



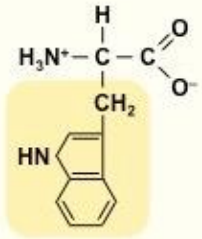
Leucine
(Leu / L)



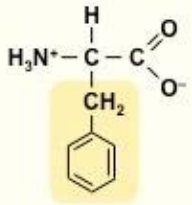
Isoleucine
(Ile / I)



Methionine
(Met / M)

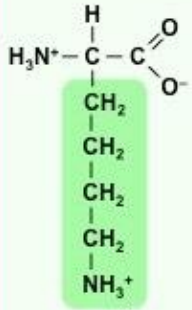


Tryptophan
(Trp / W)

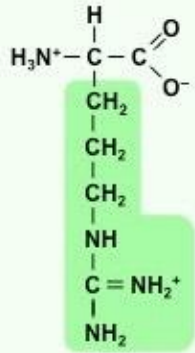


Phenylalanine
(Phe / F)

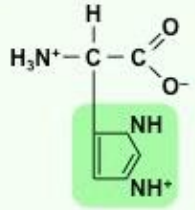
+ CHARGE



Lysine
(Lys / K)

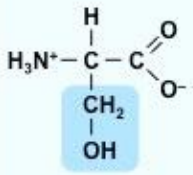


Arginine
(Arg / R)

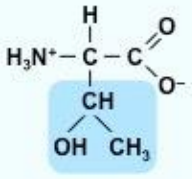


Histidine
(His / H)

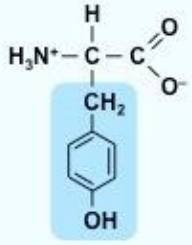
POLAR



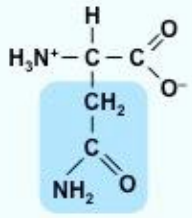
Serine
(Ser / S)



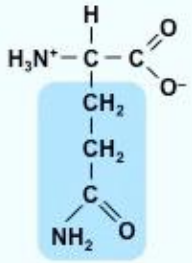
Threonine
(Thr / T)



Tyrosine
(Tyr / Y)

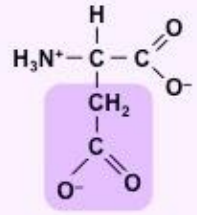


Asparagine
(Asn / N)

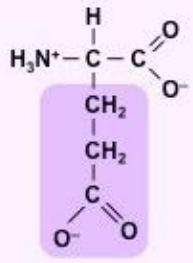


Glutamine
(Gln / Q)

– CHARGE



Aspartic Acid
(Asp / D)



Glutamic Acid
(Glu / E)

Protein Domain Information

- A single protein can have more than one functional unit
  - Proteins are annotated with functional ‘domains’
  - A domain is normally linked with a globular folded structure

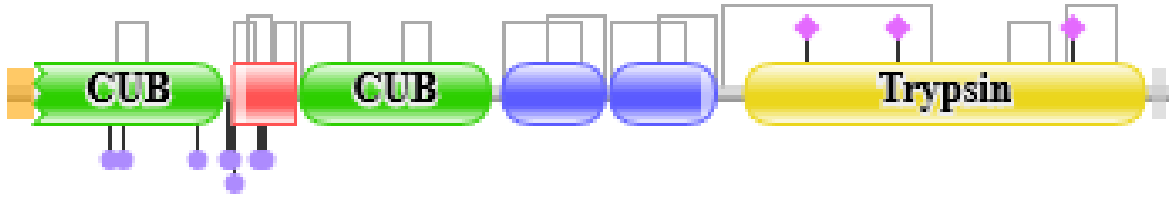
- Domain structures are re-used to provide modular functionality across multiple proteins.
  - Often linked to exon structures or splice variation

- It can be useful to know the key functional amino acids
  - Binding pockets
  - Active sites

Protein Domain Databases

http://smart.embl-heidelberg.de/

https://www.ebi.ac.uk/interpro/



Tryp_SPc domain

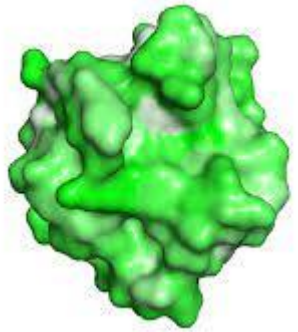
This is a SMART **Tryp_SPc** domain (full annotation).

Position:	**437** to **675**
E-value:	**4.3565597296112e-75** (HMMER2)

SMART ACC:	SM000020
Definition:	Trypsin-like serine protease
Description:	Many of these are synthesised as inactive precursor zymogens that are cleaved during limited proteolysis to generate their active forms. A few, however, are active as single chain molecules, and others are inactive due to substitutions of the catalytic triad residues.

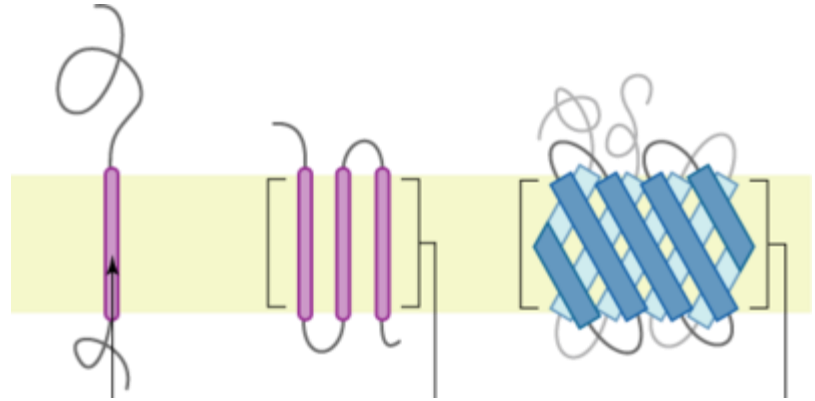
Types of domain

- Globular
  - Forms a concerted 3D structure
  - Most catalytic and some binding domains

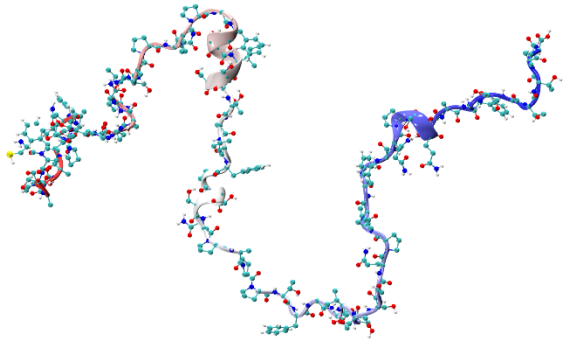


- Semi-ordered
  - Coiled coil
  - Many binding domains

- Transmembrane
  - Threaded through a membrane
  - Transmembrane regions, then internal and external segments



- Disordered / Low Complexity
  - Linker regions
  - Intrinsically disordered proteins



Key Residue Databases

RecName: Full=Tyrosine-protein kinase Tec; EC=2.7.10.2;. *Homo sapiens (Human)*

RecName: Full=Tyrosine-protein kinase Tec; EC=2.7.10.2;. *Homo sapiens (Human)*

PS00107 **PROTEIN_KINASE_ATP** *Protein kinases ATP-binding region signature :*

376 - 398: [confidence level: (0)] `LGSGLFGVVRlGkwraqyk..........VAIK`

PS00109 **PROTEIN_KINASE_TYR** *Tyrosine protein kinases specific active-site signature :*

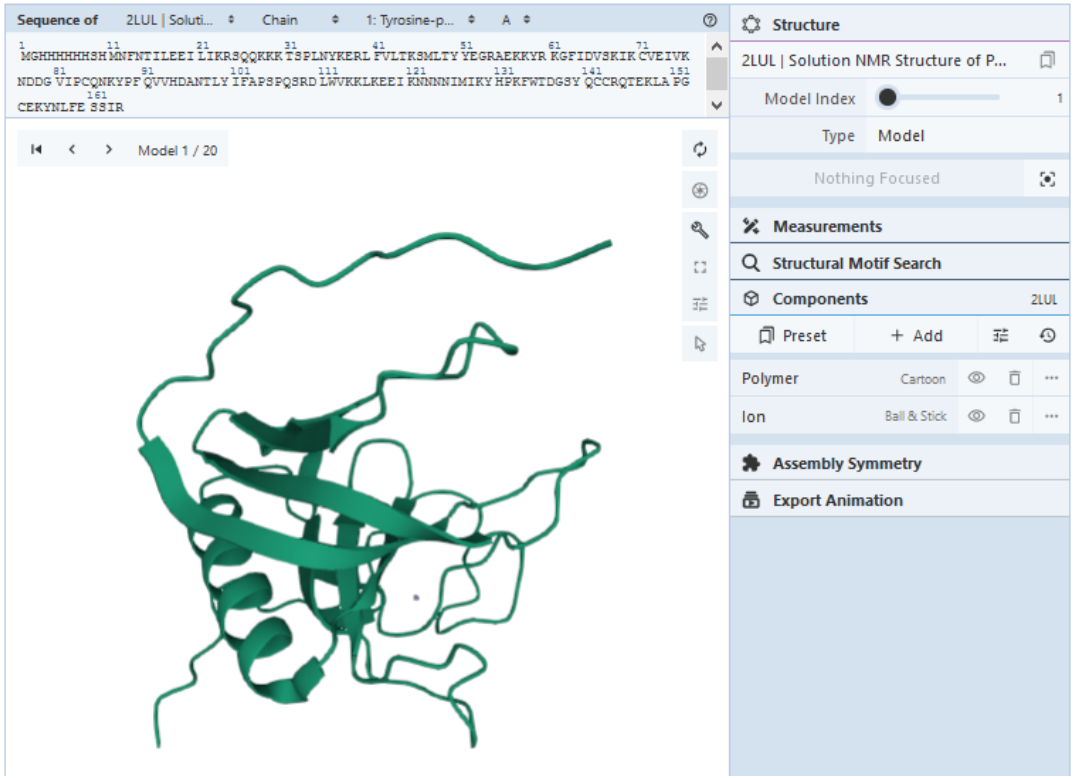
485 - 497: [confidence level: (0)] `FIHrDLAARNCLV`

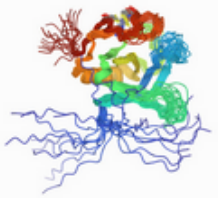
Predicted feature:

ACT_SITE	489	Proton acceptor	[condition: none]

Protein Structure Databases

RCSB PDB
PROTEIN DATA BANK





3D View

2LUL

Download File | View File

Solution NMR Structure of PH Domain of Tyrosine-protein kinase Tec from Homo sapiens, Northeast Structural Genomics Consortium (NESG) Target HR3504C

Liu, G., Xiao, R., Janjua, H., Hamilton, K., Shastry, R., Kohan, E., Acton, T.B., Everett, J.K., Lee, H., Pederson, K., Huang, Y.J., Montelione, G.T., Northeast Structural Genomics Consortium (NESG)

To be published

Released	2012-08-15
Method	SOLUTION NMR
Organisms	Homo sapiens
Macromolecule	Tyrosine-protein kinase Tec (protein)
Unique Ligands	ZN

3D View

1GL5

Download File | View File

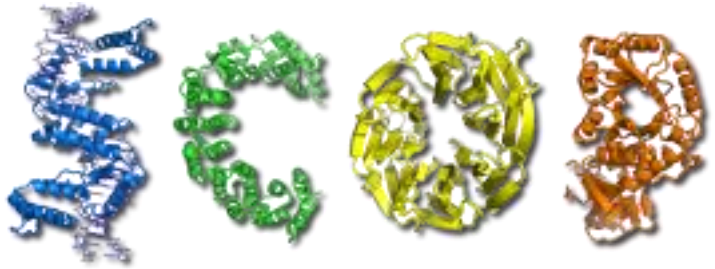
NMR structure of the SH3 domain from the Tec protein tyrosine kinase

Mulhern, T.D., Pursglove, S.E., Booker, G.W.

(2002) J Biol Chem **277**: 755-762

Released	2001-11-28
Method	SOLUTION NMR
Organisms	Mus musculus
Macromolecule	TYROSINE-PROTEIN KINASE TEC (protein)

Protein Structure Classification Databases

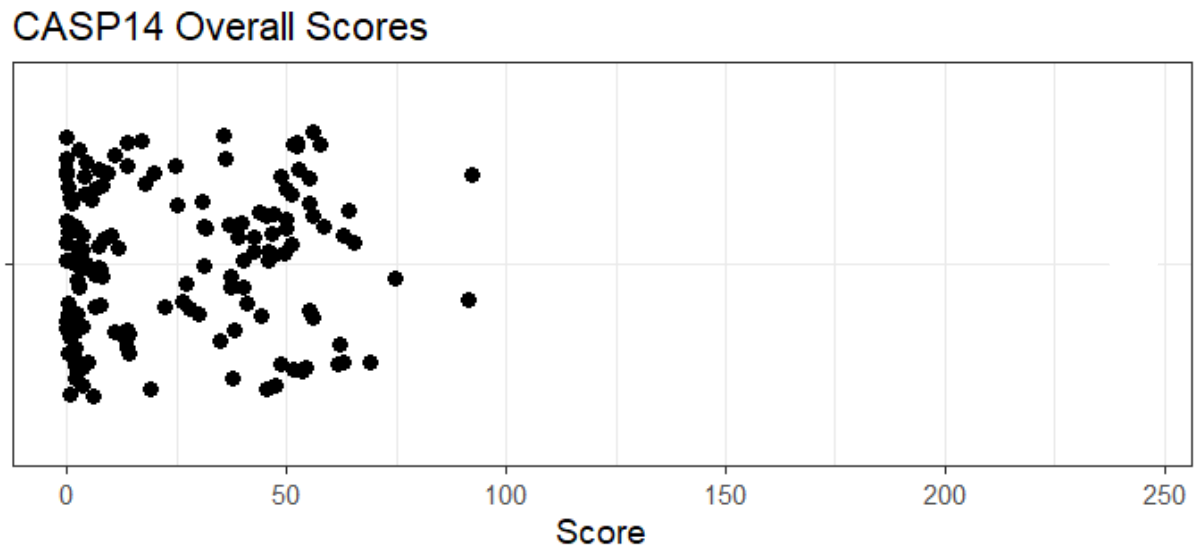


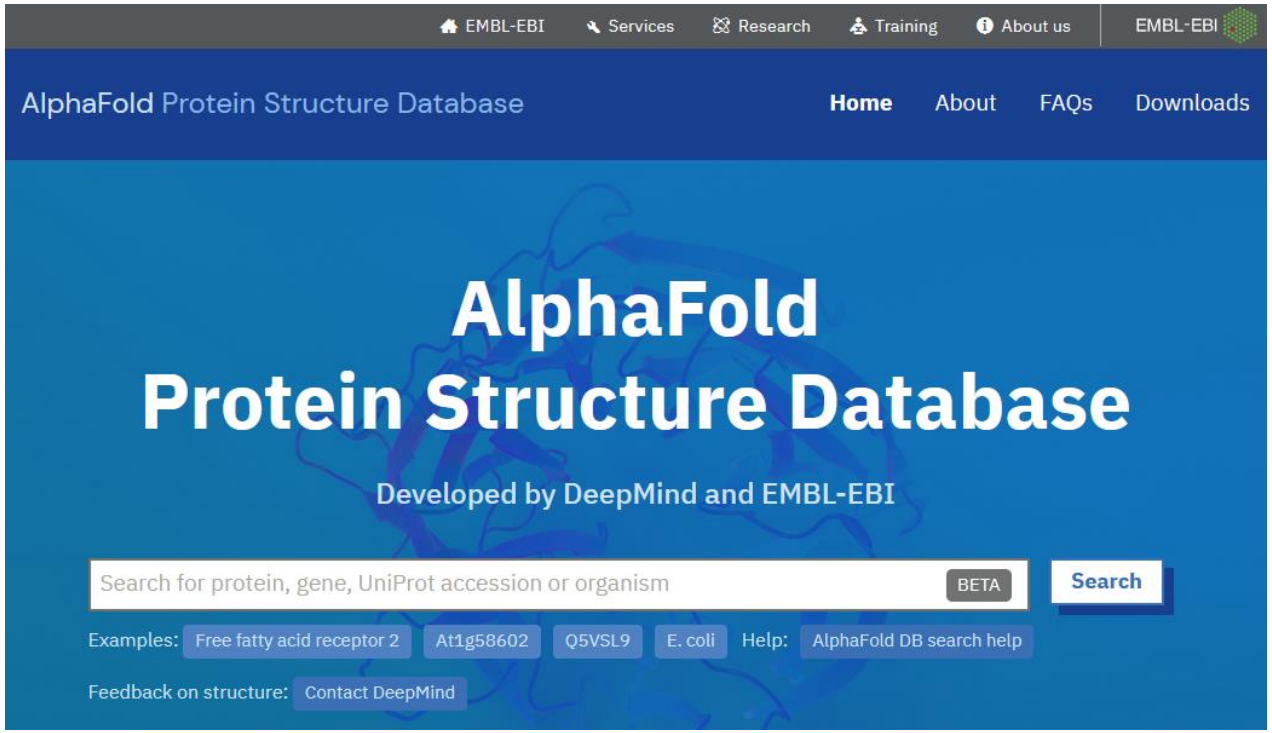
https://scop.mrc-lmb.cam.ac.uk/

https://www.cathdb.info/

Predicted Structure Database

Currently (Mar 2022), only 7914/22818 protein coding genes have an experimental 3D structure available





ARTIFICIAL INTELLIGENCE

DeepMind's protein-folding AI has solved a 50-year-old grand challenge of biology

AlphaFold can predict the shape of proteins to within the width of an atom. The breakthrough will help scientists design drugs and understand disease.

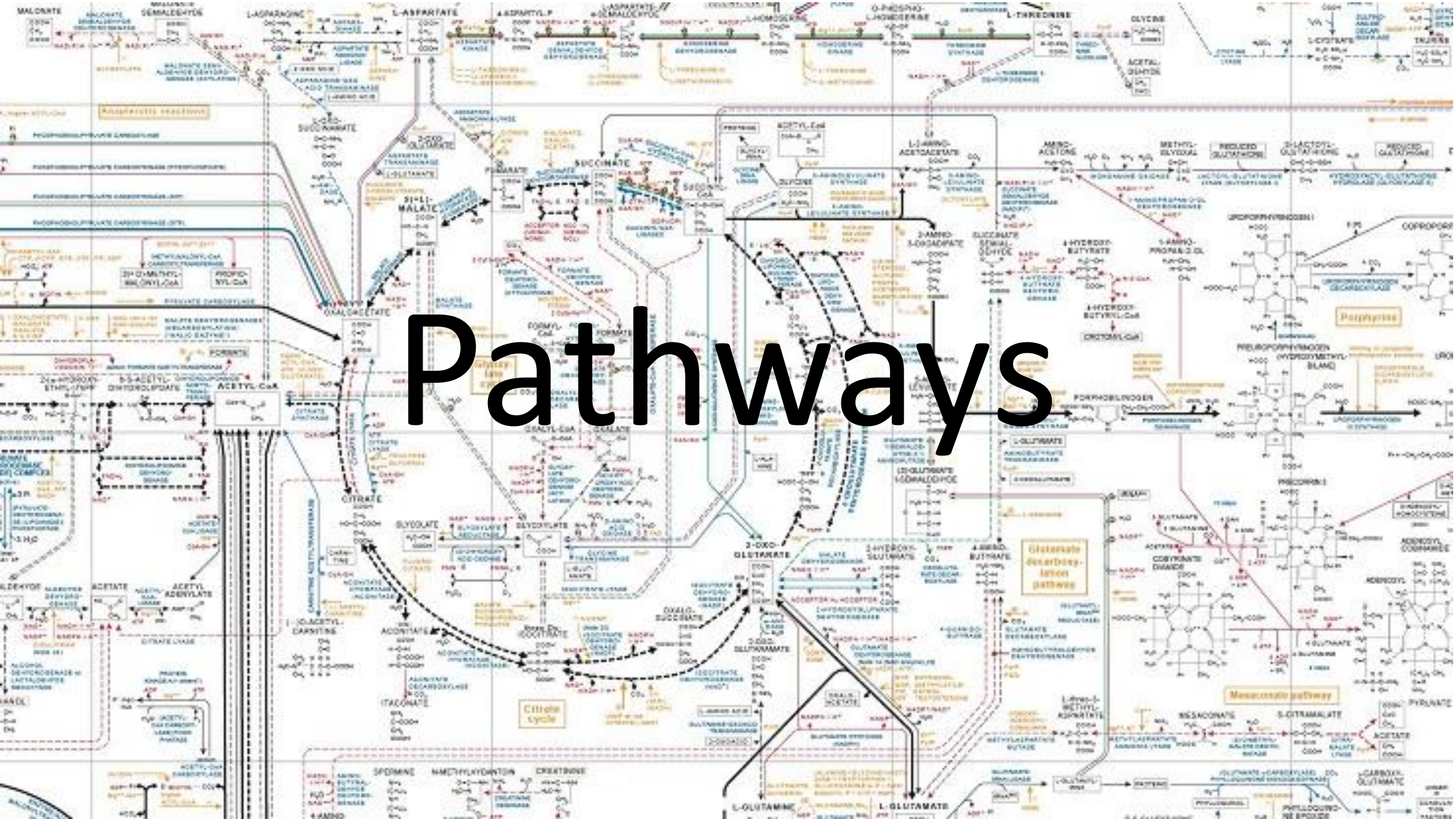
By Will Douglas Heaven

November 30, 2020

https://alphafold.ebi.ac.uk/

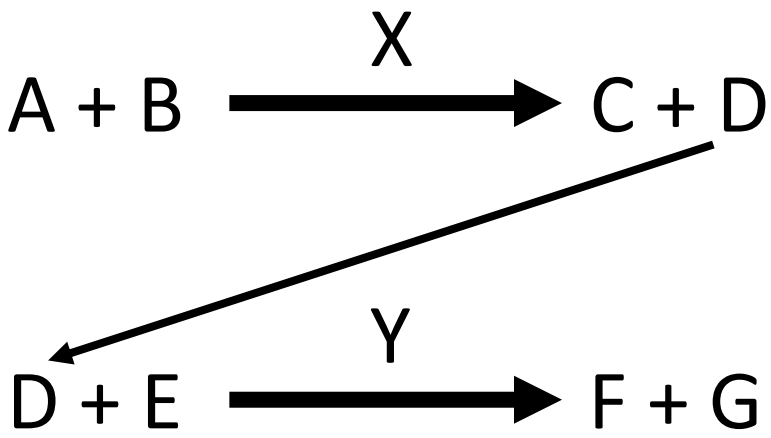
Protein Annotation Exercise

Babraham Bioinformatics


Pathways

Hierarchy of Reaction Annotations

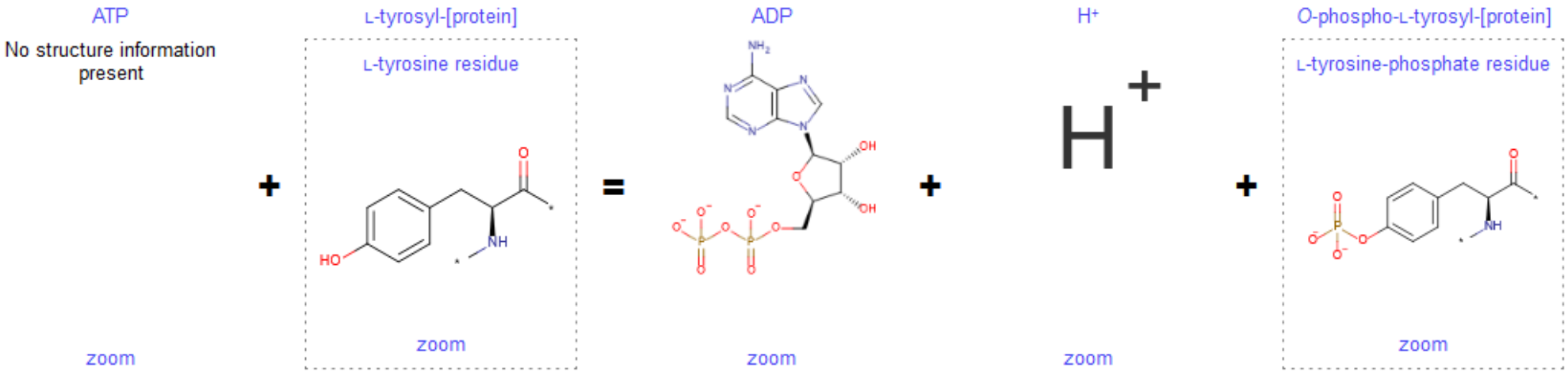
- Components (Reactants / Products)
- Proteins (Enzymes)
- Reactions
- Pathways
- Processes



Reactions

RHEA:10596

Enzymes	82,966 proteins (UniProtKB)
Enzyme classes	EC 2.7.10.1 Receptor protein-tyrosine kinase EC 2.7.10.2 Non-specific protein-tyrosine kinase EC 2.7.12.1 Dual-specificity kinase EC 2.7.12.2 Mitogen-activated protein kinase kinase



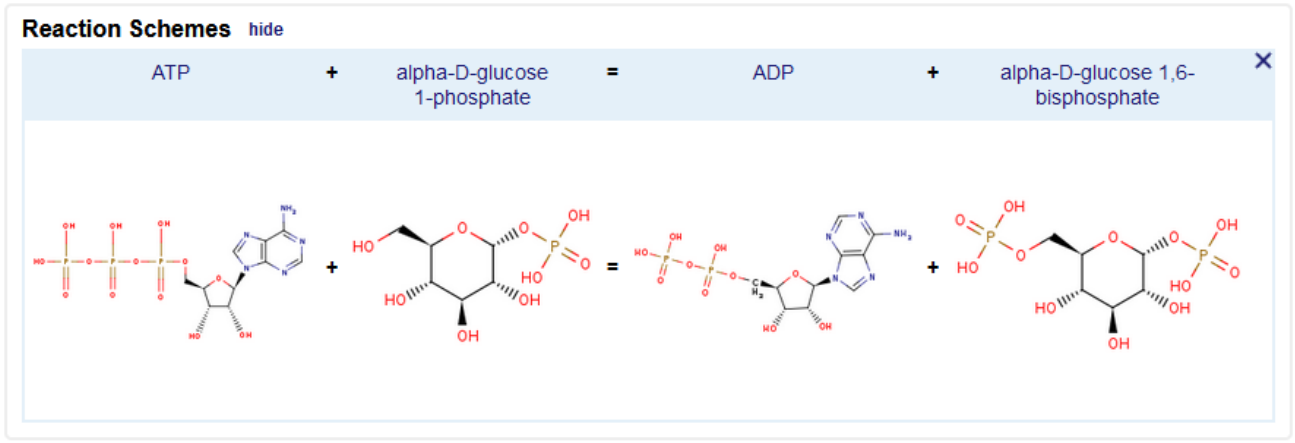
Enzyme Databases

- Enzymes are described by an Enzyme Commission (EC) number
  - EC `2.7.1.10` is phosphoglucokinase
  - Hierarchical structure

- Main Enzyme databases
  - Expasy Enzyme
  - BRENDA The Comprehensive Enzyme Information System

EC Tree

└ 2 Transferases
  └ 2.7 Transferring phosphorus-containing groups
    └ 2.7.1 Phosphotransferases with an alcohol group as acceptor
      └ 2.7.1.10 phosphoglucokinase

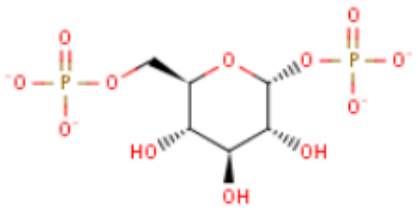


Chemical entities of biological interest

A database of "small" molecules with biological relevance
Natural or synthetic products which intervene in the processes of living organisms

CHEBI:58392 - α-D-glucose 1,6-bisphosphate(4−)

Main | ChEBI Ontology | Automatic Xrefs | Reactions | Pathways | Models



ChEBI Name	**α-D-glucose 1,6-bisphosphate(4−)**
ChEBI ID	**CHEBI:58392**
ChEBI ASCII Name	alpha-D-glucose 1,6-bisphosphate(4-)
Definition	A quadruply-charged organophosphate oxoanion arising from deprotonation of the phosphate OH groups of α-D-glucose 1,6-bisphosphate; major species at pH 7.3.
Stars	★★★ This entity has been manually annotated by the ChEBI Team.
Supplier Information	No supplier information found for this compound.
Download	Molfile XML SDF

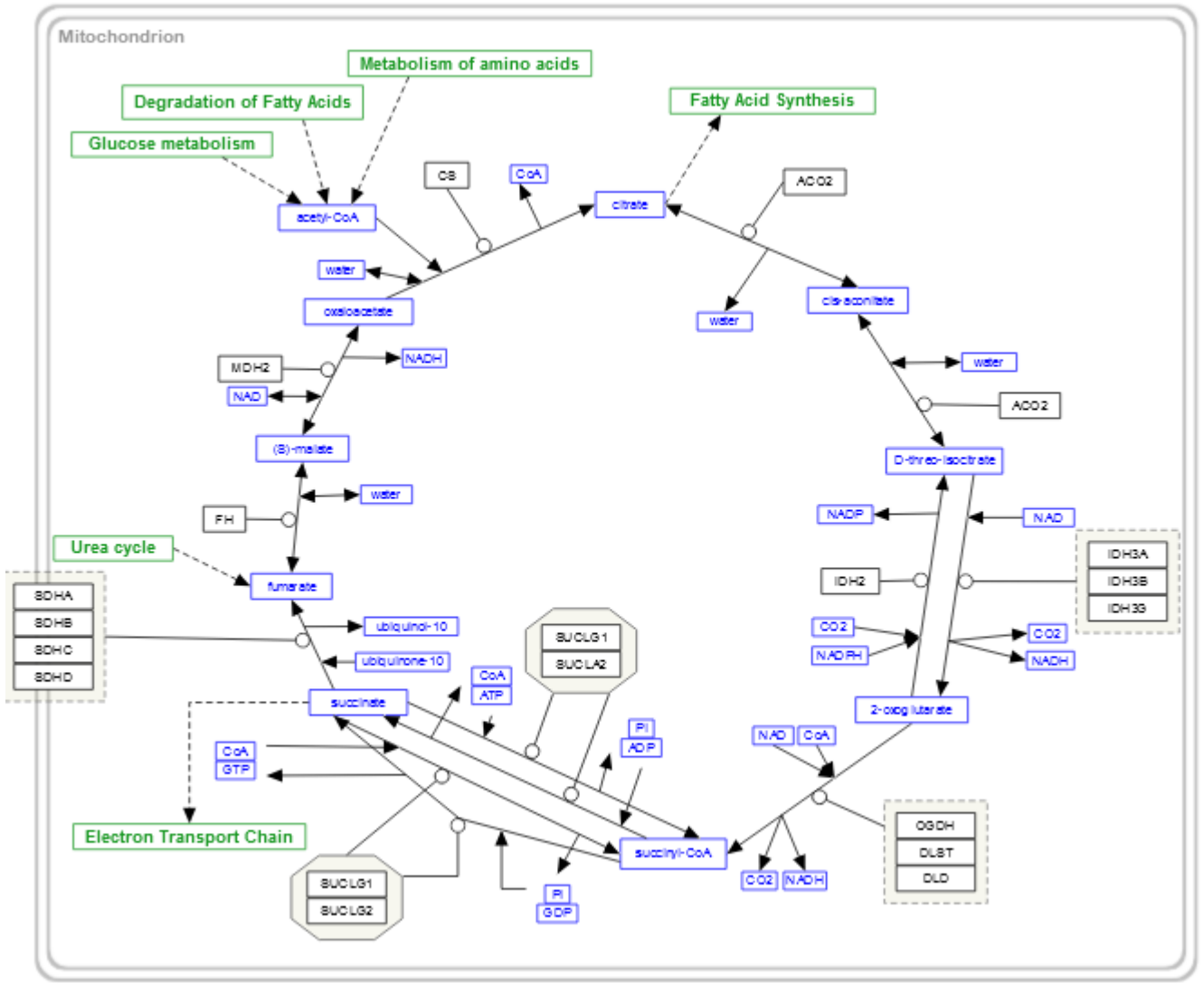
- Find compounds which contain this structure
- Find compounds which resemble this structure
- Take structure to the Advanced Search

Pathways

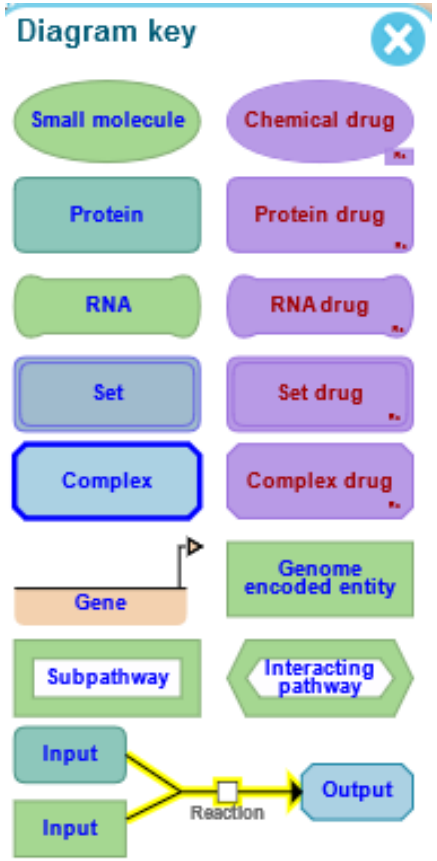
WikiPathways
Pathways for the People

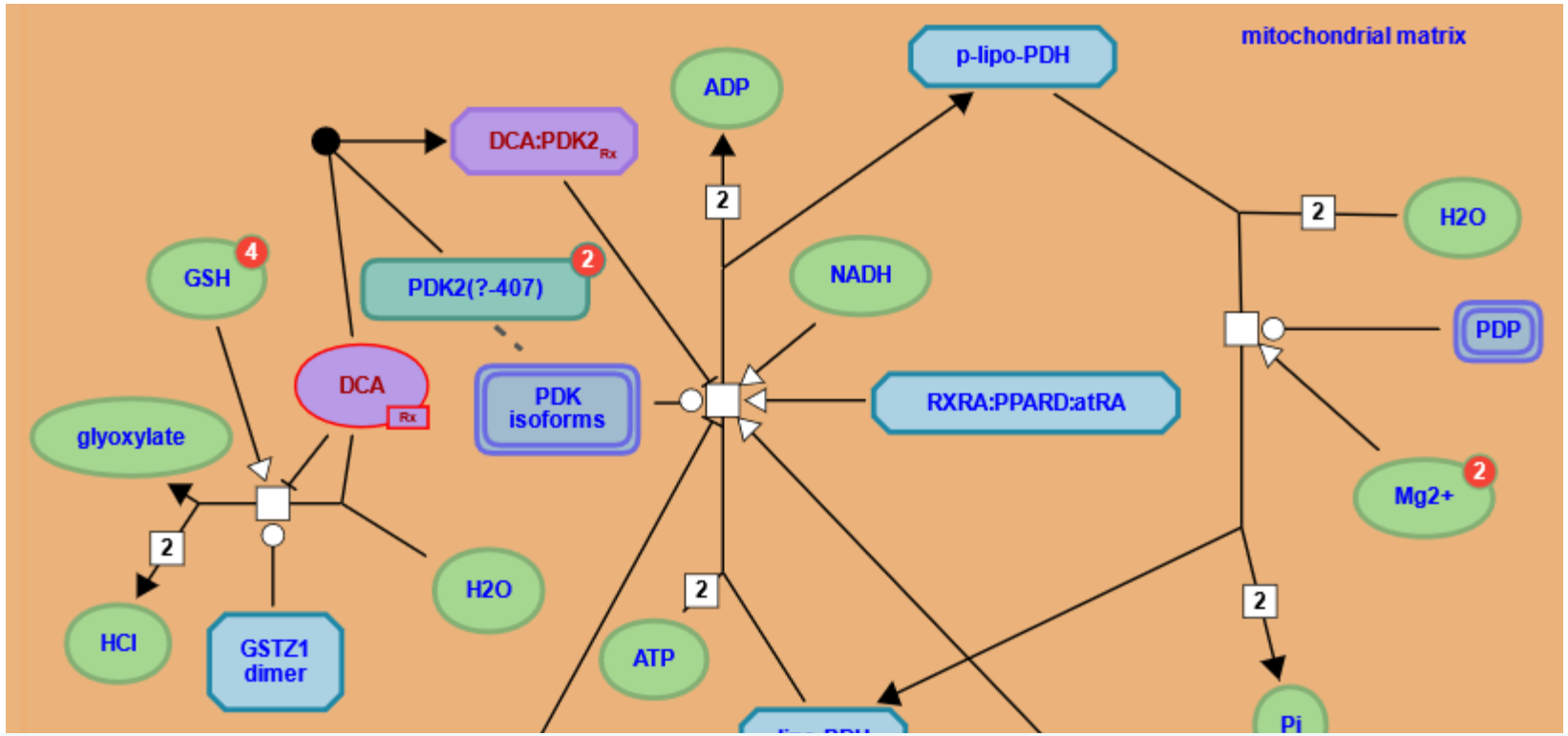
reactome

Kyoto Encyclopedia of Genes and Genomes
KEGG



Reactome





ME1:Mg2+ tetramer oxidatively decarboxylates MAL to PYR | Id: R-HSA-9012036.1 | Species: Homo sapiens

Summation

One hallmark of cancer is altered cellular metabolism. Malic enzymes (MEs) are a family of homotetrameric enzymes that catalyse the reversible oxidative decarboxylation of L-malate to pyruvate, with a simultaneous reduction of NAD(P)+ to NAD(P)H. As MEs generate NADPH and NADH, they may play roles in energy production and reductive biosynthesis. Humans possess three ME isoforms; ME1 is cytosolic and utilises NADP+, ME3 is mitochondrial and can utilise NADP+ and ME2 is mitochondrial and can utililse either NAD+ or NADP+ (Chang & Tong 2003, Murugan & Hung 2012).

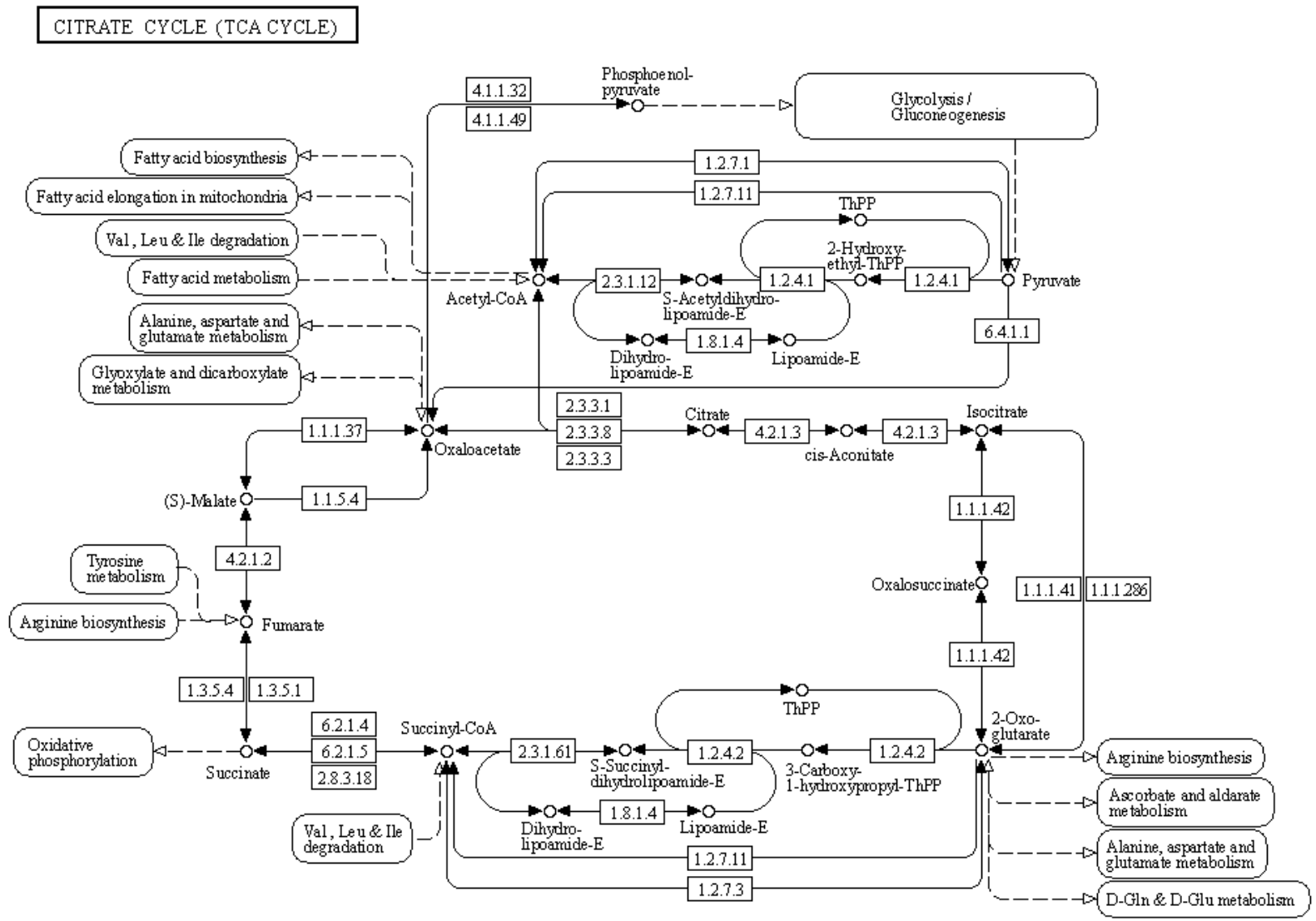
NADP-dependent malic enzyme (ME1, aka c-NADP-ME) is a cytosolic enzyme that oxidatively decarboxylates (s)-malate (MAL) to pyruvate (PYR) and CO2 using NADP+ as cofactor (Zelewski & Swierczynski 1991). ME1 exists as a dimer of dimers (Murugan & Hung 2012, Hsieh et al. 2014) and a divalent metal such as Mg2+ is essential for catalysis (Chang & Tong 2003).

▶ **Background literature references...**

KEGG databases

Category	Entry point
Systems information	KEGG PATHWAY
	KEGG BRITE
	KEGG MODULE KEGG RModule
Genomic information	KEGG ORTHOLOGY KEGG Annotation
	KEGG GENES KEGG SeqData
	KEGG GENOME KEGG Virus
Chemical information	KEGG COMPOUND
	KEGG GLYCAN
	KEGG REACTION
	KEGG Enzyme
Health information	KEGG NETWORK
	KEGG DISEASE
	KEGG DRUG

KEGG



Functional Gene Sets

GENEONTOLOGY
Unifying Biology

Molecular Function

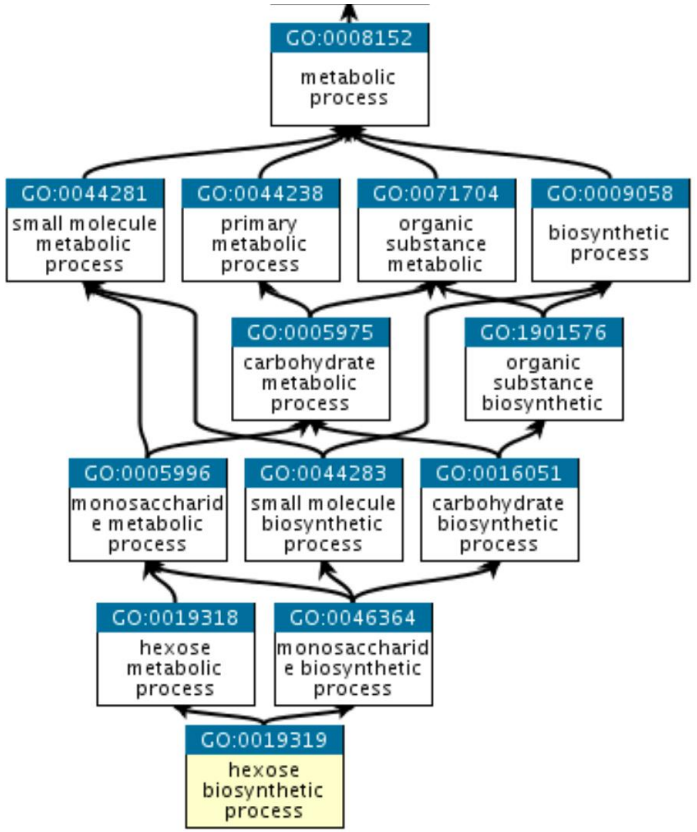
Molecular-level activities performed by gene products. Molecular function terms describe activities that occur at the molecular level, such as “catalysis” or “transport”. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where, when, or in what context the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products (*i.e.* a protein or RNA), but some activities are performed by molecular complexes composed of multiple gene products. Examples of broad functional terms are *catalytic activity* and *transporter activity*; examples of narrower functional terms are *adenylate cyclase activity* or *Toll-like receptor binding*. To avoid confusion between gene product names and their molecular functions, GO molecular functions are often appended with the word “activity” (a *protein kinase* would have the GO molecular function *protein kinase activity*).

Cellular Component

The locations relative to cellular structures in which a gene product performs a function, either cellular compartments (*e.g.*, *mitochondrion*), or stable macromolecular complexes of which they are parts (*e.g.*, the *ribosome*). Unlike the other aspects of GO, cellular component classes refer not to processes but rather a cellular anatomy.

Biological Process

The larger processes, or ‘biological programs’ accomplished by multiple molecular activities. Examples of broad biological process terms are *DNA repair* or *signal transduction*. Examples of more specific terms are *pyrimidine nucleobase biosynthetic process* or *glucose transmembrane transport* . Note that a biological process is not equivalent to a pathway. At present, the GO does not try to represent the dynamics or dependencies that would be required to fully describe a pathway.

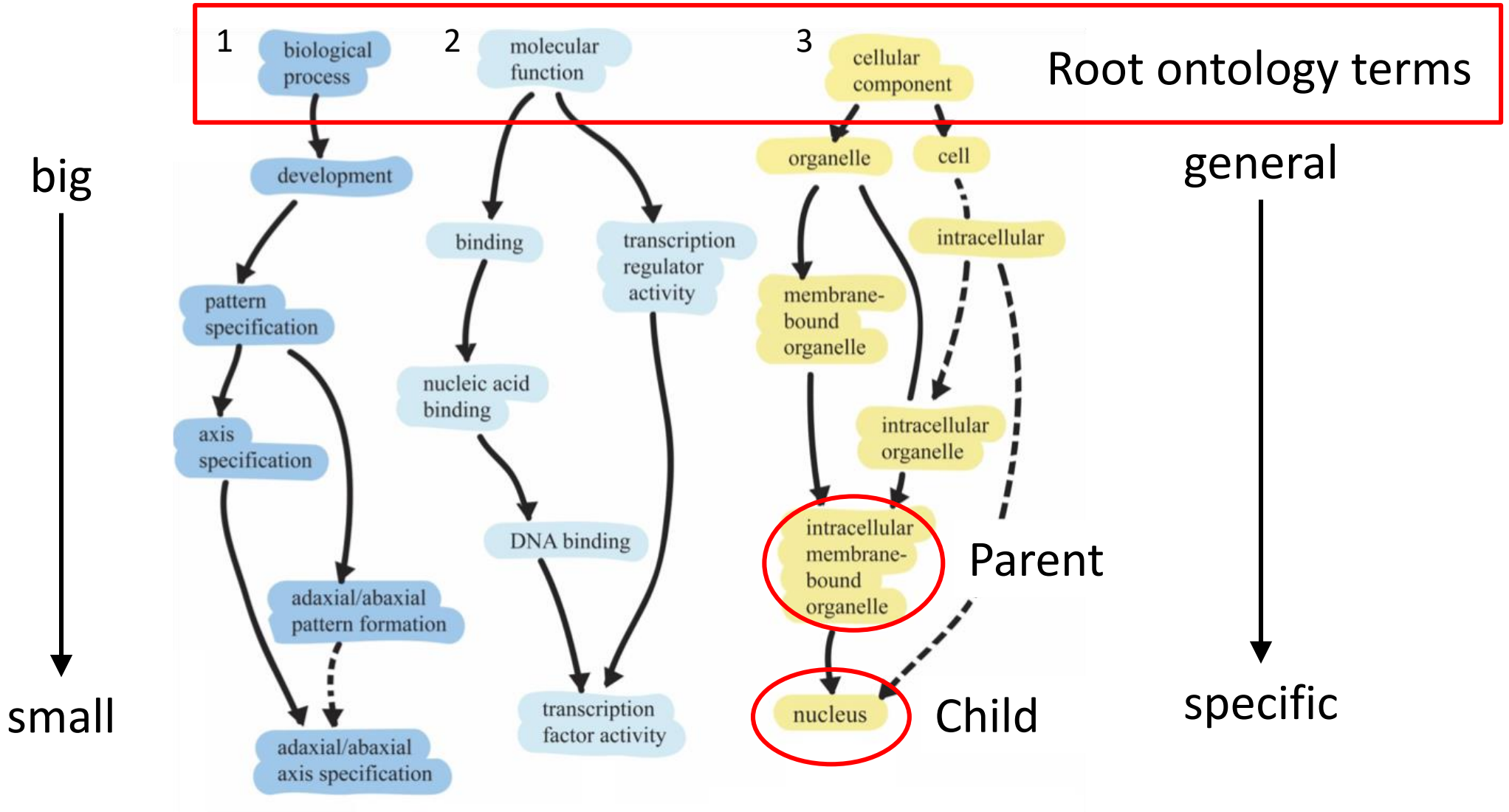


Term Information

Accession GO:0019319
Name hexose biosynthetic process
Ontology biological_process
Synonyms hexose anabolism, hexose biosynthesis, hexose formation, hexose synthesis
Alternate IDs None
Definition The chemical reactions and pathways resulting in the formation of hexose, any monosaccharide with a chain of six carbon atoms in the molecule. *Source:* ISBN:0198506732

Data health

Gene/product	Gene/product name	Organism
Sds	serine dehydratase	Mus musculus
G6pc	glucose-6-phosphatase, catalytic	Mus musculus
Gnpda1	glucosamine-6-phosphate deaminase 1	Mus musculus
Nr3c1	nuclear receptor subfamily 3, group C, member 1	Mus musculus
Gpt	glutamic pyruvic transaminase, soluble	Mus musculus
Ranbp2	RAN binding protein 2	Mus musculus
Ptpn2	protein tyrosine phosphatase, non-receptor type 2	Mus musculus
Stk11	serine/threonine kinase 11	Mus musculus
Gm10768	predicted gene 10768	Mus musculus
Fbp1	fructose bisphosphatase 1	Mus musculus


1
biological process
development
pattern specification
axis specification
adaxial/abaxial pattern formation
adaxial/abaxial axis specification
2
molecular function
binding
nucleic acid binding
DNA binding
transcription regulator activity
transcription factor activity
3
cellular component
organelle
cell
intracellular
membrane-bound organelle
intracellular organelle
intracellular membrane-bound organelle
nucleus
Root ontology terms
big
small
general
specific
Parent
Child

Genes assigned to ontology terms

Nanog homeobox [Source:HGNC Symbol;Acc:HGNC:20857]

- Cellular Component
  - GO:0005634 nucleus
  - GO:0005654 nucleoplasm
  - GO:0005730 nucleolus

- Molecular Function
  - GO:0003677 DNA binding
  - GO:0003700 transcription factor activity, sequence-specific DNA binding
  - GO:0003714 transcription corepressor activity
  - GO:0005515 protein binding
  - GO:0043565 sequence-specific DNA binding

- Biological Process
  - GO:0001714 endodermal cell fate specification
  - GO:0006351 transcription, DNA-templated
  - GO:0006355 regulation of transcription, DNA-templated
  - GO:0007275 multicellular organism development
  - GO:0008283 cell proliferation
  - GO:0019827 stem cell population maintenance
  - GO:0030154 cell differentiation
  - GO:0035019 somatic stem cell population maintenance
  - GO:0045595 regulation of cell differentiation
  - GO:0045944 positive regulation of transcription from RNA polymerase II promoter
  - GO:1903507 negative regulation of nucleic acid-templated transcription

Reactions and Pathways Exercise

Babraham Bioinformatics

Regulation and Interactions

- The regulation of genes is as important as their structure or function
- Several sources of useful information
  - Regulatory binding proteins, mostly transcription factors
  - Interactions with other proteins to form complexes
  - Composition of known complexes

Transcription Factor Information

JASPAR 2020

Detailed information of matrix profile **MA0605.2**

Home > Matrix > MA0605.2

Profile summary

Add

Name:	ATF3
Matrix ID:	MA0605.2
Class:	Basic leucine zipper factors (bZIP)
Family:	Jun-related factors
Collection:	CORE
Taxon:	Vertebrates
Species:	Homo sapiens
Data Type:	HT-SELEX
Validation:	12815047
Uniprot ID:	P18847
Source:	28473536
Comment:	

Sequence logo

Download SVG

Frequency matrix

JASPAR | TRANSFAC | MEME | RAW PFM | Reverse comp.

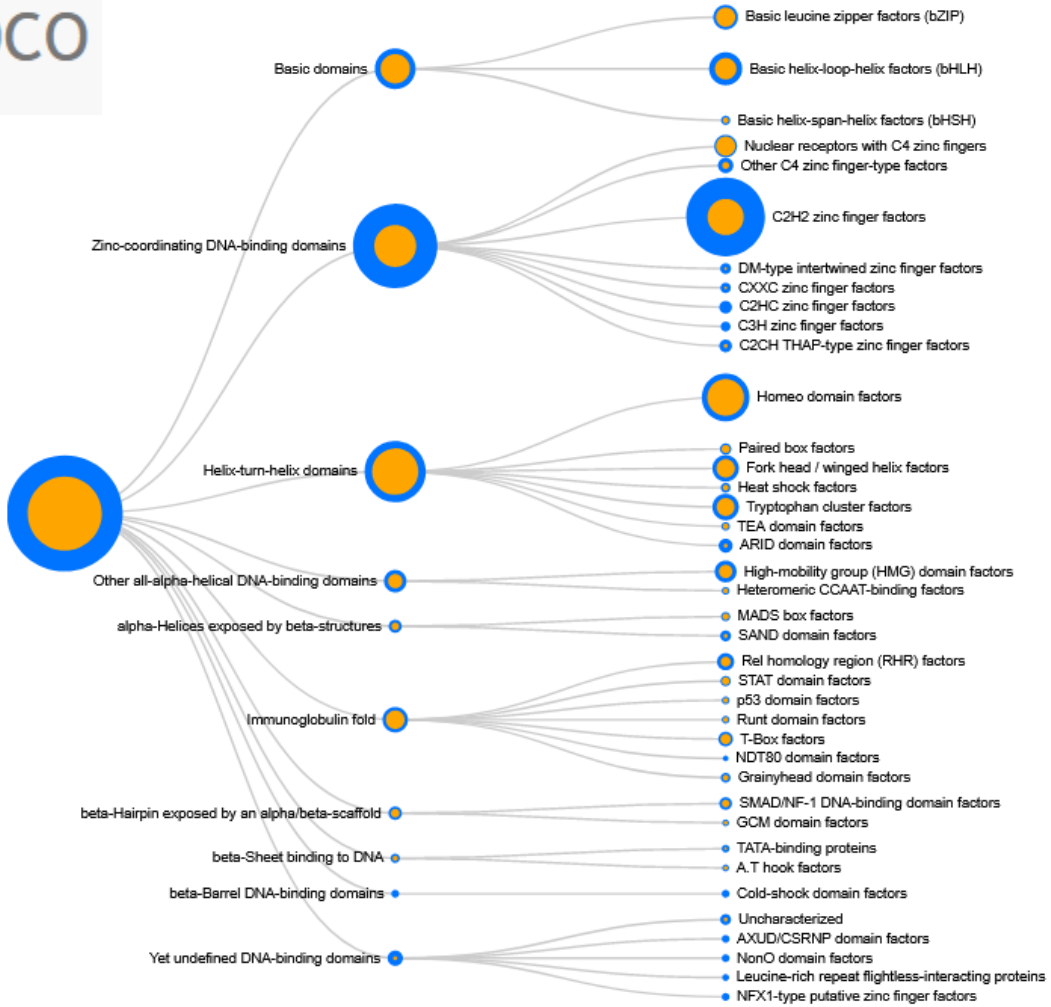
A [	8505	24741	0	0	40546	0	891	0	1520	40546	0	6919	]
C [	8220	1354	0	0	0	40546	0	0	40546	0	15737	16820	]
G [	16894	15805	1	40546	0	0	40546	0	0	0	1094	8242	]
T [	6926	0	40546	1366	0	556	0	40546	0	0	24808	8564	]

Transcription Factor Information

HOCOMOCO

Model info

Transcription factor	BCL6B (GeneCards)
Model	BCL6B_HUMAN.H11MO.0.D
Model type	Mononucleotide PWM
LOGO	
LOGO (reverse complement)	
Data source	HT-SELEX
Model release	HOCOMOCOv10
Model length	11
Quality	D
Motif rank	0
Consensus	nYGCTTTCTAG



Genes Regulated by a Transcription Factor

- Difficult to predict - lots of false positives
  - Swiss Regulon
  - 

  - ReMap2022
  - GTRD
    Gene Transcription Regulation Database

BRCA2

Entrez	Description	Chrom.	Strand	Promoter (Start - Stop)	TSS
675	breast cancer 2, early onset	chr13	+	32884616 - 32890116	32889616

Transcription Factor Binding Sites

Download all TFBS in the BRCA2 promoter

Show 10 entries Search:

Motif	Source	Strand	Start	Stop	PValue	Match Sequence	Overlap w/ Footprints
Pax4_MA0068.1	JASPAR	+	32888990	32889019	0.0E+00	AAAAAAAAAAGCAAAAGATACTACCAAGCC	30
V_GC_01_M00255	TRANSFAC	-	32889167	32889180	0.0E+00	AGTGGGCGGGGCTG	14
V_LDSPOLYA_B_M00317	TRANSFAC	-	32889437	32889452	0.0E+00	AGTGTGTGTTCTCTTC	16
V_SOX2_Q6_M01272	TRANSFAC	-	32889284	32889299	0.0E+00	AATACCTTTGTTCTGA	16
V_SP1_Q4_01_M00932	TRANSFAC	-	32889989	32890001	0.0E+00	AAGGGGCGGGGCT	13
V_STAT5A_01_M00457	TRANSFAC	+	32890101	32890115	0.0E+00	AATTTCTTGGAAACA	15
V_STAT5A_Q6_M01890	TRANSFAC	+	32890100	32890112	0.0E+00	AAATTTCTTGGAA	13
V_STAT5B_01_M00459	TRANSFAC	+	32890101	32890115	0.0E+00	AATTTCTTGGAAACA	15
V_STAT_Q6_M00777	TRANSFAC	+	32890099	32890111	0.0E+00	GAAATTTCTTGGA	13
SP1_C2H2_DBD_monomeric_11_1	SELEX	+	32889169	32889179	1.0E-05	GCCCCGCCCAC	11

Showing 1 to 10 of 196 entries Previous Next

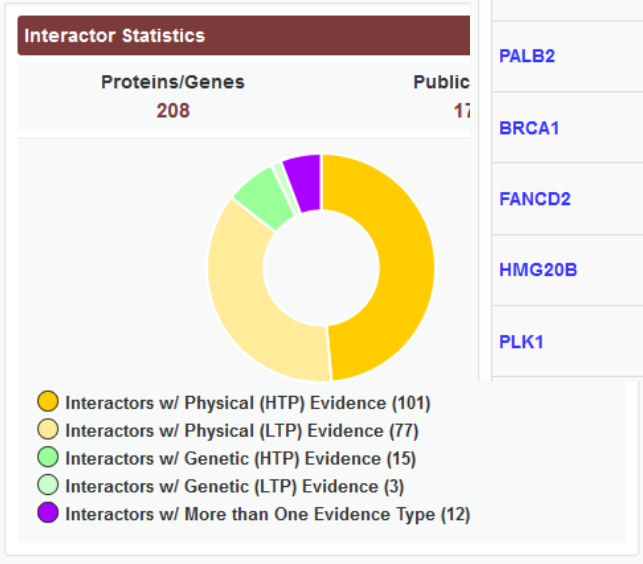
Gene Interactions

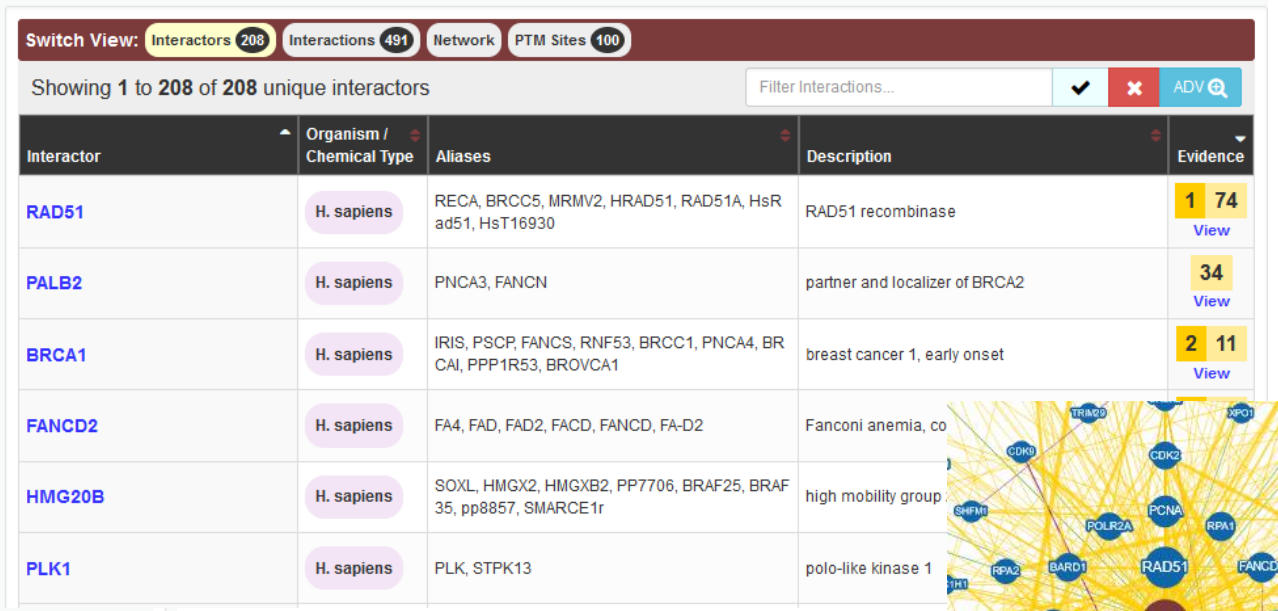
- Many genes form stable or transitory interactions with others
- Knowing the genes that interact helps understand biology

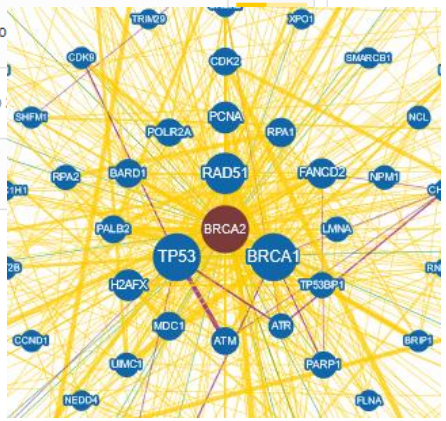
BioGRID 4.4

IntAct

STRING







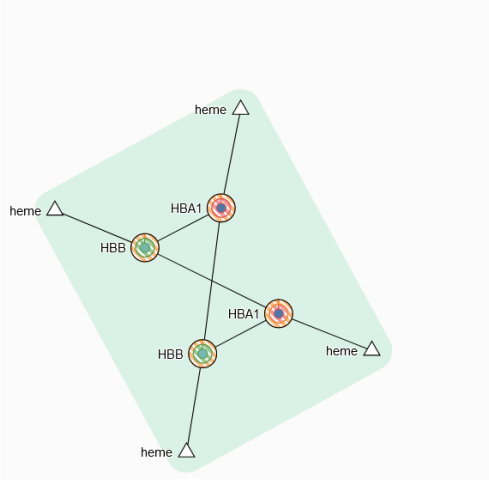
Types of Interaction

- Physical
  - Two proteins directly interact, either stably or transiently
- Genetic
  - One gene influences another, normally after modification
    - Co-expression
    - Knockout compensation

Complex Prediction

- Many proteins interact with several others, but at different times
- Complexes suggest that multiple proteins directly associate
  - Can't always be clearly predicted from pairwise interactions
  - Other experimental methods are required

Complex Portal



Legend	Description	Stoichiometry
●	**protein - HBA1 (unspecified role)** P69905 Hemoglobin subunit alpha	2
●	**protein - HBB (unspecified role)** P68871 Hemoglobin subunit beta	2
△	**small molecule - heme (cofactor)** CHEBI:30413 heme	4

Regulatory Information Exercise

Babraham Bioinformatics

Sequence Variants

Babraham Bioinformatics

Sequence Variants

Reference	GATCTTA**G**CTGA
Variant	GATCTTA**C**CTGA

- Germline variants
  - Happen in sperm or eggs
  - Completely inherited into the next generation
  - Can cause genetic disease
- Somatic variants
  - Happen in other tissues
  - Partially penetrant
  - Common cause of cancer

Types of Variant

Functional Variant Consequences

- Within Coding Region
  - Silent (codon changes, but same translation)
  - Missense (change translation from one amino acid to another)
  - Nonsense (change translation from one amino acid to STOP)
  - Frameshift (InDel changing the translation frame)

- Outside CDS
  - Breaks or adds splice junction
  - Changes functional binding site

Structural variants

- Chromosomal copy number change
  - Gain or loss of a chromosome
  - Leads to serious genetic disease

- Segmental Deletion / Duplication
  - Large parts of chromosomes deleted, duplicated, inverted, translocated
  - 1kb to 3Mbp
  - Affects many genes, can lead to gene fusions

Databases of Variants

- Common genomic variants
  - Measured across a large population
  - Shows natural variation
  - Not necessarily linked to disease
  - Used for studying populations and families

- Functional variants
  - Variants with an associated phenotype
  - Often disease related but can be any measurable phenotype

Variant Databases

- Single Variants
  - dbSNP (https://www.ncbi.nlm.nih.gov/snp/)
    - Full reference for any reported SNPs, mix of functional and non-functional
  - HGMD (http://www.hgmd.cf.ac.uk)
    - Human genetic disease focussed database
  - COSMIC (https://cancer.sanger.ac.uk/cosmic)
    - Mutations observed in Cancer
    - Also has details of mutations in immortalised cell lines

- Larger Regions
  - dbVar (https://www.ncbi.nlm.nih.gov/dbvar/)
    - Counterpart to dbSNP for larger variants
  - ClinVar (https://www.ncbi.nlm.nih.gov/clinvar/)
    - Larger variants with clinical relevance
  - OMIM (https://www.ncbi.nlm.nih.gov/omim)
    - A more wide ranging collection of the phenotypic variation linked to genes

Variant Terminology

- Minor Allele Frequency (MAF)
  - How prevalent the variant is in the population
- Impact scores (SIFT / PolyPhen etc)
  - A quantitative value assessing the likely biological impact of a variant

Variant Exercise

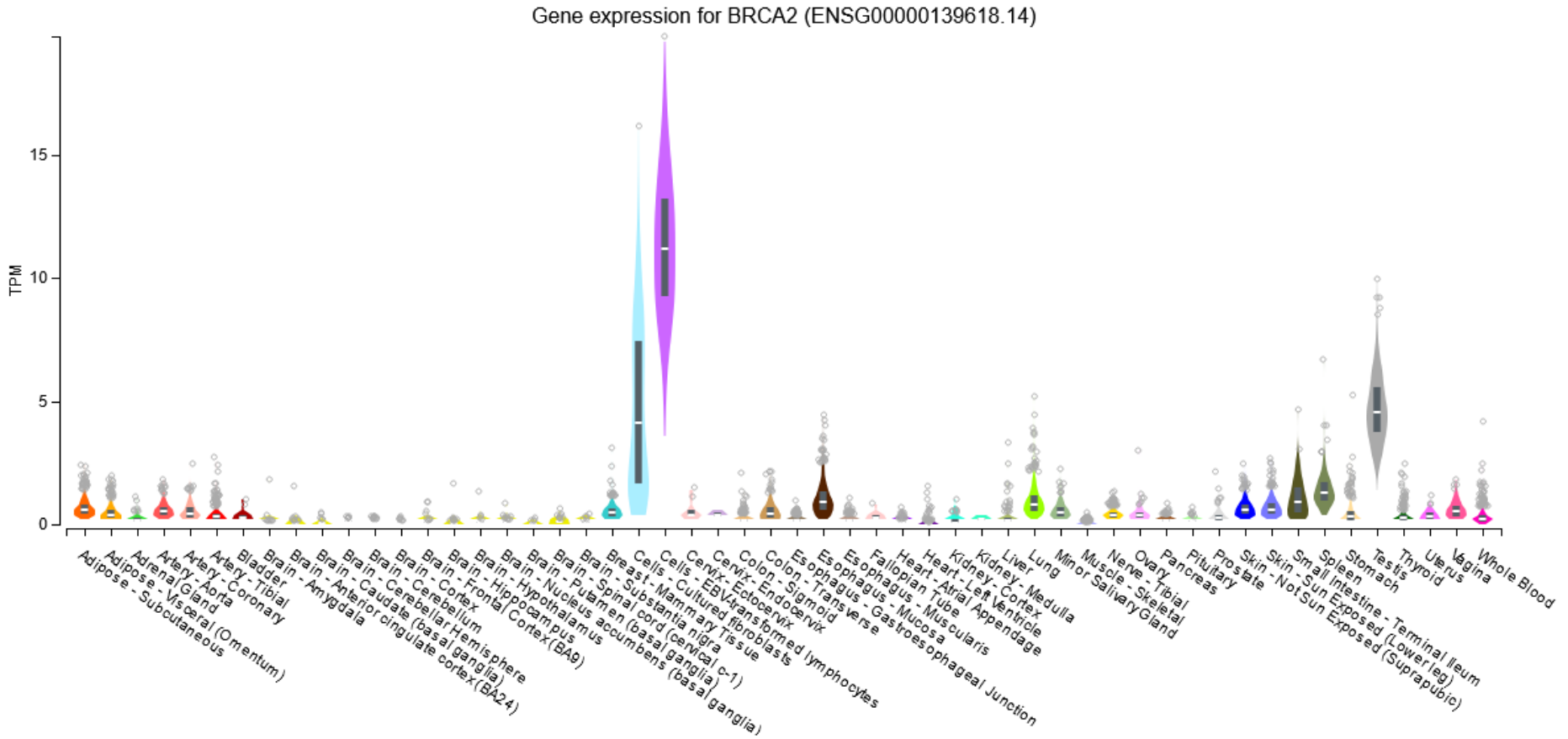
Babraham Bioinformatics

Other Information and Data Sources

Babraham Bioinformatics

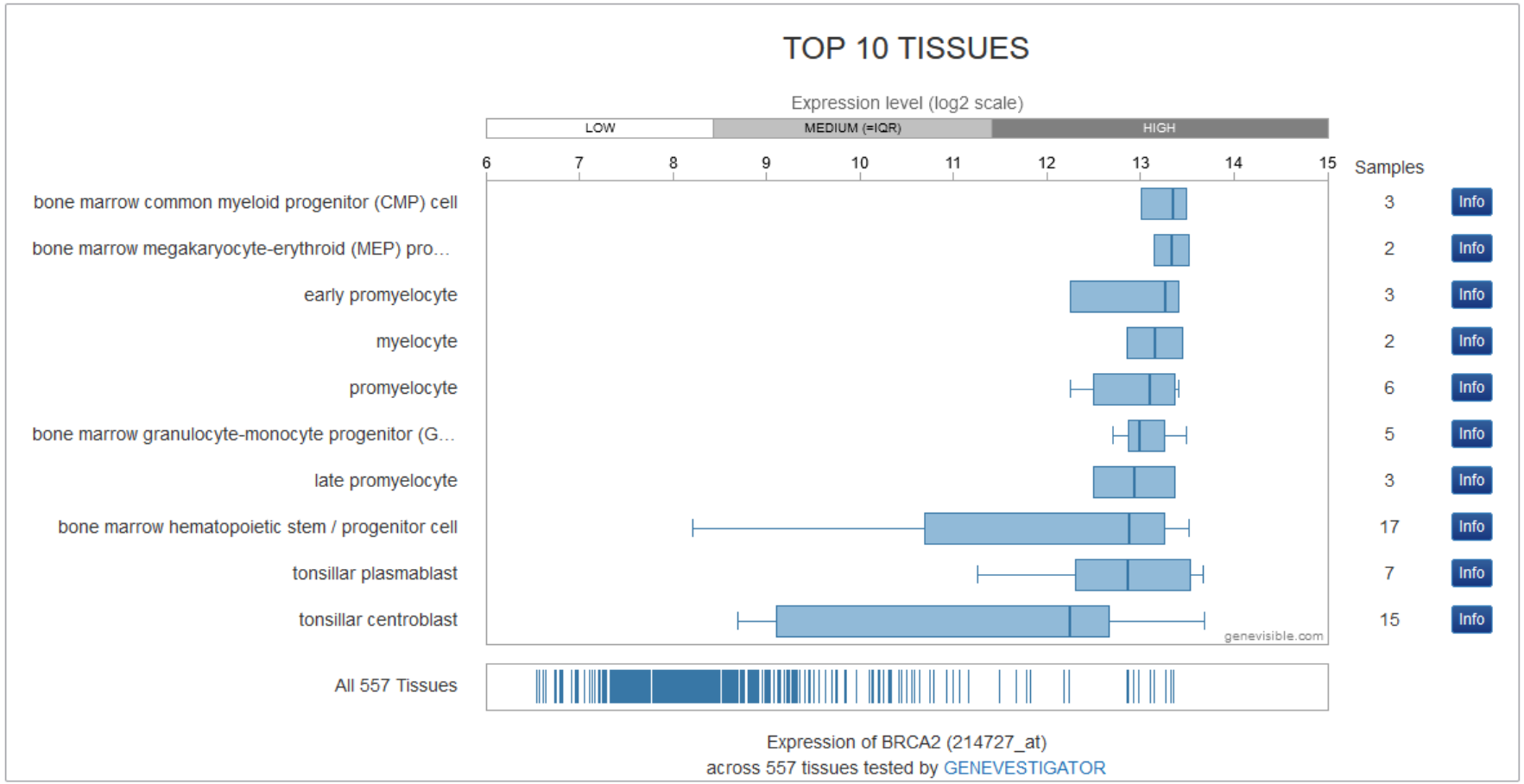
Gene Expression Information

GTEx Portal

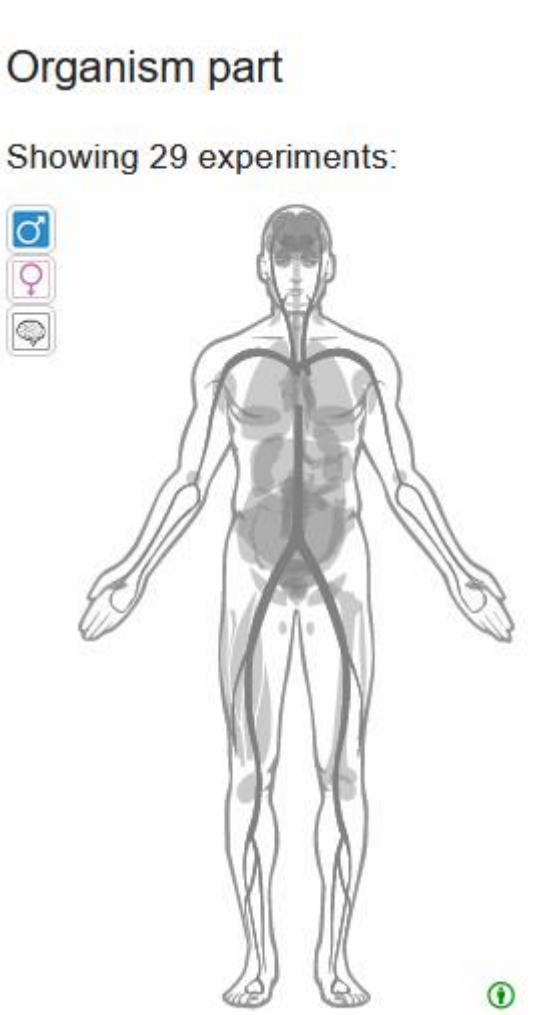


Gene Expression Information

Genevisible

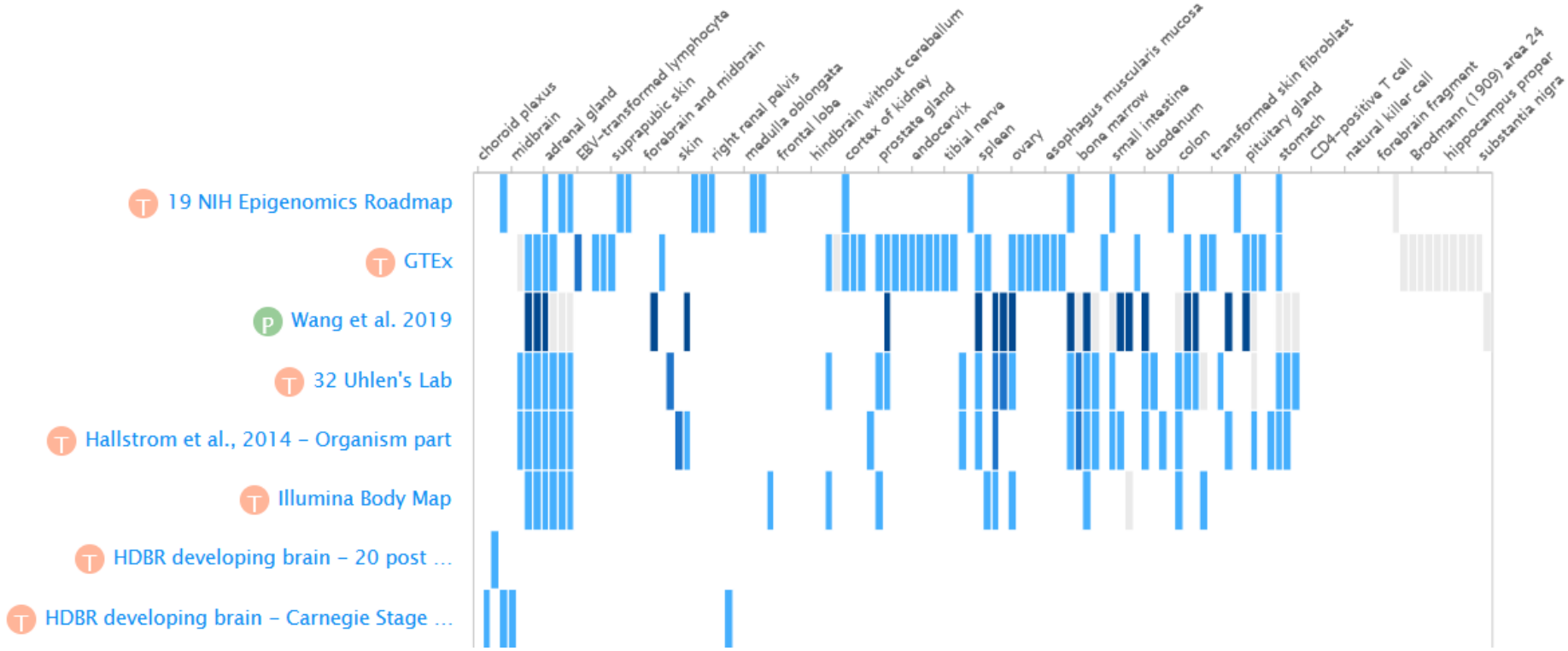


Gene Expression Information



Expression Atlas

Gene expression across species and biological conditions



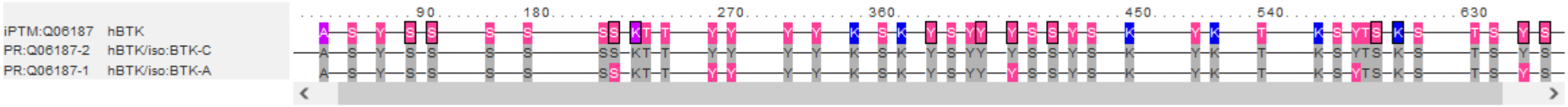
Post translational Modifications

- Many proteins are modified after they have been translated
  - Phosphorylation
  - Glycosylation
  - Ubiquitination
  - Nitrosylation
  - Methylation
  - Acetylation
  - Lipidation
  - Proteolysis

iPTMnet

Both PTMs observed on a protein and proteins modified by a query gene.

Site	PTM Type	PTM Enzyme	Score
All	All		2 selected
S21	Phosphorylation		★★★☆
S115	Phosphorylation		★★★☆
S180	Phosphorylation	P05771 (PRKCB)	★★★★
Y223	Phosphorylation	Q06187 (BTK) , Q08881 (ITK) , P07948 (LYN) , P00519 (ABL1) , A0A173G4P4 (Abl fusion) , P42680 (TEC)	★★★★
Y551	Phosphorylation	P07948 (LYN) , Q06187 (BTK) , P12931 (SRC) , P43405 (SYK)	★★★★



Combined Gene/Protein Centric Datasources

https://www.uniprot.org/

https://www.genecards.org

https://www.ncbi.nlm.nih.gov/gene/

https://www.wikigenes.org/

UniProt

- Function
- Names & Taxonomy
- Subcell. location
- Pathol./Biotech
- PTM / Processing
- Expression
- Interaction
- Structure
- Family & Domains
- Sequences (1+)
- Similar proteins
- Cross-references
- Entry information
- Miscellaneous

wikigenes
collaborative publishing

Disease Relevance
High Impact publication summaries
Biological Context
Anatomical Context
Chemical Compound Associations
Physical Interactions
Enzymatic Interactions
Regulatory relationships
Analytical, diagnostic and therapeutic context
References

NCBI Gene

- Summary
- Genomic context
- Genomic regions, transcripts, and products
- Expression
- Bibliography
- Variation
- Pathways from PubChem
- Interactions
- General gene information
  - Markers, Homology, Gene Ontology
- General protein information
- NCBI Reference Sequences (RefSeq)
- Related sequences
- Additional links

GeneCards®
THE HUMAN GENE DATABASE

Jump to section									
	Aliases	Disorders	Domains	Drugs	Expression	Function	Genomics	Localization	Orthologs
	Paralogs	Pathways	Products	Proteins	Publications	Sources	Summaries	Transcripts	Variants
Research Products	Antibodies	Assays	Proteins	Inhib. RNA	CRISPR	Exp. Assays	miRNA	Drugs	Animal Models
	Cell Lines	Clones	Primers	Genotyping					

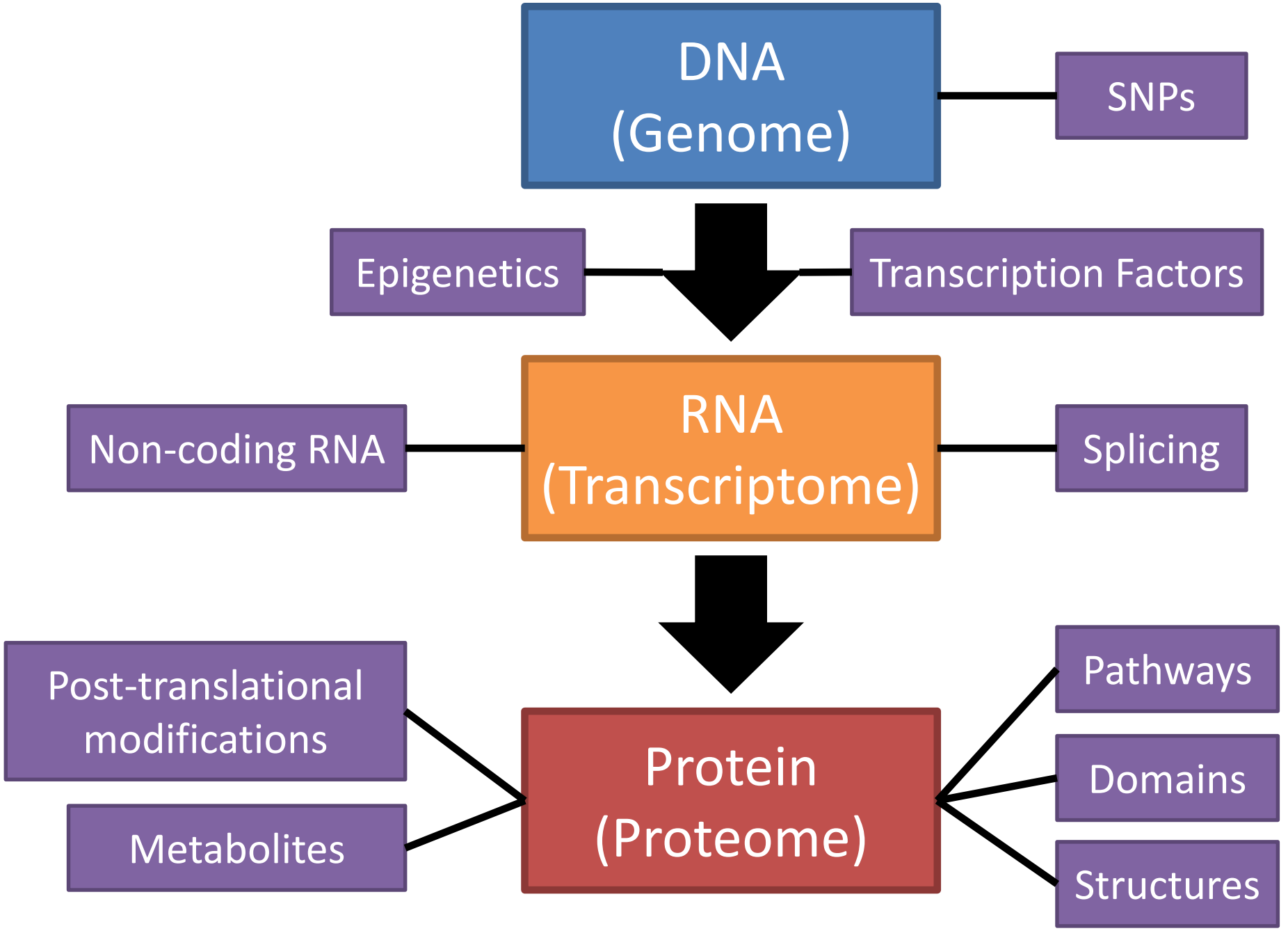
Final Summary Exercise

Babraham Bioinformatics

Experimental Data Types and Repositories

Simon Andrews, Chris Hall, Judith Webster, Eoin Fahy, Laura Biggins, Hanneke Okkenhaug, Simon Walker

Babraham Bioinformatics


DNA
(Genome)
SNPs
Epigenetics
Transcription Factors
RNA
(Transcriptome)
Non-coding RNA
Splicing
Post-translational
modifications
Metabolites
Protein
(Proteome)
Pathways
Domains
Structures

Big Data Generation

- High throughput sequencing
  - Genomics, Transcriptomics, Epigenetics
- Multi-channel Flow Cytometry
  - Cell surface proteomics
- Mass Spectrometry
  - Proteomics, Metabolomics
- Biological Imaging
  - Cell / Tissue structure, Proteomics, Metabolomics

Data Repositories

- For many techniques deposition of data in a suitable repository is a condition of publication
- Repositories are more developed and complete for some techniques than others
- Still a growing area

Mandatory deposition	Suitable repositories
Protein sequences	Uniprot
DNA and RNA sequences	Genbank
	DNA DataBank of Japan (DDBJ)
	EMBL Nucleotide Sequence Database (ENA)
DNA and RNA sequencing data	NCBI Trace Archive
	NCBI Sequence Read Archive (SRA)
Genetic polymorphisms	dbSNP
	dbVar
	European Variation Archive (EVA)
Linked genotype and phenotype data	dbGAP
	The European Genome-phenome Archive (EGA)
Macromolecular structure	Worldwide Protein Data Bank (wwPDB)
	Biological Magnetic Resonance Data Bank (BMRB)
	Electron Microscopy Data Bank (EMDB)
Gene expression data (must be MIAME compliant)	Gene Expression Omnibus (GEO)
	ArrayExpress
Crystallographic data for small molecules	Cambridge Structural Database
Proteomics data	PRIDE
*Earth, space & environmental sciences	Recommended Repositories

FAIR Data Principles

- Designed to make data as useful as possible to future researchers
  - Findable
    - Unique accession code
    - Rich metadata
  - Accessible
    - Automated query and download API
  - Iteroperable
    - Use of open formats
    - Standard Ontologies for descriptions
  - Reusable
    - Clear licensing
    - Annotated to common community standards

High Throughput Sequencing

Illumina Novaseq

ONT MinION

PacBio Revio

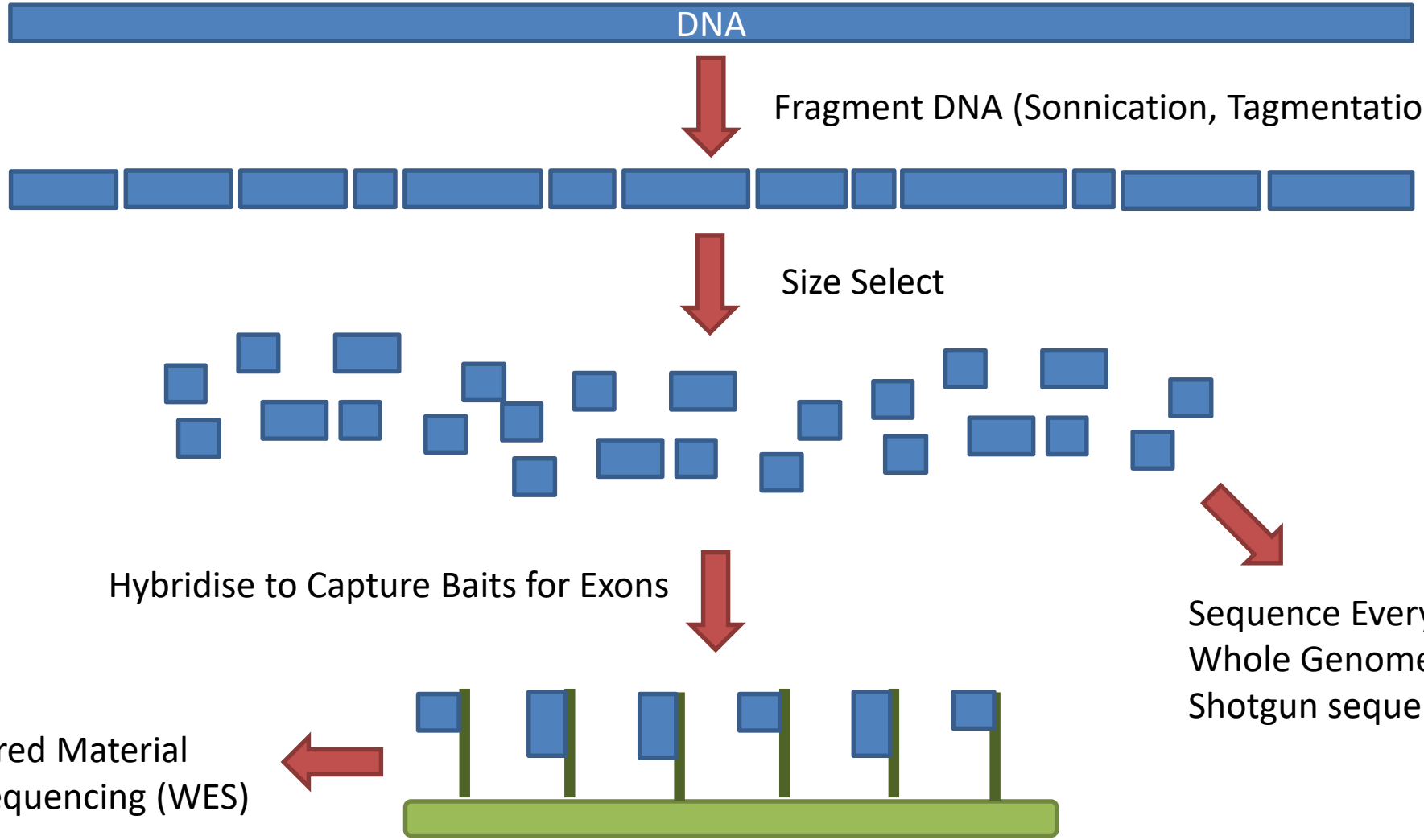
Data Generation Capacity

Sequencer	Read Length	Bases per run
Illumina NovaSeq	50-250bp	3000 Gbp
ONT Promethion 48	1kb - 80Mbp	48 x 20-90 Gbp
PacBio Revio	1kb - 20kb	90 Gbp

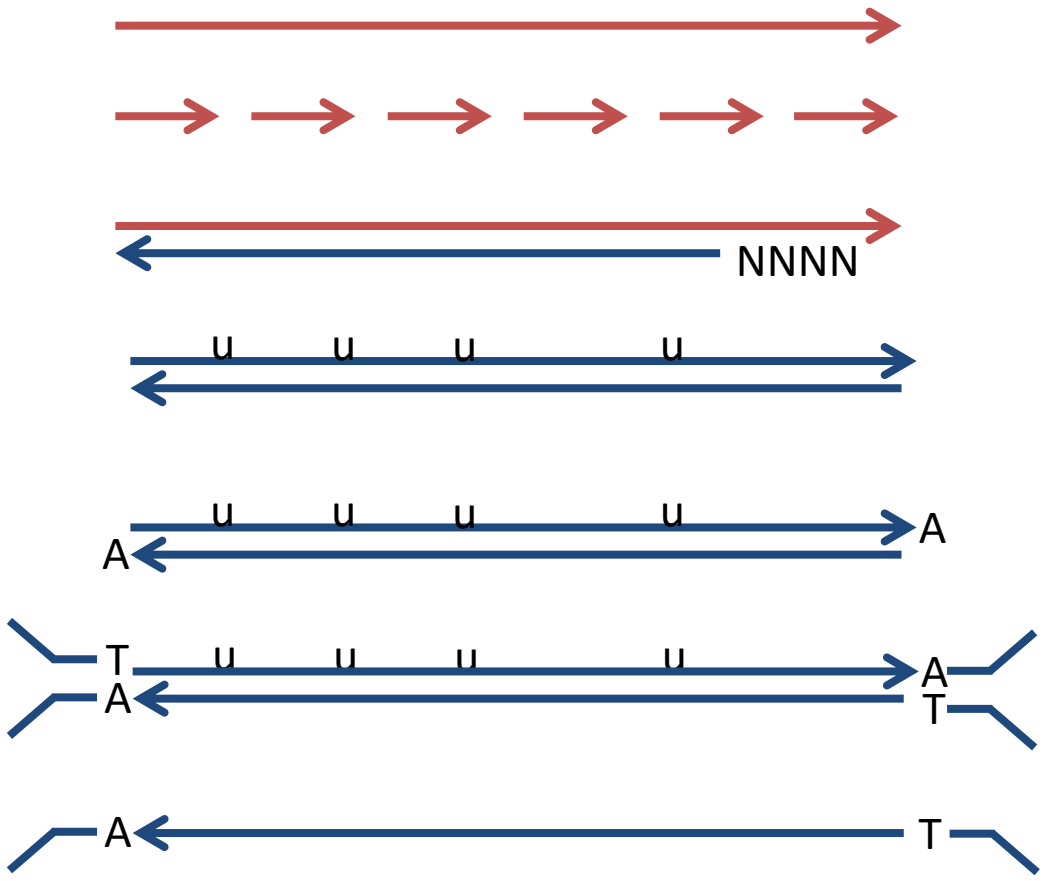
What can you measure?

- Genomics
  - Whole genome sequencing, Targeted Sequencing
- Transcriptomics
  - RNA-Sequencing
- Regulation
  - Accessible DNA (ATAC-Seq), Histone Modifications, Transcription Factor binding sites
- Epigenetics
  - DNA Methylation, Chromatin Structure

Genome Sequencing



RNA-Sequencing



rRNA depleted mRNA

Fragment

Random prime + RT

2nd strand synthesis (+ U)

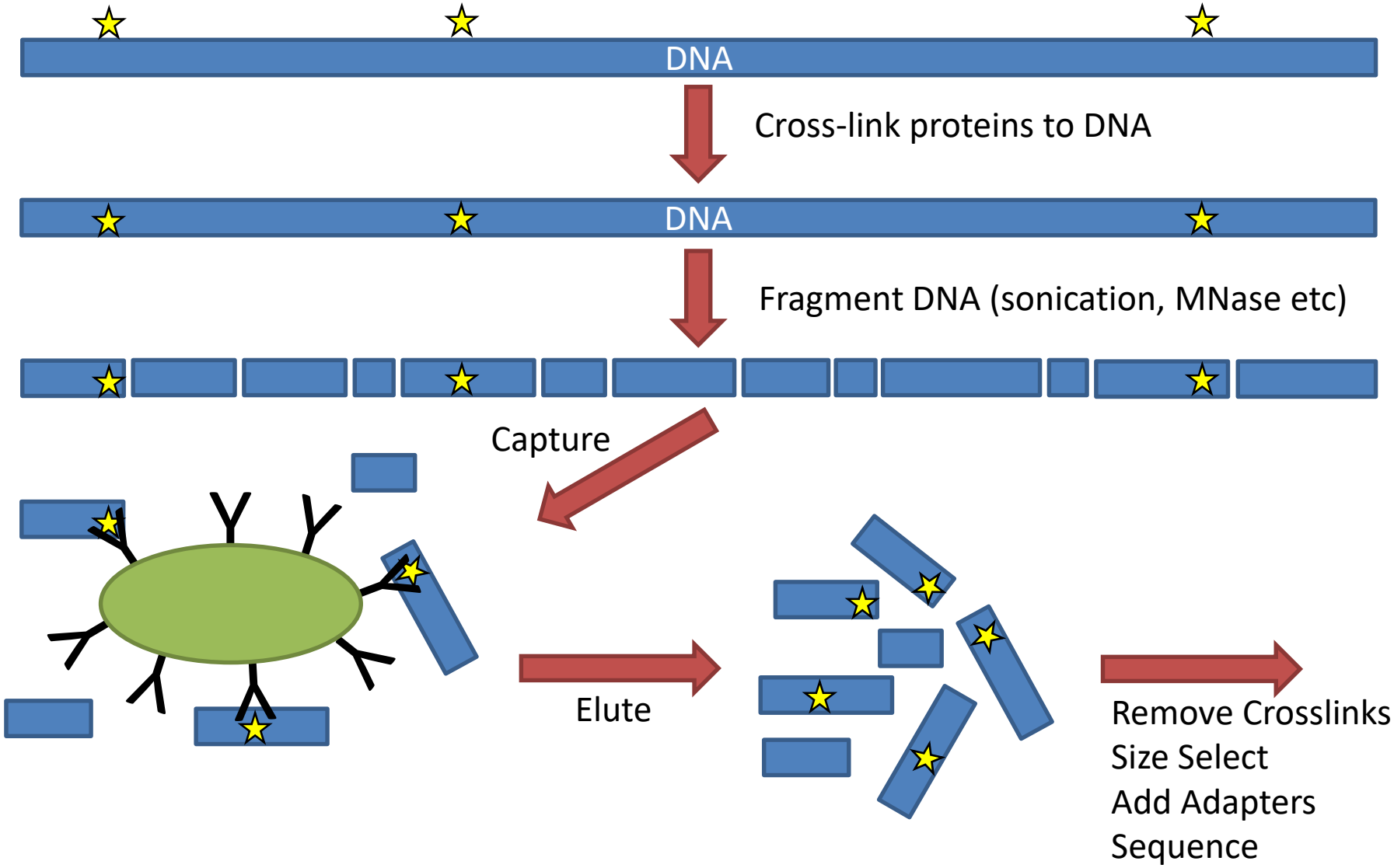
A-tailing

Adapter Ligation

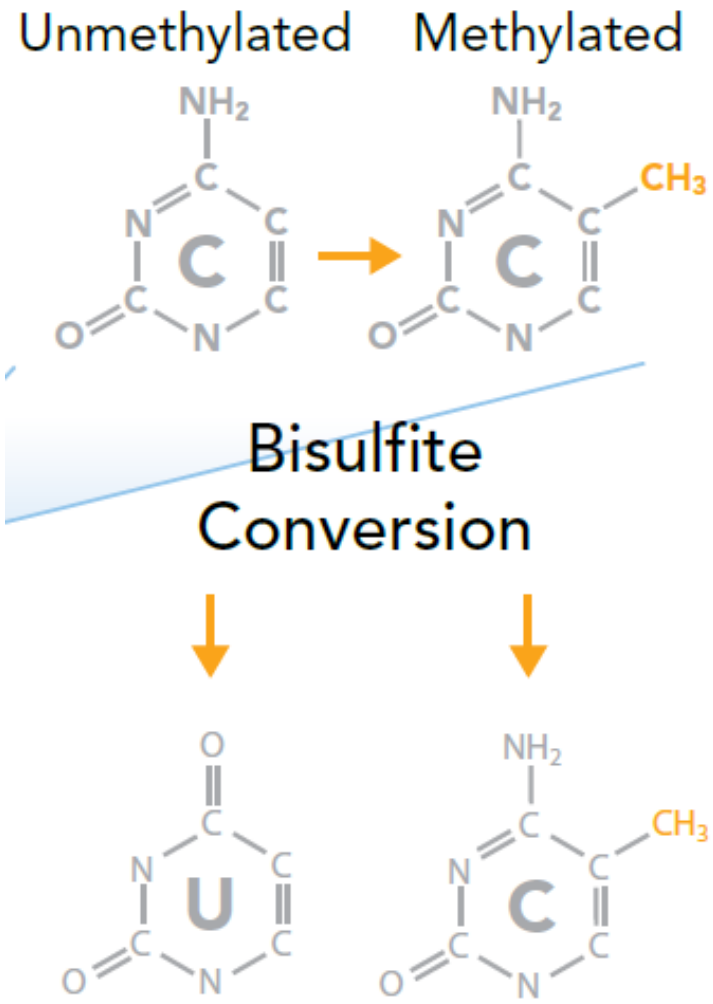
(U strand degradation)

Sequencing

Enrichment Sequencing



Bisulphite Sequencing



me me

CCAGTCGCTATAGCGCGATATCGTA

Convert

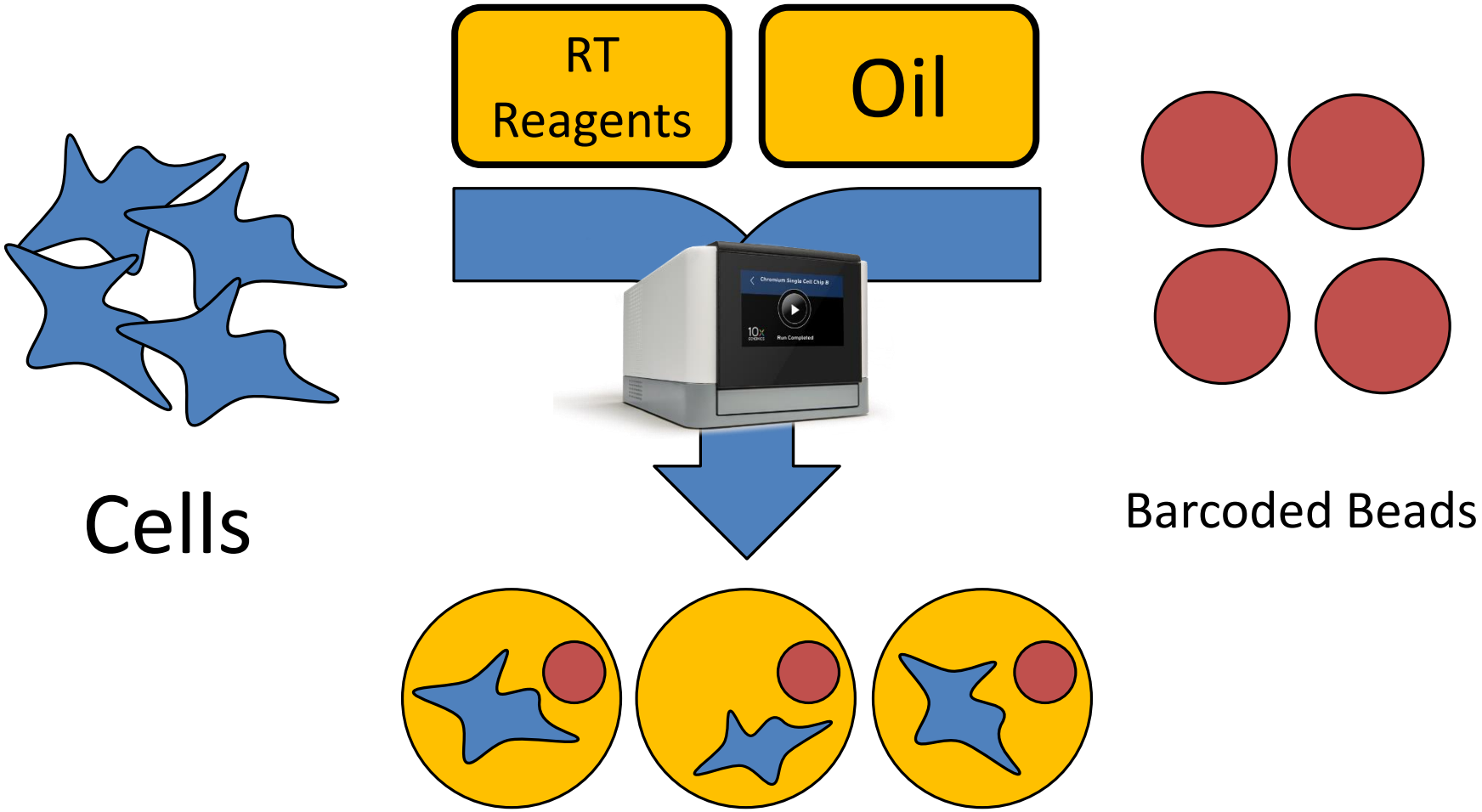
TTAGTTGCTATAGTGCGATATTGTA

Map

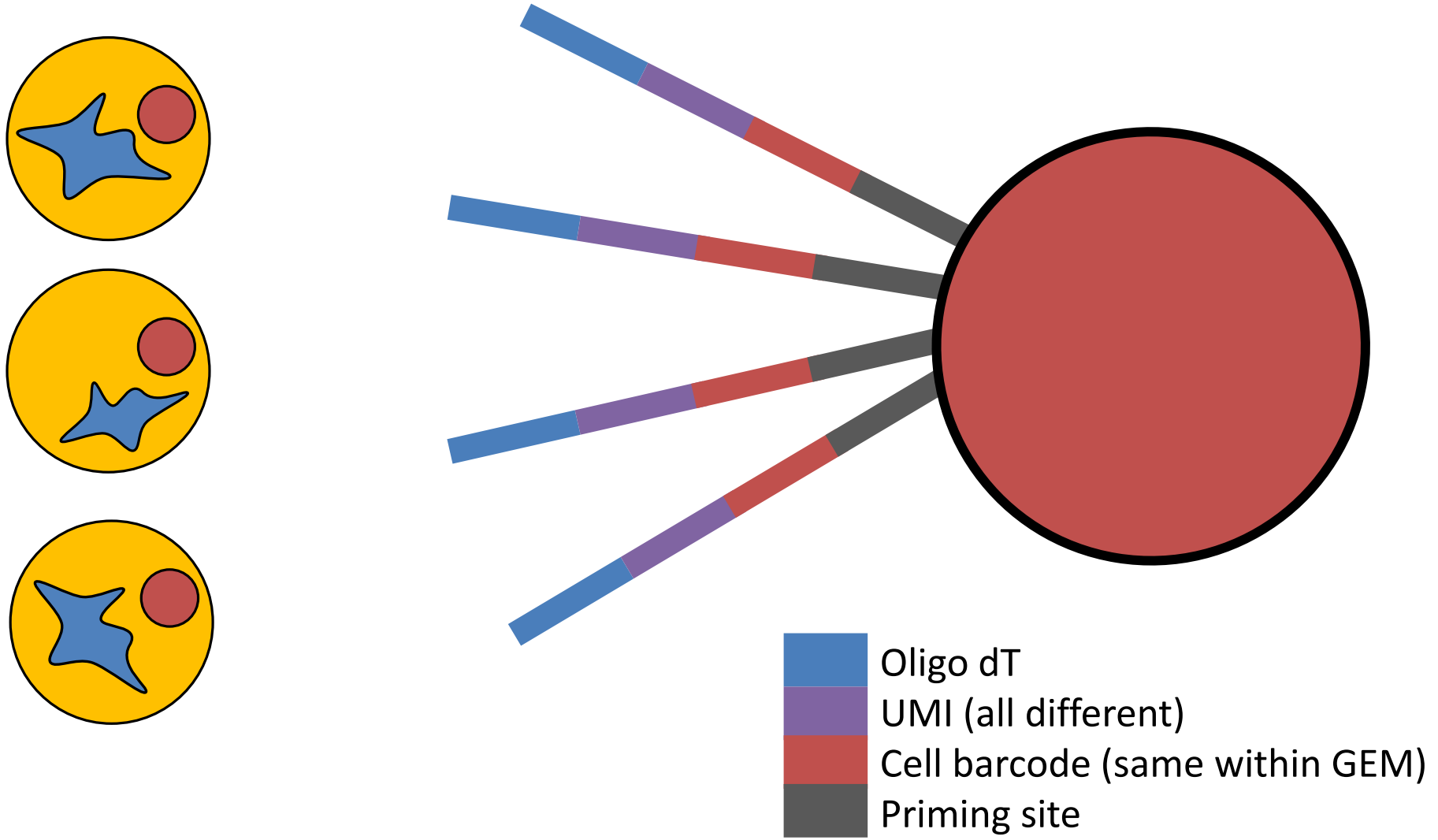
TTAGTTGCTATAGTGCGATATTGTA

. . . CCAGTCGCTATAGCGCGATATCGTA . . .

10X Single Cell RNA-Seq



10X Single Cell RNA-Seq Adapter System

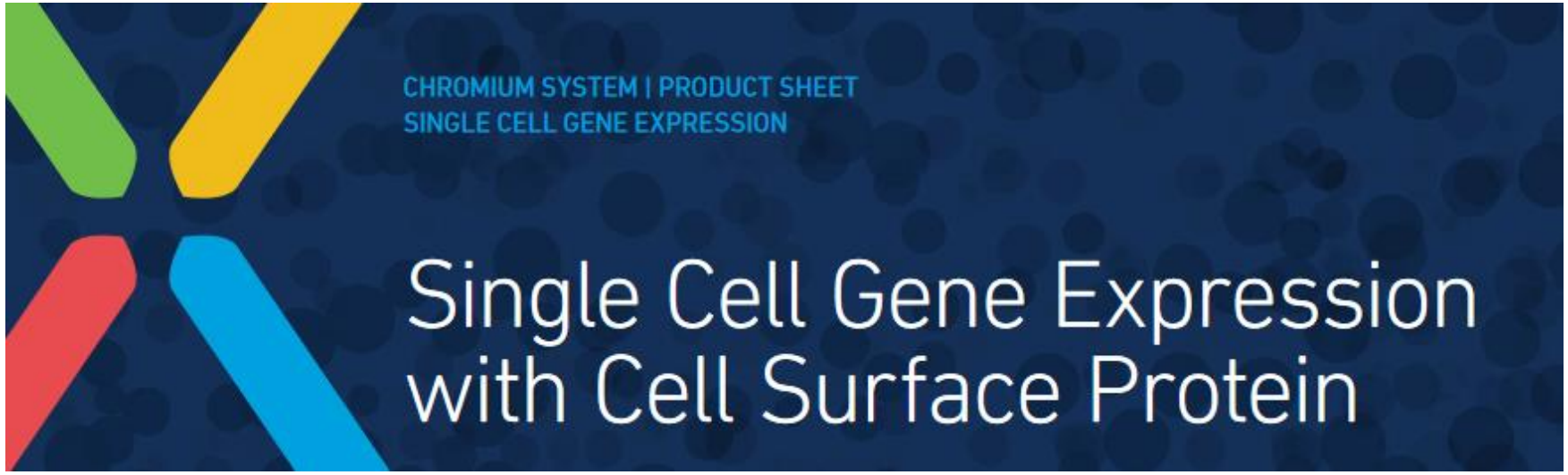


Multi-measure single cell

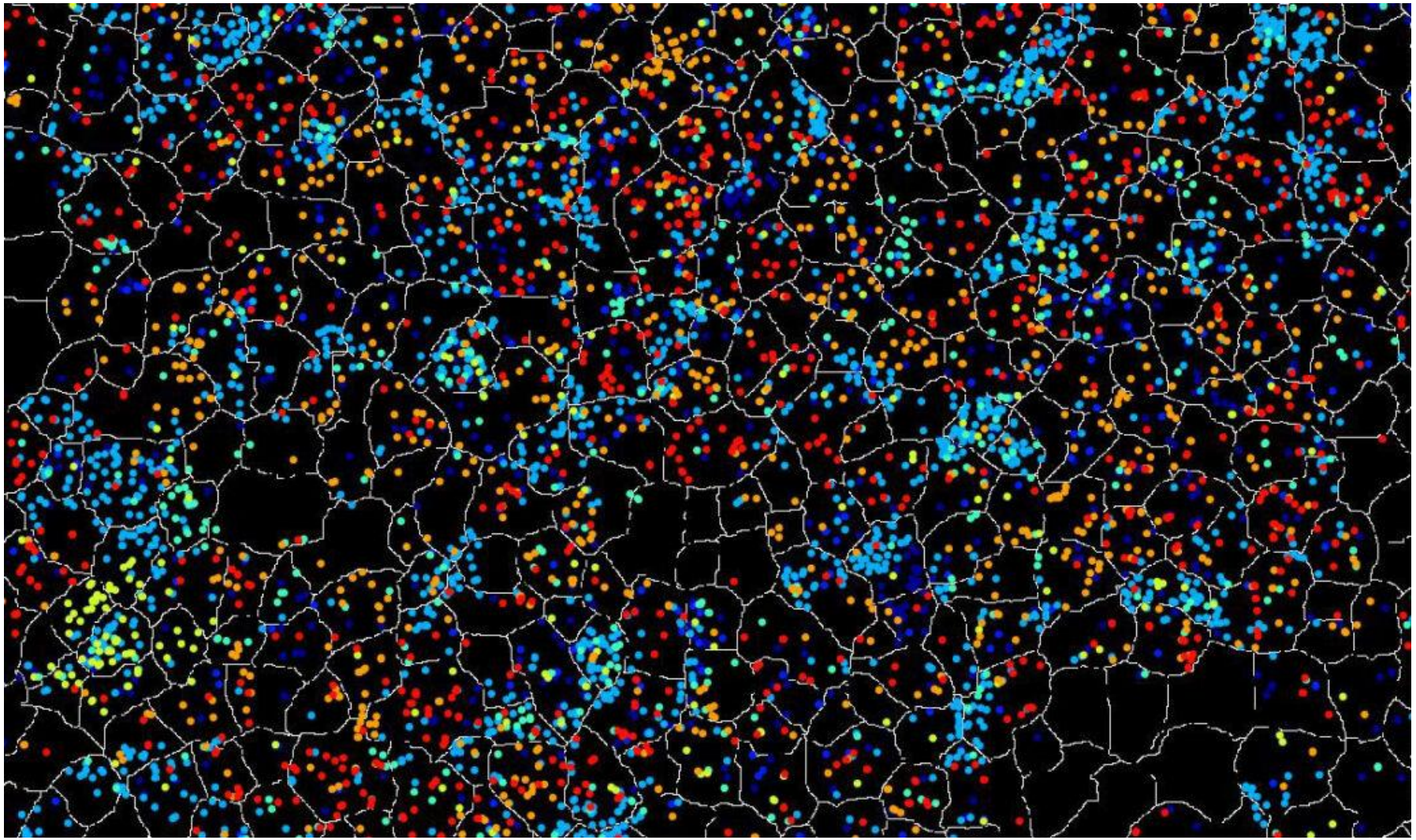
› Nat Commun. 2018 Feb 22;9(1):781. doi: 10.1038/s41467-018-03149-4.

scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells

Stephen J Clark [1], Ricard Argelaguet [2 3], ChantriolInt-Andreas Kapourani [4], Thomas M Stubbs [5], Heather J Lee [5 6 7], Celia Alda-Catalinas [5], Felix Krueger [8], Guido Sanguinetti [4], Gavin Kelsey [5 9], John C Marioni [10 11 12], Oliver Stegle [13], Wolf Reik [14 15 16]



Spatial Transcriptomics



- 10X Visium
- Nanostring CosMX
- Vizgen Merscope

FastQ Format Data

```
@HWUSI-EAS611:34:6669YAAXX:1:1:5069:1159 1:N:0:
TCGATAATACCGTTTTTTTCCGTTTGATGTTGATACCATT
+
IIHIIHIIIIIIIIIIIIIIIIIIIIIIIIIHIIIIHIIIII
@HWUSI-EAS611:34:6669YAAXX:1:1:5243:1158 1:N:0:
TATCTGTAGATTTCACAGACTCAAATGTAAATATGCAGAG
+
DF=DBD<BBFGGGGGGGBD@GGGD4@CA3CGG>DDD:D,B
@HWUSI-EAS611:34:6669YAAXX:1:1:5266:1162 1:N:0:
GGAGGAAGTATCACTTCCTTGCCTGCCTCCTCTGGGGCCT
+
:GBGGGGGGGGGDGGDEDGGDGGGGDHHDHGHHGBGG:GG
```

Public Sequencing Databases

- GEO (NCBI)
- Array Express (EBI)
  - Databases for quantitated sequencing data. Provide experimental annotation and metadata and processed quantitated data

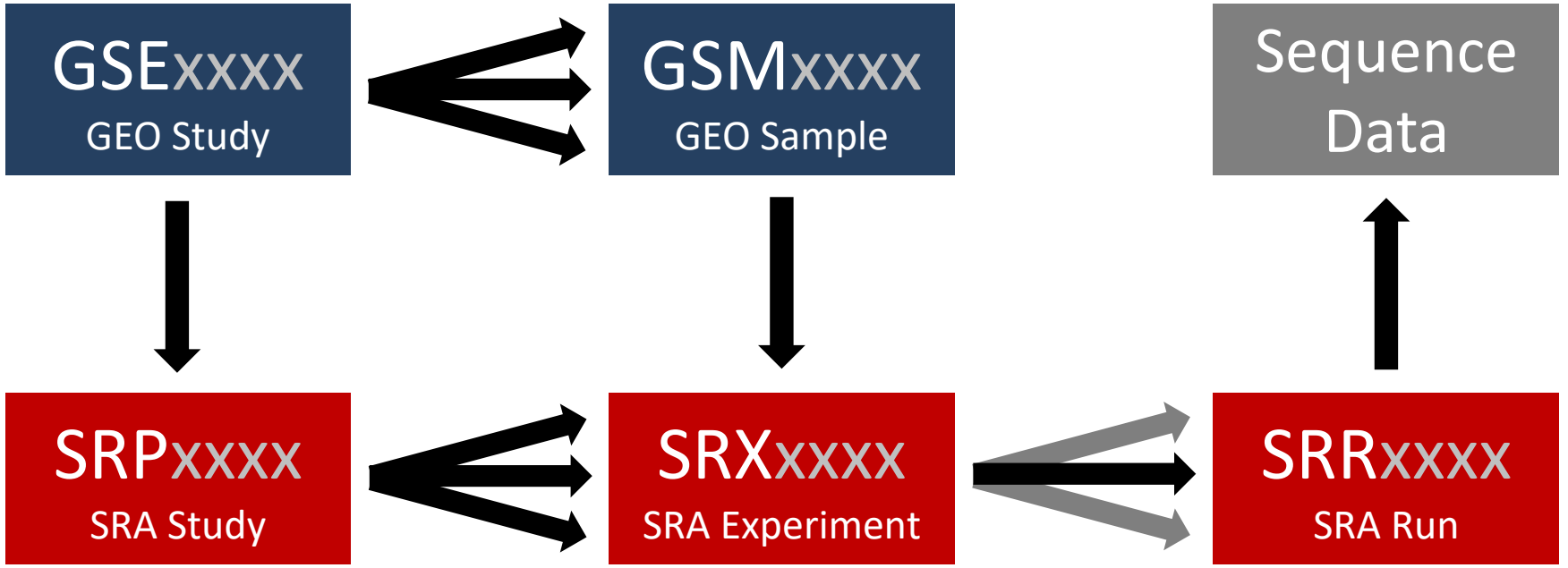
- SRA (NCBI)
- ENA (EBI)
  - Provide raw sequencing data as fastq files

Accession Codes

Transcription-induced formation of extrachromosomal DNA during yeast ageing

Ryan M. Hull[1¤a], Michelle King[1], Grazia Pizza[1¤b], Felix Krueger[2], Xabier Vergara[1¤c], Jonathan Houseley[1]*

Data Availability Statement: All relevant data are within the paper and its Supporting Information files. All sequencing files are available from the GEO database (accession number GSE135542).



Series GSE135542

Query DataSets for GSE135542

Status	Public on Oct 18, 2019
Title	Transcription-induced formation of extrachromosomal DNA during yeast ageing
Organism	Saccharomyces cerevisiae
Overall design	Aged cell samples analysed in pairs of -/+ Cu, for both wt and various mutants. 3 replicates of the 3xCUP1 experiment are included
Contributor(s)	Hull R, King M, Houseley J
Platforms (1)	GPL17342 Illumina HiSeq 2500 (Saccharomyces cerevisiae)
Samples (30)	GSM4015617 3xCUP1_24hr_1_REC-seq
	GSM4015618 3xCUP1_24hr_2_REC-seq
	GSM4015619 3xCUP1_24hr_300uM_Cu_1_REC-seq

⊞ More...

Relations

BioProject	PRJNA559191
SRA	SRP217740

Supplementary file	Size	Download	File type/resource
GSE135542_3xCUP1_processed_data_report.txt.gz	1.3 Mb	(ftp)(http)	TXT
GSE135542_cu_and_gal_processed_data_report.txt.gz	12.4 Mb	(ftp)(http)	TXT
GSE135542_mutants_processed_data_report.txt.gz	1.4 Mb	(ftp)(http)	TXT

SRA Run Selector

Raw data are available in SRA

Processed data are available on Series record

SRA Run Selector

		BioSample	Bases	Bytes	Experiment	GEO_Accession	Sample Name	source_name	strain
1	SRR9924096	SAMN12529574	1.01 G	344.69 Mb	SRX6673092	GSM4015624	GSM4015624	Cells aged 24 hours in SD media	MEP mus81
2	SRR9924097	SAMN12529572	994.71 M	338.32 Mb	SRX6673093	GSM4015625	GSM4015625	Cells aged 24 hours in SD media	MEP mus81
3	SRR9924098	SAMN12529570	838.88 M	294.83 Mb	SRX6673094	GSM4015626	GSM4015626	Cells aged 24 hours in SD media	MEP mus81
4	SRR9924099	SAMN12529569	631.87 M	250.37 Mb	SRX6673095	GSM4015627	GSM4015627	Cells aged 48 hours in YPD media	MEP [Pgal1-3HA cup1]/CUP1
5	SRR9924100	SAMN12529567	1.11 G	407.87 Mb	SRX6673096	GSM4015628	GSM4015628	Cells aged 48 hours in YPD media	MEP [Pgal1-3HA cup1]/CUP1
6	SRR9924101	SAMN12529565	903.88 M	343.05 Mb	SRX6673097	GSM4015629	GSM4015629	Cells aged 48 hours in YPGal media	MEP [Pgal1-3HA cup1]/CUP1
7	SRR9924102	SAMN12529564	1.45 G	529.28 Mb	SRX6673098	GSM4015630	GSM4015630	Cells aged 48 hours in YPGal media	MEP [Pgal1-3HA cup1]/CUP1
8	SRR9924103	SAMN12529561	646.51 M	227.76 Mb	SRX6673099	GSM4015631	GSM4015631	Cells aged 24 hours in SD media	MEP sae2
9	SRR9924104	SAMN12529560	1.05 G	357.64 Mb	SRX6673100	GSM4015632	GSM4015632	Cells aged 24 hours in SD media	MEP sae2
10	SRR9924105	SAMN12529558	829.11 M	281.75 Mb	SRX6673101	GSM4015633	GSM4015633	Cells aged 24 hours in SD media	MEP sae2
11	SRR9924106	SAMN12529557	823.58 M	284.92 Mb	SRX6673102	GSM4015634	GSM4015634	Cells aged 24 hours in SD media	MEP sae2
12	SRR9924107	SAMN12529555	1.03 G	354.90 Mb	SRX6673103	GSM4015635	GSM4015635	Cells aged 24 hours in SD media	MEP spt3
13	SRR9924108	SAMN12529553	994.63 M	344.93 Mb	SRX6673104	GSM4015636	GSM4015636	Cells aged 24 hours in SD media	MEP spt3
14	SRR9924109	SAMN12529608	551.82 M	242.67 Mb	SRX6673105	GSM4015637	GSM4015637	Cells aged 24 hours in SD media	MEP
15	SRR9924110	SAMN12529606	961.46 M	360.10 Mb	SRX6673106	GSM4015638	GSM4015638	Cells aged 24 hours in SD media	MEP
16	SRR9924111	SAMN12529604	1.33 G	454.02 Mb	SRX6673107	GSM4015639	GSM4015639	Cells aged 24 hours in SD media	MEP
17	SRR9924112	SAMN12529602	1.06 G	395.94 Mb	SRX6673108	GSM4015640	GSM4015640	Cells aged 24 hours in SD media	MEP
18	SRR9924113	SAMN12529601	563.99 M	258.47 Mb	SRX6673109	GSM4015641	GSM4015641	Cells aged 24 hours in SD media	MEP
19	SRR9924114	SAMN12529599	886.36 M	336.01 Mb	SRX6673110	GSM4015642	GSM4015642	Cells aged 24 hours in SD media	MEP
20	SRR9924115	SAMN12529598	1.30 G	446.44 Mb	SRX6673111	GSM4015643	GSM4015643	Cells aged 24 hours in SD media	MEP
21	SRR9924116	SAMN12529596	1.33 G	483.90 Mb	SRX6673112	GSM4015644	GSM4015644	Cells aged 24 hours in SD media	MEP
22	SRR9924117	SAMN12529594	1.13 G	389.68 Mb	SRX6673113	GSM4015645	GSM4015645	Cells aged 24 hours in SD media	MEP
23	SRR9924119	SAMN12529593	921.09 M	324.51 Mb	SRX6673114	GSM4015646	GSM4015646	Cells aged 24 hours in SD media	MEP

ENA
European Nucleotide Archive

Home | Submit | Search | Rulespace | About | Support

Project: PRJNA559191

Extrachromosomal circular DNA (eccDNA) facilitates adaptive evolution by allowing rapid and extensive gene copy number variation, and is implicated in the pathology of cancer and ageing. Here, we demonstrate that yeast aged under environmental copper accumulate high levels of eccDNA containing the copper resistance gene CUP1. Transcription of CUP1 causes CUP1 eccDNA accumulation, which occurs in the absence of phenotypic selection. We have developed a sensitive and quantitative eccDNA sequencing pipeline that reveals CUP1 eccDNA accumulation on copper exposure to be exquisitely site specific, with no other detectable changes across the eccDNA complement. eccDNA forms de novo from the CUP1 locus through processing of DNA double-strand breaks (DSBs) by Sae2 / Mre11 and Mus81, and genome-wide analyses show that other protein coding eccDNA species in aged yeast share a similar biogenesis pathway. Although abundant we find that CUP1 eccDNA does not replicate efficiently, and high copy numbers in aged cells arise through frequent formation events combined with asymmetric DNA segregation. The transcriptional stimulation of CUP1 eccDNA formation shows that age-linked genetic change varies with transcription pattern, resulting in gene copy number profiles tailored by environment. Overall design: Aged cell samples analysed in pairs of -/+ Cu, for both wt and various mutants. 3 replicates of the 3xCUP1 experiment are included

Show More

Organism:	Saccharomyces cerevisiae (baker's yeast)
Secondary Study Accession:	SRP217740
Study Title:	Transcription-induced formation of extrachromosomal DNA during yeast ageing
Center Name:	Bioinformatics, The Babraham Institute
Study Name:	Transcription-induced formation of extrachromosomal DNA during yeast ageing

Read Files

Show Column Selection

Download report: JSON TSV

Download Files as ZIP | Download selected files

Study Accession	Sample Accession	Experiment Accession	Run Accession	Tax Id	Scientific Name	Download All FASTQ FTP	Sub
PRJNA559191	SAMN12529574	SRX6673092	SRR9924096	4932	Saccharomyces cerevisiae	SRR992409...fastq.gz SRR992409...fastq.gz	

SRA Explorer

This tool aims to make datasets within the Sequence Read Archive more accessible.

Search for: SRP217740[All Fields]

Max Results 100 **Start At Record** 0

Need inspiration? Try GSE30567, SRP043510, PRJEB8073, ERP009109 or human liver miRNA.

Select relevant datasets and click *add to collection*. When you're finished, view all saved datasets with the button in the top right of the page, where you can copy the SRA URLs.

Showing 30 results.

Filter results: Enter search term | All Fields

Add 0 to collection

Title	Accession	Instrument	Total Bases (Mb)	Date Created
GSM4015617: 3xCUP1_24hr_1_REC-seq; Saccharomyces cerevisiae; OTHER	SRR9924120	Illumina HiSeq 2500	8685	21 Oct 2019
GSM4015618: 3xCUP1_24hr_2_REC-seq; Saccharomyces cerevisiae; OTHER	SRR9924121	Illumina HiSeq 2500	10212	21 Oct 2019
GSM4015619: 3xCUP1_24hr_300uM_Cu_1_REC-seq; Saccharomyces cerevisiae; OTHER	SRR9924122	Illumina HiSeq 2500	9693	21 Oct 2019
GSM4015620: 3xCUP1_24hr_300uM_Cu_2_REC-seq; Saccharomyces cerevisiae; OTHER	SRR9924123	Illumina HiSeq 2500	9602	21 Oct 2019

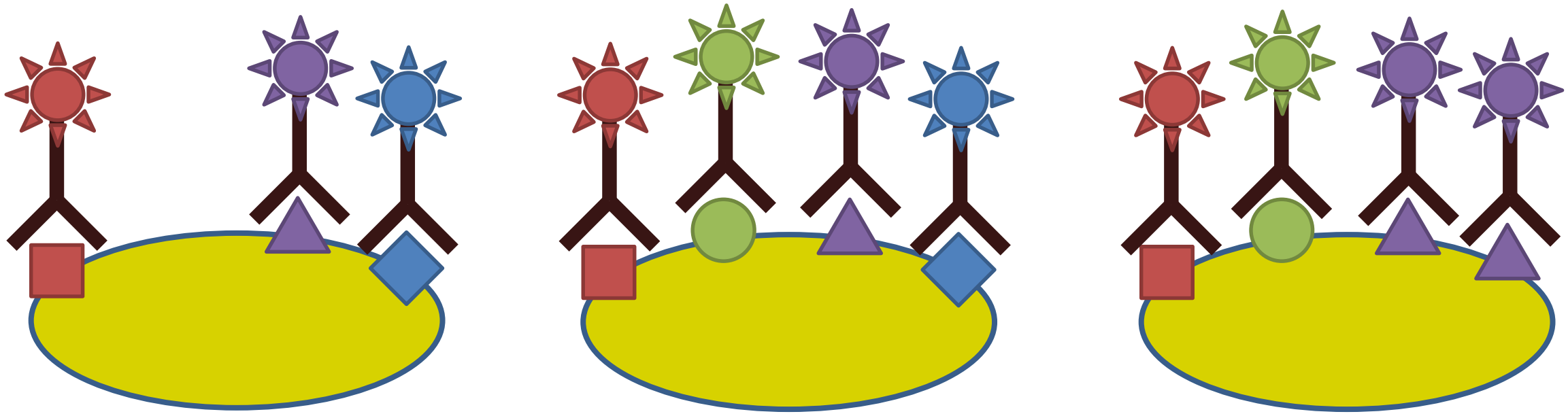
SRA Downloader

```
sradownloader SRR9924120
```

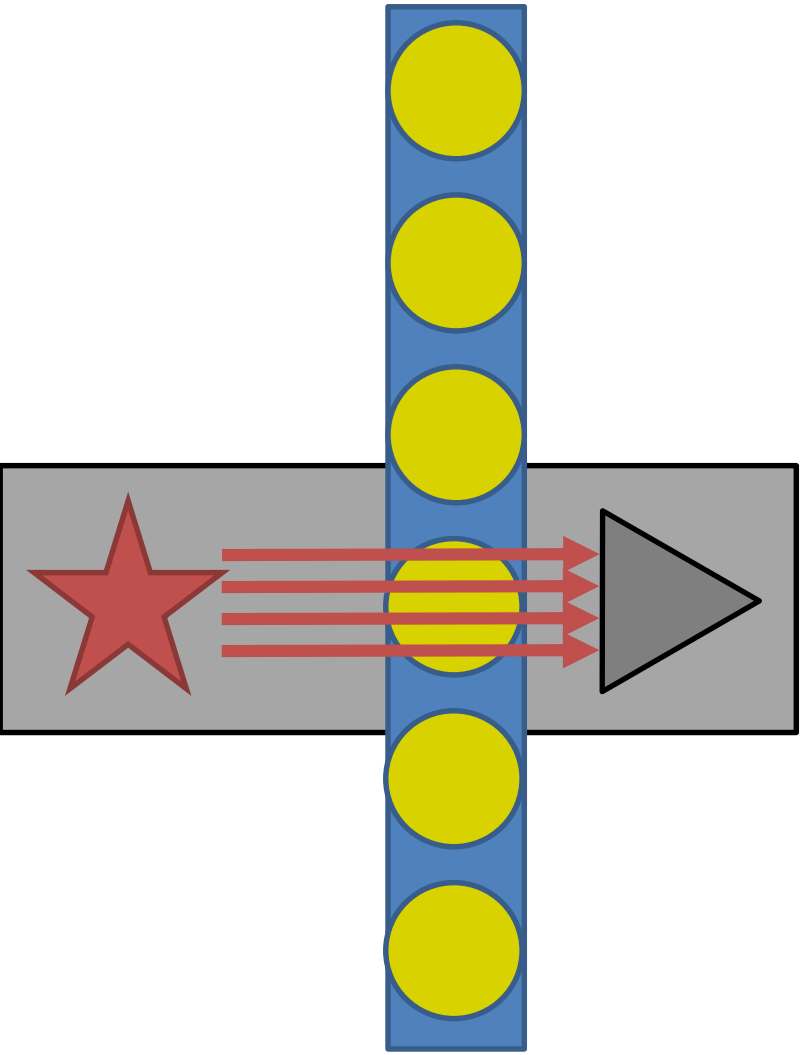
Sequencing Data Exercise

Flow Cytometry

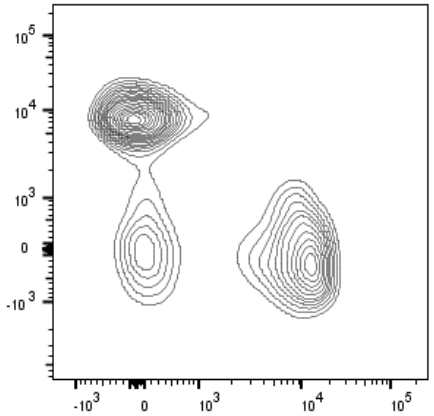
Flow Cytometry



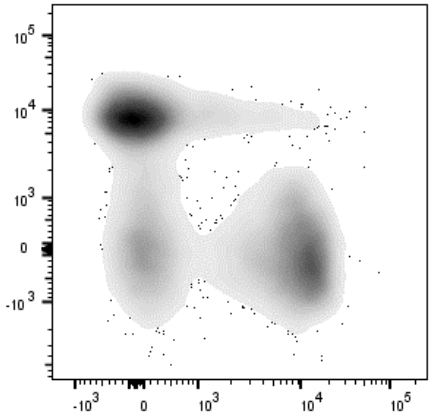
Small Scale Measurement



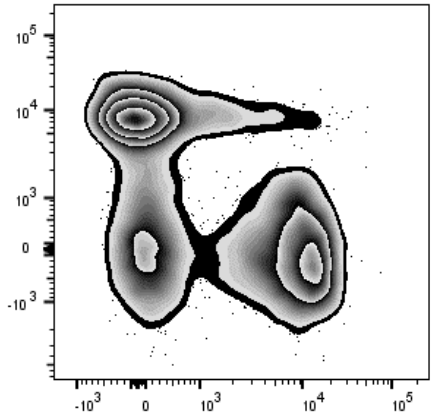
Contour plot



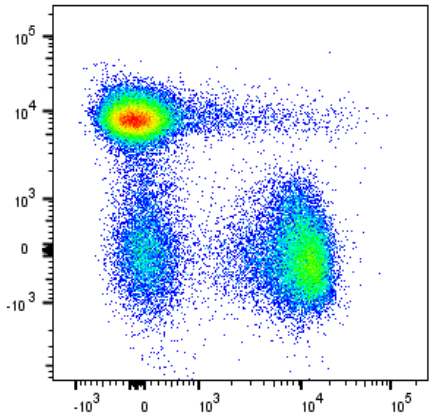
Density plot



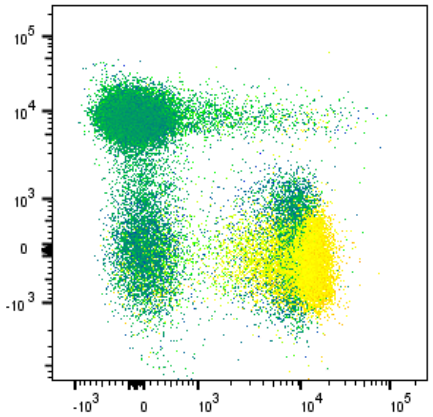
Zebra plot



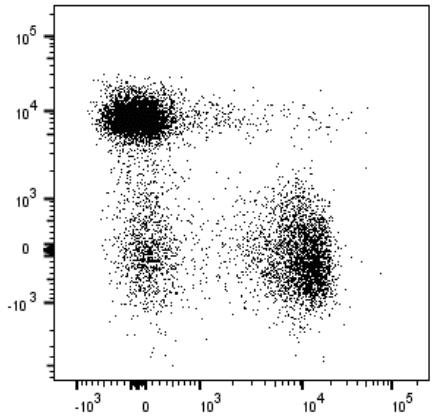
Pseudocolor



Heatmap Statistic

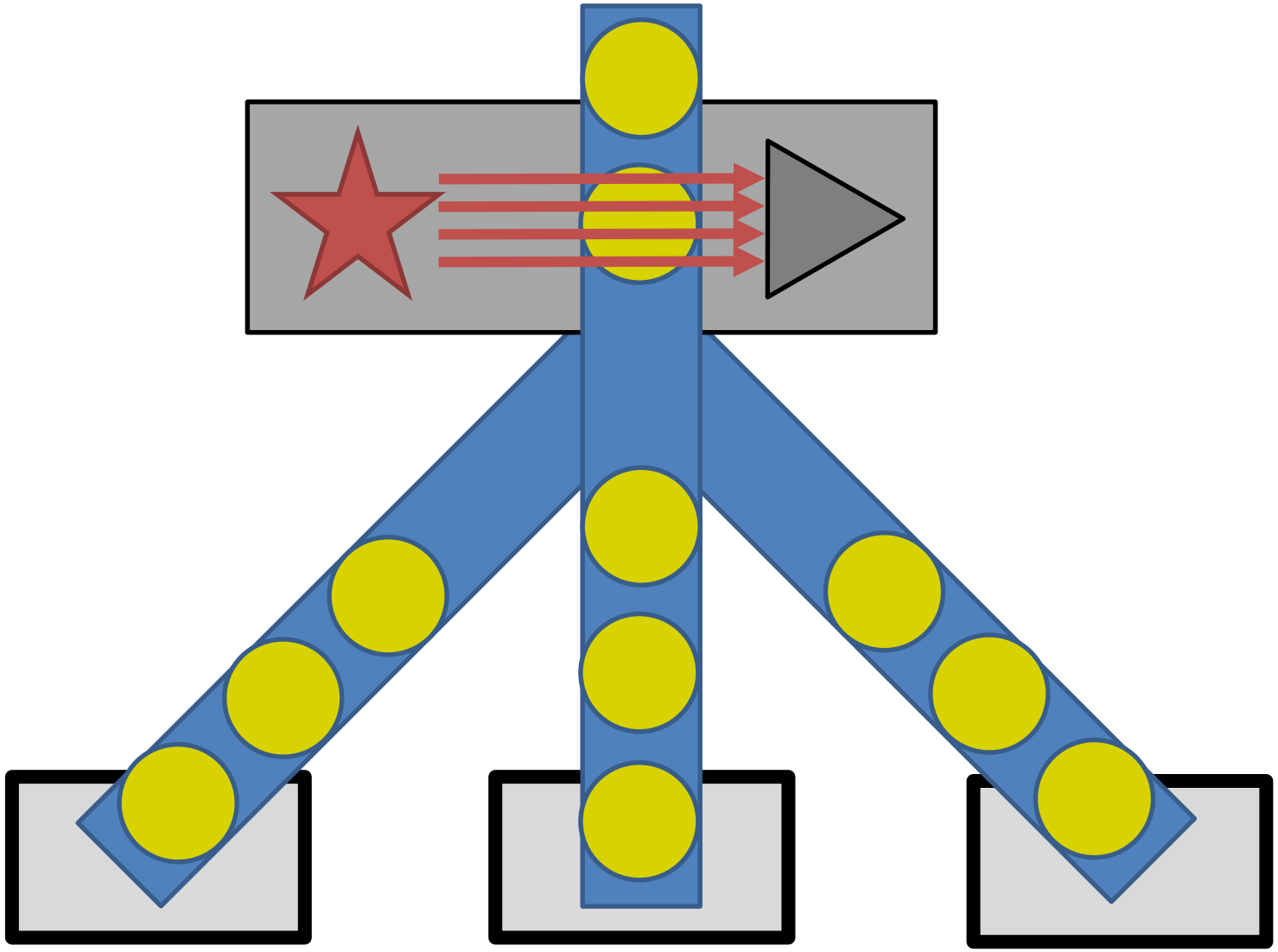


Dot Plot



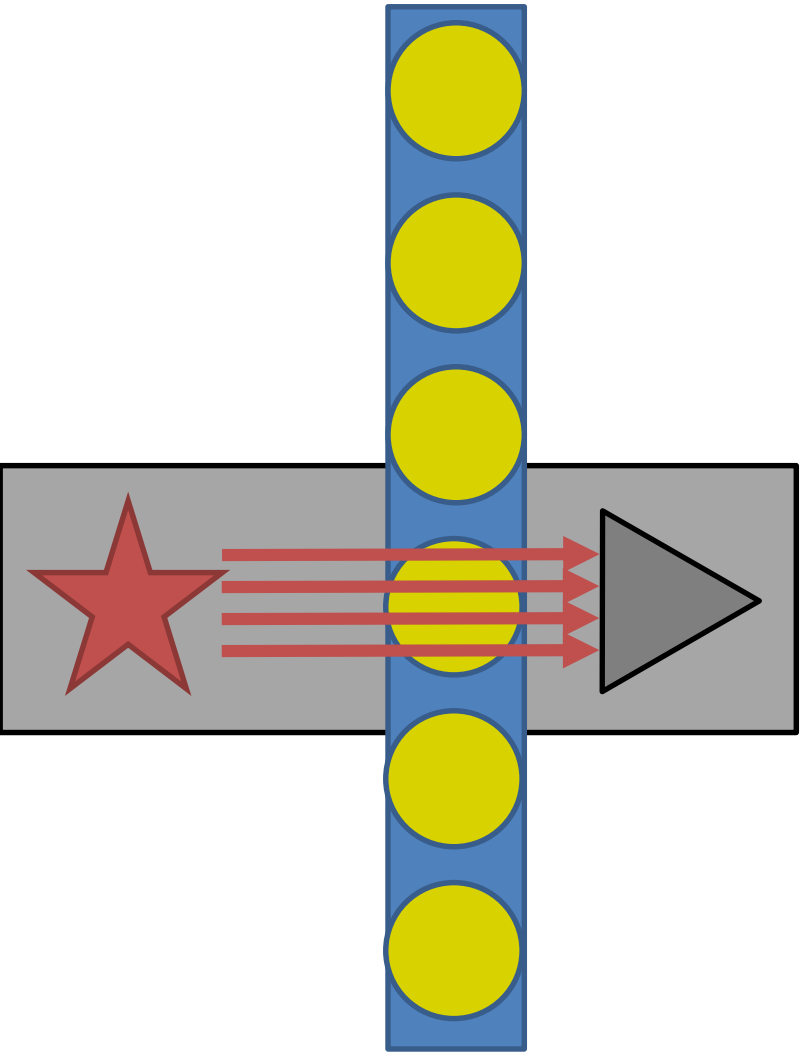
https://flowjo.com

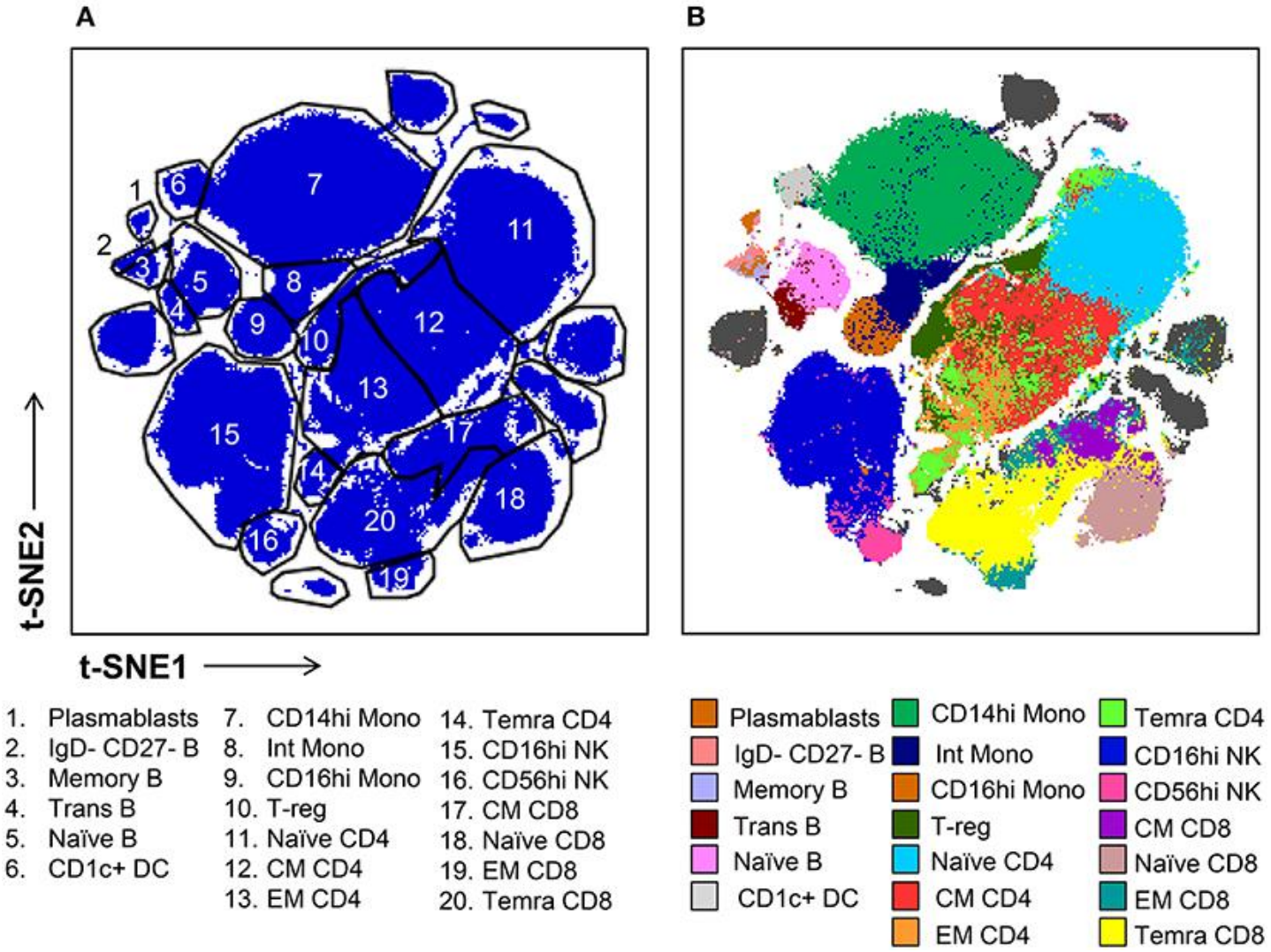
Using Flow for Sorting Cells



- Cell subpopulations
- CRISPR screens
- Cow sexing!

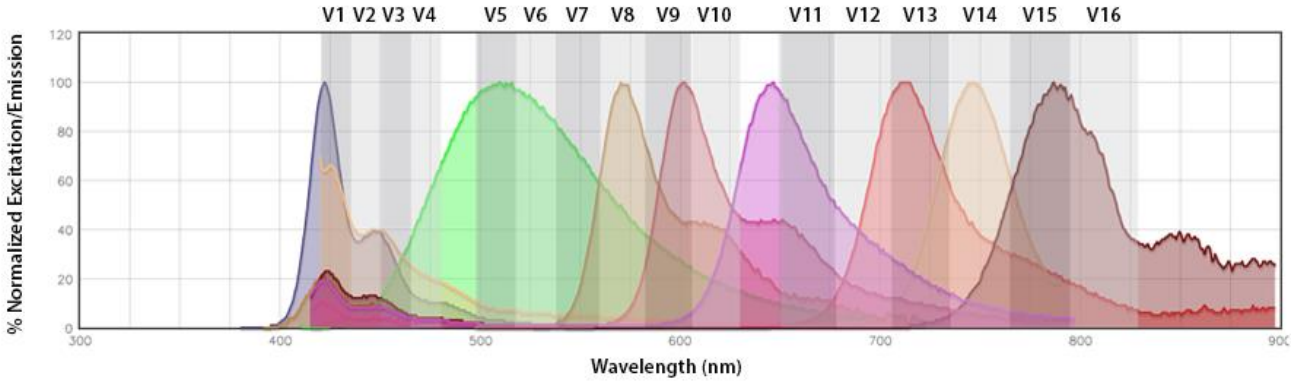
Large Scale Measurement





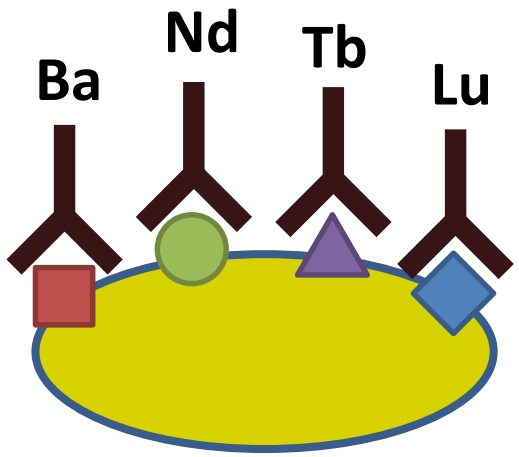
https://doi.org/10.3389/fimmu.2019.01194

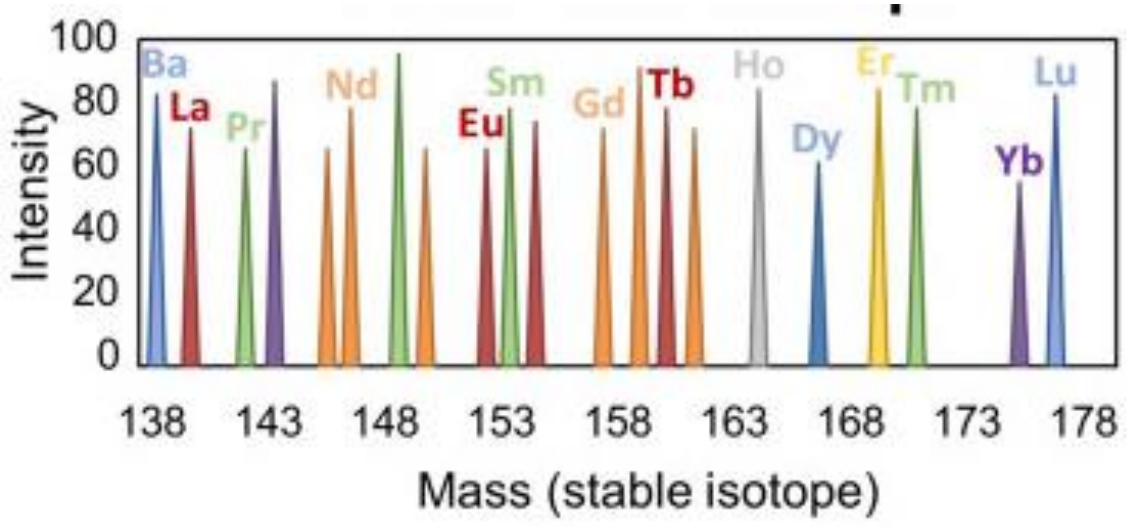
Problems with multiple fluorescent markers



Traditionally filters measure one wavelength per fluor

Spectral Flow Cytometry measures the whole spectrum and can deconvolve overlapping emissions spectra
Allows for 40+ markers to be used simultaneously.





Public Flow Data Repository

FLOWRepository

- Deposition of FCS files
  - Instrument details
  - Raw data
  - Analysis details

- Basic description of experiment structure

Experiment: Human COVID-19 Immune Phenotyping **ID:** FR-FCM-Z2KP **Primary Researcher:** Stephanie Humblet-Baron MIFlowCyt Score: 97.60%

« Back to All Public Experiments

« Back to Start Page

DOWNLOAD NOW!

Help

The following open access article describes how to upload and annotate flow cytometry data sets: Spidlen J, Breuer K and Brinkman R. Preparing a Minimum Information about a Flow Cytometry Experiment (MIFlowCyt) Compliant Manuscript Using the International Society for Advancement of Cytometry (ISAC) FCS File Repository (FlowRepository.org). Current Protocols in Cytometry, UNIT 10.18, July 2012.

We also have a Quick start guide and a FAQ section.

You may download slides from our Workshop at CYTO 2012: Publishing MIFlowCyt Compliant Data to ISAC's FlowRepository.org for Cytometry A and Other Journals

Experiment Overview

Repository ID:	FR-FCM-Z2KP	**Experiment name:**	Human COVID-19 Immune Phenotyping	**MIFlowCyt score:**	97.60%
Primary researcher:	Stephanie Humblet-Baron	**PI/manager:**	Adrian Liston	**Uploaded by:**	Oliver Burton
Experiment dates:	2020-04-21 - 2020-04-24	**Dataset uploaded:**	May 2020	**Last updated:**	May 2020
Keywords:	[Intracellular Cytokine Staining] [human PBMCs] [COVID-19] [SARS-CoV2] [Coronavirus]	**Manuscripts:**			
Organizations:	VIB/KU Leuven, Leuven, Leuven (Belgium) Babraham Institute, Babraham Institute, Cambridge, Cambridge (United Kingdom)				
Purpose:	Analysis of cytokine production by PBMC from COVID-19 patients				
Conclusion:	Cytokines are produced by PBMC from SARS-CoV2-infected patients				
Comments:	None				
Funding:	VIB, KU Leuven				
Quality control:	Unstimulated controls, healthy controls, Automated compensation				

FLOWRepository

Experiment variables

Conditions

· Healthy

export_COVID19 samples 23_04_20_ST3_COVID19_HC_001 ST3 230420_017_Live_cells.fcs · export_COVID19 samples 23_04_20_ST3_COVID19_HC_005 ST3 230420_016_Live_cells.fcs · export_COVID19 samples 23_04_20_ST3_COVID19_HC_006 ST3 230420_015_Live_cells.fcs · export_COVID19 samples 23_04_20_ST3_COVID19_HC_007 ST3 230420_014_Live_cells.fcs · export_COVID19 samples 23_04_20_ST3_COVID19_HC_008 ST3 230420_013_Live_cells.fcs · export_COVID19 samples 23_04_20_ST3_COVID19_HC_009 ST3 230420_012_Live_cells.fcs

export_COVID19 samples 21_04_20_ST3_COVID19_ICU_changedW_019_O ST3 210420_040_Live_cells.fcs · export_COVID19 samples 21_04_20_ST3_COVID19_ICU_changedW_027_O ST3 210420_039_Live_cells.fcs · export_COVID19 samples 21_04_20_ST3_COVID19_ICU_changedW_036_O ST3 210420_035_Live_cells.fcs · export_COVID19 samples 21_04_20_ST3_COVID19_W_033_O ST3 210420_036_Live_cells.fcs · export_COVID19

Flow Exercise

Babraham Bioinformatics

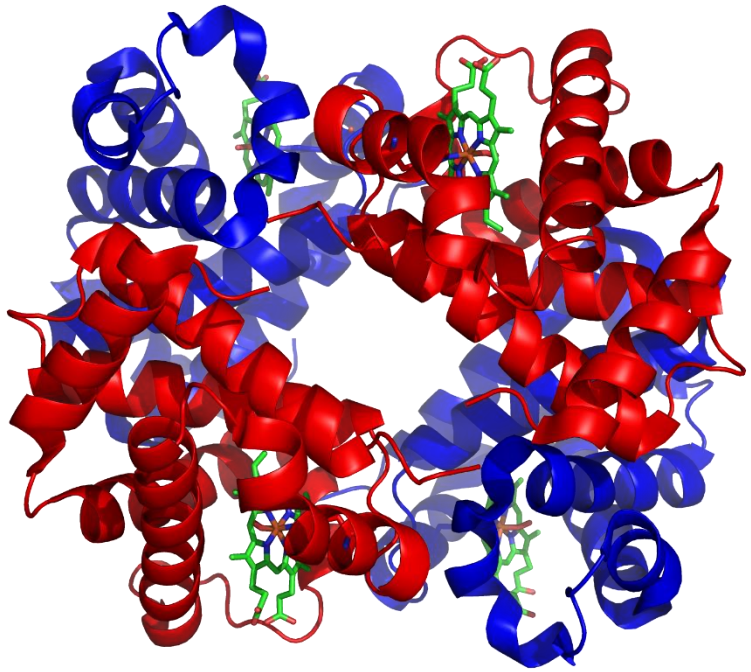
Mass Spec

- General purpose method to measure the accurate masses of small molecules

- Can be used to identify
  - Proteins (plus modifications)
  - Metabolites
    - Sugars
    - Nucleotides
    - Amino Acids
    - Lipids

Protein Mass Spec

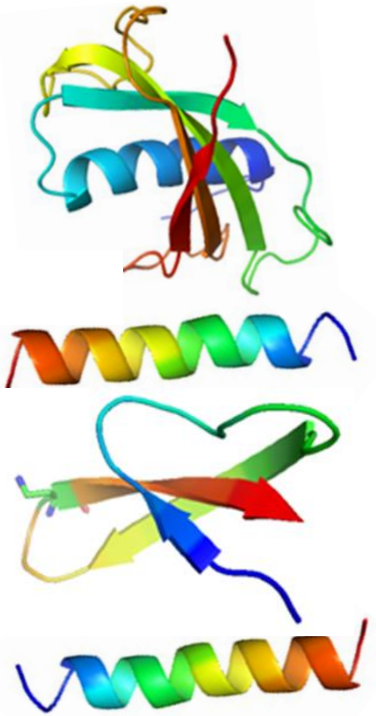
Proteins



Too Big

Digest

Peptides



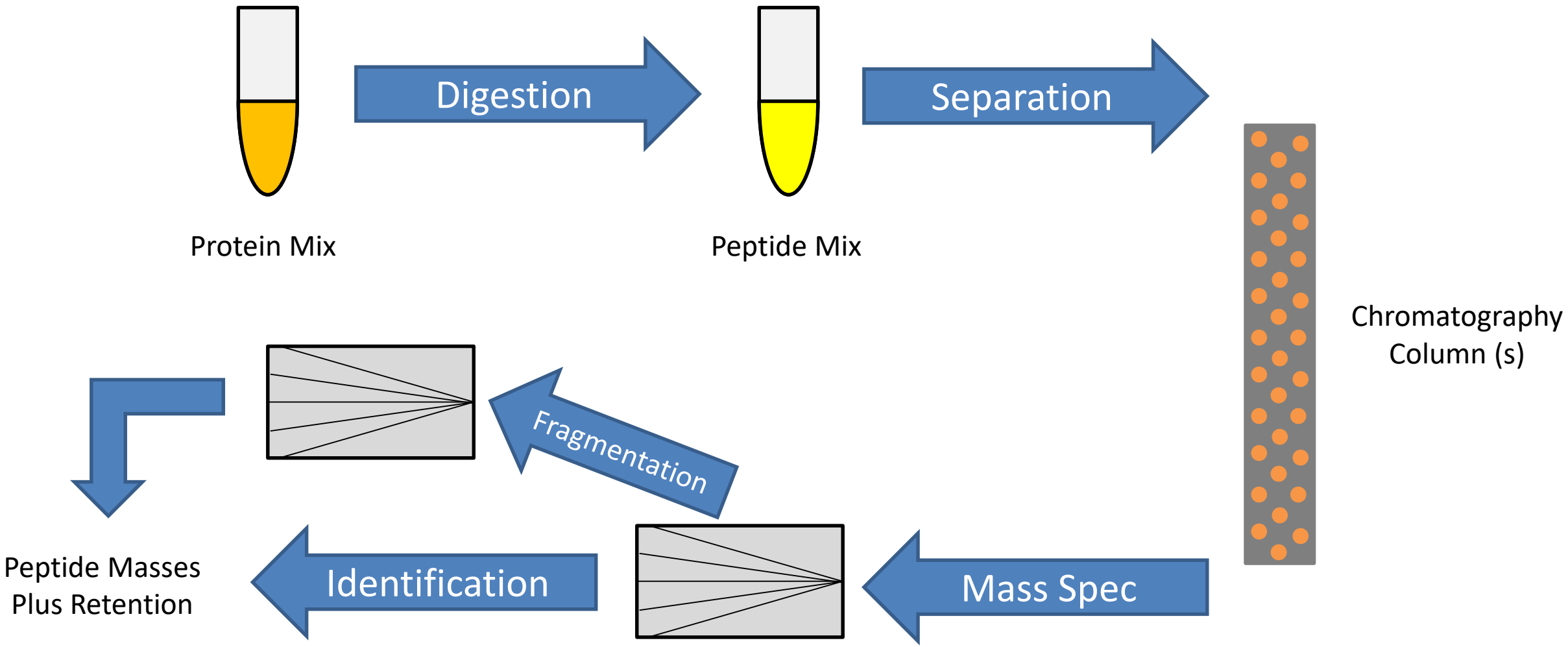
Too Many

Separate

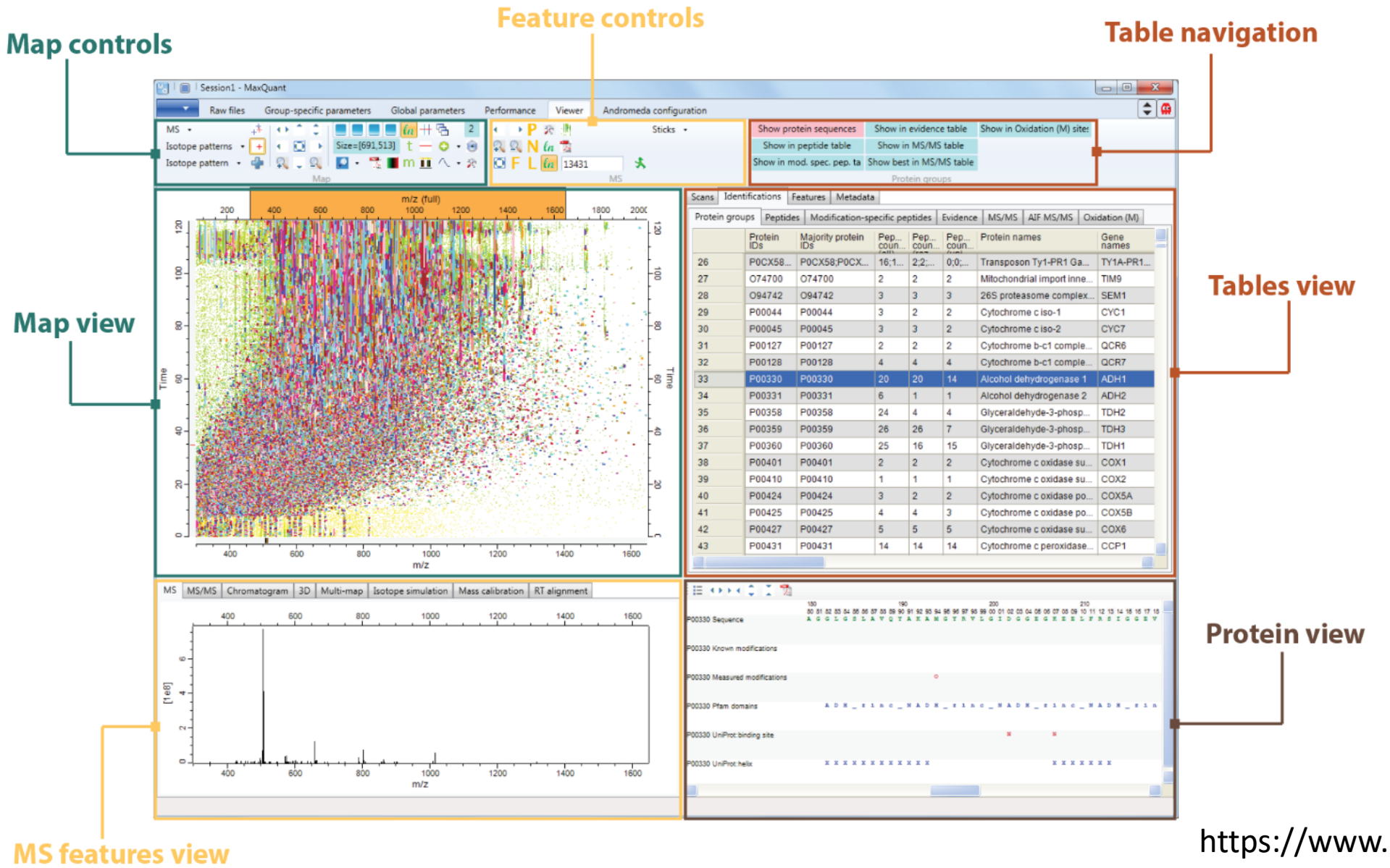
A peptide

Non-specific

Protein Mass Spec Workflow

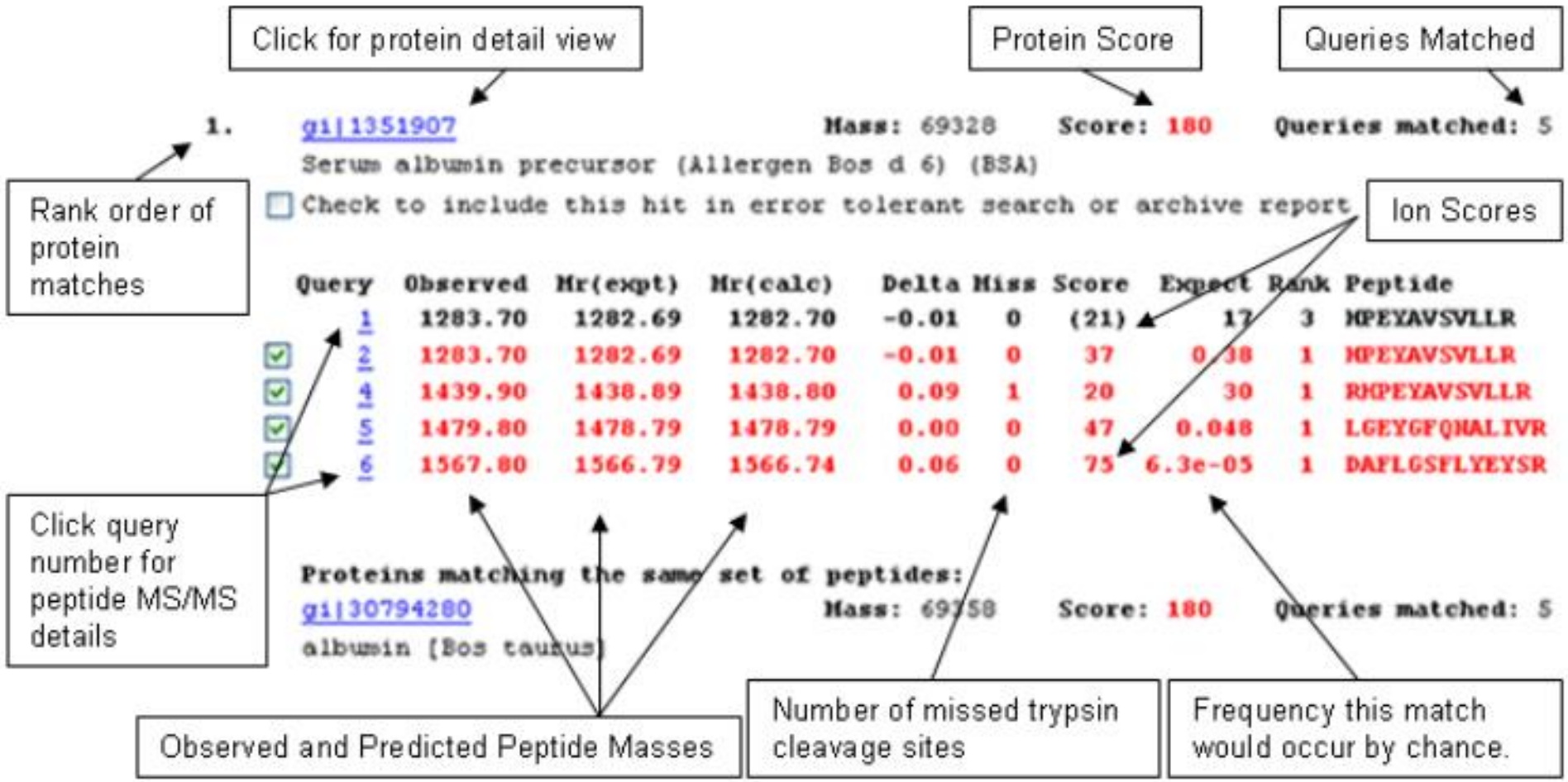


Protein Mass Spec Results



https://www.maxquant.org/

Protein Identification

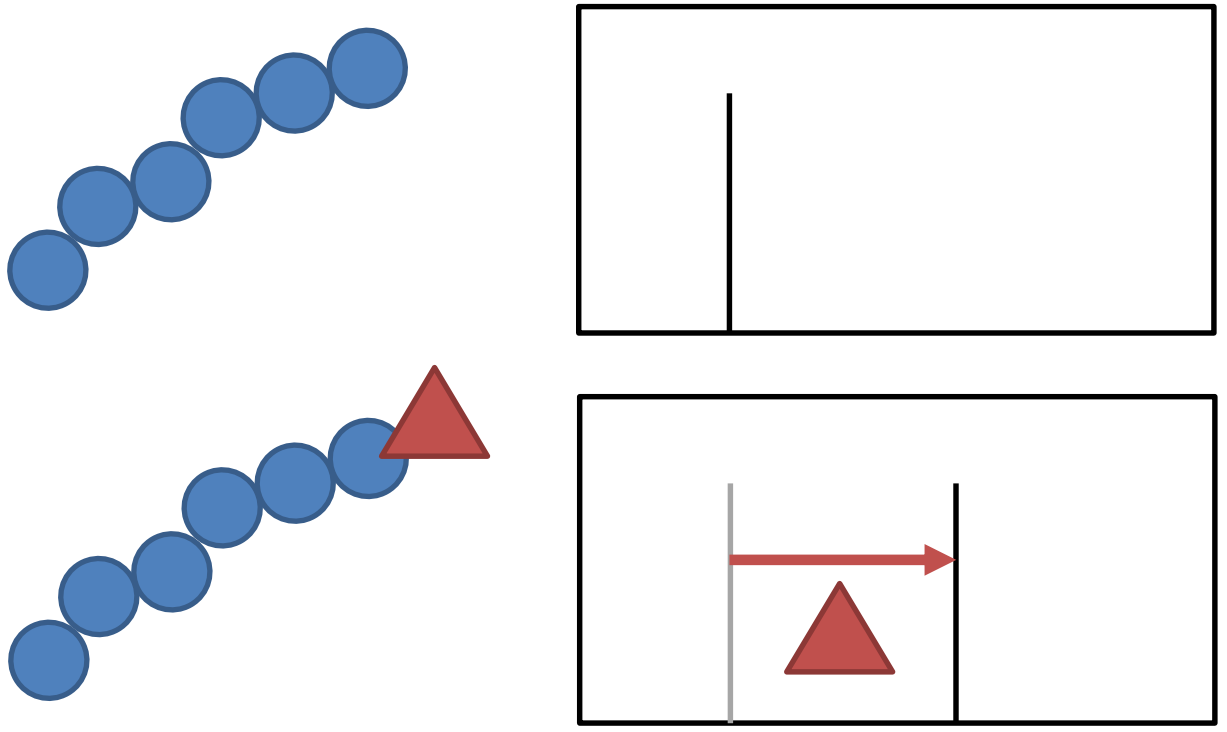


http://www.ohri.ca/proteomics/

Post Translational Modifications

- When doing tandem mass spectrometry you can also identify modified peptides

- Phosphorylation
- Acetylation
- Methylation
- Palmitylation
- Acylation
- Ubiquitination
- etc.



High throughput proteomics

J Proteome Res. 2019 May 3;18(5):2346-2353. doi: 10.1021/acs.jproteome.9b00082. Epub 2019 Apr 12.

Evosep One Enables Robust Deep Proteome Coverage Using Tandem Mass Tags while Significantly Reducing Instrument Time

Jonathan R Krieger, Leanne E Wybenga-Groot, Jiefei Tong, Nicolai Bache [1], Ming S Tsao [2 3 4], Michael F Moran [5]

30 samples per day Evosep workflow,
>12 000 proteins were identified in 48 h
of mass spectrometry time

J Proteome Res. 2021 May 7;20(5):2964-2972. doi: 10.1021/acs.jproteome.1c00168. Epub 2021 Apr 26.

TMTpro-18plex: The Expanded and Complete Set of TMTpro Reagents for Sample Multiplexing

Jiaming Li [1], Zhenying Cai [2 3], Ryan D Bomgarden [4], Ian Pike [5], Karsten Kuhn [5], John C Rogers [4], Thomas M Roberts [2 3], Steven P Gygi [1], Joao A Paulo [1]

Nat Protoc. 2018 Jul;13(7):1632-1661. doi: 10.1038/s41596-018-0006-9.

Reproducible workflow for multiplexed deep-scale proteome and phosphoproteome analysis of tumor tissues by liquid chromatography-mass spectrometry

Philipp Mertins [1 2 3], Lauren C Tang [1], Karsten Krug [1], David J Clark [4], Marina A Gritsenko [5], Lijun Chen [4], Karl R Clauser [1], Therese R Clauss [5], Punit Shah [4], Michael A Gillette [1], Vladislav A Petyuk [5], Stefani N Thomas [4], D R Mani [1], Filip Mundt [1], Ronald J Moore [5], Yingwei Hu [4], Rui Zhao [5], Michael Schnaubelt [4], Hasmik Keshishian [1], Matthew E Monroe [5], Zhen Zhang [4], Namrata D Udeshi [1], Deepak Mani [1], Sherri R Davies [6], R Reid Townsend [6], Daniel W Chan [4], Richard D Smith [5], Hui Zhang [4], Tao Liu [5], Steven A Carr [7]

10,000 proteins per sample
37,000 phosphosites per sample

Expanding Mass Spec Technology

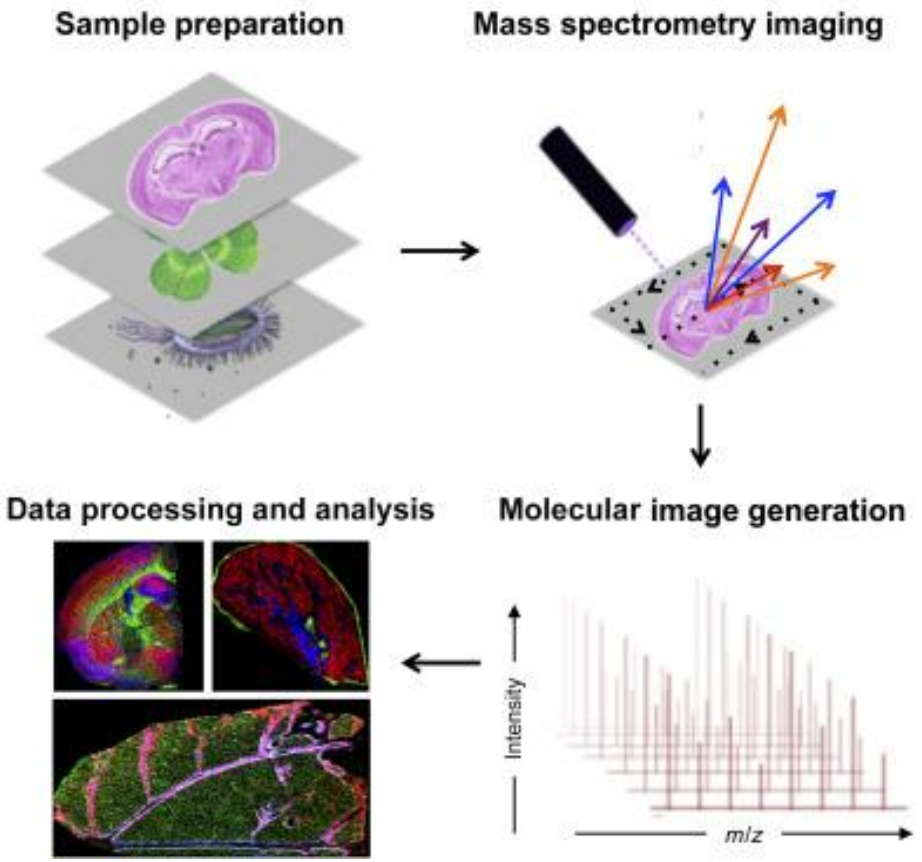
Ultra-high sensitivity mass spectrometry quantifies single-cell proteome changes upon perturbation

Andreas-David Brunner, Marvin Thielert, Catherine G. Vasilopoulou, Constantin Ammar, Fabian Coscia, Andreas Mund, Ole B. Hoerning, Nicolai Bache, Amalia Apalategui, Markus Lubeck, Sabrina Richter, David S. Fischer, Oliver Raether, Melvin A. Park, Florian Meier, Fabian J. Theis, Matthias Mann

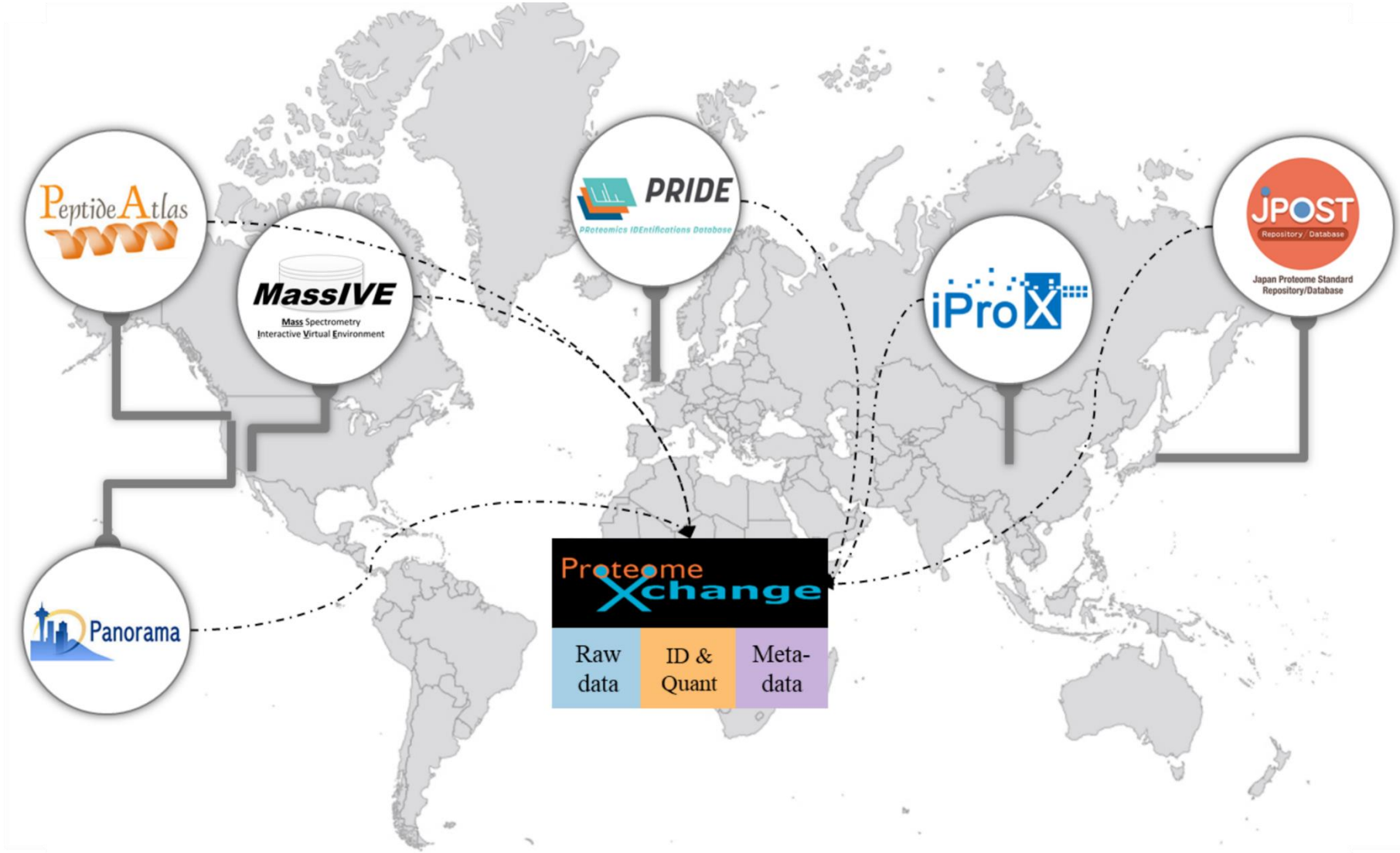
1,400 proteins measured from a single cell

Mass Spectral Imaging
Fix a sample to a surface and then do scanning Ionisation over it to get a spectrum for each point.

You can then pick any fragment and image its distribution over the original sample



Data Repositories for Proteomics Mass Spec



Data Repositories for Proteomics Mass Spec

- Varying amounts of experimental annotation
- Good description of processing and preparation
- Raw data files available
  - Mass spec still uses a lot of proprietary vendor file formats
  - Open mzML format is defined but often not used
  - Converters exist but often lose information.

Plasma X

Dataset Identifier	Title	Repos	Species	Instrument	Publication	LabHead	Announce Date	Keywords
PXD026962	Multi-omic Profiling of Plasma Identify Biomarkers and Pathogenesis of COVID-19 in Children	iProX	Homo sapiens	QTRAP 6500+	Dataset with its publication pending	Xi Zhou	2021-06-28	Multi-omic Profiling, COVID-19, Children,
PXD026928	Mycoplasma gallisepticum WhiA knockdown and overexpression	PRIDE	Mycoplasma gallisepticum S6	Q Exactive Plus	Dataset with its publication pending	Gleb Fisunov	2021-06-25	mycoplasma, transcription factor, WhiA, overexpression, knockdown,
PXD022361	Recombinant SWATH library for identification of low abundant human plasma proteins	MassIVE	Homo sapiens	TripleTOF 6600	Ahn et al. (2021)	Prof Mark S. Baker	2021-06-24	SWATH, Recombinant Protein, Plasma Proteome,
PXD021581	Prognostic accuracy of Mass Spectrometric Analysis of Plasma in COVID-19	PRIDE	Homo sapiens	LTQ Orbitrap Velos	Dataset with its publication pending	Giuseppe Palmisano	2021-06-21	Sars-cov-2, Covid-19, COVID-19, SARS-CoV-2, Mass spectrometry, Biomarker, Plasma, Prognosis,

PXD026962

PXD026962 is an **original dataset** announced via ProteomeXchange.

Dataset Summary

Title	Multi-omic Profiling of Plasma Identify Biomarkers and Pathogenesis of COVID-19 in Children
Description	Although people of all ages are susceptible to COVID-19, children usually develop less severe disease than adults. Little is known about the pathogenesis of COVID-19 in children. Herein, we conduct the plasma proteomic and metabolomic profiling of a cohort of COVID-19 children patients with mild symptoms, and uncovered that many proteins involved in immune response are significantly up-regulated in a stronger extent than in adults with COVID-19. Interestingly, more molecules involved in protective processes of reducing inflammation are also stimulated to antagonize the deleterious effect in both proteomic and metabolomic levels. By developing a machine learning-based pipeline, we prioritize two set of biomarker combinations, and identify 5 proteins and 5 metabolites as potentially children-specific biomarkers. Further experiments demonstrate these protective metabolites not only inhibit the expression of pro-inflammatory factors, but also suppress the viral replication. Taken together, our study not only discover the protective mechanisms in children with COVID-19, but also shed light on potential therapies targets for treating COVID-19.
HostingRepository	iProX
AnnounceDate	2021-06-28
AnnouncementXML	Submission_2021-06-28_01:26:39.414.xml
DigitalObjectIdentifier	
ReviewLevel	Peer-reviewed dataset
DatasetOrigin	Original dataset
RepositorySupport	Unsupported dataset by repository
PrimarySubmitter	Yang Qiu
SpeciesList	scientific name: Homo sapiens; NCBI TaxID: 9606;

Project Information

Project ID

IPX0002673000

ProteomeXchange ID

PXD026962

Project Title

Multi-omic Profiling of Plasma Identify Biomarkers and Pathogenesis of COVID-19 in Children

XML File

PX_IPX0002673000.xml Download

Download All Files (7.55G)

Aspera Download *recommended

Http Download

Metabolite Mass Spectrometry

- Similar concepts to protein mass spec
- Range of starting material
  - Serum
  - Urine
  - Cerebrospinal fluid
  - Saliva
- Different separations
- Up to 5000 different metabolites to find

Data Repository for Metabolomics Data

MoNA - MassBank of North America

- Reference spectra for biological molecules
  - Used for searching and quantitation

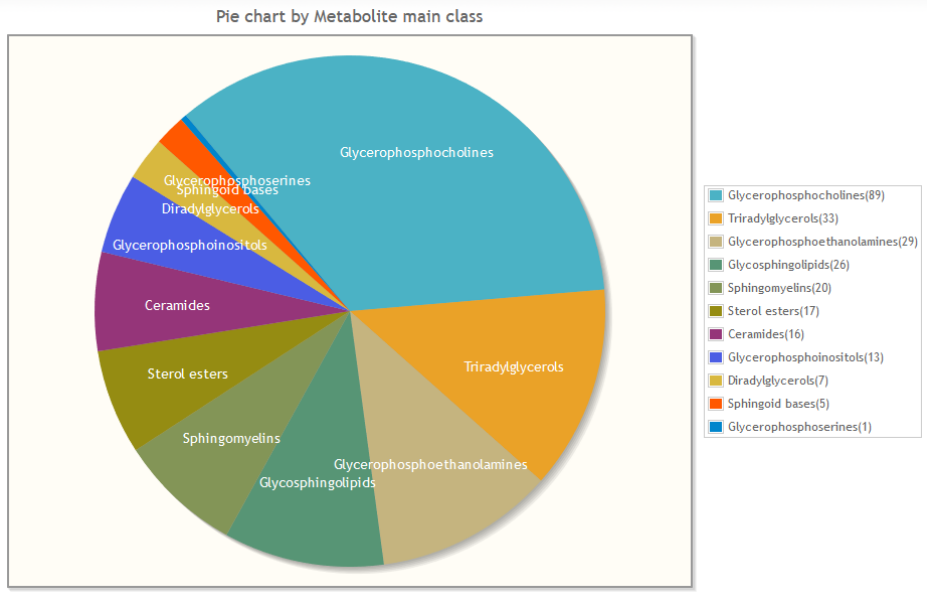
- Experimental datasets of MassSpec Studies
  - Used to answer biological questions
  - Also provides visualisations and tools

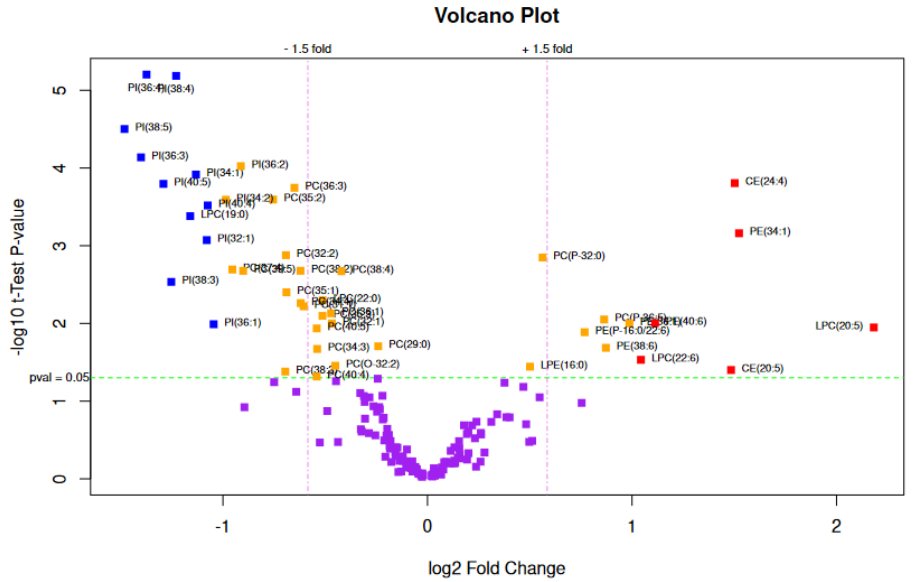
Metabolomics Workbench

Data for ST001140 Perform analyte scaling Perform sample normalization

(Analysis AN001870): Average values per metabolite and experimental factor (Units:uM)

Metabolite structure	All data	F1	F2	F3	F4
CE(16:0)	ME271966	2.67	2.94	2.03	1.56
CE(16:1)	ME271967	0.47	0.52	0.44	0.37
CE(17:0)	ME271968	0.06	0.06	0.05	0.03
CE(17:1)	ME271969	0.05	0.06	0.06	0.05
CE(18:0)	ME271970	0.52	0.66	0.46	0.38
CE(18:1)	ME271971	22.69	22.21	22.53	20.25
CE(18:2)	ME271972	85.96	85.68	95.16	74.09





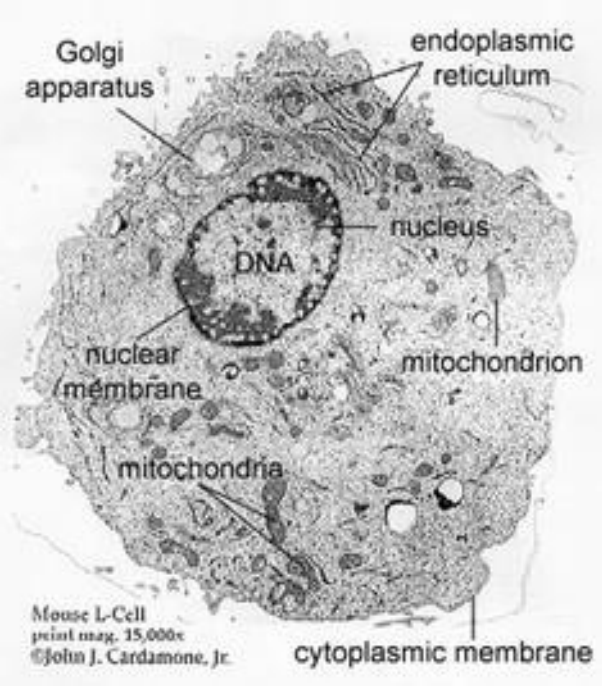
Mass Spec Data Exercise

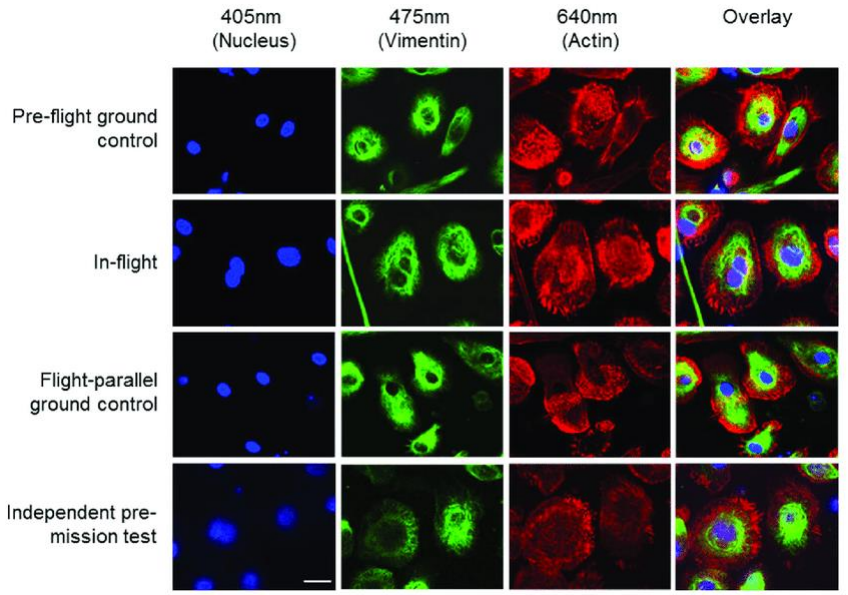
Babraham Bioinformatics

Imaging Analysis

- What can you measure with imaging?

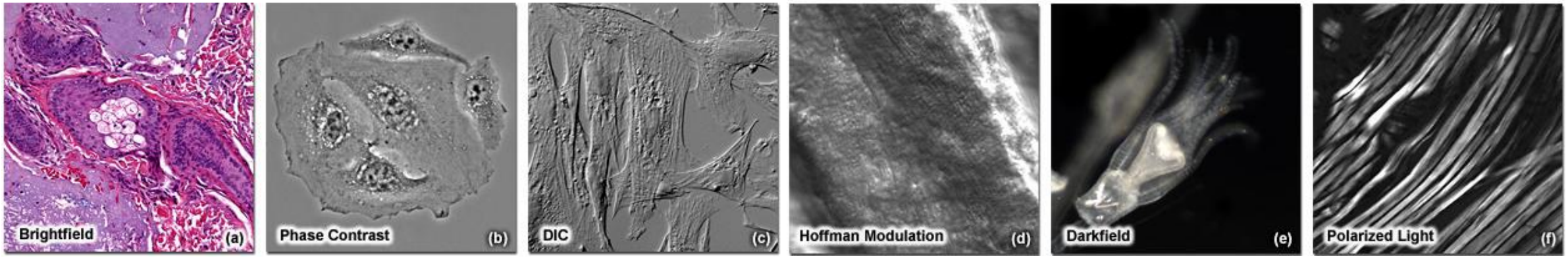
- Cell structure and morphology
  - In both live and fixed cells
- Targeted molecules (fluorescence microscopy)
  - Antibodies to proteins
  - Fluorescent fusion proteins
- Functional readouts
  - Redox state
  - pH





Types of Microscopy

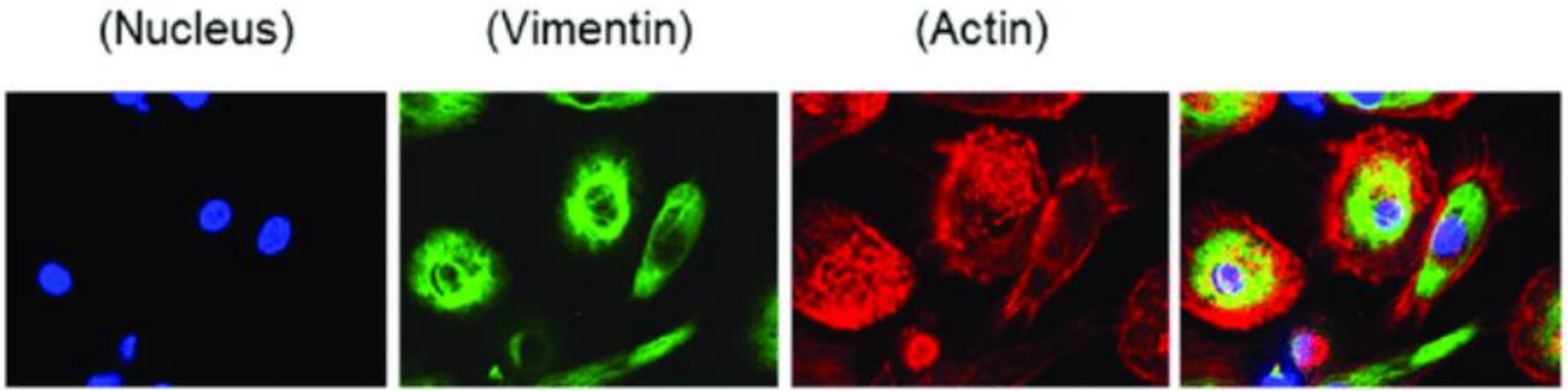
- Light Microscopy
  - Sample is illuminated, some light goes to the viewer
  - Biological samples are generally clear, so hard to see
  - Can use stains (often toxic) or reflection or phase shift to see better



https://zeiss-campus.magnet.fsu.edu/articles/basics/contrast.html

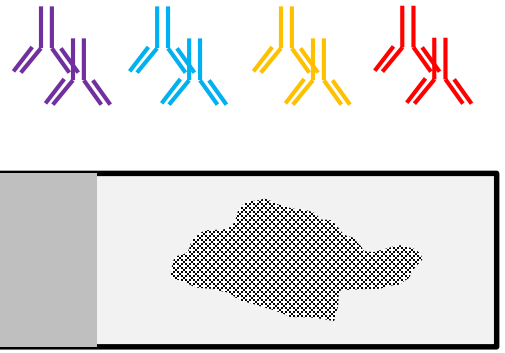
Types of Microscopy

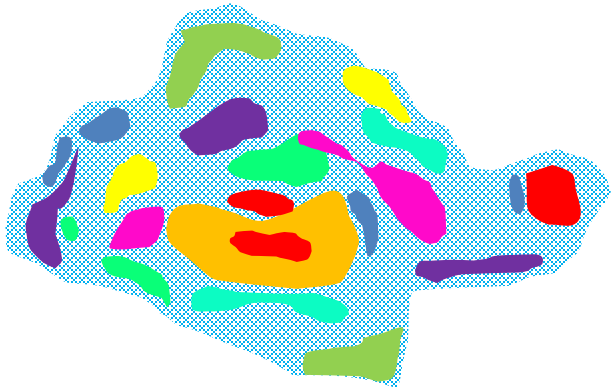
- Fluorescence Microscopy
  - Uses molecules which excite at one wavelength and emit at another
  - Allow the tagging of specific biological molecules
  - Confocal microscopes allow clear views of a single plane in the sample

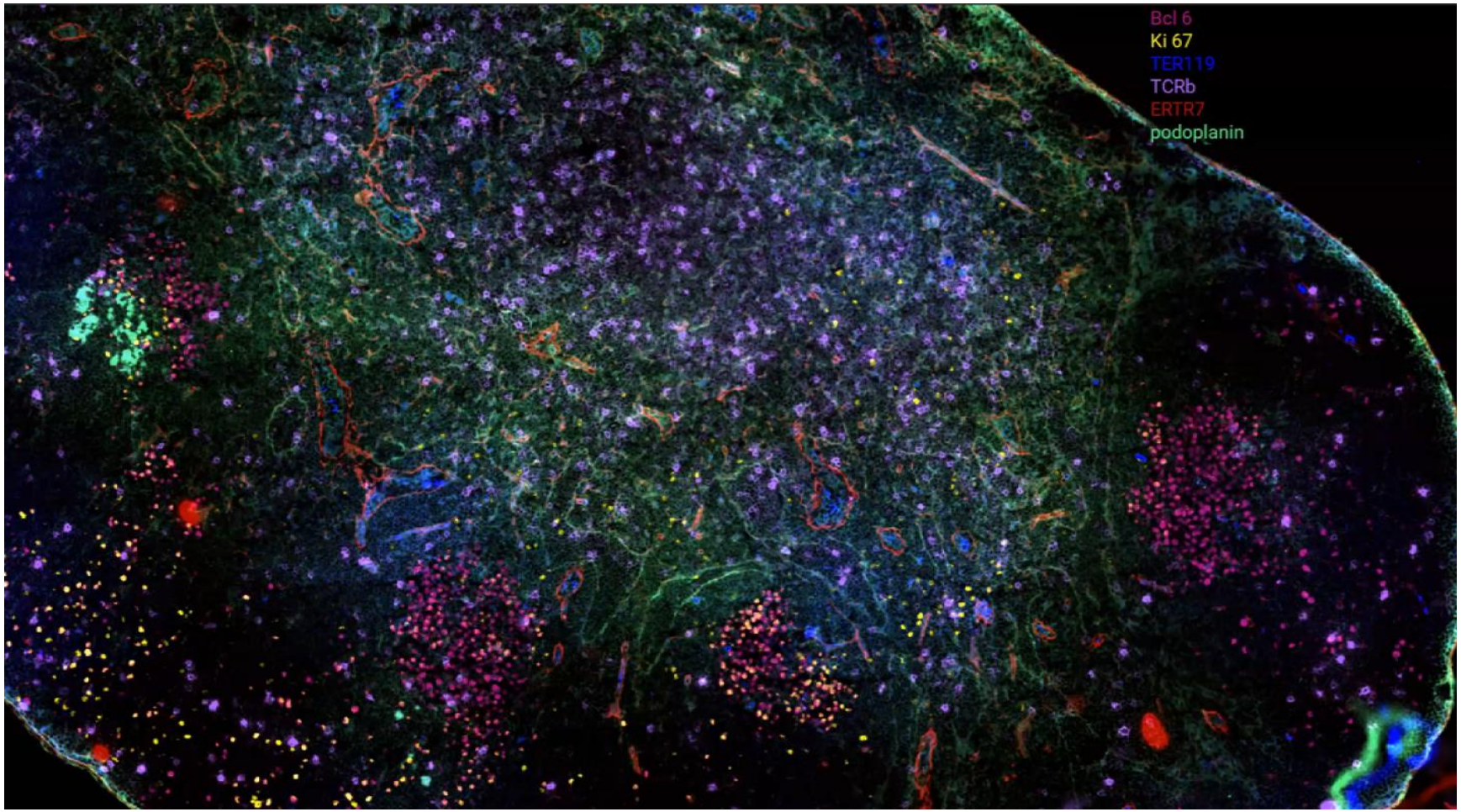


DOI:10.3390/ijms20082033

Ultra-plex fluorescence imaging



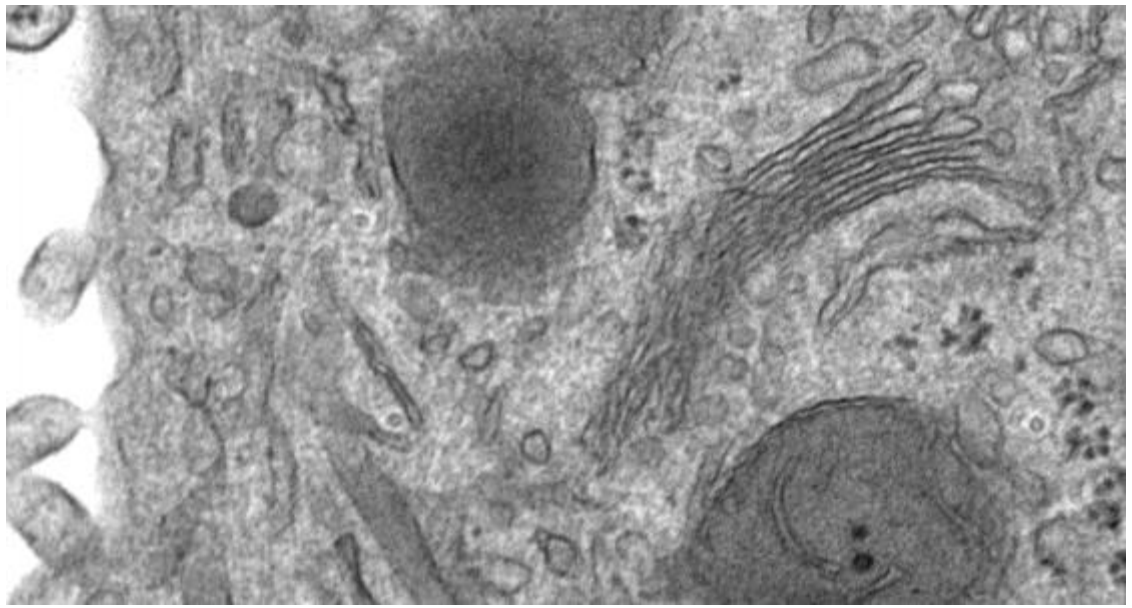


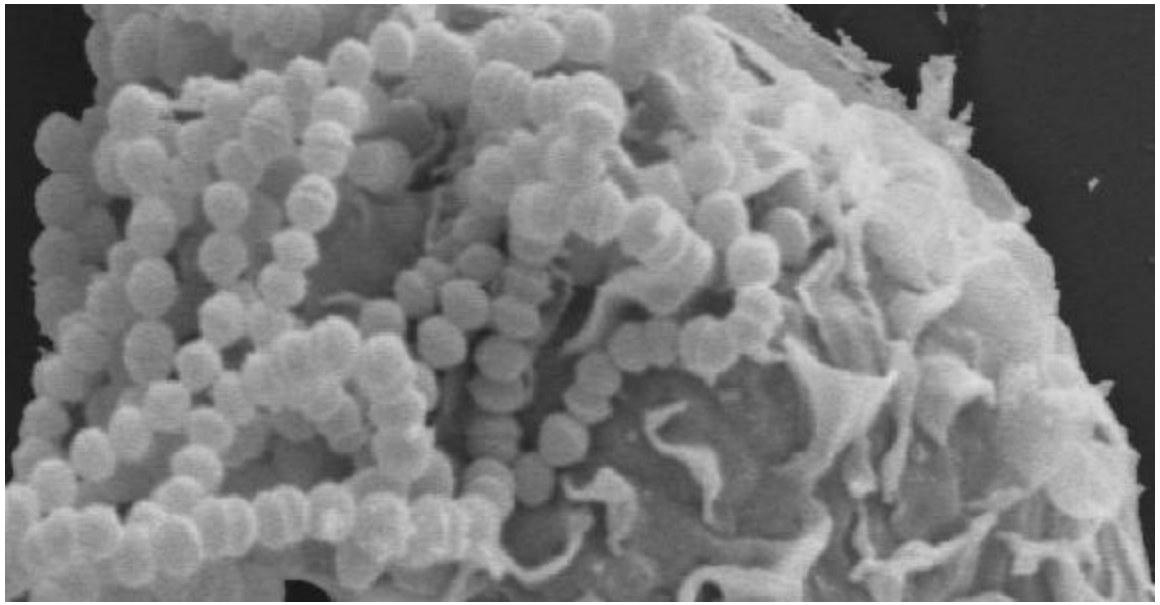


Simon Walker – BI Imaging Facility

Types of Microscopy

- Electron Microscopy
  - Fixed and processed samples only (not live)
  - Very high resolution



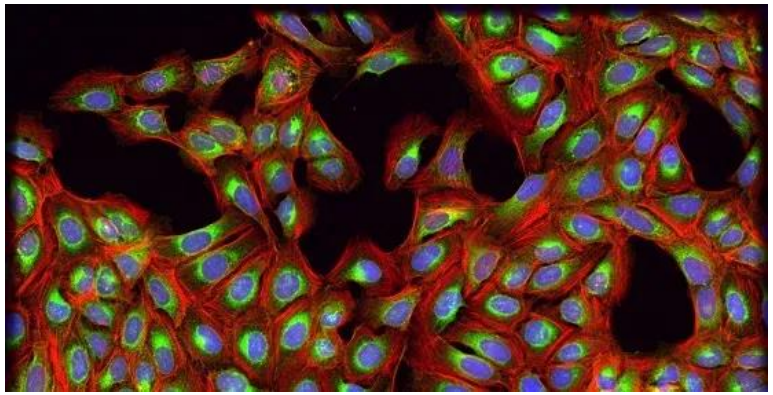


https://www.ncl.ac.uk/emrs/

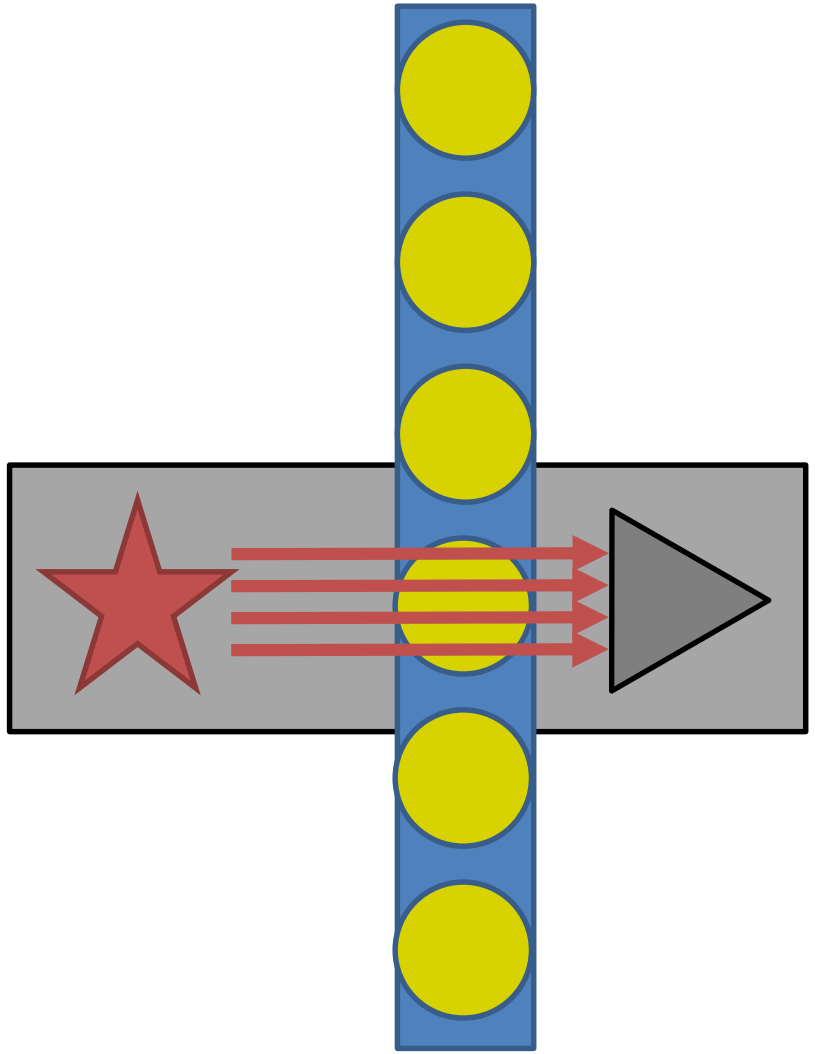
High Content Imaging

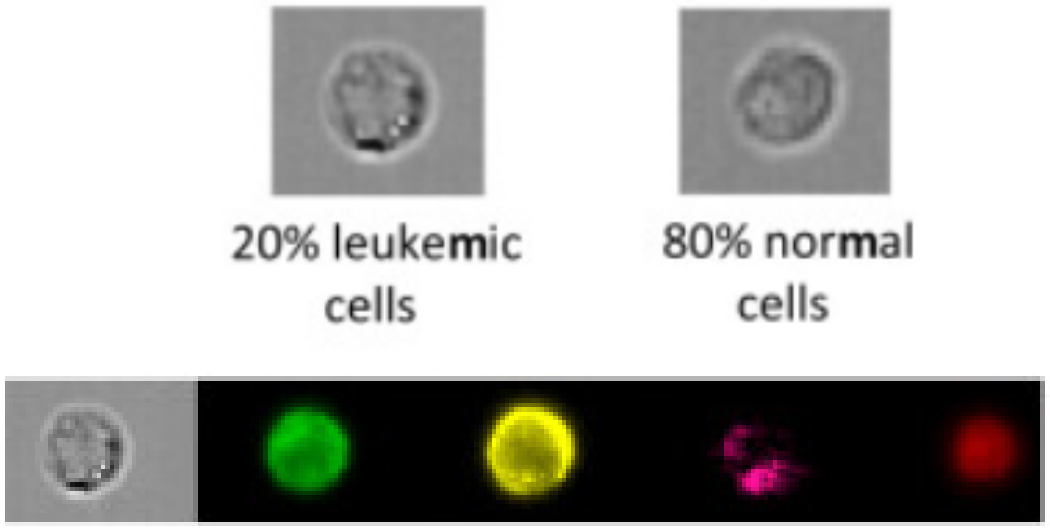
- Microscopy traditionally operated on small numbers of individual samples

- Improved equipment and automation now allows for more ambitious studies
  - 384 well plates
  - 30 images per well
  - 5 different markers
  - Thousands of images
  - Hundreds of measured features per cell



Imaging Flow Cytometry
High Content Imaging from Flowed Cells





Up to 5000 cells per second

Automated real-time feature extraction

High Content Applications

- Screening for drugs with specific phenotypic effects
- Measuring CRISPR library phenotypes
- Measuring RNAi library phenotypes

The BioImage
that are useful
archiving servi
including adde
IDR and Tissue

Release Date: 1 July 2019

[Cite] {JSON} <XML> →PageTab↓ FTP

Recordings of locomotor behaviour in wild-type and mutant Caenorhabditis elegans

Akihiro Mori [1], Yee Lian Chew [2], Laura Grundy [1], Eviatar Yemini [3], Andr? E.X. Brown [4], William R. Schafer [1]

[1] Division of Neurobiology, MRC Laboratory of Molecular Biology [2] [3] [4] [5] Current Address: Illawarra Health and Medical Research Institute & School of Chemistry and Molecular Bioscience, University of Wollongong, Wollongong, Australia [6] Current Address: Columbia University, New York, NY USA [7] Current Address: Imperial College London, UK

Accession S-BIAD9

Description The analysis of behaviou
genes affecting nervous system functio
have been identified whose loss of func
high-content, quantitative phenotypes
underlying this database, consisting of
wild-type controls. Each genotype is re
accessory files containing records of st
represent a useful resource for investig
specific genes on locomotion.

Study type high content screen

Key words C. elegans, locomotor be

Study Organism Caenorhabditis elegan

Data files

Show 5 entries Search:

	Name	Size	Section	Sample Name	Protocol REF	Assay Name	Raw Data File	Comment[Data Repository]	Comment[ventral side]
	AQ2947_AQ2947_on_food_L_2012_02_09__10_25_36___1___1_seg.avi	129.4 MB	Screen A	AQ2947	Tracking of wild-type and mutant Caenorhabditis elegans	AQ2947_mutant_replicate	AQ2947_AQ2947_on_food_L_2012_02_09__10_25_36___1___1.avi	Biostudies	Left
	AQ2947_AQ2947_on_food_L_2012_02_09__10_57_38___8___2_seg.avi	60.7 MB	Screen A	AQ2947	Tracking of wild-type and mutant Caenorhabditis elegans	AQ2947_mutant_replicate	AQ2947_AQ2947_on_food_L_2012_02_09__10_57_38___8___2.avi	Biostudies	Left
	AQ2947_AQ2947_on_food_L_2012_02_09__11_19_46__3_seg.avi	129.5 MB	Screen A	AQ2947	Tracking of wild-type and mutant Caenorhabditis elegans	AQ2947_mutant_replicate	AQ2947_AQ2947_on_food_L_2012_02_09__11_19_46__3.avi	Biostudies	Left
	AQ2947_AQ2947_on_food_L_2012_02_09__11_48___3___4_seg.avi	86.2 MB	Screen A	AQ2947	Tracking of wild-type and mutant Caenorhabditis elegans	AQ2947_mutant_replicate	AQ2947_AQ2947_on_food_L_2012_02_09__11_48___3___4.avi	Biostudies	Left
	AQ2947_AQ2947_on_food_L_2012_02_09__12_13_27___2___5_seg.avi	346.0 MB	Screen A	AQ2947	Tracking of wild-type and mutant Caenorhabditis elegans	AQ2947_mutant_replicate	AQ2947_AQ2947_on_food_L_2012_02_09__12_13_27___2___5.avi	Biostudies	Left

Showing 1 to 5 of 10,105 entries

Previous 1 2 3 4 5 ... 2021 Next

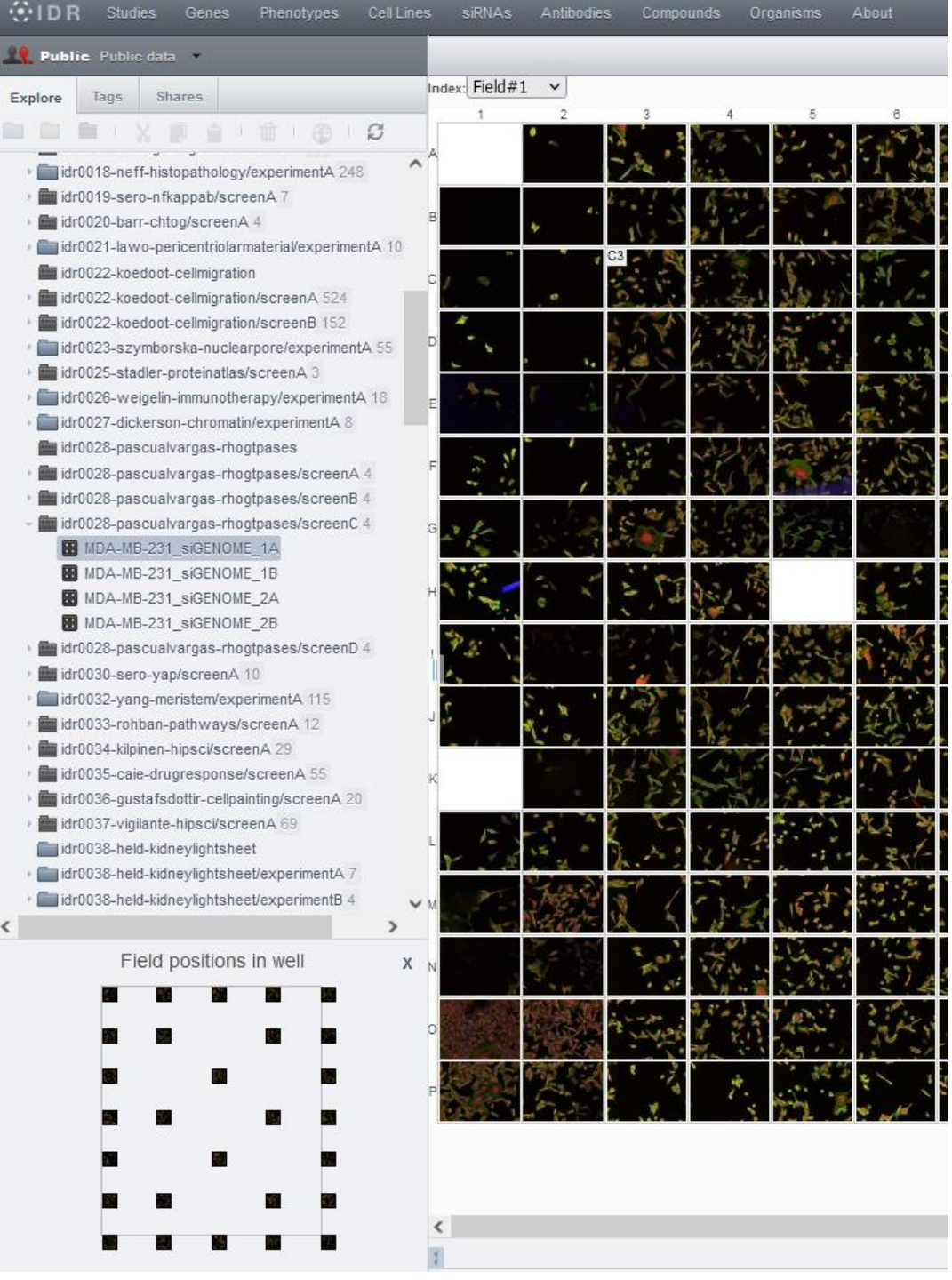
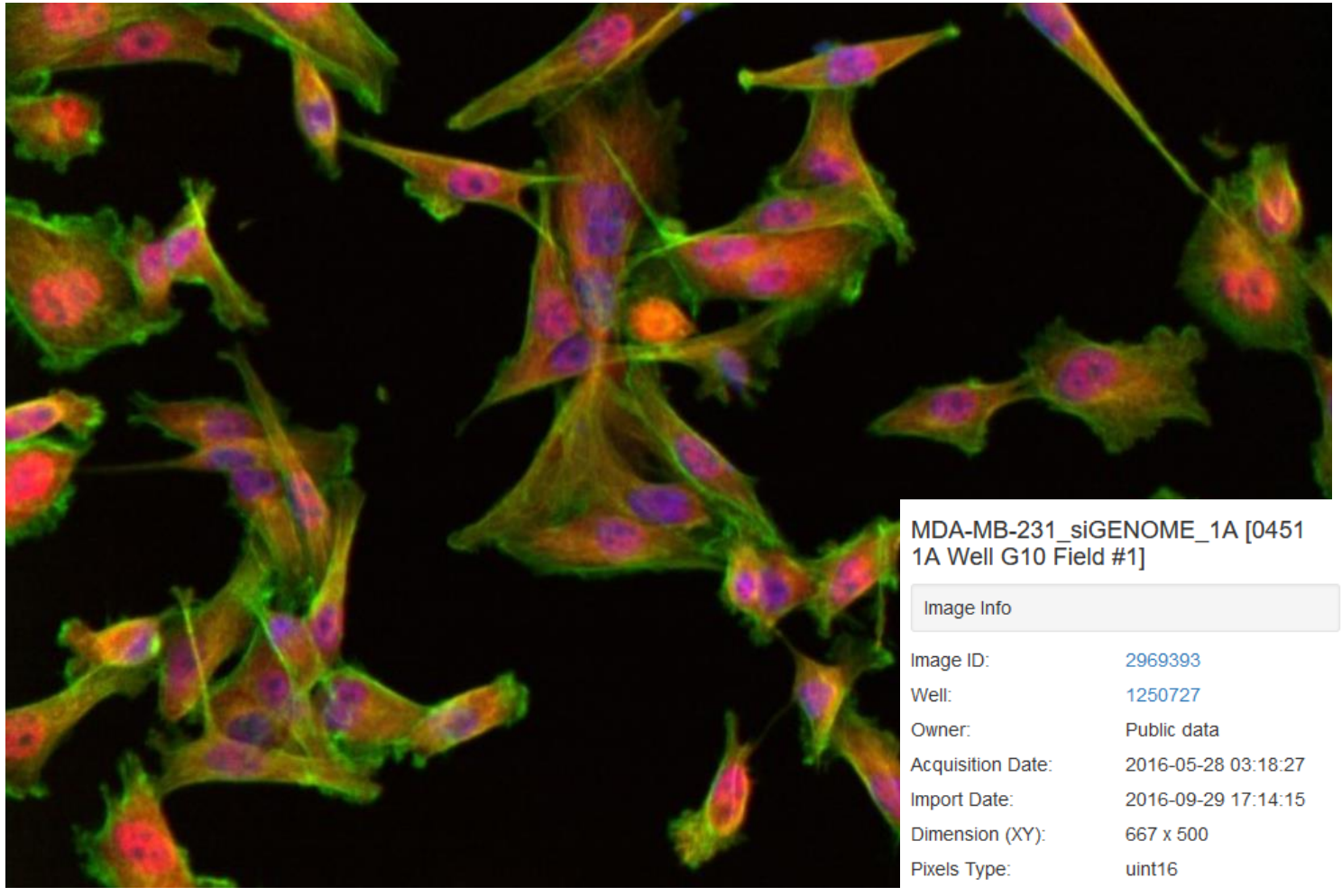


Image Data Resource



MDA-MB-231_siGENOME_1A [0451 1A Well G10 Field #1]

Image Info

Image ID:	2969393
Well:	1250727
Owner:	Public data
Acquisition Date:	2016-05-28 03:18:27
Import Date:	2016-09-29 17:14:15
Dimension (XY):	667 x 500
Pixels Type:	uint16
Pixels Size (XYZ) (µm):	0.65 x 0.65 x -
Z-sections:	1
Timepoints:	1
Channels:	Hoechst, AlexaFluor568, Phalloidin488, AlexaFluor647

Imaging Data Exercise

Babraham Bioinformatics

Graphical Software for Sequence Exploration

- IGV
  - Viewer for multiple library types
  - Generally works with BAM or VCF files
  - Looks at sequence level alignments of reads against genomes

- SeqMonk
  - Visualisation and analysis for mapped datasets
  - Looks at positions rather than sequence
  - RNA-Seq, ChIP, ATAC, BS-Seq etc
  - Works with BAM files

Using IGV

Babraham Bioinformatics

Integrative Genomics Viewer (IGV)

Software from the Broad Institute http://software.broadinstitute.org/software/igv/home

Interactive tool for the visual exploration of genomic data

Available to download and run as a desktop java application

Also available as an online application https://igv.org/app/

James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, Jill P. Mesirov. **Integrative Genomics Viewer**. Nature Biotechnology 29, 24–26 (2011).

IGV

Can use it without data to explore genes in the genome (similar to Ensembl / UCSC)

Upload bam files for data exploration – must have accompanying index file in the same location as the bam file (.bai)

Upload VCF files for variant analysis – must have accompanying index file in the same location at the VCF file (.tbi)

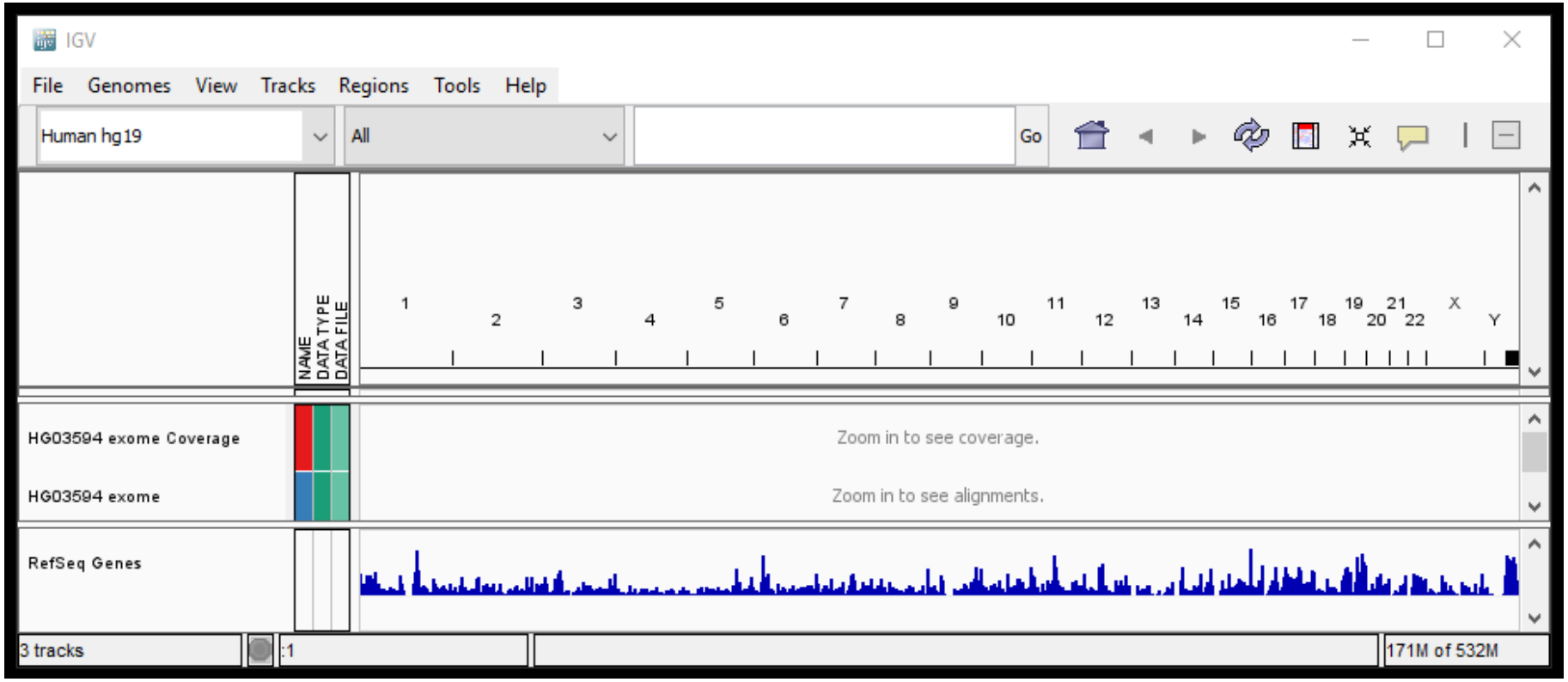
IGV web app with no data loaded

IGV desktop – initial view

Zoomed right out showing all the chromosomes

No reads are shown at this zoom level

Track at the bottom shows gene density



Gene level view – ‘expanded’

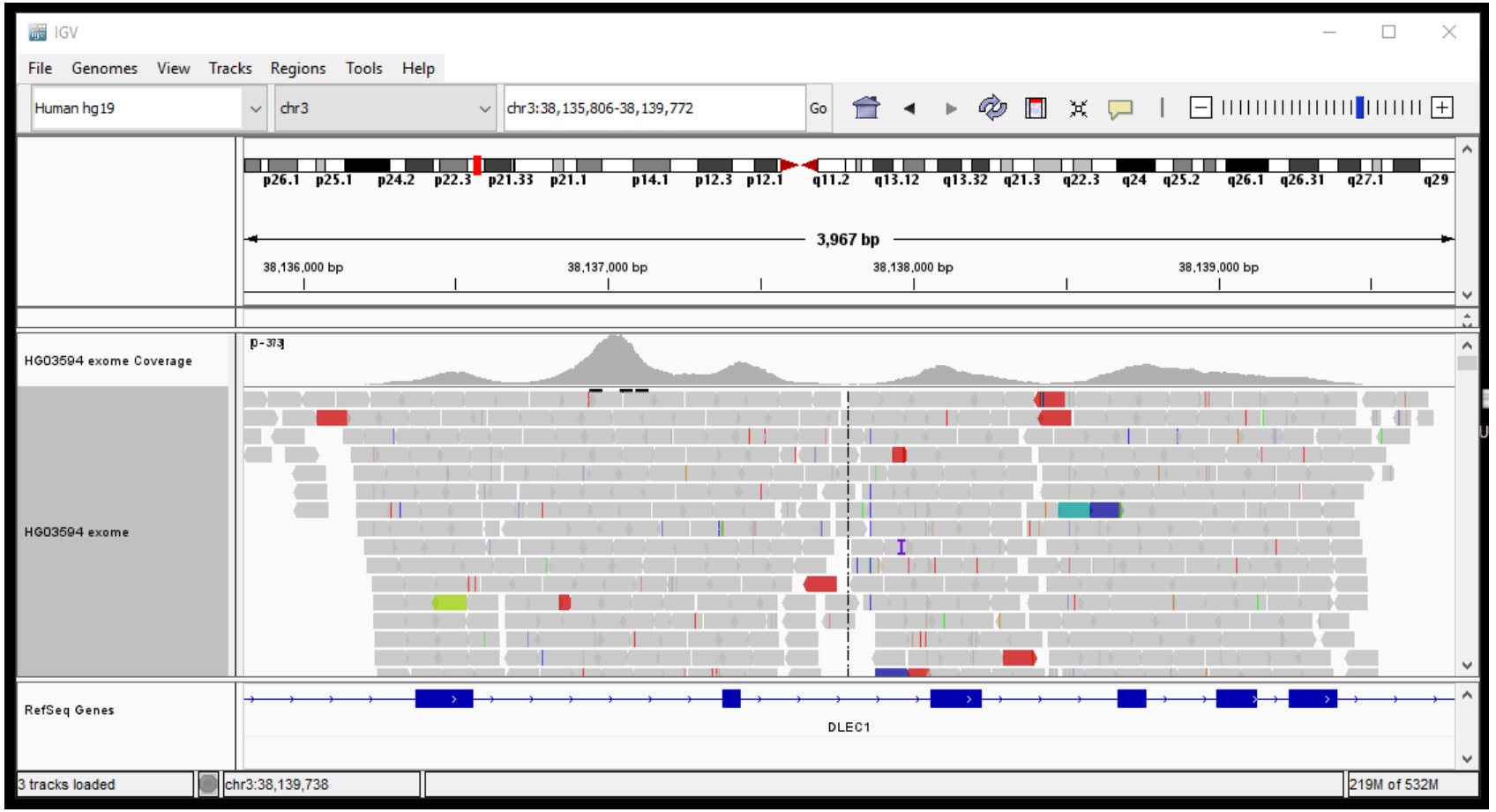
Zoom in to see coverage track and aligned reads.

Track at the bottom shows genes.

Exons are solid rectangles, strand is shown by arrows

Can click on gene to see more info, link to ncbi

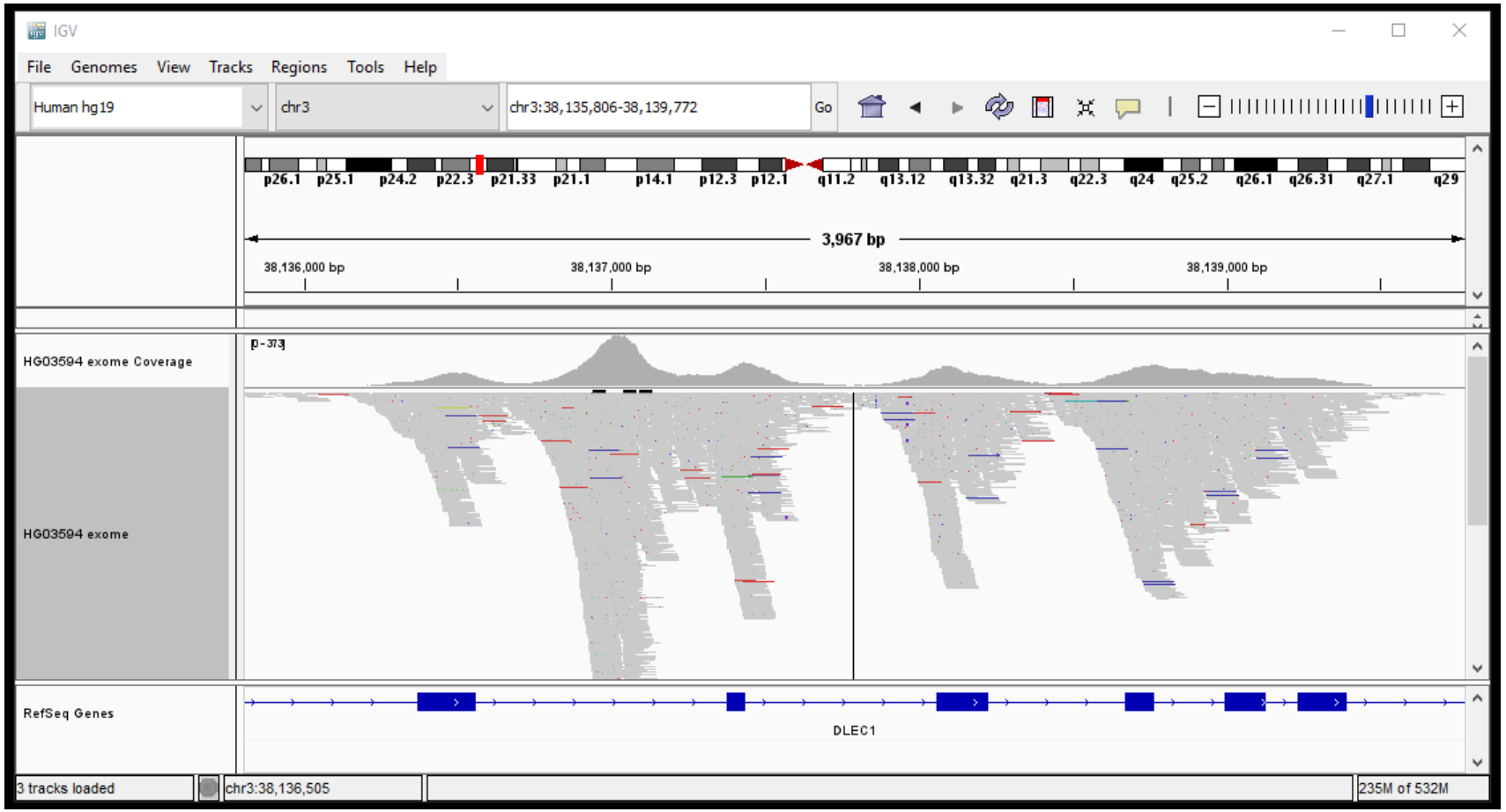
To see splice variants, right click on gene and select “Expanded”



Gene level view – ‘squished’

Colours represent different chromosomal events

- Blue - inserts that are smaller than expected
- Red - inserts that are larger than expected.
- Inter-chromosomal rearrangements are color-coded by chromosome.



Sequence level view

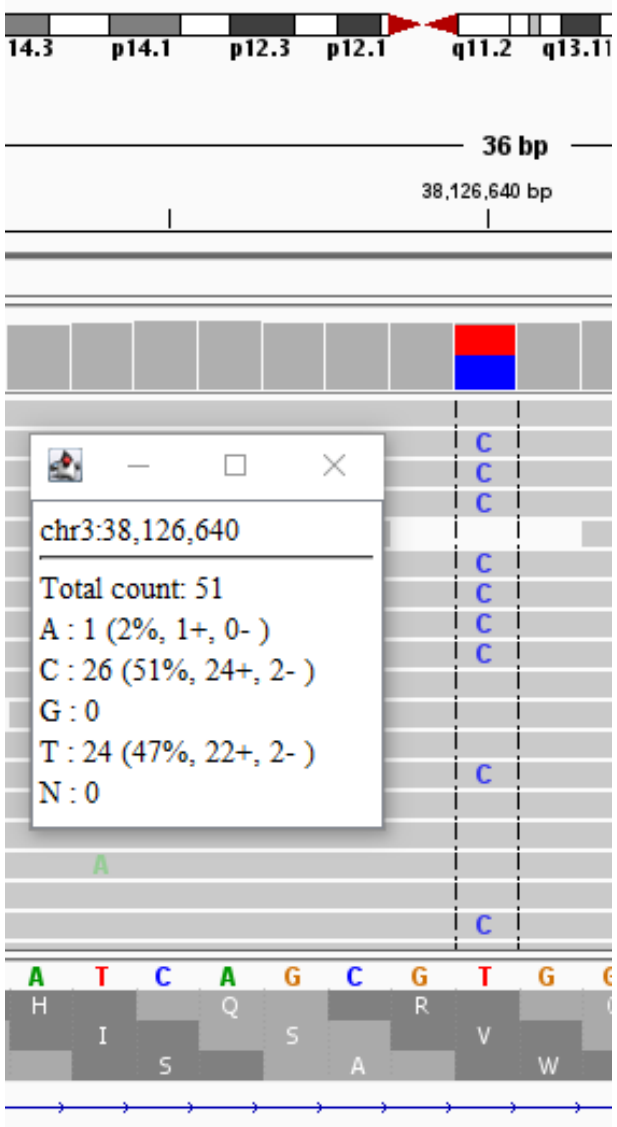
Zoom right in to base pair resolution.

Clicking on reference nucleotides shows or hides the 3 frame aa translation

Forward strand is shown – change this by clicking small arrow to the left of the DNA sequence

Sequence level view

SNPs are highlighted in the coverage track if the nucleotide differs from the reference in >= 20% of reads

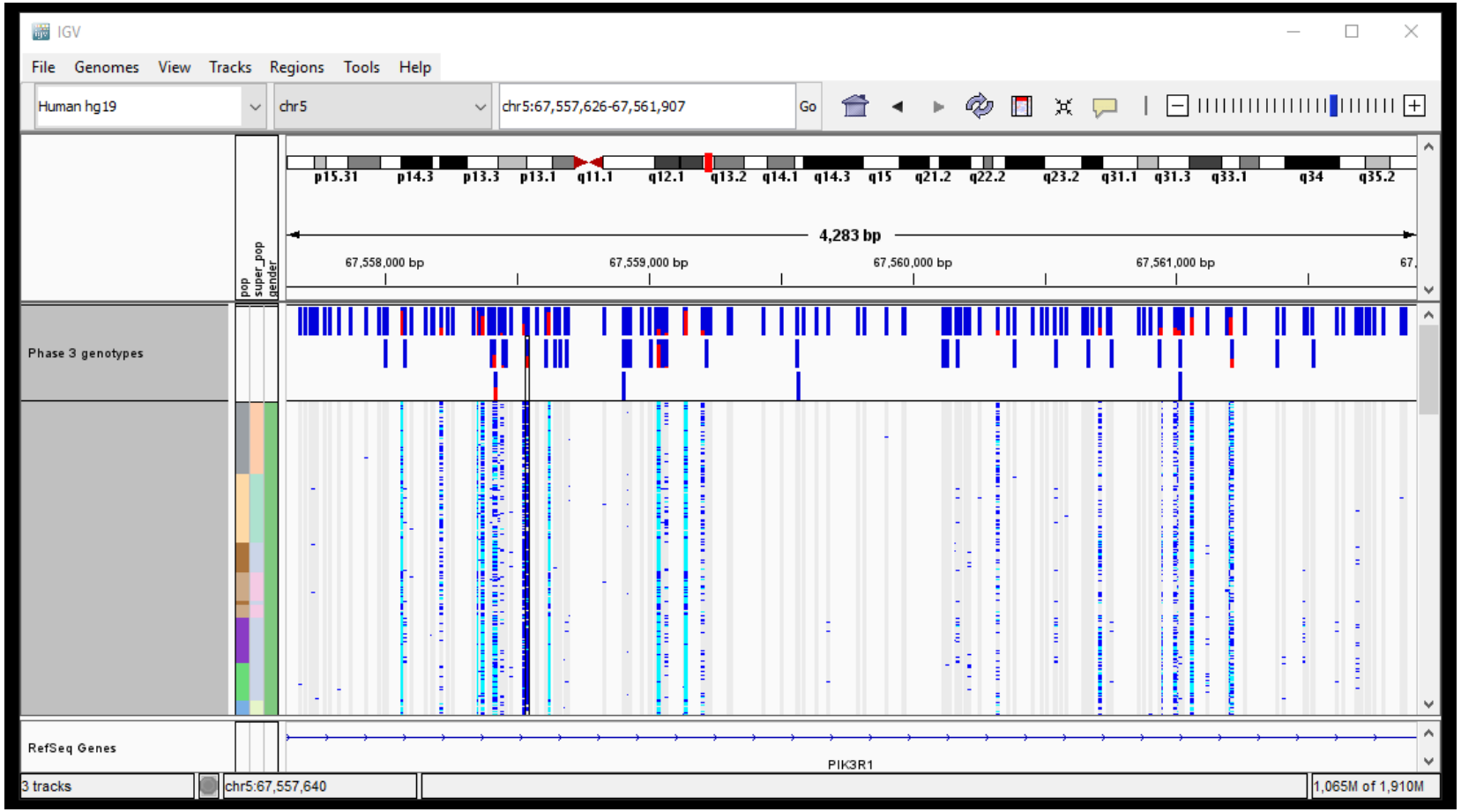


Variants – variant call format (VCF) files

VCF files only show variation from reference

Default – dark blue is heterozygous and light blue is homozygous for SNP

Can supply a metadata file - tab delimited file with sample names the same as the tracks

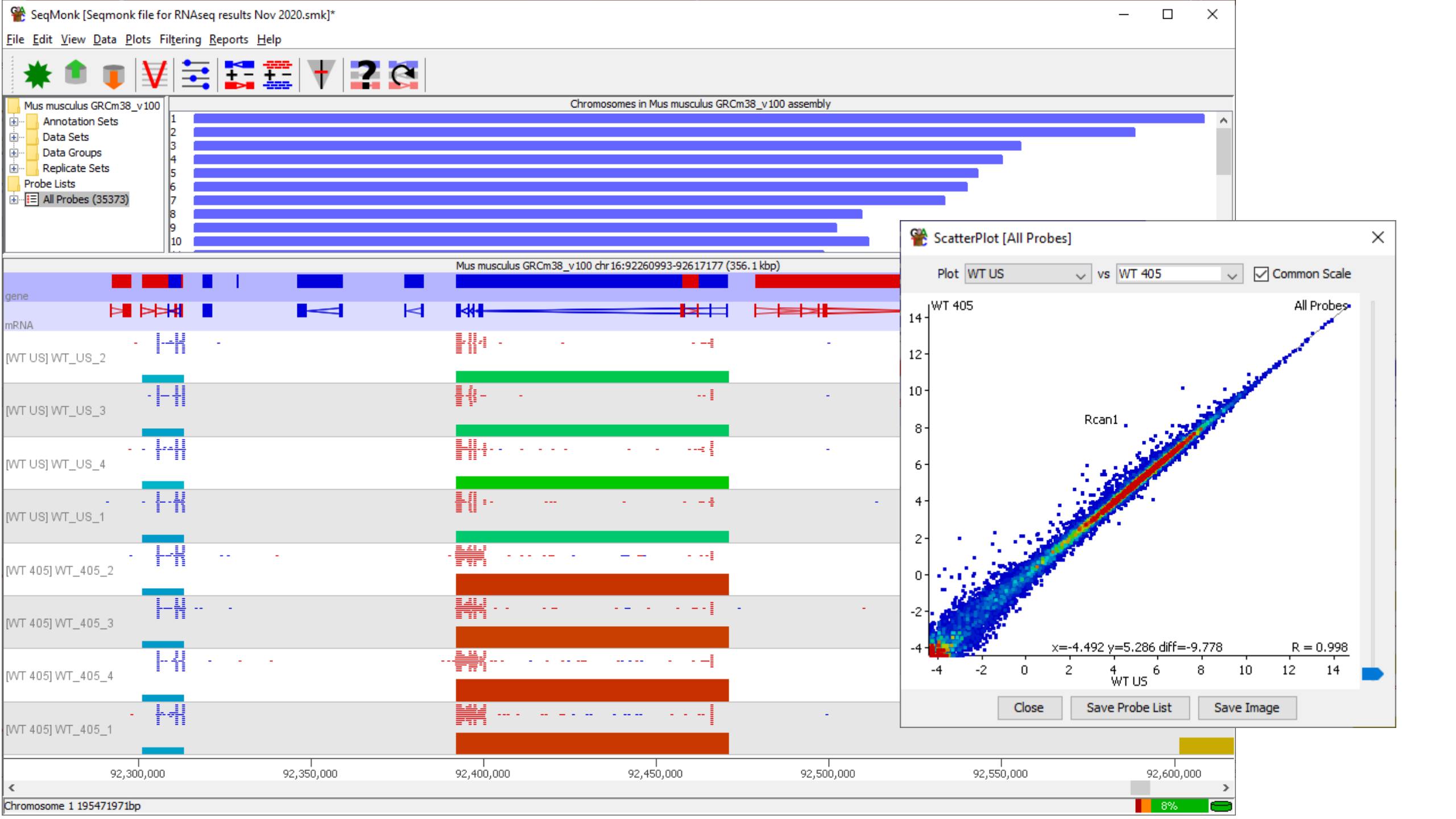


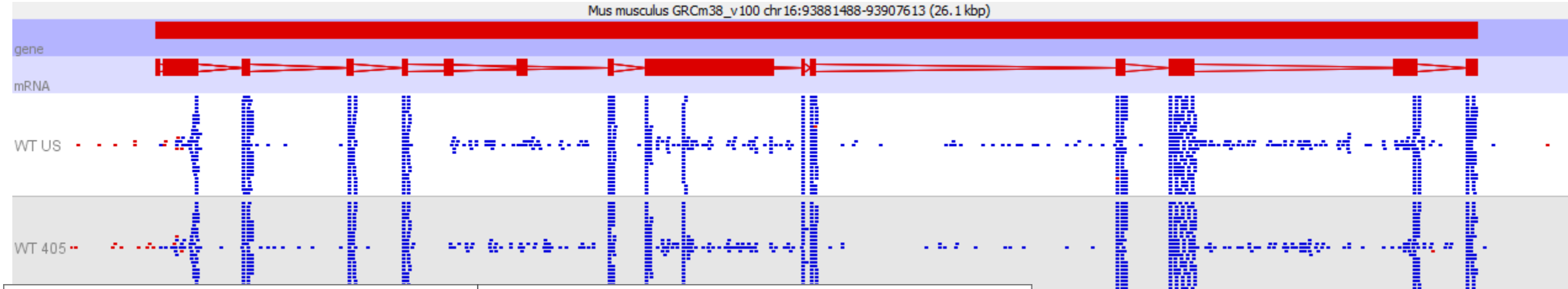
IGV Exercise

Babraham Bioinformatics

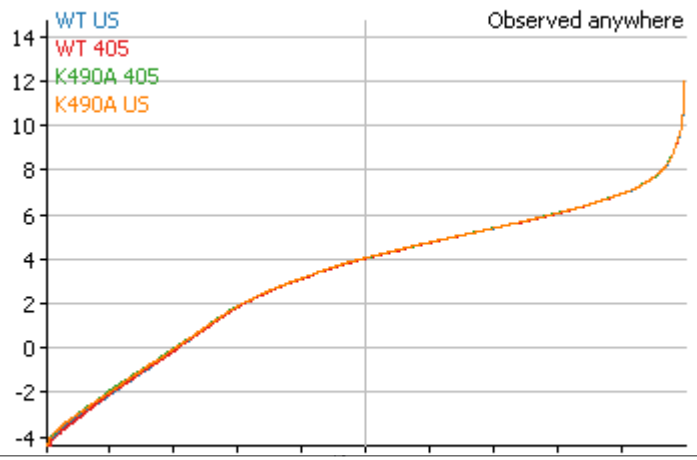
SeqMonk

Babraham Bioinformatics

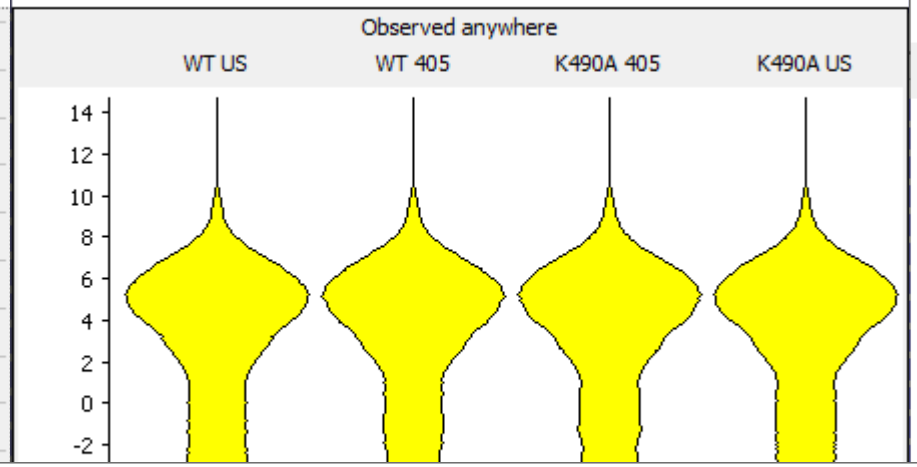

SeqMonk [Seqmonk file for RNAseq results Nov 2020.smk]*
File Edit View Data Plots Filtering Reports Help
Mus musculus GRCm38_v100
Annotation Sets
Data Sets
Data Groups
Replicate Sets
Probe Lists
All Probes (35373)
Chromosomes in Mus musculus GRCm38_v100 assembly
1
2
3
4
5
6
7
8
9
10
Mus musculus GRCm38_v100 chr16:92260993-92617177 (356.1 kbp)
gene
mRNA
[WT US] WT_US_2
[WT US] WT_US_3
[WT US] WT_US_4
[WT US] WT_US_1
[WT 405] WT_405_2
[WT 405] WT_405_3
[WT 405] WT_405_4
[WT 405] WT_405_1
92,300,000
92,350,000
92,400,000
92,450,000
92,500,000
92,550,000
92,600,000
Chromosome 1 195471971bp
8%
ScatterPlot [All Probes]
Plot WT US vs WT 405
Common Scale
WT 405
All Probes
Rcan1
x=-4.492 y=5.286 diff=-9.778
R = 0.998
WT US
Close
Save Probe List
Save Image



Cumulative Distribution Graph [Observed anywhere]

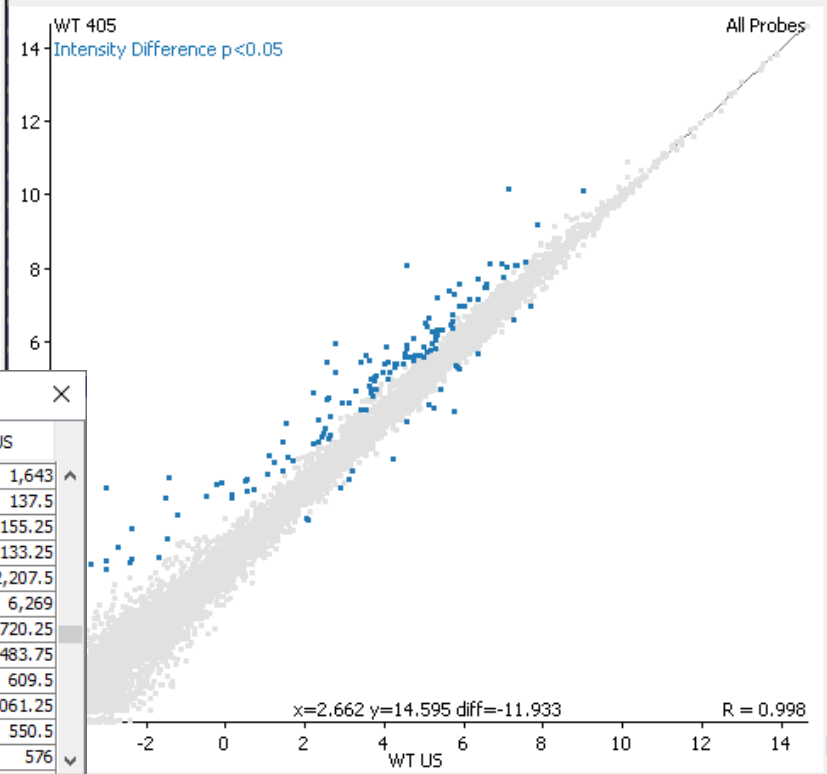


Bean Plots



ScatterPlot [All Probes]

Plot WT US vs WT 405 ☑ Common Scale [Highlight Sublists]



Close | Save Probe List | Save Image

Annotated Probe Report for DESeq stats p<0.05 after correction

Probe	Chrom...	Start	End	Probe ...	Feature	ID	Descri...	Featur...	...	...	Distance	P-v...	FDR (...	Log2 Fo...	Shrun...	WT US	WT 405	K490A 405	K490A US
Swap70	7	110221...	110283...	+	Swap70	ENSMUS...	SWA-70...	+	...	...	0	0	0.002	0.262	0.237	1,578.5	1,376	1,513.75	1,643
Ankrd33b	15	31291478	31367726	-	Ankrd33b	ENSMUS...	ankyrin ...	-	...	...	0	0	0.002	-0.573	-0.377	106.25	165.25	174.5	137.5
Plaur	7	24462484	24475968	+	Plaur	ENSMUS...	plasmin...	+	...	...	0	0	0.002	-0.192	-0.182	7,709	9,085.25	7,929.75	7,155.25
Gm49510	15	78899667	78919032	+	Gm49510	ENSMUS...	predicte...	+	...	...	0	0	0.002	0.239	0.219	1,036	914.25	1,069.75	1,133.25
Spint1	2	119237...	119249...	+	Spint1	ENSMUS...	serine p...	+	...	...	0	0	0.002	0.225	0.208	2,223.25	1,987.5	1,813.75	2,207.5
Slfn2	11	83065112	83070678	+	Slfn2	ENSMUS...	schlafe...	+	...	...	0	0	0.002	-0.313	-0.271	6,181.5	8,104.75	7,791	6,269
Hlx	1	184727...	184732...	-	Hlx	ENSMUS...	H2.0-lik...	-	...	...	0	0	0.002	0.326	0.279	651.5	540.25	645.75	720.25
Fabp4	3	10204088	10208576	-	Fabp4	ENSMUS...	fatty ac...	-	...	...	0	0	0.002	-0.417	-0.327	1,679.25	2,284.25	1,915.25	1,483.75
Aar2	2	156547...	156568...	+	Aar2	ENSMUS...	AAR2 s...	+	...	...	0	0	0.002	-0.27	-0.242	608.25	757.75	690	609.5
Clcn5	X	7153810	7319358	-	Clcn5	ENSMUS...	chloride...	-	...	...	0	0	0.002	-0.274	-0.245	1,032	1,297	1,334.75	1,061.25
Gramd4	15	86057695	86137634	+	Gramd4	ENSMUS...	GRAM d...	+	...	...	0	0	0.002	0.412	0.324	552.75	429.25	442.75	550.5
Lyl1	8	84701449	84704940	+	Lyl1	ENSMUS...	lympho...	+	...	...	0	0	0.002	0.39	0.314	582	467.75	444	576

Close | Save to File | Save to Vistory

SeqMonk Exercise

Babraham Bioinformatics

Computational Environments for Processing and Analysing Big Data

Simon Andrews

Babraham Bioinformatics

Computation for Big Data

- Physical
  - What sort of machine / storage can I use?
  - What will I need?
- Software
  - What programs exist to process / analyse my data?
  - What operating system will they run under?
- Programming / Analysis
  - How can I write new analysis tools or perform programmatic analyses?

Topics for Today

- Running programs in a command line environment
- How to select which programs / methods to use?
- Programmatic analysis with R and Tidyverse
- Developing new analysis tools with python

Big Data Operating Systems

What (exactly) is Linux?

UNIX®

FreeBSD

GNU Operating System

redhat®

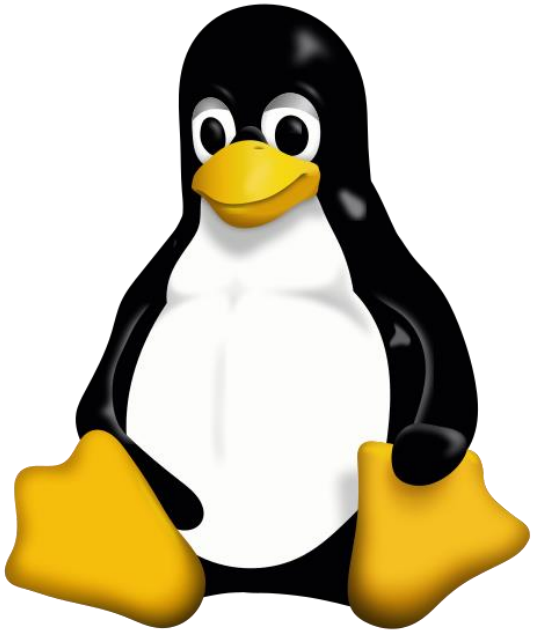
ubuntu®

CentOS

Why Linux?

- Programs are long running and require automation
  - Need a command line driven operating system
  - Easy text based remote access

- Computers need to operate at scale
  - Free and open source are a real benefit
  - Can tinker with everything to tune performance



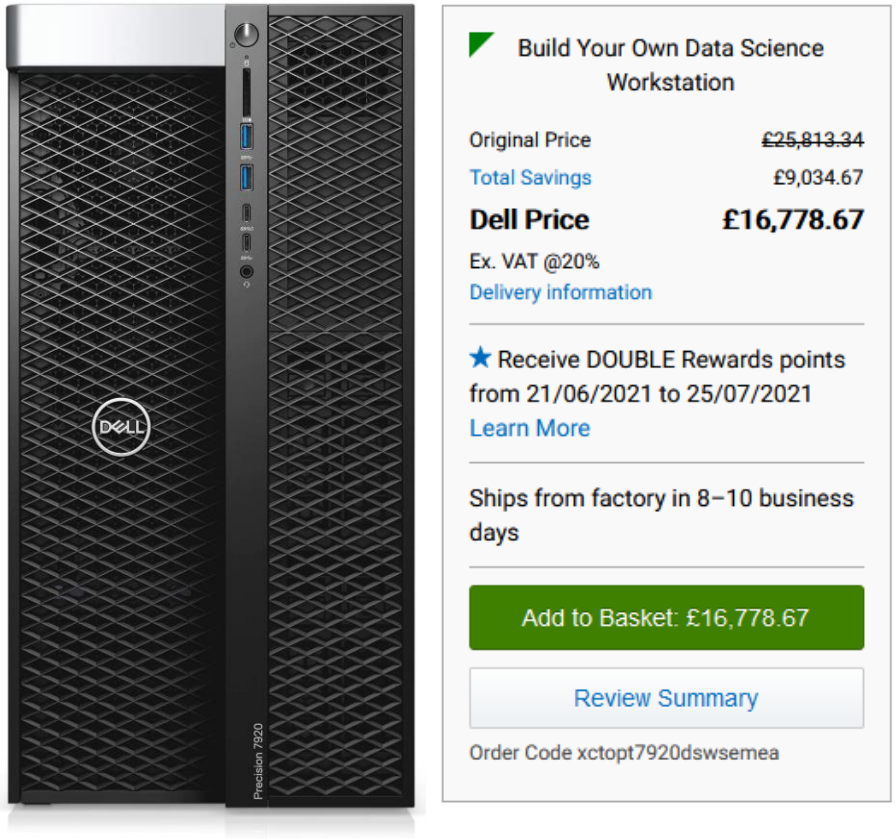
Babraham Bioinformatics

Types of Linux installation

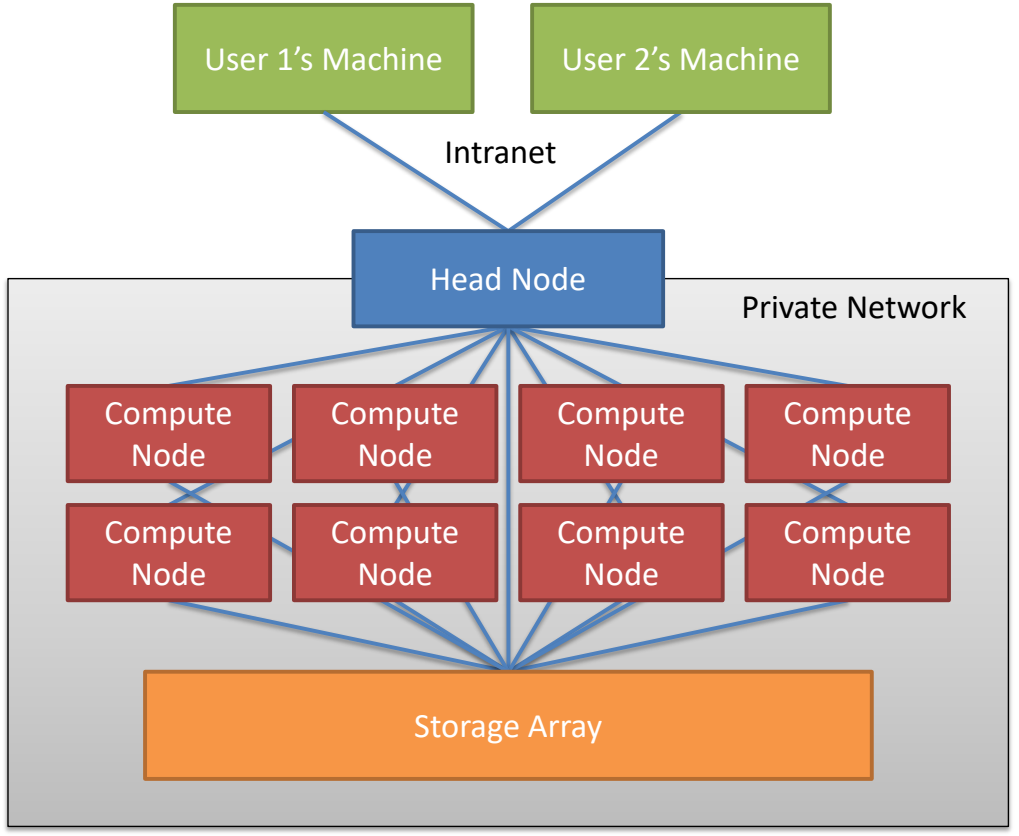
- Bare metal
  - Physical hardware
  - CD / DVD / USB / Network installation
  - Can be physically accessible (desktop) or remote (server / cluster)
- Virtual Machine
  - Runs within another operating system
  - Portable / disposable
  - Install from ISO / Network
- Cloud
  - Virtual machine on someone else's hardware
  - Amazon / Google are the main providers
  - Range of available hardware and OS images available
  - Pay by the hour

Single Machines vs Clusters

1 physical box
28 CPU cores
512GB RAM



20 physical boxes
~700 CPU cores
7TB RAM



Cluster Queues


```
fastqc data.fq.gz

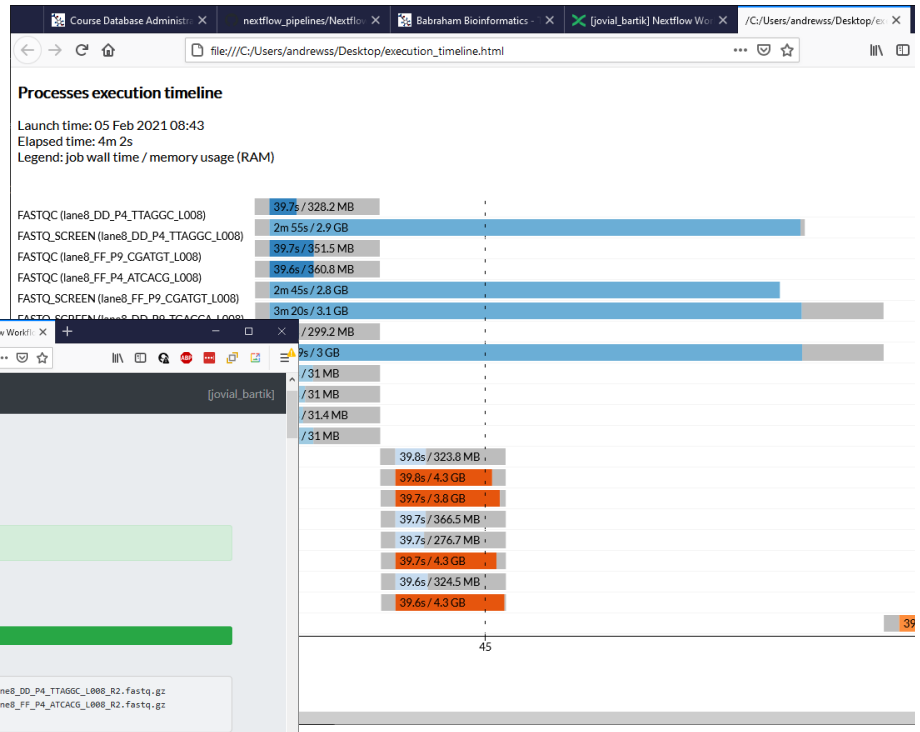
ssub
    -o f.log
    --cores=2
    --mem=5G
    fastqc data.fq.gz
```

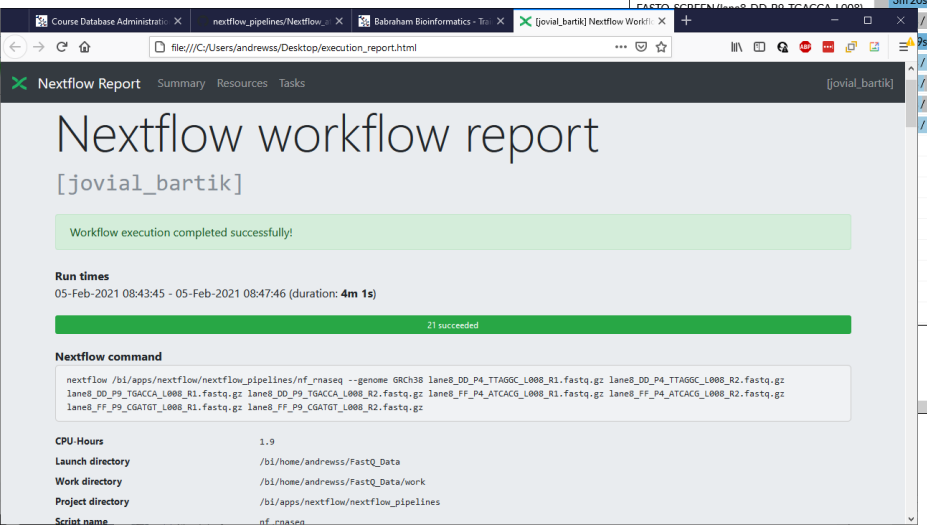
Workflows

- Larger Scale Automation
- Multiple Programs
- Multiple Files
- Integrates with Clusters


```
nf_rnaseq --genome GRCh38 *fastq.gz
```

```
executor >  slurm (21)
[15/929bd5] process > FASTQC (lane8_DD_P9_TGACCA_L008)          [100%] 4 of 4 ✔
[b9/674ced] process > FASTQ_SCREEN (lane8_FF_P4_ATCACG_L008)    [100%] 4 of 4 ✔
[ca/b39d14] process > TRIM_GALORE (lane8_FF_P9_CGATGT_L008)     [100%] 4 of 4 ✔
[c0/4dcaf9] process > FASTQC2 (lane8_FF_P9_CGATGT_L008)         [100%] 4 of 4 ✔
[58/879cf5] process > HISAT2 (lane8_FF_P9_CGATGT_L008)          [100%] 4 of 4 ✔
[c4/cfe1f1] process > MULTIQC                                   [100%] 1 of 1 ✔
Completed at: 05-Feb-2021 08:47:47
Duration    : 4m 2s
CPU hours   : 1.9
Succeeded   : 21
```





Workflow completion notification

Run Name: jovial_bartik

Execution completed successfully!

The command used to launch the workflow was as follows:

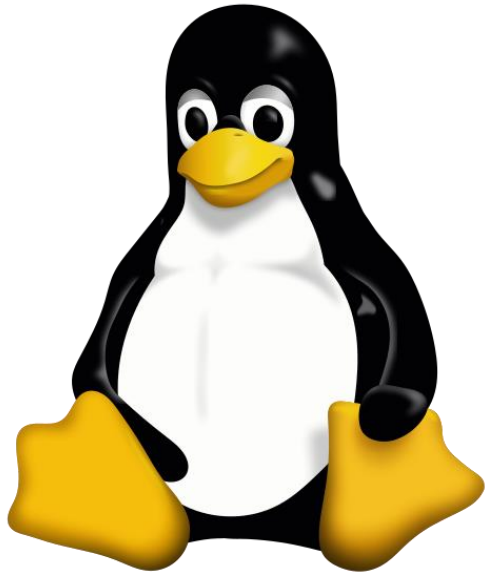
```
nextflow /bi/apps/nextflow/nextflow_pipelines/nf_rnaseq --genome GRCh38
lane8_DD_P4_TTAGGC_L008_R1.fastq.gz lane8_DD_P4_TTAGGC_L008_R2.fastq.gz
lane8_DD_P9_TGACCA_L008_R1.fastq.gz lane8_DD_P9_TGACCA_L008_R2.fastq.gz
lane8_FF_P4_ATCACG_L008_R1.fastq.gz lane8_FF_P4_ATCACG_L008_R2.fastq.gz
lane8_FF_P9_CGATGT_L008_R1.fastq.gz lane8_FF_P9_CGATGT_L008_R2.fastq.gz
```

Execution summary

Launch time	05-Feb-2021 08:43:45
Ending time	05-Feb-2021 08:47:46 (duration: 4m 1s)
Total CPU-Hours	1.9
Tasks stats	Succeeded: 21 Cached: 0 Ignored: 0 Failed: 0

Running programs in the BASH shell

Babraham Bioinformatics



Running programs in Linux

- Two major methods
  - Graphical
  - Command line
- Graphical launches only work for graphical programs accessed through a graphical environment
- Most data processing will be command line based, as will most remote access
  - Graphical programs can still be launched from the command line

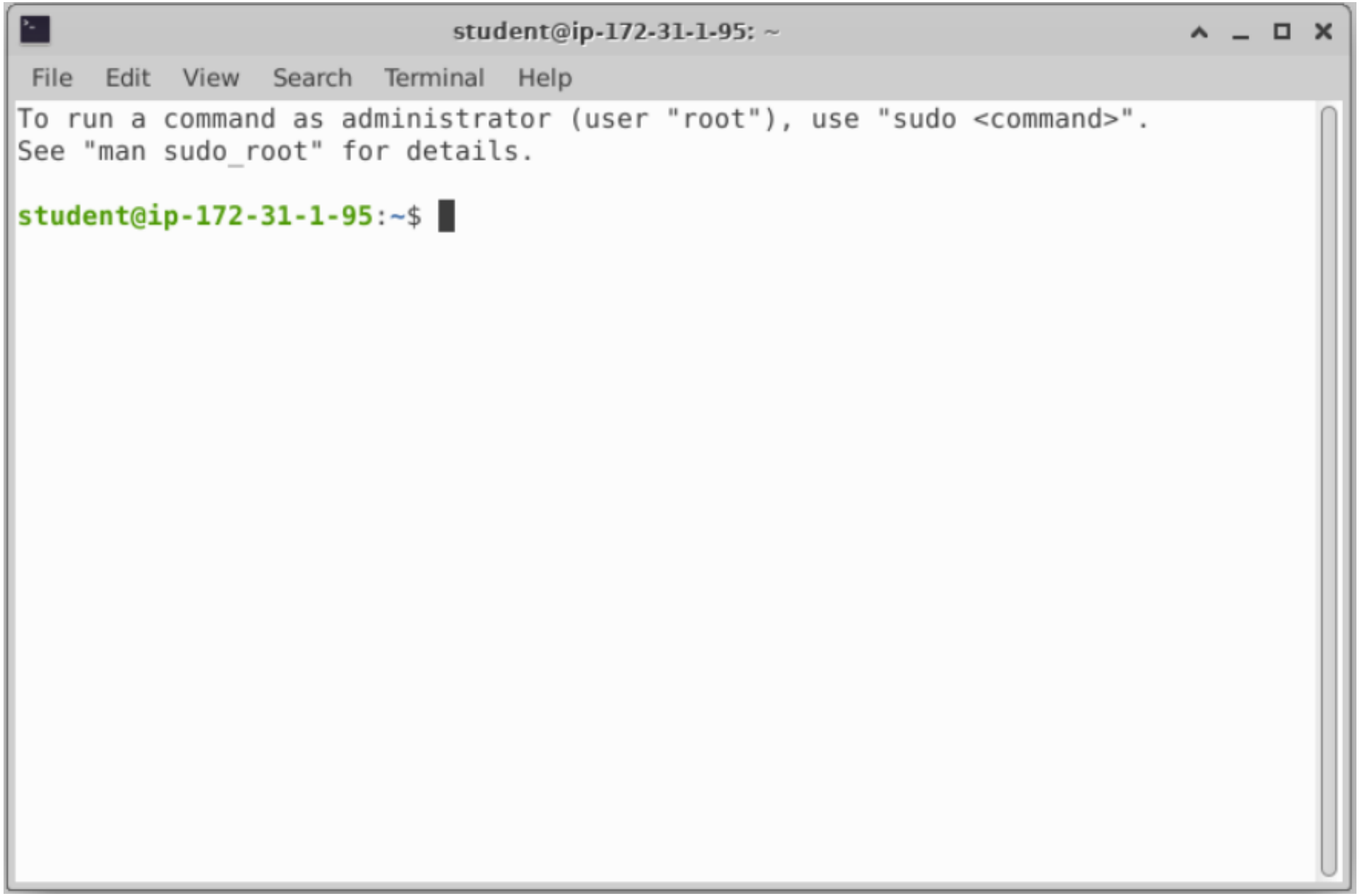
Shells

- A shell is a command line interpreter, used to launch software in Linux
- Text commands are used to launch programs
- We will use the most popular shell, BASH

What does a shell provide

- Command line editing and construction tools
- History
- Job control

- Automation
  - Scripting language
  - Variables, functions etc


student@ip-172-31-1-95: ~
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
student@ip-172-31-1-95:~$

Running programs

- Type the name of the program you want to run
  - Add on any options the program needs
  - Press return - the program will run
- When the program ends control will return to the shell
- Run the next program!

Running programs

```
student@ip1-2-3-4:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music
Pictures  Public  Templates  Videos

student@ip1-2-3-4:~$
```

- Command prompt - you can't enter a command unless you can see this
- The command we're going to run (`ls` in this case, to list files)
- The output of the command - just text in this case

Running graphical programs

```
student@ip1-2-3-4:~$ xeyes
```

```
student@ip1-2-3-4:~$
```

Note that you can't enter another command
until you close the program you launched

The structure of a unix command

Program name

Switches

Data
(normally files)

Each option or section is separated by spaces. Options or files with spaces in must be put in quotes.

Command line switches

- Change the behaviour of the program
- Come in two flavours (each option usually has both types available)
  - Minus plus single letter (eg `-x -c -z`)
    - Can be combined (eg `-xcz`)
  - Two minuses plus a word (eg `--extract --gzip`)
    - Can't be combined
- Some take an additional value, this can be an additional option, or use an = to separate (it's up to the program)
  - `-f somfile.txt` (specify a filename)
  - `--width=30` (specify a value)

Manual pages

- All core programs will have a manual page to document the options for the command
- Manual pages are accessible using the man program followed by the program name you want to look up.
- All manual pages have a common structure

Manual Pages (`man cat`)

```
CAT(1)                          User Commands                          CAT(1)

NAME
       cat - concatenate files and print on the standard output

SYNOPSIS
       cat [OPTION]... [FILE]...

DESCRIPTION
       Concatenate FILE(s) to standard output.

       With no FILE, or when FILE is -, read standard input.

       -A, --show-all
              equivalent to -vET

       -n, --number
              number all output lines

       -T, --show-tabs
              display TAB characters as ^I

       --help display this help and exit

EXAMPLES
       cat f - g
              Output f's contents, then standard input, then g's contents.

       cat    Copy standard input to standard output.
```

Help Pages

- For non core programs (ie analysis / processing) you won't have a man page
- Instead use `--help` to get the help page

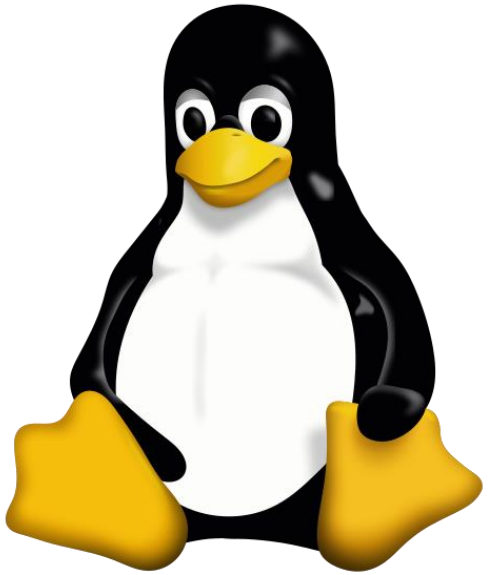
```
$ hisat2 --help
HISAT2 version 2.1.0 by Daehwan Kim (infphilo@gmail.com, www.ccb.jhu.edu/people/infphilo)
Usage:
  hisat2 [options]* -x <ht2-idx> {-1 <m1> -2 <m2> | -U <r> | --sra-acc <SRA accession number>} [-S <sam>]

  <ht2-idx>  Index filename prefix (minus trailing .X.ht2).
  <m1>       Files with #1 mates, paired with files in <m2>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>       Files with #2 mates, paired with files in <m1>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <r>        Files with unpaired reads.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <SRA accession number>        Comma-separated list of SRA accession numbers, e.g. --sra-acc SRR353653,SRR353654.
  <sam>      File for SAM output (default: stdout)

  <m1>, <m2>, <r> can be comma-separated lists (no whitespace) and can be
  specified many times.  E.g. '-U file1.fq,file2.fq -U file3.fq'.

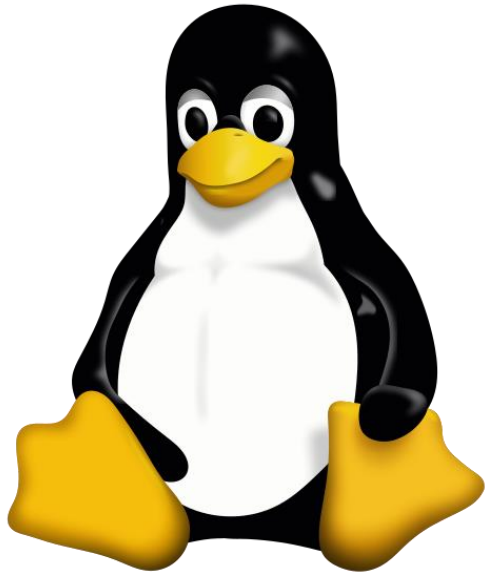
Options (defaults in parentheses):
```

Exercise 12
Running Programs in Bash



Understanding Unix File Systems

Babraham Bioinformatics



Unix File Systems

- Consists of a hierarchical set of directories (folders)
- Each directory can contain files

- No drive letters (drives can appear at arbitrary points in the file system)

- No file extensions (you can add them, but they're not required)

A simple unix filesystem

/ (Always the top of the file system)

- home/ (Directory containing all home directories)
  - anne/
  - simon/
    - Documents/ (All names are case sensitive)
      - test.txt (A file we want to work with) **/home/simon/Documents/test.txt**
- media/
  - myusb/ (A USB stick added to the system)

Creating and moving into directories

- Every Unix session has a ‘working directory’ which is a folder where the shell looks for file paths
- You can see your current working directory with `pwd`
- Your initial working directory will be your home directory (`/home/user`)
- You can change your working directory with `cd [new working directory]`
- Running `cd` on its own takes you back home
- You can create a new directory with `mkdir [new directory name]`

Specifying file paths

- Some shortcuts
  - ~ (tilde, just left of the return key) - the current user's home directory
  - . (single dot) - the current directory
  - .. (double dot) - the directory immediately above the current directory

Specifying file paths

- Absolute paths from the top of the file system
  - `/home/simon/Documents/Course/some_file.txt`
- Relative paths from whichever directory you are currently in
  - If I'm working in `/home/simon/Course/`
  - `big_data.csv = /home/simon/Course/big_data.csv`
- Paths using the home shortcut
  - `~/Documents/Course/some_file.txt` will work for user simon anywhere on the system

Command line completion

- Most errors in commands are typing errors in either program names or file paths
- Shells (ie BASH) can help with this by offering to complete path names for you
- Command line completion is achieved by typing a partial path and then pressing the TAB key (to the left of Q)

Command line completion

Actual files in a folder:

```
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
Videos
```

If I type the following and press tab:

`De` [TAB] will complete to `Desktop` as it is the only option

`T` [TAB] will complete to `Templates` as it is the only option

`Do` [TAB] will no nothing (just beep) as it is ambiguous

`Do` [TAB] [TAB] will show `Documents` and `Downloads` since those are the only options

`Do` [TAB] [TAB] `c` [TAB] will complete to `Documents`

You should ALWAYS use TAB completion to fill in paths for locations which exist so you can't make typing mistakes

(it obviously won't work for output files though)

Wildcards

- Another function provided by your shell (not your application)
- A quick way to be able to specify multiple related file paths in a single operation
- There are two main wildcards
  - ***** = Any number of any characters
  - **?** = One of any character
- You can include them at any point in a file path and the shell will expand them before passing them on to the program
- Multiple wildcards can be in the same path.
- Command line completion won't work after the first wildcard

Wildcard examples

```
$ ls Monday/*txt
Monday/mon_1.txt  Monday/mon_2.txt  Monday/mon_3.txt
Monday/mon_500.txt

$ ls Monday/mon_?.txt
Monday/mon_1.txt  Monday/mon_2.txt  Monday/mon_3.txt

$ ls */*txt
Friday/fri_1.txt  Monday/mon_1.txt  Monday/mon_3.txt
Tuesday/tue_1.txt
Friday/fri_2.txt  Monday/mon_2.txt  Monday/mon_500.txt
Tuesday/tue_2.txt

$ ls */*1.txt
Friday/fri_1.txt  Monday/mon_1.txt  Tuesday/tue_1.txt
```

The structure of a Unix command

Each option or section is separated by spaces. Options or files with spaces in must be put in quotes.

Manipulating files

- You will spend a lot of time managing files on a Linux system.
  - Viewing files (normally text files)
  - Editing text files
  - Moving or renaming files
  - Copying files
  - Deleting files
  - Finding files

Viewing Files

- Simplest solution
  - `cat [file]` Sends the entire contents of a file (or multiple files) to the screen.

- Quick look
  - `head` or `tail` will look at the start/end of a file
    - `head -10 [file]`
    - `tail -20 [file]`

- More scalable solution
  - `less` is a 'pager' program, sends output to the screen one page at a time

    - Return / j = move down one line
    - k = move up one line
    - Space = move down one page
    - b = go back one page
    - /[term] = search for [term] in the file
    - q = quit back to the command prompt

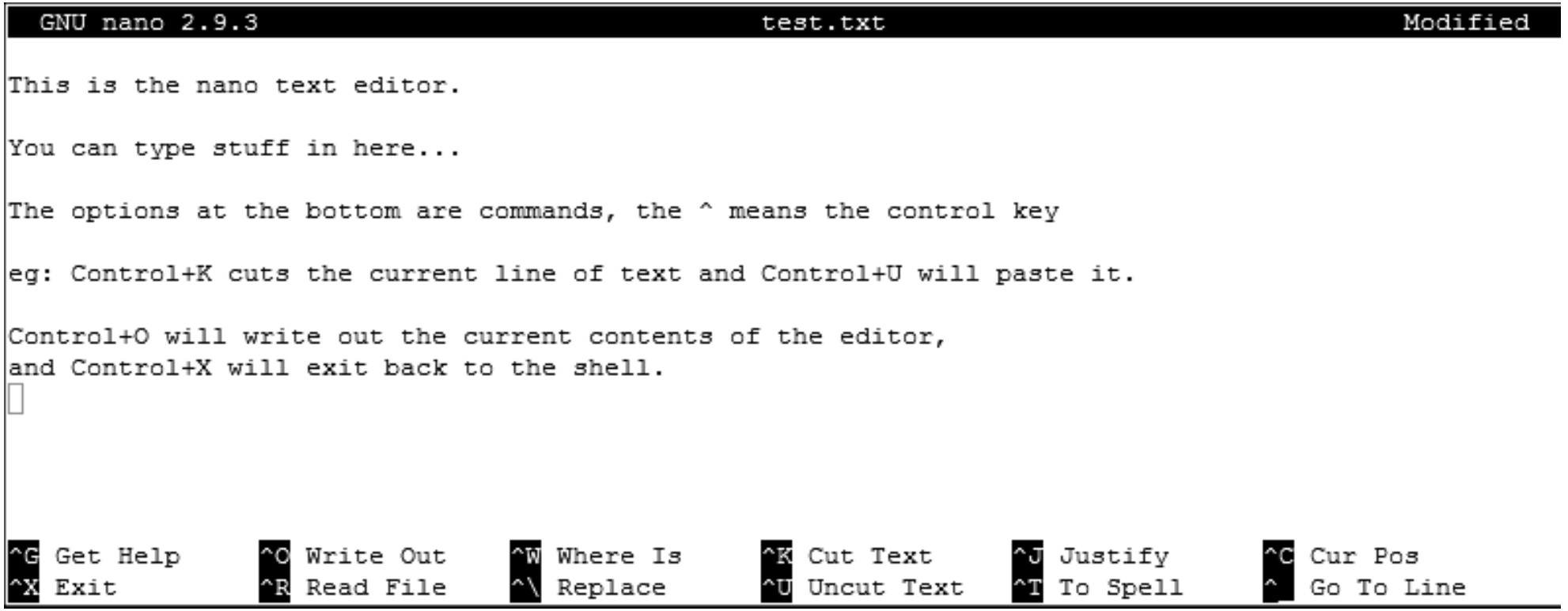
`less -S` (no wrapping)

Editing files

- Lots of text editors exist, both graphical and command line
- Many have special functionality for specific content (C, HTML etc)
- `nano` is a simple command line editor which is always present

Using nano to edit text files

- `nano [filename]` (edits if file exists, creates if it doesn't)



Moving / Renaming files

- Uses the `mv` command for both (renaming is just moving from one name to another)
- `mv [file or directory] [new name/location]`
- If new name is a directory then the file is moved there with its existing name
- Moving a directory moves all of its contents as well
- Examples
  - `mv old.txt new.txt`
  - `mv old.txt ../Saved/`
  - `mv old.txt ../Saved/new.txt`
  - `mv ../Saved/old.txt .`

Copying files

- Uses the `cp` command
- `cp [file] [new file]`
- Operates on a single file
- Can copy directories using recursive copy (`cp -r`)

- Examples
  - `cp old.txt new.txt`
  - `cp old.txt ../Saved/`
  - `cp old.txt ../Saved/new.txt`
  - `cp ../Saved/old.txt .`
  - `cp -r ../Saved ./NewDir`
  - `cp -r ../Saved ./ExistingDir/` (only if ExistingDir exists)

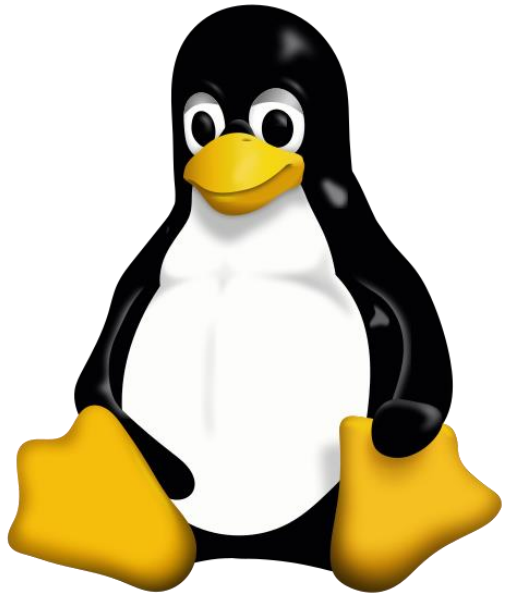
Deleting files

- Linux has no undo.
- Deleting files has no recycle bin.
- Linux will not ask you "are you sure"

- Files can be deleted with the `rm` command
- Directories (and all of their contents) can be deleted with `rm -r`

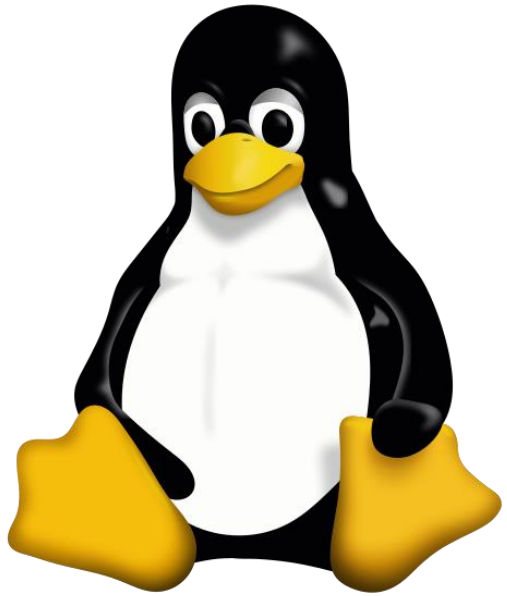
- Examples
  - `rm test_file.txt test_file2.txt`
  - `rm *.txt` (be VERY careful using wildcards. Always run `ls` first to see what will go)
  - `rm -r Old_directory/`

Exercise 13:
Using the filesystem



More clever BASH usage

Babraham Bioinformatics



What we know already

- How to run programs
- How to modify the options for a program using switches
- How to supply data to programs using file paths and wildcards

What else can we do

- Record the output of programs
- Check for errors in programs which are running
- Link programs together into small pipelines
- Automate the running of programs over batches of files

- All of these are possible with some simple BASH scripting

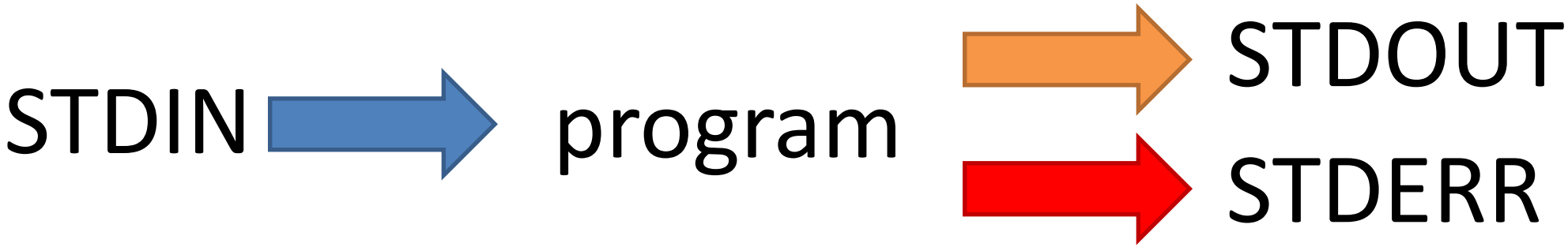
Recording the output of programs

- One of the aspects of POSIX is a standard system for sending data to and from programs.

- Three data streams exist for all Linux programs (though they don't have to use them all)
  - STDIN (Standard Input - a way to send data into the program)
  - STDOUT (Standard Output - a way to send expected data out of the program)
  - STDERR (Standard Error - a way to send errors or warnings out of the program)

- By default STDOUT and STDERR are connected to your shell, so when you see text coming from a program, it's coming from these streams.

Recording the output of programs



- Rather than leaving these streams connected to the screen, you can link them to either files, or other programs to either create logs, or to build small pipelines

Redirecting standard streams

- You can redirect using arrows at the end of your command
  - `> [file]` Redirects STDOUT
  - `< [file]` Redirects STDIN
  - `2> [file]` Redirects STDERR
  - `2>&1` Sends STDERR to STDOUT so you only have one output stream

```
$ find . -print > file_list.txt 2> errors.txt
```

```
$ ls
Data  Desktop  Documents  Downloads  errors.txt  examples.desktop  file_list.txt  Music  Pictures  Public  Templates  Videos

$ head file_list.txt
.
./Downloads
./Pictures
./Public
./Music
./.bash_logout
./.local
./.local/share
./.local/share/icc
./.local/share/icc/edid-33d524c378824a7b78c6c679234da6b1.icc
```

Throwing stuff away

- Sometimes you want to be able to hide output
  - STDOUT - I just want to test whether something worked
  - STDERR - I want to hide progress / error messages

- Linux defines a special file /dev/null which you can write to but just discards all data sent to it
  - `might_fail > /dev/null`
  - `chatty_app 2> /dev/null`

Linking programs together with pipes

- Part of the original UNIX design was to have lots of small programs doing specific jobs, and then to link them together to perform more advanced tasks.

- Pipes are designed to do this by connecting STDOUT from one program to STDIN on another

Linking programs together using pipes

- Pipes are a mechanism to connect the STDOUT of one program to the STDIN of another. You can use them to build small pipelines
- To create a pipe just use a pipe character | between programs

```
$ ls | head -2
Data
Desktop
```

Useful programs for pipes

- Whilst you can theoretically use pipes to link any programs, there are some which are particularly useful, these are things like:

  - `wc` to do word and line counting
  - `grep` to do pattern searching
  - `sort` to sort things
  - `uniq` to deduplicate things
  - `less` to read large amounts of output
  - `zcat/gunzip/gzip` - to do decompression or compression

Small example pipeline

- Take a compressed fastq sequence file, extract from it all of the entries containing the telomere repeat sequence (TTAGGG) and count them

- `zcat file.fq.gz | grep TTAGGGTTAGGG | wc -l`

```
$ zcat file.fq.gz | wc -l
179536960

$ zcat file.fq.gz | grep TTAGGGTTAGGG | wc -l
3925
```

Iterating over files

- When processing data it is common to need to re-run the same command multiple times for different input/output files.
- Some programs will support being provided with multiple input files, but many will not.
- You can use the automation features of the BASH shell to automate the running of these types of programs

The BASH `for` loop

- Simple looping construct
  - Loop over a set of files
  - Loop over a set of values

- Creates a temporary environment variable which you can use when creating commands

Examples of `for` loops

```
for value in {5,10,20,50}
  do
  run_simulation --iterations=$value > ${value}_iterations.log 2>&1
done
```

```
for value in {10..100}
  do
  run_simulation --iterations=$value > ${value}_iterations.log 2>&1
done
```

```
for file in *txt
  do
  echo $file
  grep .sam $file | wc -l
done
```

Job Control

- By default you run one job at a time in a shell
  - Shells support multiple running jobs

- States of job
  - Running - foreground (shell has the attention of the job)
  - Running - background (output goes to the shell but other jobs can run)
  - Suspended - background (job exists but is paused, consumes no CPU)
  - Running - disconnected (output is no longer attached to the shell)

Job Control

- `prog_to_run` (starts in foreground)
- `prog_to_run &` (starts in background)

- Control of running jobs
  - `jobs` lists the jobs in this shell
  - Control + Z suspends the current job
  - `bg` puts the current suspended job into the background
  - `fg [num]` puts the selected job back in the foreground

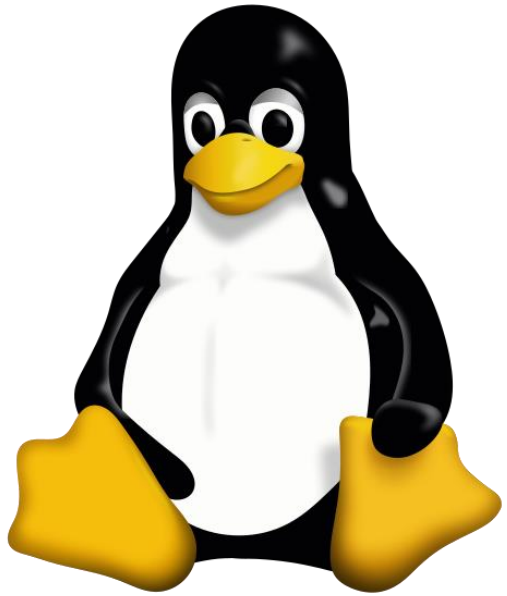
Job Control - nohup

- `nohup prog_to_run`
  - Merges STDOUT and STDERR
  - Disconnects from the terminal
  - Can't be killed when the terminal exits
  - Output appended to `nohup.out`
  - Can redirect with `> logfile.txt`

Exercise 14: Automation in BASH

Selecting Analysis Tools

Babraham Bioinformatics



RNA-Seq Aligner Selection

- ABMapper
- BBMap
- ContextMap
- CRAC
- GSNAP
- GMAP
- Hisat
- Hisat2
- HMMSplicer
- MapSplice
- MapNext
- Olego
- PALMapper
- Pass
- PASSion
- PASTA
- QPALMA
- RAZER
- SeeSaw
- SoapSplice
- SpliceMap
- SplitSeq
- STAR
- Subjunc
- SuperSplat
- TopHat

Article | Open Access | Published: 12 November 2020

Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis

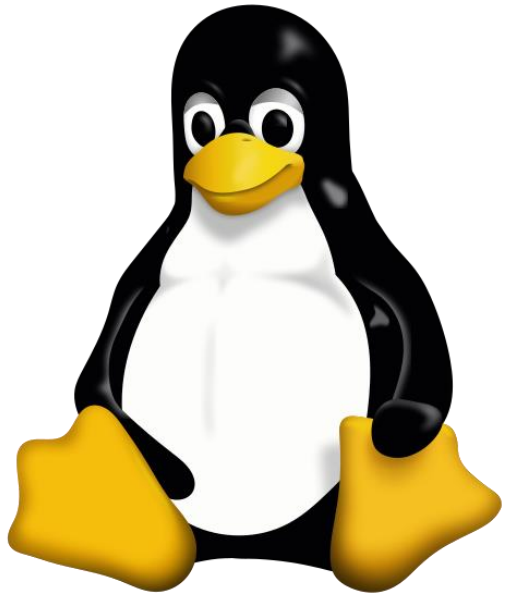
Luis A. Corchete, Elizabeta A. Rojas, Diego Alonso-López, Javier De Las Rivas, Norma C. Gutiérrez & Francisco J. Burguillo

Scientific Reports **10**, Article number: 19737 (2020) | Cite this article

- Local Knowledge
- Relevant increases in sensible metrics
- Ease of installation / use / defaults
- Documentation
- Longevity and support

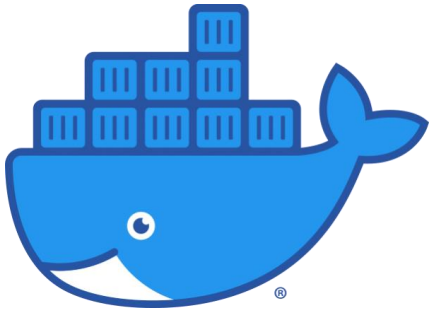
Installing New Software

Babraham Bioinformatics



Different Options

- Ask someone to do it for you
  - Best option if you're on a managed system
- Manual Installation
  - Look for install instructions - sometimes trivial, sometimes horrific
- Containerised Applications
  - Docker or Singularity
- Automated Installation
  - BioConda



Containers

- Single applications or pipelines in a VM
- Lighter than normal virtual machines
- Every app operates in an isolated environment
- All dependencies handled for you
- Not easy to modify or debug
- Software black box

BIOCONDA®

- Collection of recipes to install applications on different systems
- Handles dependencies and versioning
- Local installation per user
- Options to install mutually incompatible software
- Great when it works
  - Users love it
  - Sysadmins hate it
- A nightmare when it doesn't work
  - Debugging is really complex

[Optional BioConda Exercise]

Babraham Bioinformatics

Programmatic Environments

Different Types of Programmatic Environment

Data Analysis

- Alternative to GUI exploration/analysis
- Interactive Environment
- Graphing and Statistics
- Report Generation

- Reproducible
- Automatable
- Flexible

- Often one-off

Application Development

- Data processing and extraction
- Automation and pipelining
- Use of remote resources
- Interaction with users

- Non-interactive use
- Advanced command line options
- Longer term development

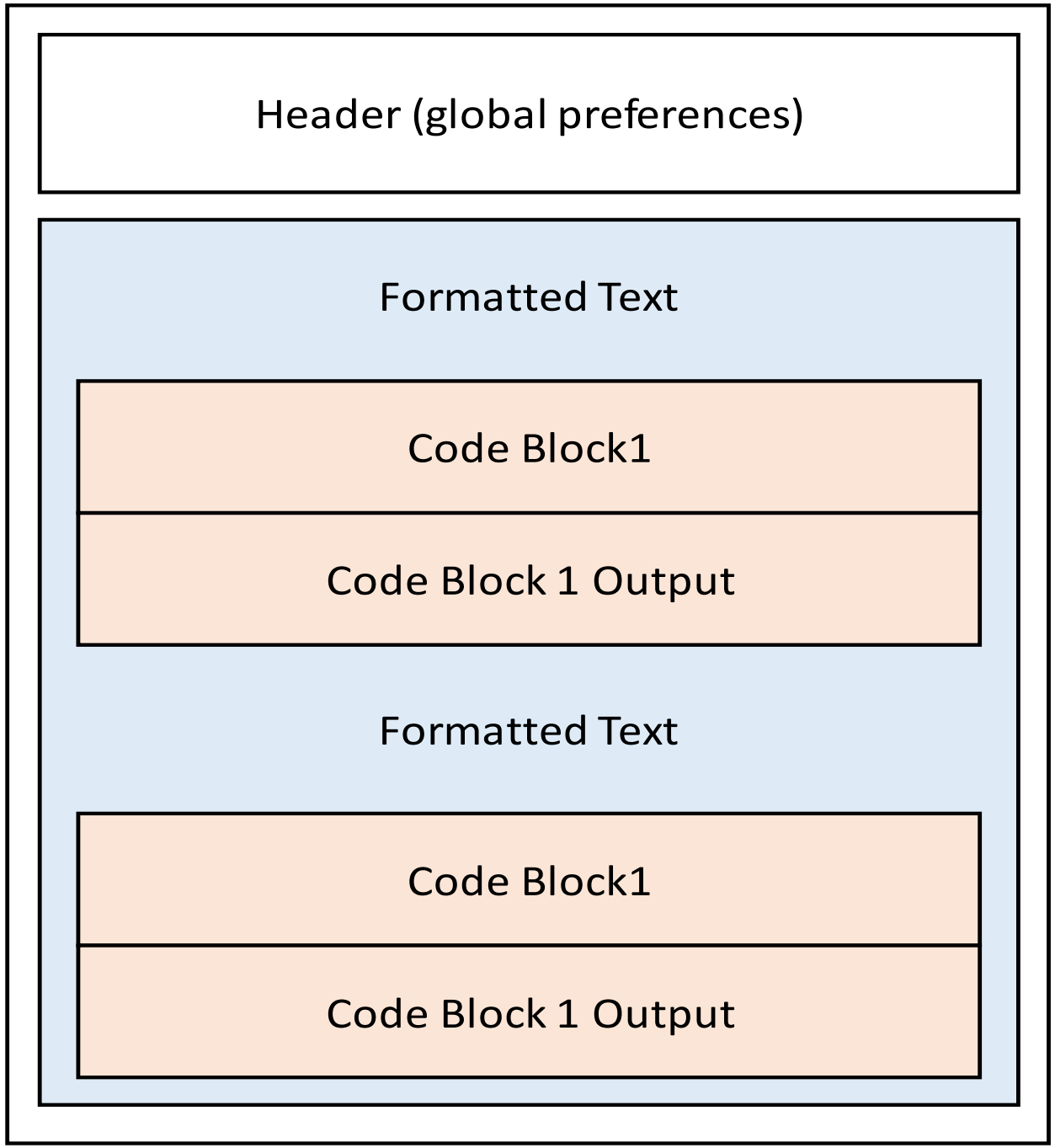
Programmatic Analysis

- Alternative and complement to exploratory graphical tools
- Positives
  - Reproducible and automatable
  - Completely flexible and scalable
- Negatives
  - Tends to encourage repetition without exploration
  - Can be difficult to spot unusual behaviour / bugs

R, Rstudio, Tidyverse, Notebooks

Notebook Structure

- Single overall text document, split into sections
  - Header (mostly preferences)
  - Body
    - Commentary (default)
    - R Code
    - Output (graphical and text)



Code

Introduction
Processing
Read the data
Summarise
Plot


```
---
title: "Example Notebook"
output:
  html_document:
    df_print: paged
    toc: true
    toc_float: true
---

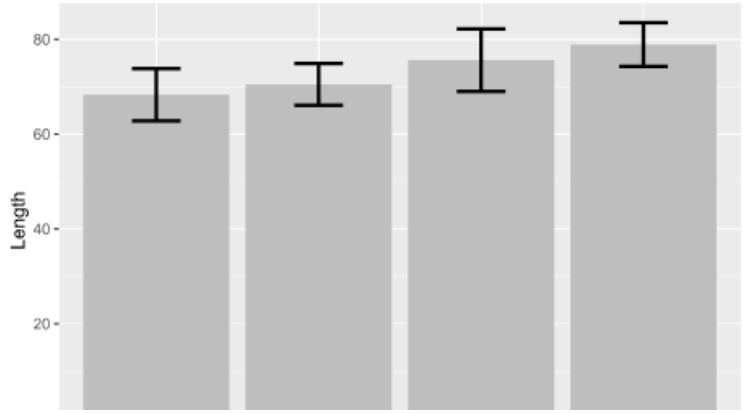
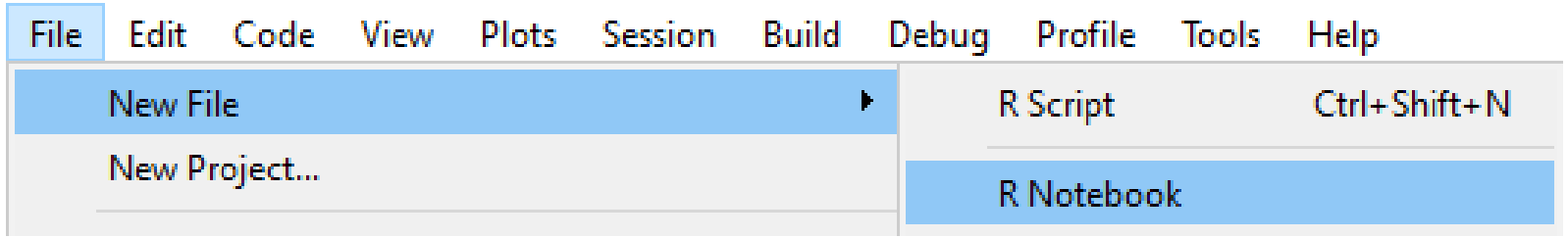
Introduction
============

This is an example of a notebook to show how they work.

```{r message=FALSE}
library(tidyverse)
```

Processing
==========

Read the data

```{r message=FALSE}
read_tsv("small_file.txt") -> small
head(small)
```
```

Sample <chr>	Length <dbl>	Category <chr>
x_1	45	A
x_2	82	B
x_3	81	C
x_4	56	D
x_5	96	A

Summarise

We're going to calculate the mean of the lengths per category

```
small %>%
  group_by(Category) %>%
  summarise(
    count=n(),
    length=mean(Length)
  )
```

Category <chr>	count <int>	length <dbl>
A	10	68.3
B	10	70.5
C	10	75.6
D	10	78.9

4 rows

Plot

```
small %>%
  ggplot(aes(x=Category, y=Length)) +
  geom_bar(stat="summary", fun="mean", fill="grey")+
  stat_summary(geom="errorbar", width=0.3, size=1, fun.data=mean_se)
```



Output

Creating a Notebook in RStudio



- You may need to install some packages (Rstudio will prompt you if you do)

- Opens a default template which you can then edit

```
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```{r}
plot(cars)
```

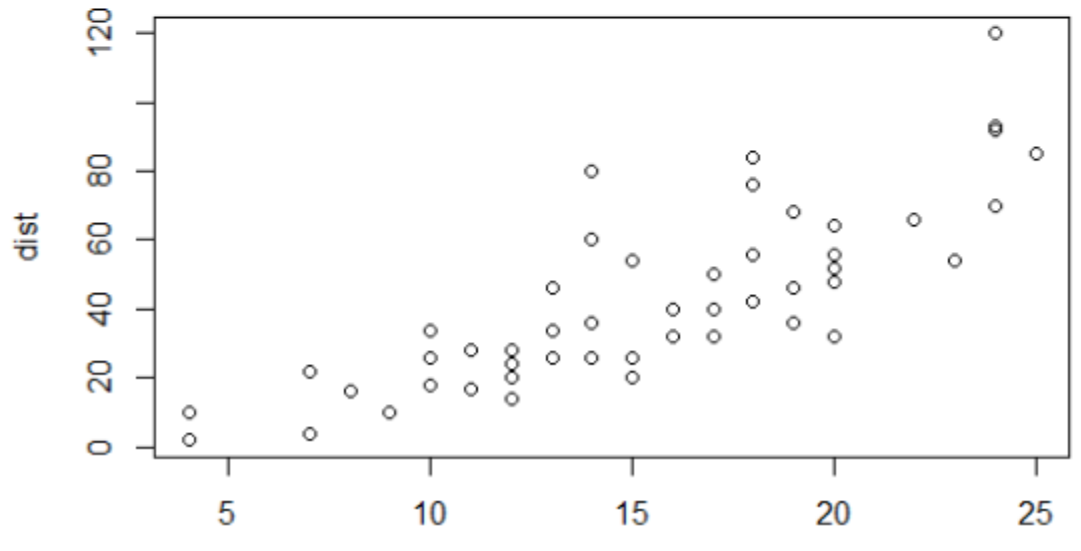
Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.
```

Notebook sections

Header

Commentary

Code

```
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

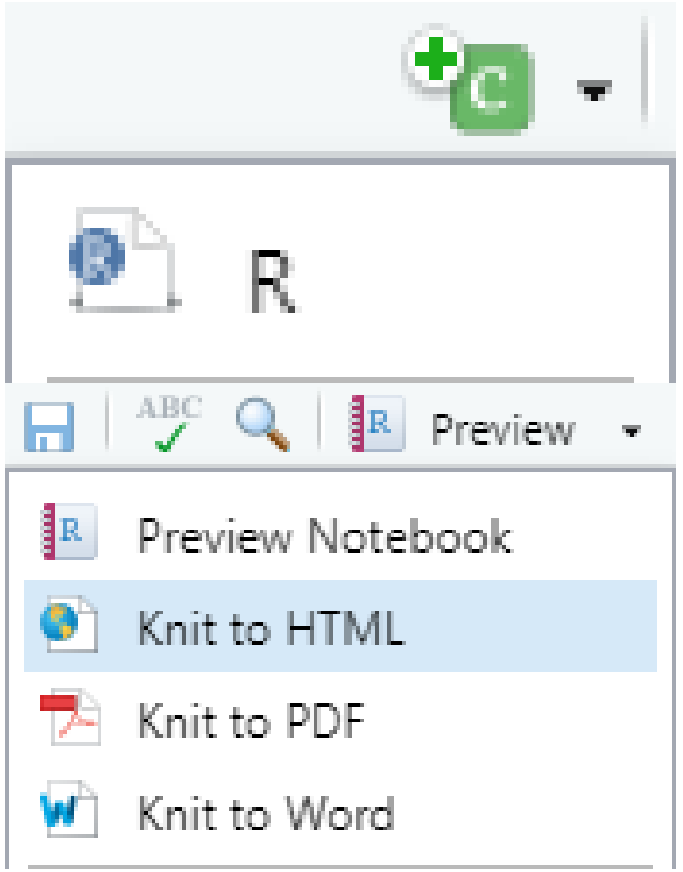
```{r}
plot(cars)
```
```



Sections are marked by special quotes

```
---
```
`---` for header

```
```{r}
```
```` ```{r} ```` for R code

Default for unquoted text is commentary

Notebook workflow

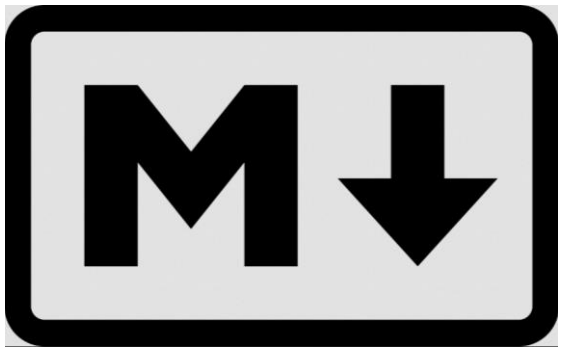
- Create new notebook document
- Save it straight away (use a .Rmd extension)
- Add commentary in Markdown format
- Add R sections using Insert > R
- Run code blocks to generate output

- Knit document to HTML / PDF / Word



Commentary sections use ‘Markdown’

- Simple markup language
- Designed to be nicely readable as plain text
- Compiles to properly formatted text
- Simple syntax



Markdown basics

- Headings

```
# Heading1
## Heading 2
### Heading 3 etc.

Heading 1
=========

Heading 2
---------
```

- Lists (need a blank line first)

```
* Bullet 1
    * Sub-bullet 1
* Bullet 2

1. Numbered 1
2. Numbered 2
```

Headings also give you navigation for your document, so they're worth using!

Data Processing with Tidyverse

Basic Structures in R

`myfunc(x, value=y)`	Runs `myfunc` using data `x` and `y`
`100 -> saveme`	Saves a value under a name
`myfunc(x,y) -> saveme`	Saves the output of `myfunc`
`saveme`	Shows the contents of `saveme`
`funca() %>% funcb()`	Passes data from `funca` to `funcb`

Reading Files

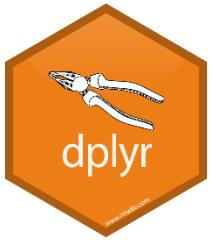
- Tidyverse functions for reading text files into data structures

```
read_delim("file.csv") -> data

read_tsv("file.tsv") -> data
```

Reading files with readr

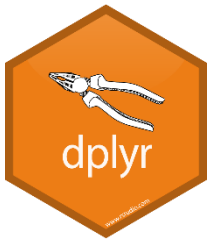

```
> read_delim("trumpton.txt") -> trumpton
Rows: 7 Columns: 5

-- Column specification ----------------
Delimiter: "\t"
chr (2): LastName, FirstName
dbl (3): Age, Weight, Height

> trumpton
# A tibble: 7 x 5
  LastName FirstName   Age Weight Height
  <chr>    <chr>     <dbl>  <dbl>  <dbl>
1 Hugh     Chris        26     90    175
2 Pew      Adam         32    102    183
3 Barney   Daniel       18     88    168
4 McGrew   Chris        48     97    155
5 Cuthbert Carl         28     91    188
6 Dibble   Liam         35     94    145
7 Grub     Doug         31     89    164
```

Tidyverse Data Processing



- `select` pick columns by name/position
- `filter` pick rows based on the data
- `arrange` sort rows

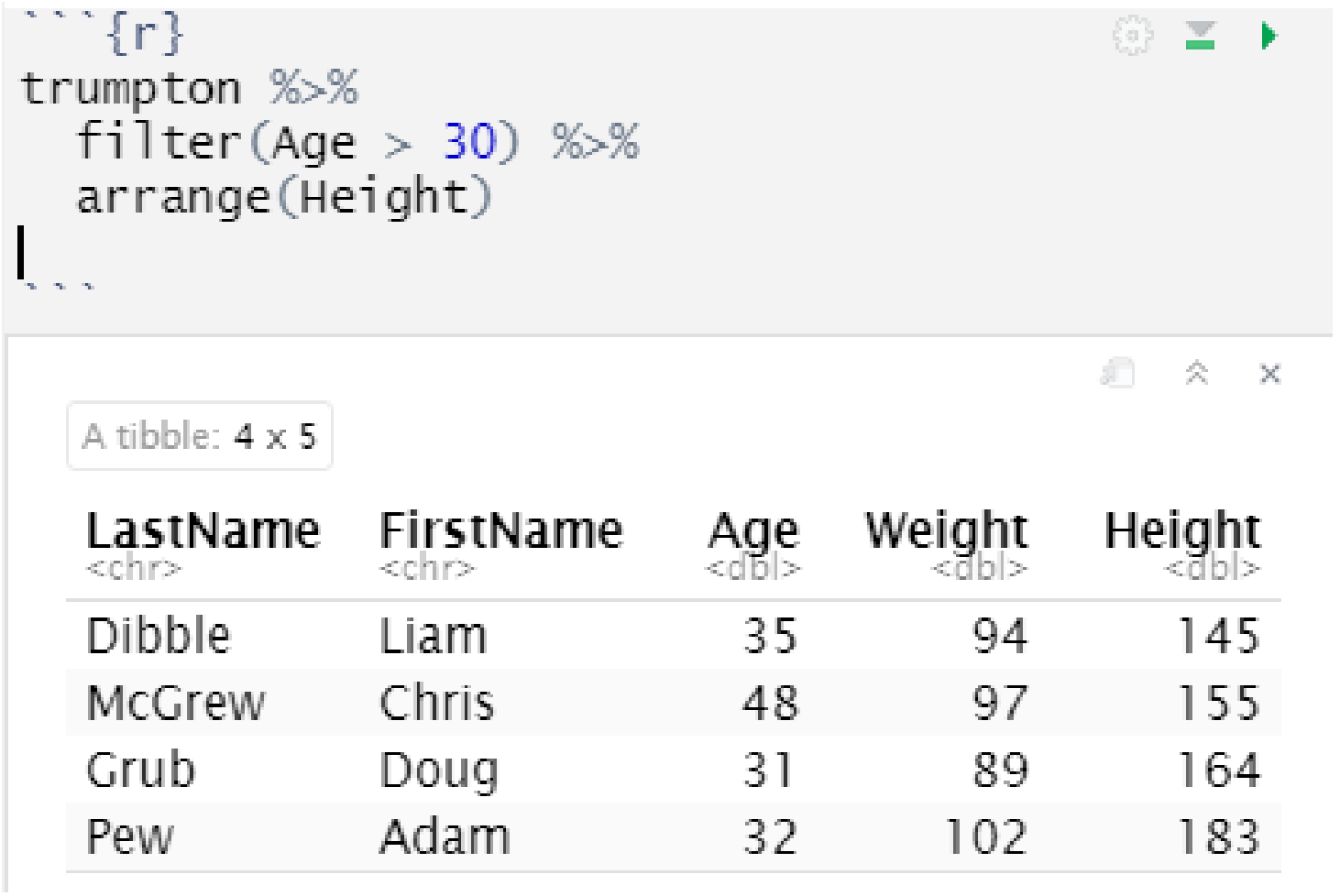
Combining multiple operations




```
trumpton %>%
  filter(Age > 30) %>%
  arrange(Height)
```

```
# A tibble: 4 x 5
  LastName FirstName   Age Weight Height
  <chr>    <chr>     <dbl>  <dbl>  <dbl>
1 Dibble   Liam         35     94    145
2 McGrew   Chris        48     97    155
3 Grub     Doug         31     89    164
4 Pew      Adam         32    102    183
```

Running R code in a notebook

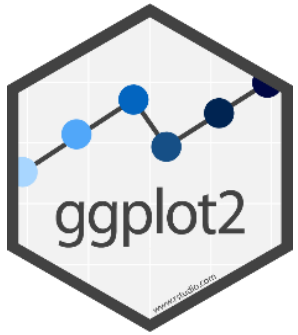


Run Button

Code Block

Inserted Output

Plotting Graphs with GGPlot



- Say what data you want to use
- Say what graph type you want to use
- Say how you want the data to affect the graph
- Plot the graph

Geometries and Aesthetics

- Geometries are types of plot

`geom_point()`	Scatterplots
`geom_jitter()`	Stripcharts
`geom_boxplot()`	Box plots
`geom_col()`	Barplots

- Aesthetics are graphical parameters in a given geometry
  - Size
  - Colour
  - Fill
  - X/Y position

Setting Aesthetics

- Aesthetic Mappings
  - A column in your data defines the value for the aesthetic
    - Height is the position on the x, Weight is the position on the y
    - Colour the graph by experimental condition
  - Set an aesthetic to a fixed value
    - Fill all bars with yellow
    - Make all of the points size 5

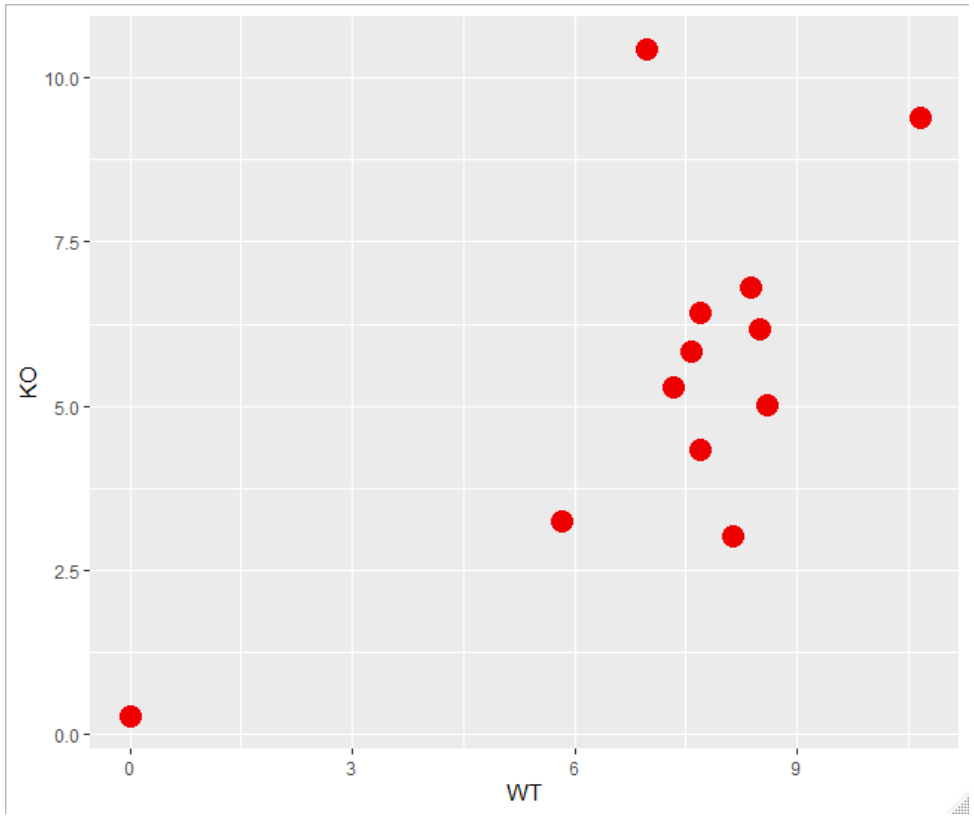
An Example GGplot

Set the data to use

```
expression %>%
  ggplot (aes(x=WT, y=KO)) +
  geom_point(color="red2", size=5)
```

Set the aesthetic mappings

Set the plot type and fixed aesthetics



Statistics in Tidyverse

- Just more functions

  `anova_test` Comparison of multiple means

  `tukey_hsd` Multiple pairwise comparisons

```
data %>% anova_test (x~y)
```

  x is a quantitative column

  y is a categorical column

  Test how well we can predict x if we know y

R / Tidyverse / GGPlot / Notebook Exercise

- Create a Notebook
- Write some commentary in markdown
- Load in a dataset
- Plot out the data
- Calculate some summaries
- Run some statistics

Babraham Bioinformatics

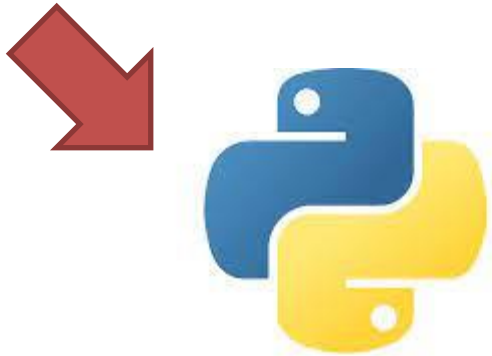
Exercise 15: Interactive Data Analysis in R

Application Development in Python

Python is a ‘scripting’ language

```
#!/usr/bin/env python

print("I am a python program")
```

VS Code



python™

```
python
python.exe
python3
python3.exe
```

https://www.python.org/

```
C:\Introduction to Python>python example.py
I am a python program
```

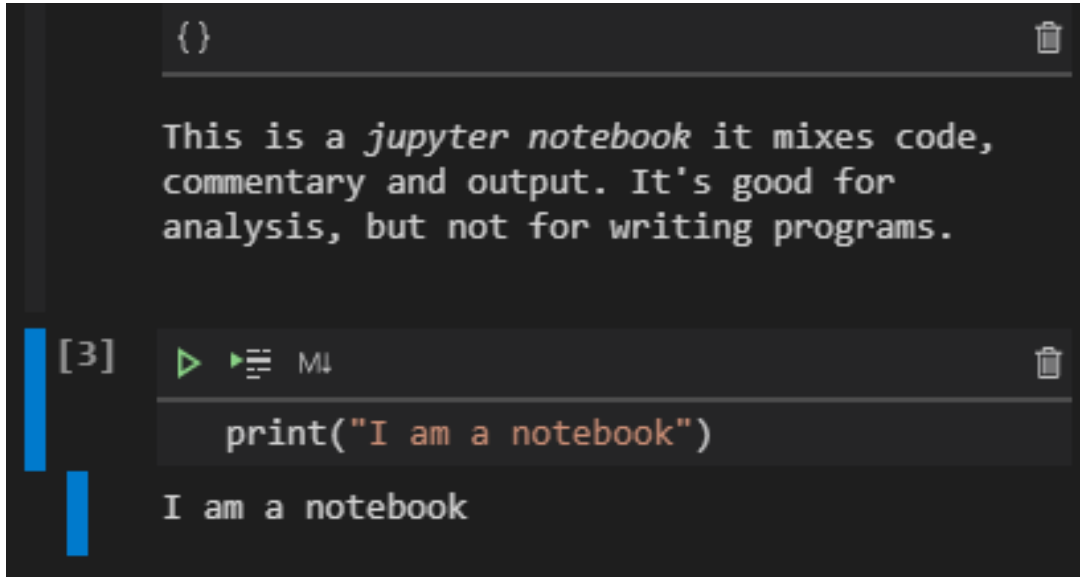
Different environments for writing python

```
#!/usr/bin/env python

print("I am a python program")
```

Scripted: code in text file, output in console

```
C:\Users\andrewss\>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020,
17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or
"license" for more information.
>>>
>>> print("I am an interactive session")
I am an interactive session
>>>
```

Interactive: code and output in console



Notebook: code, commentary and output in a single file

Python script basics

Where to find an interpreter (optional)

Series of python ‘statements’. One per line (generally). These are executed in order, from the top of the file to the bottom.

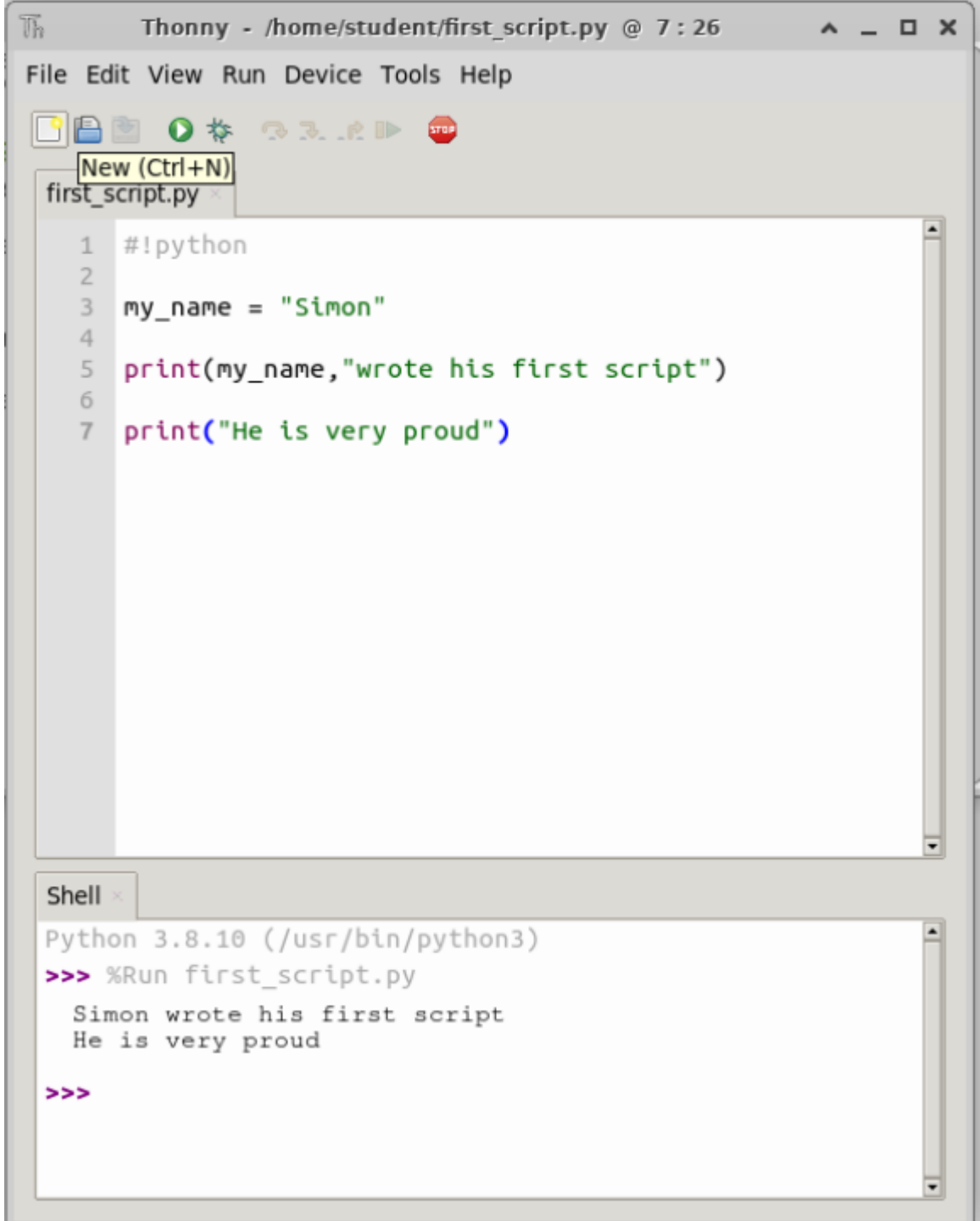
Your program finishes at the end of the file

```
#!/usr/bin/env python

my_name = "Simon"

print (my_name,"wrote his first python program")

print ("He is very proud")
```

Thonny

- Simple python editor
- Editor at the top
- Interaction at the bottom

- Write
- Save
- Run
  - Debug!



Functions vs Methods

- Functions
  - Named pieces of code. All data must be passed in to them.

```
len("Simon")
5
```

- Methods
  - Functions which are associated with a piece of data. Called via the data, you don't need to pass the data in to the method

```
"Simon".upper()
'SIMON'
```

Text manipulation

`string` — Common string operations
`re` — Regular expression operations

Data Types

`datetime` — Basic date and time types
`zoneinfo` — IANA time zone support
`calendar` — General calendar-related functions
`array` — Efficient arrays of numeric values
`copy` — Shallow and deep copy operations
`pprint` — Data pretty printer
`graphlib` — Operate with graph-like structures

Numeric and Mathematical Modules

`math` — Mathematical functions
`random` — Generate pseudo-random numbers
`statistics` — Mathematical statistics functions

File and Directory Access

`os.path` — Common pathname manipulations
`stat` — Interpreting stat() results
`tempfile` — Generate temporary files and directories
`glob` — Unix style pathname pattern expansion
`shutil` — High-level file operations

Data Persistence

`pickle` — Python object serialization
`sqlite3` — DB-API 2.0 interface for SQLite databases

Data Compression and Archiving

`gzip` — Support for gzip files
`bz2` — Support for bzip2 compression
`zipfile` — Work with ZIP archives
`csv` — CSV File Reading and Writing

Generic Operating System Services

`os` — Miscellaneous operating system interfaces
`io` — Core tools for working with streams
`time` — Time access and conversions
`argparse` — Parser for command-line options

Internet Data Handling

`email` — An email and MIME handling package
`json` — JSON encoder and decoder

Graphical User Interfaces with Tk

`tkinter` — Python interface to Tcl/Tk

Software Packaging and Distribution

`distutils` — Building and installing Python modules
`venv` — Creation of virtual environments

Packages we're going to use

- `requests` - fetches data from a web resource and saves it into your program

- `biopython` - lots of functionality related to bioinformatics. We're using it to parse a sequence file, but there's lots of stuff in there

Using functions from packages

Use functions via the package

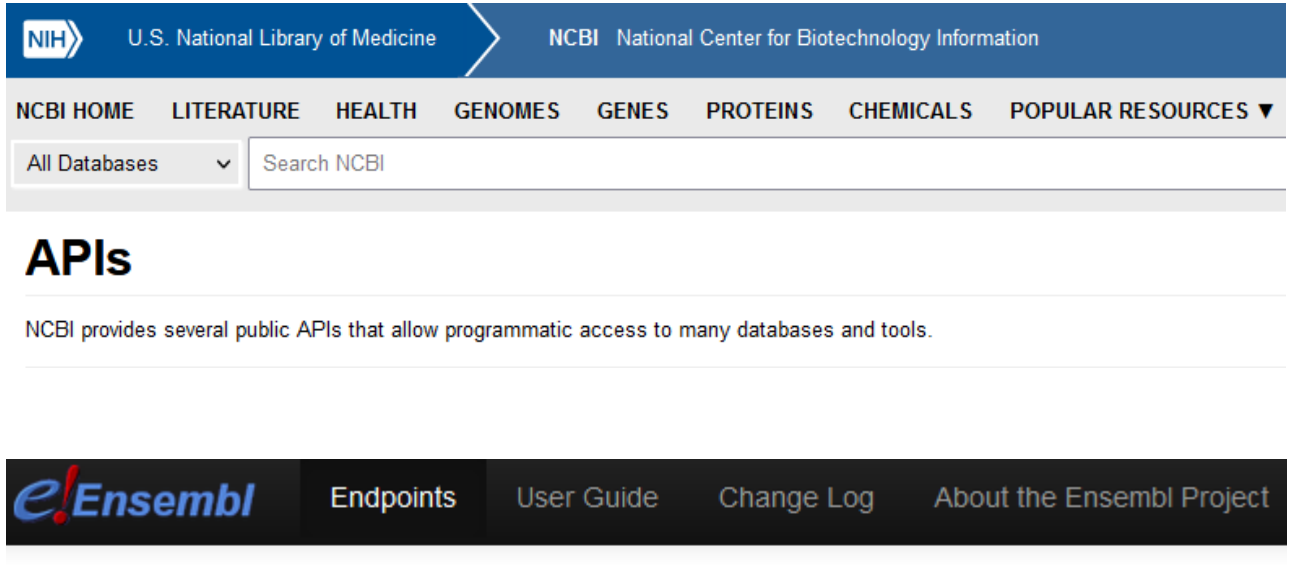
```
import math
math.sqrt(10)

3.162277
```

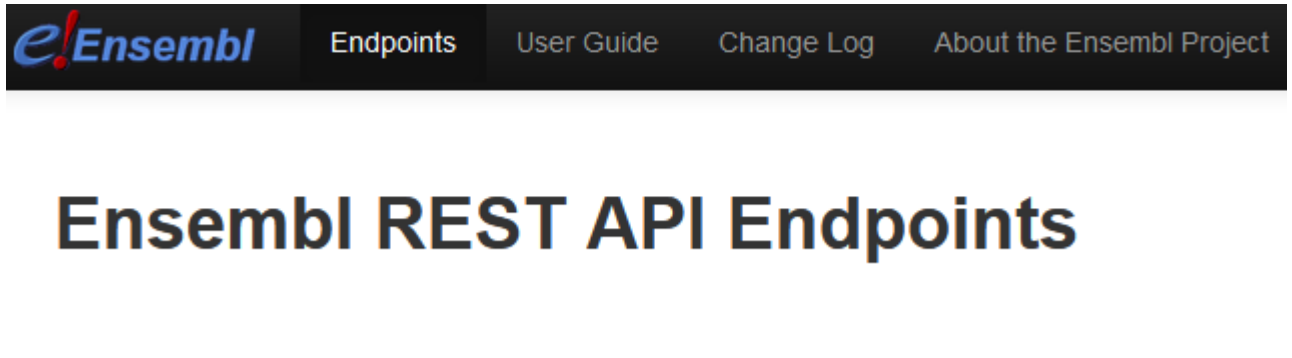
Import individual functions

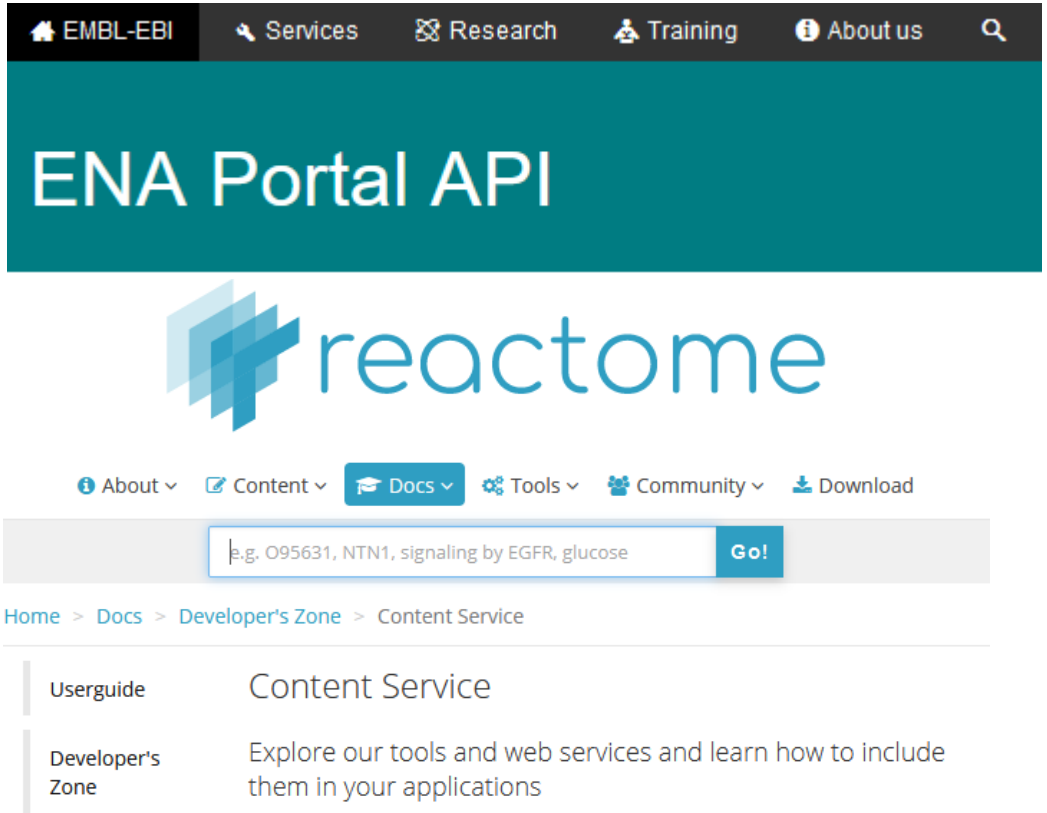
```
from math import sqrt
sqrt(10)

3.162277
```

APIs

- Lots of resources make their data available programmatically
- An API describes how to query and access the data







Using APIs

e!Ensembl | Endpoints | User Guide | Change Log | About the Ensembl Project | Contact Ensembl | Privacy Notice

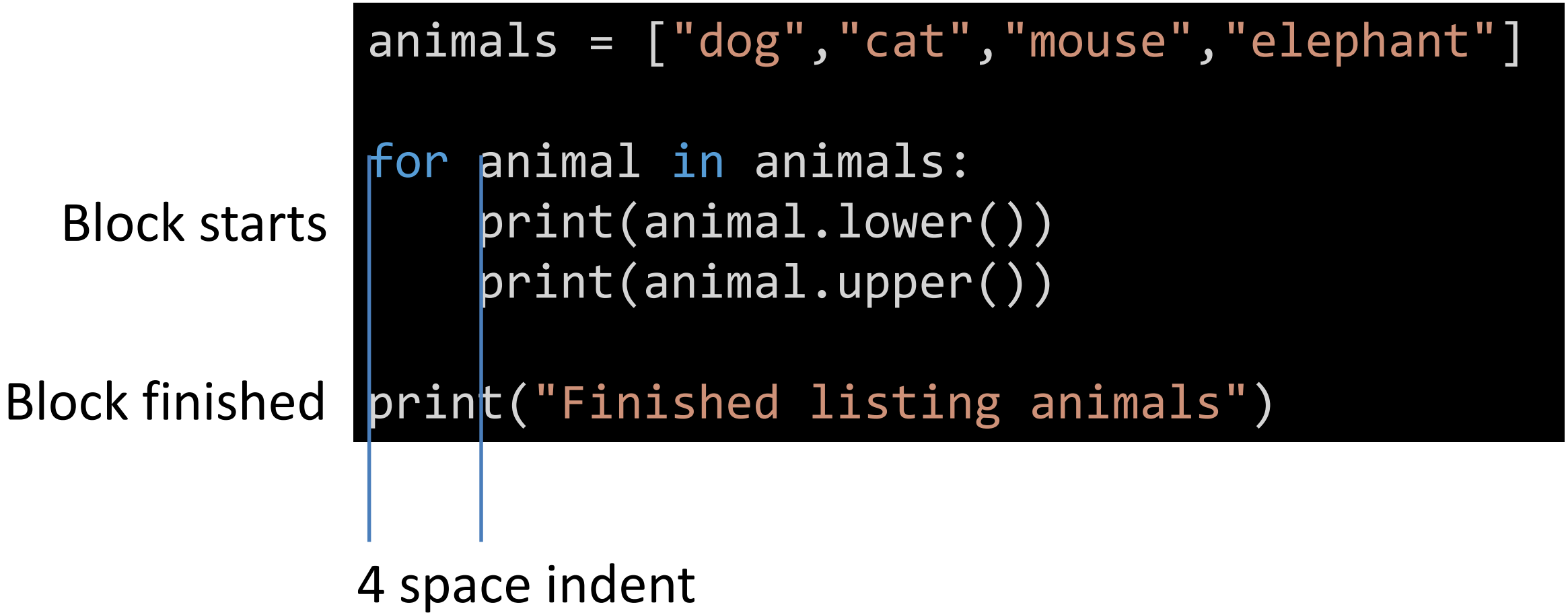
Example Requests

/xrefs/symbol/homo_sapiens/BRCA2?content-type=application/json

Example output | Perl | Python2 | **Python3** | Ruby | Java | R | Curl | Wget

```
1.  import requests, sys
2.
3.  server = "https://rest.ensembl.org"
4.  ext = "/xrefs/symbol/homo_sapiens/BRCA2?"
5.
6.  r = requests.get(server+ext, headers={ "Content-Type" : "application/json"})
7.
8.  if not r.ok:
9.    r.raise_for_status()
10.   sys.exit()
11.
12. decoded = r.json()
13. print(repr(decoded))
14.
```

Code Blocks in Python



Exercise 16
Application Development in Python Exercise

- Ask the user for the name of a gene
- Use the Ensembl API to get the Ensembl ID for that gene
- Use the Ensembl API to get the transcript sequences
- Use BioPython to parse the sequences

- Write out a list of the transcripts and their lengths