Funded by the European Union - Project 101079467

grinn aqua

green innovation strategies for animal health management: towards sustainable aquaculture

Increasing researcher´s skills on the use of omics tools workshop

16th and 17th of March 2023
Auditório Infante D. Henrique
Marina de Leça, Leça da Palmeira (Portugal)

Free but limited to 30 attendants

ciimar
Centro Interdisciplinar de Investigação Marinha e Ambiental

UNIVERSITAS BERGENSIS

ROSLIN INNOVATION CENTRE

CSIC
Consejo Superior de Investigaciones Científicas

GRINNAQUA AIMS TO STRENGTHEN THE PERFORMANCE OF CIIMAR IN AQUACULTURE AND ANIMAL HEALTH.

The synergy between Research institutions that integrate the GRINNAQUA consortium will boost the Scientific excellence and innovation capacity at CIIMAR, which can then be transferred to the portuguese aquaculture sector.

grinnaqua

green innovation strategies for animal health management: towards sustainable aquaculture

Increasing researcher´s skills on the use of omics tools workshop

Thursday 16th March		
8:30-8:45	Welcome	Tim Bean & Diego Robledo
8:45-9:15	Genetic and genomic technologies in aquaculture research	Tim Bean & Diego Robledo
9:15-9:30	RNA-Seq workshop overview	Diego Robledo
9:30-9:45	RNA-Seq – definition and technologies	Tim Regan
9:45-10:30	Sample types, RNA extraction and library preparation	Tim Bean
10:30-11:00	*Coffee break*	
11:00-11:30	Introduction to linux *	Clemence Fraslin
11:30-12:30	Quality control of RNA-Seq raw data *	Diego Robledo
12:30-14:00	*Lunch*	
14:00-14:30	Alignment and quantification pipelines	Diego Robledo
14:30-15 :30	Quantification using Kallisto *	Diego Robledo
15:30-16:00	*Coffee break*	
16:00-17:00	Alignment using STAR *	Diego Robledo
17:00-17:30	Visualization using IGV *	Tim Regan
17:30-19:00	Open session and reception	

ciimar Centro Interdisciplinar de Investigação Marinha e Ambiental

UNIVERSITAS BERGENSIS

ROSLIN INNOVATION CENTRE

CSIC Consejo Superior de Investigaciones Científicas

Funded by the European Union - Project 101079467

grinnaqua

green innovation strategies
for animal health management:
towards sustainable aquaculture

Increasing researcher´s skills on the use of omics tools workshop

Friday 17th March		
9:00-9:15	Recap: the workshop so far	Tim Bean & Diego Robledo
9:15-9:45	Experimental design	Tim Bean
9:45-10:15	Introduction to R *	Jennifer Nascimento Schulze
10:15-11:00	Differential expression analysis *	Diego Robledo
11:00-11:30	*Coffee break*	
11:30-12:30	Visualization *	Diego Robledo
12:30-14:00	*Lunch*	
14:00-15:30	Beyond differential expression	Diego Robledo
15:30-17:00	*Farewell drinks and impressions about workshop*	

* denotes practical work – the participants are expected to bring their own laptops

Registration form:

https://docs.google.com/forms/d/e/1FAIpQLSfQO_JhEDdXjxkqju3tgAJGQEIYJRLz2Jai5pkdf6G1t9zYQA/viewform?usp=sf_link

Meals and coffee breaks are included

ciimar Centro Interdisciplinar de Investigação Marinha e Ambiental

UNIVERSITAS BERGENSIS

ROSLIN INNOVATION CENTRE

CSIC Consejo Superior de Investigaciones Científicas

Funded by the European Union - Project 101079467

grinnaqua

green innovation strategies
for animal health management:
towards sustainable aquaculture

Increasing researcher´s skills on the use of omics tools workshop

Training team:

Diego Robledo. Group leader at Roslin Institute with interest on sex determination in turbot as well as other traits and species, generally developing genomic resources and studying immune responses to pathogens.

Tim Bean. Group leader at Roslin Institute working to improve the productivity and success of bivalve aquaculture in the UK by using the most appropriate technologies to deal with issues such as disease and environmental health.

Clemence Fraslin. Postdoctoral researcher at Roslin Institute working to implementation of genomic selection to improve rainbow trout resistance to columnaris disease

Tim Regan. Postdoctoral researcher at Roslin Institute working on the application of genome editing, screening assays and metagenomics to better understand host immunity and disease susceptibility - resistance in humans, bees and various terrestrial livestock species.

Jennifer Nascimento. Postdoctoral researcher at Roslin Institute. Oceanographer interested in ecophysiology and genomics of marine organisms. Application of statistics and bioinformatic approaches to investigate the implications of climate change in marine bivalves.

ciimar
Centro Interdisciplinar de Investigação Marinha e Ambiental

UNIVERSITAS BERGENSIS

ROSLIN INNOVATION CENTRE

CSIC
Consejo Superior de Investigaciones Científicas

Funded by the European Union - Project 101079467

grinnaqua

green innovation strategies
for animal health management:
towards sustainable aquaculture

Where?

Auditorio Infante Dom Henrique, Leça da Palmeira

How to arrive?

From CIIMAR (Rua do Godinho stop) – Bus 105,106 and 507 (around 20 min.)

Funded by the European Union - Project 101079467

ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

GRINNAQUA – GENOMICS WORKSHOP




Porto, 16th & 17th March 2023

ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

GRINNAQUA – a Twinning Network funded by the EU Horizon Widera programme

ciimar

- Functional diets for improved immune response (SC)
- Fish health status evaluation (SC)

INIA Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

CSIC Consejo Superior de Investigaciones Científicas

- Vaccine development (SC)
- Characterization of immune cells (SC)
- High Security facilities organization (ADM/SC)

UNIVERSITY OF BERGEN

- Disease impact minimization (SC)
- Host-parasite interaction (SC)
- Centre organization (ADM)

ROSLIN

- Genetic improvement (SC)
- Breeding programs (SC)
- Project management services (ADM)

ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Workshop – Increasing researcher's skills on the use of omics tools

Thursday 16th March		
8:30-8:45	Welcome	Tim Bean & Diego Robledo
8:45-9:15	Genetic and genomic technologies in aquaculture research	Tim Bean & Diego Robledo
9:15-9:30	RNA-Seq workshop overview	Diego Robledo
9:30-9:45	RNA-Seq – definition and technologies	Tim Regan
9:45-10:30	Sample types, RNA extraction and library preparation	Tim Bean
10:30-11:00	*Coffee break*	
11:00-11:30	Introduction to linux *	Clemence Fraslin
11:30-12:30	Quality control of RNA-Seq raw data *	Diego Robledo
12:30-14:00	*Lunch*	
14:00-14:30	Alignment and quantification pipelines	Diego Robledo
14:30-15 :30	Quantification using Kallisto *	Diego Robledo
15:30-16:00	*Coffee break*	
16:00-17:00	Alignment using STAR *	Diego Robledo
17:00-17:30	Visualization using IGV *	Tim Regan
17:30-19:00	Open session and reception	

Friday 17th March		
9:00-9:15	Recap: the workshop so far	Tim Bean & Diego Robledo
9:15-9:45	Experimental design	Tim Bean
9:45-10:15	Introduction to R *	Jennifer Nascimento Schulze
10:15-11:00	Differential expression analysis *	Diego Robledo
11:00-11:30	*Coffee break*	
11:30-12:30	Visualization *	Diego Robledo
12:30-14:00	*Lunch*	
14:00-15:30	Beyond differential expression	Diego Robledo
15:30-17:00	*Farewell drinks and impressions about workshop*	

* denotes practical work – the participants are expected to bring their own laptops

Workshop – Increasing researcher's skills on the use of omics tools

Training team:

Diego Robledo. Group leader at Roslin Institute with interest on sex determination in turbot as well as other traits and species, generally developing genomic resources and studying immune responses to pathogens.

Tim Bean. Group leader at Roslin Institute working to improve the productivity and success of bivalve aquaculture in the UK by using the most appropriate technologies to deal with issues such as disease and environmental health.

Clemence Fraslin. Postdoctoral researcher at Roslin Institute working to implementation of genomic selection to improve rainbow trout resistance to columnaris disease

Tim Regan. Postdoctoral researcher at Roslin Institute working on the application of genome editing, screening assays and metagenomics to better understand host immunity and disease susceptibility - resistance in humans, bees and various terrestrial livestock species.

Jennifer Nascimento. Postdoctoral researcher at Roslin Institute. Oceanographer interested in ecophysiology and genomics of marine organisms. Application of statistics and bioinformatic approaches to investigate the implications of climate change in marine bivalves.

ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Roslin aquaculture - An expanding team focussing on Aquaculture Genetics & Health

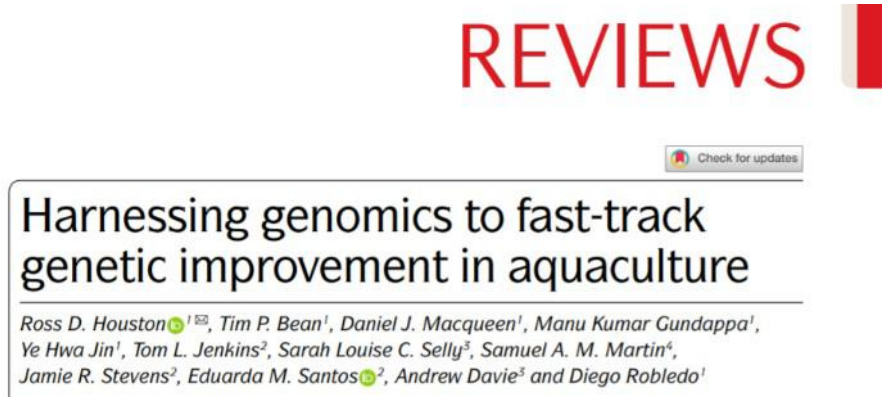
- **4** Group leaders
- **14** Postdoctoral research fellows
- **15** PhD students
- **12** Visitors in 2022

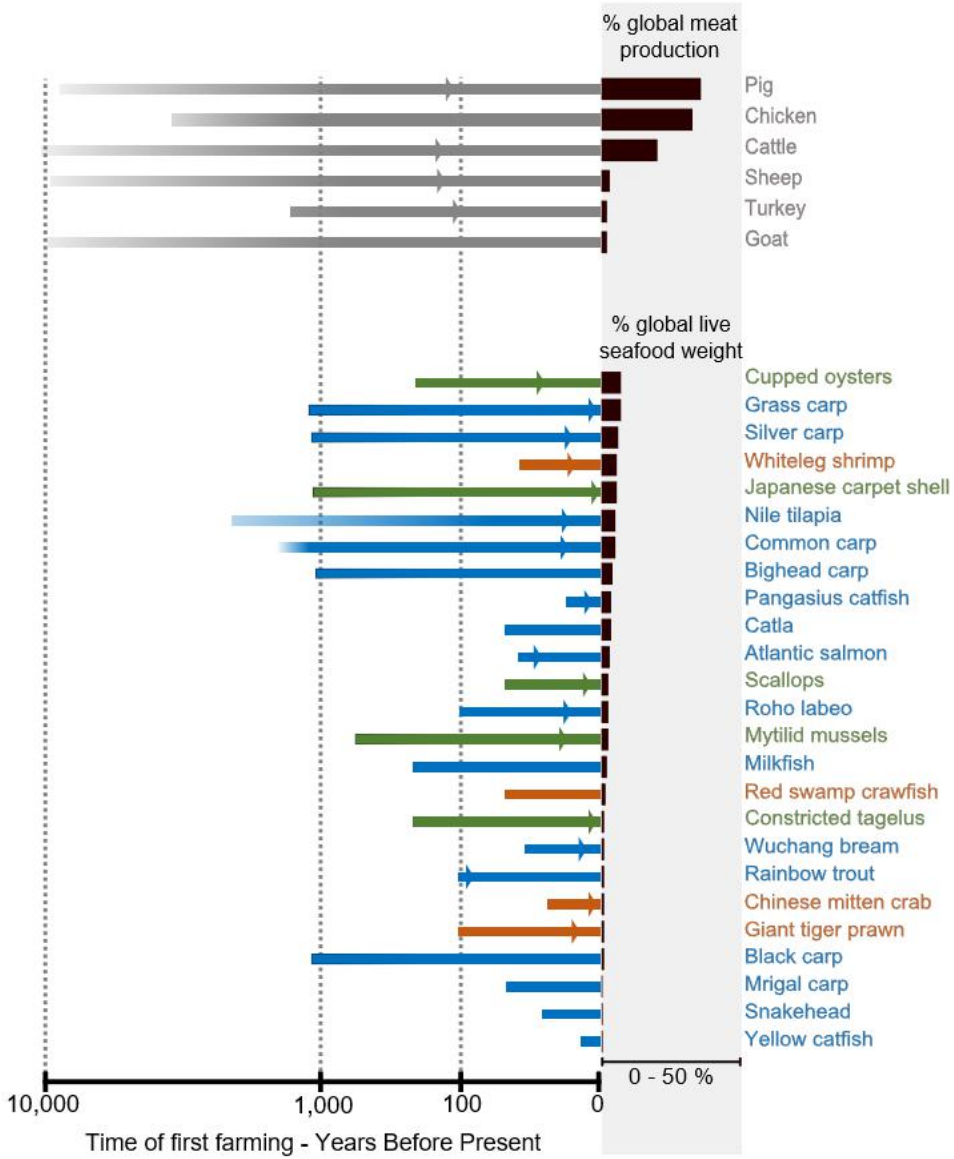
Genetics in Aquaculture

- **Domestication of aquaculture species is recent and ongoing**
  - Huge potential for genetic improvement diverse species
  - Major but under recognised role in sustainable expansion

- **Aquaculture will soon undergo a genetics revolution**
  - Uptake of selective breeding and biotechnologies
  - Enabled by data-driven genomics and phenotyping

- **Newly farmed species benefit from sharing of technology**





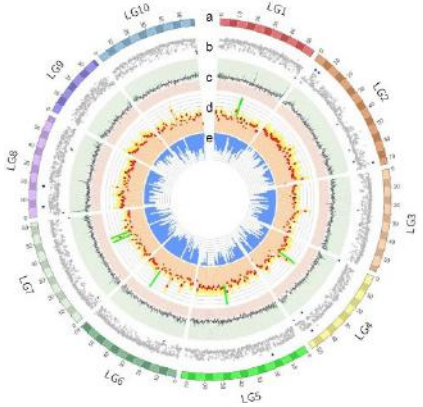
ROSLIN

grinnaqua

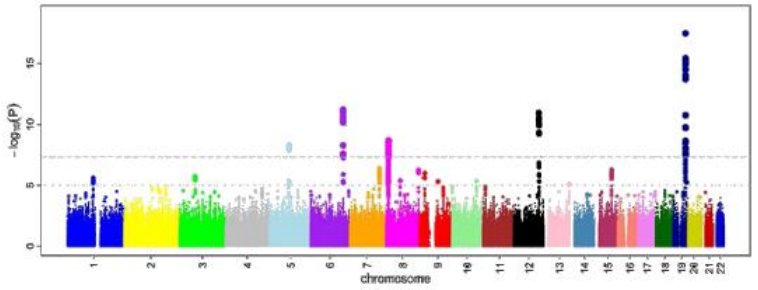
THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Full spectrum of genetic and genomics research

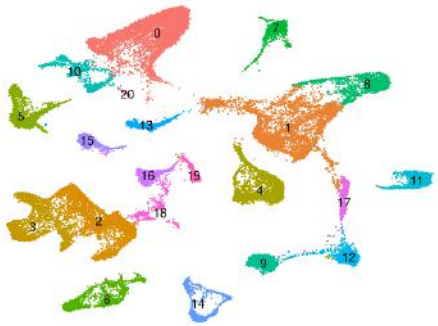
Genome assemblies



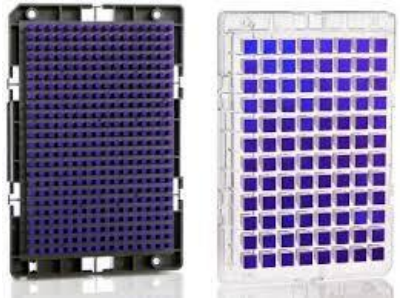
Association analyses



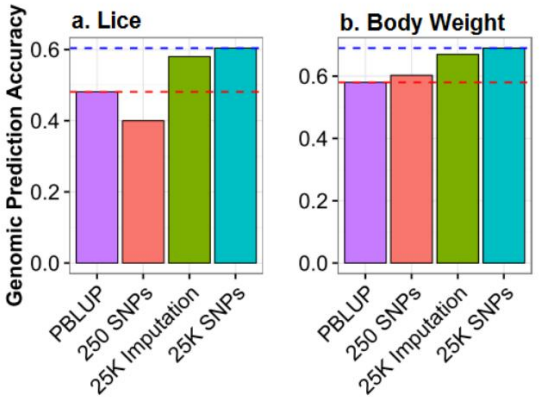
Functional genomics



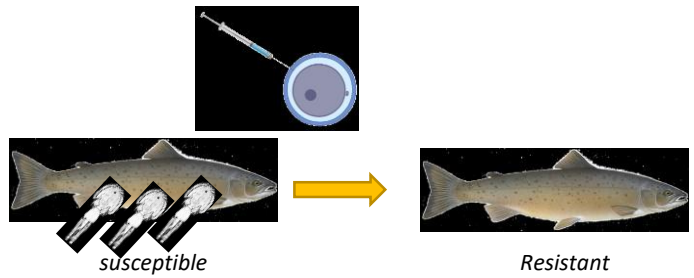
Genetic tools



Improve selection



Genome editing



ROSLIN

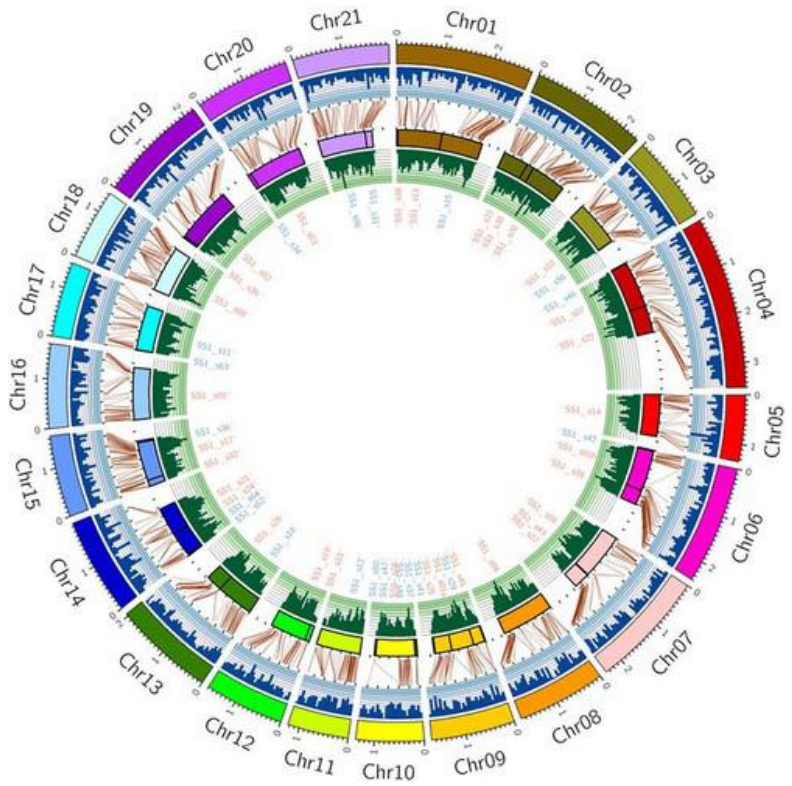
grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

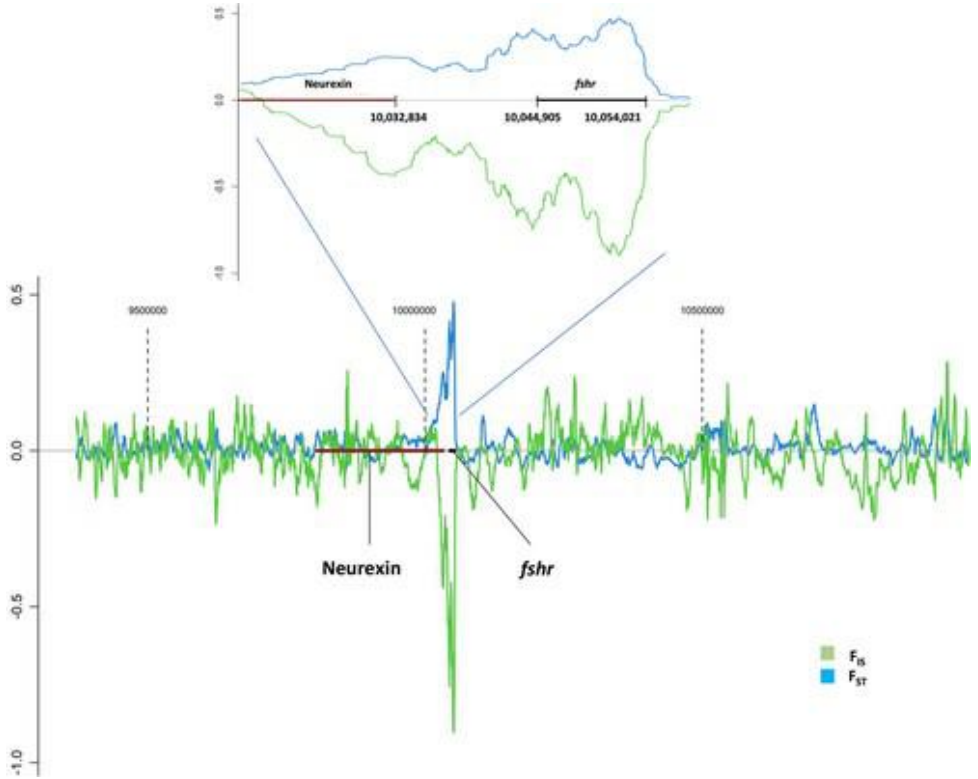
Genome assemblies

Highly-contiguous reference genomes – *Solea senegalensis*

Linkage map vs genome



Males vs females – sex determination via FSHR



Genetic tools – diverse applications

60K multi-species blue mussel array

- Platform developed for four species: *Mytilus edulis*, *M. galloprovincialis*, *M. trossulus* and *M. chilensis*

- ~20 K SNPs genotype each of the species in the array (shared between species) and ~1 K SNPs exclusive to each of the species

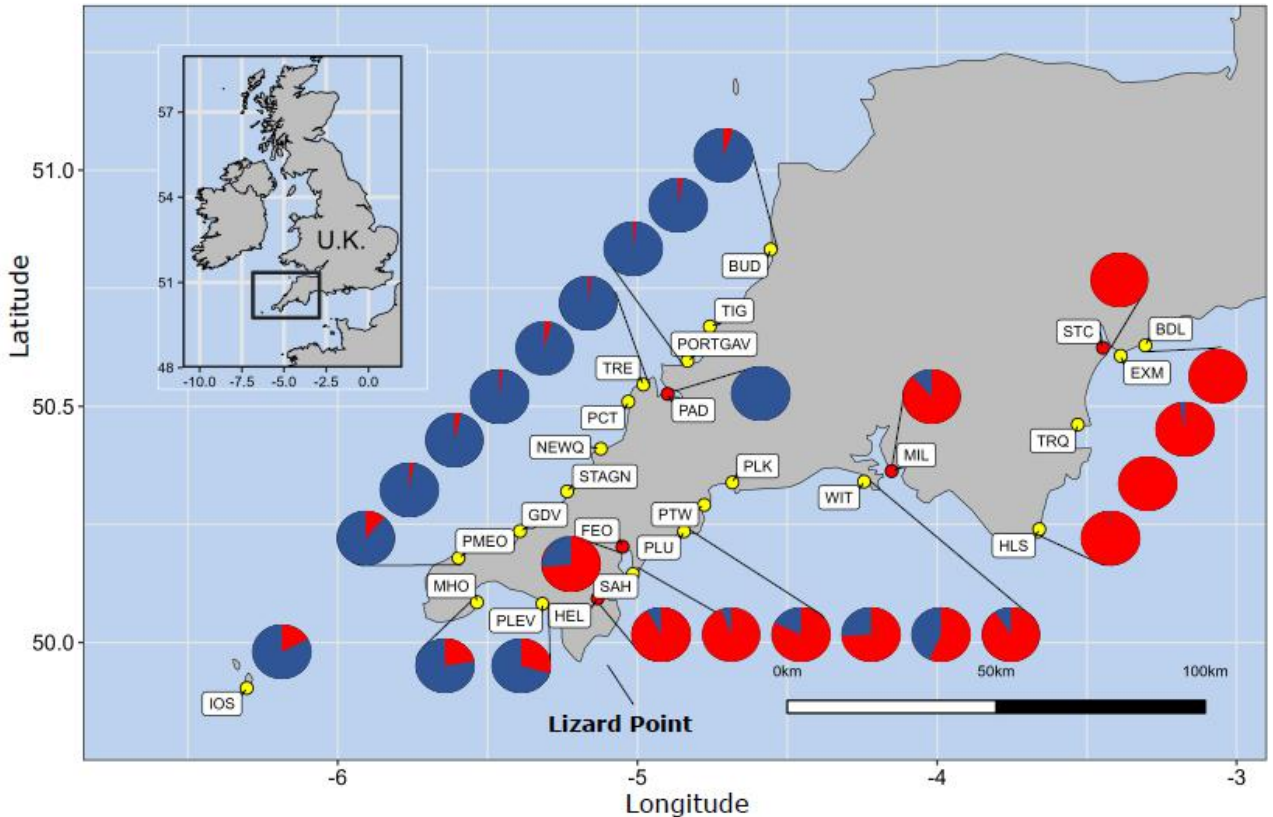
ROSLIN

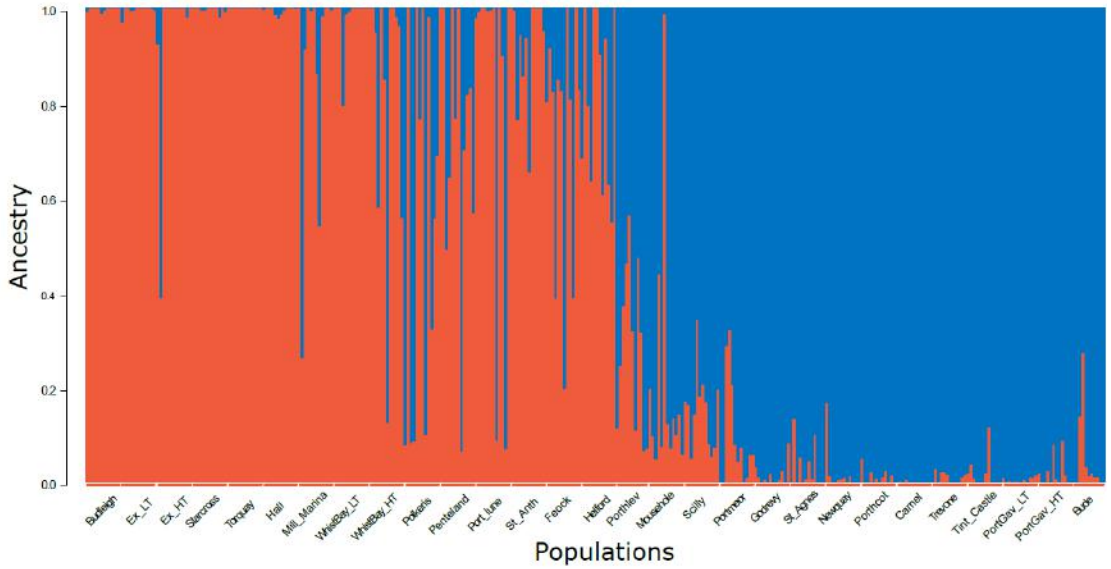
grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Genetic tools – Investigating population structure of natural mussel beds in England

- South west: hybrid zone between two species *M. edulis* and *M. galloprovincialis*
- 346 samples genotyped with 16,732 SNPs with Sample CR > 90% and marker CR > 95%





ROSLIN

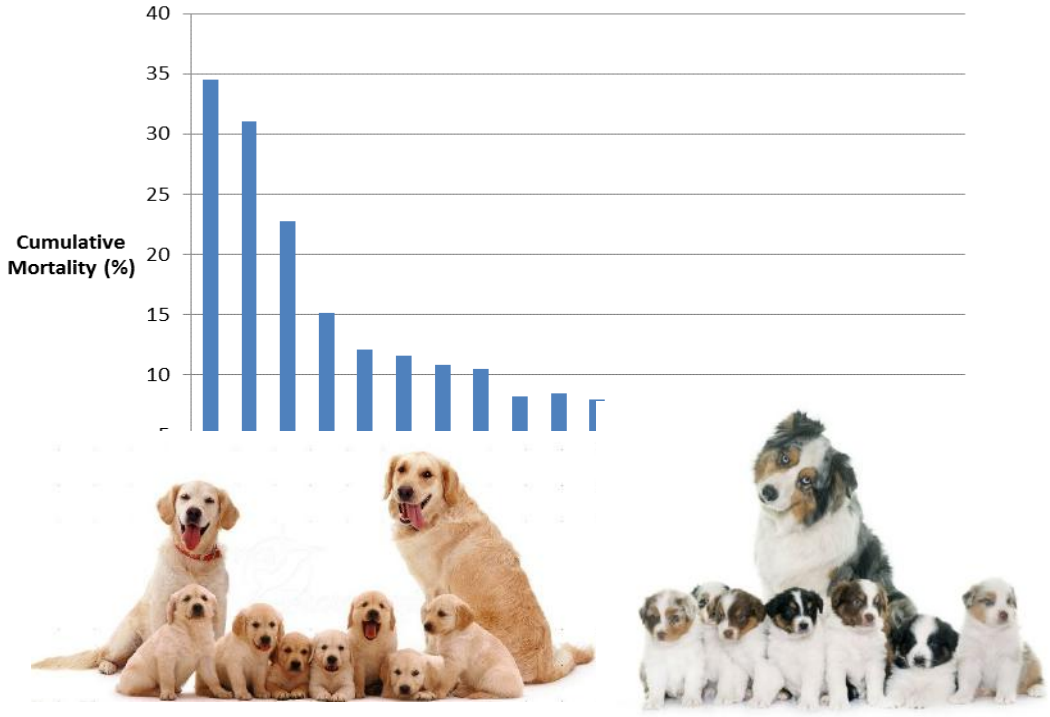
grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

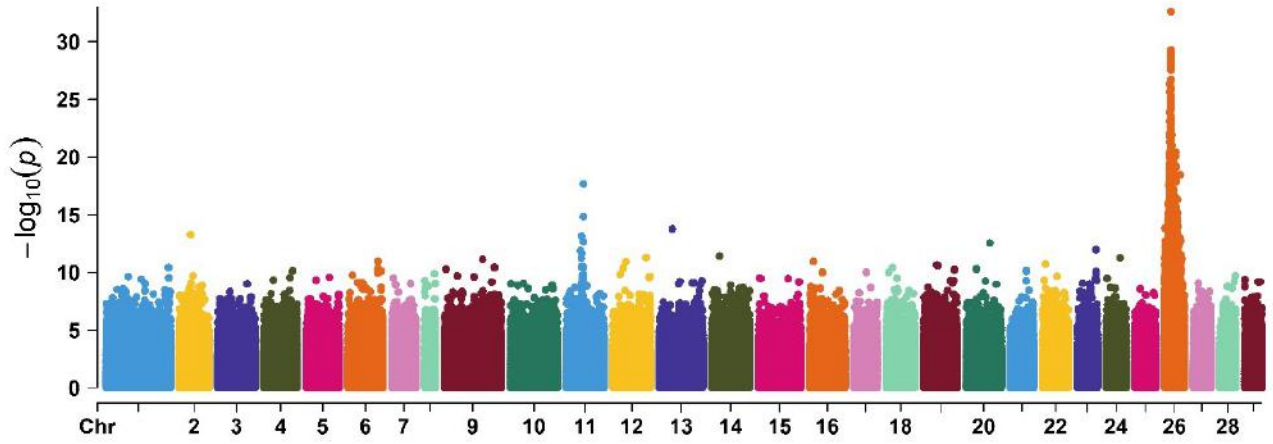
Association analysis

The curious case of resistance to IPNV in Atlantic salmon

Differences between families: genetic variation



GWAS & Manhattan plot: regions of the genome that explain those differences



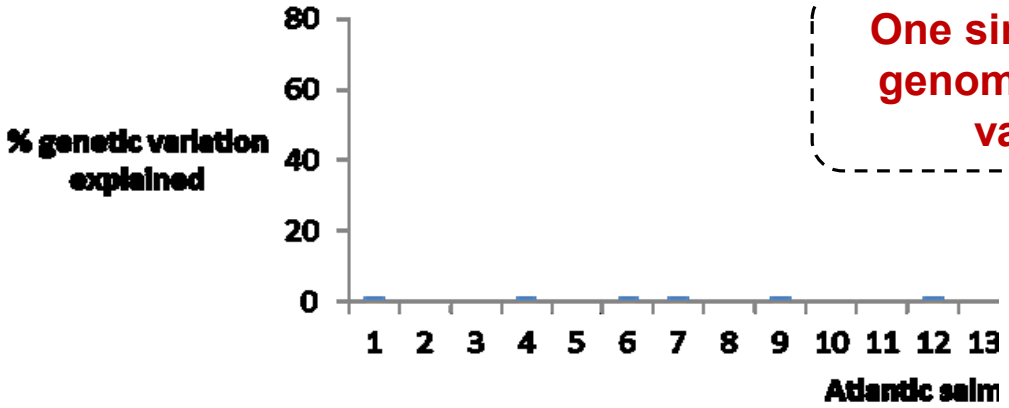
ROSLIN

grinnaqua

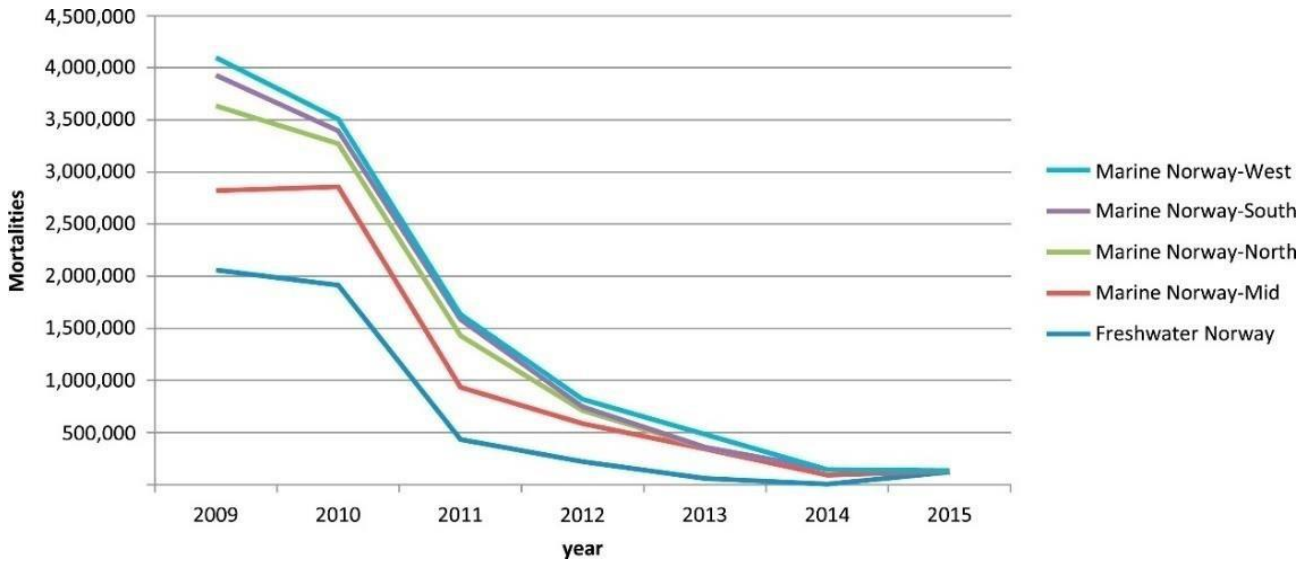
THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Association analysis

The curious case of resistance to IPNV in Atlantic salmon



One single "gene" in the salmon genome explains almost all the variation in resistance



ROSLIN

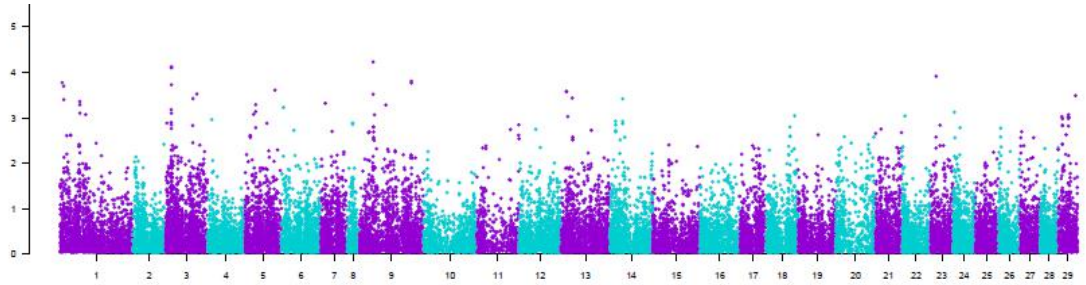
grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

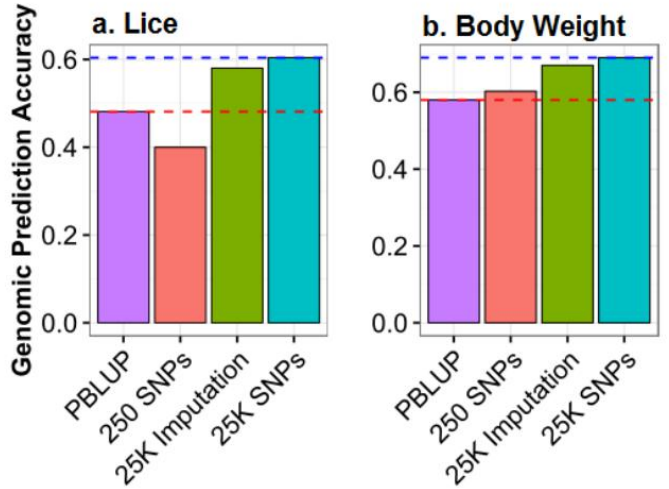
Improve selection

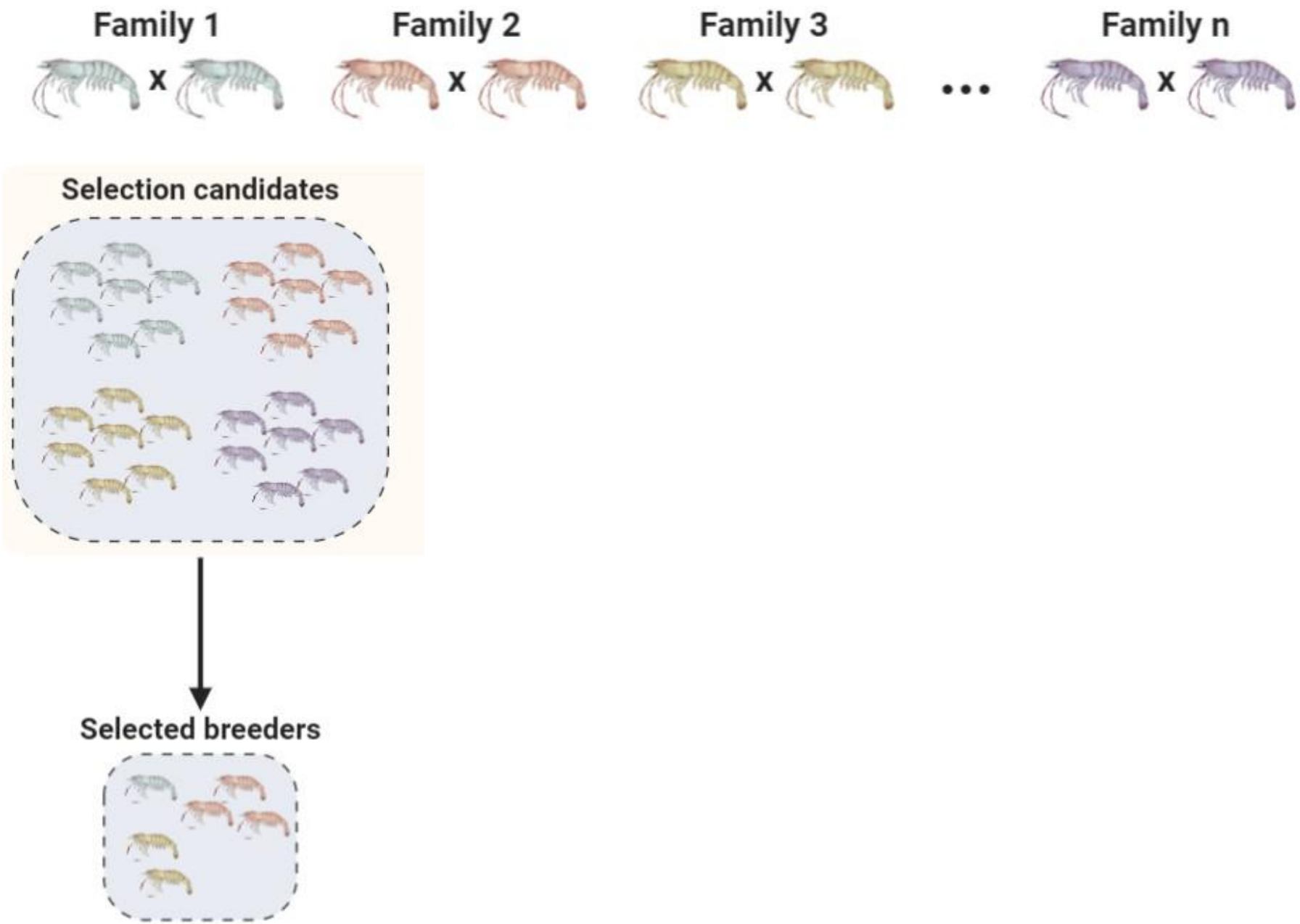
Most traits are polygenic

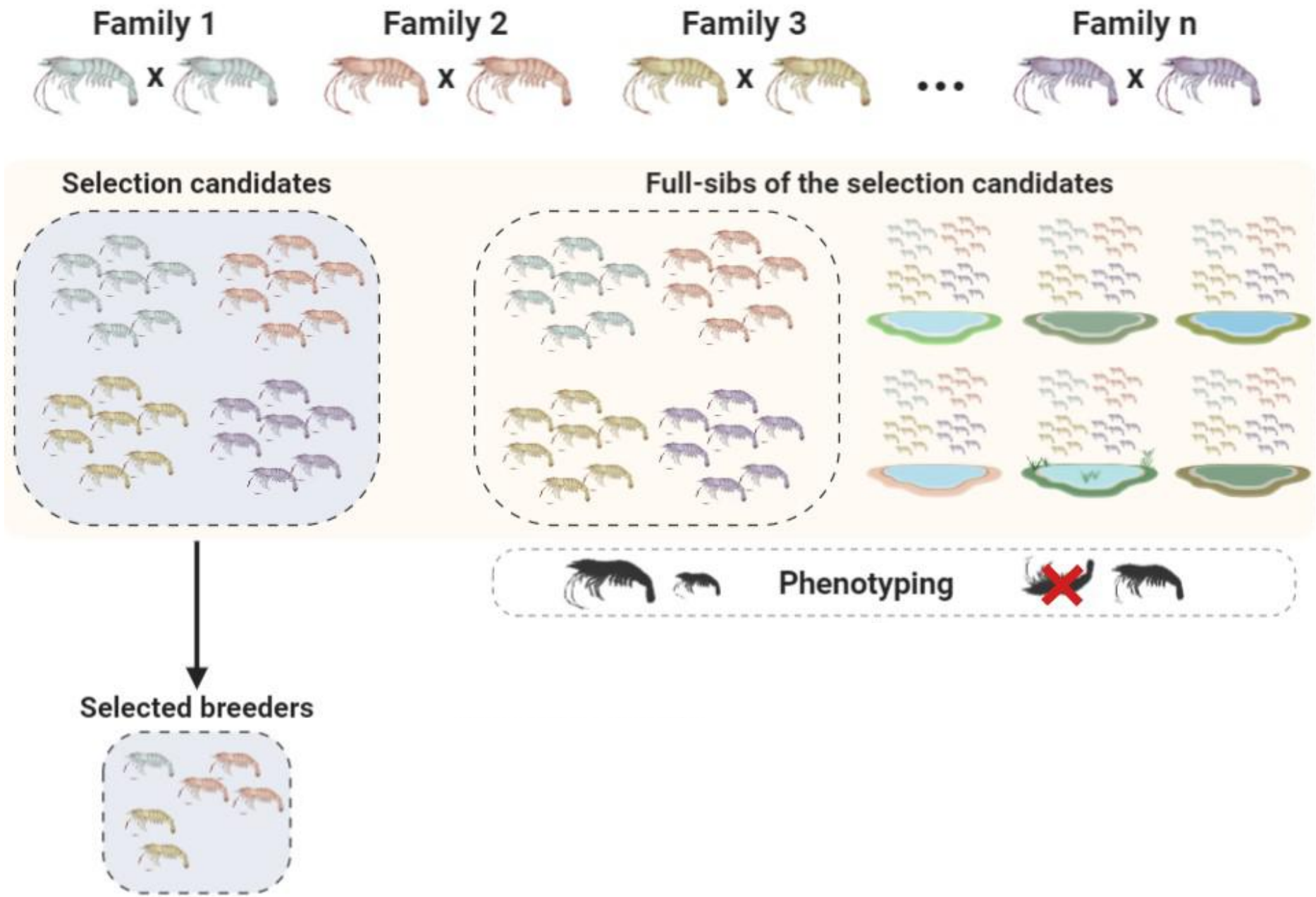
Sea lice resistance in Atlantic salmon

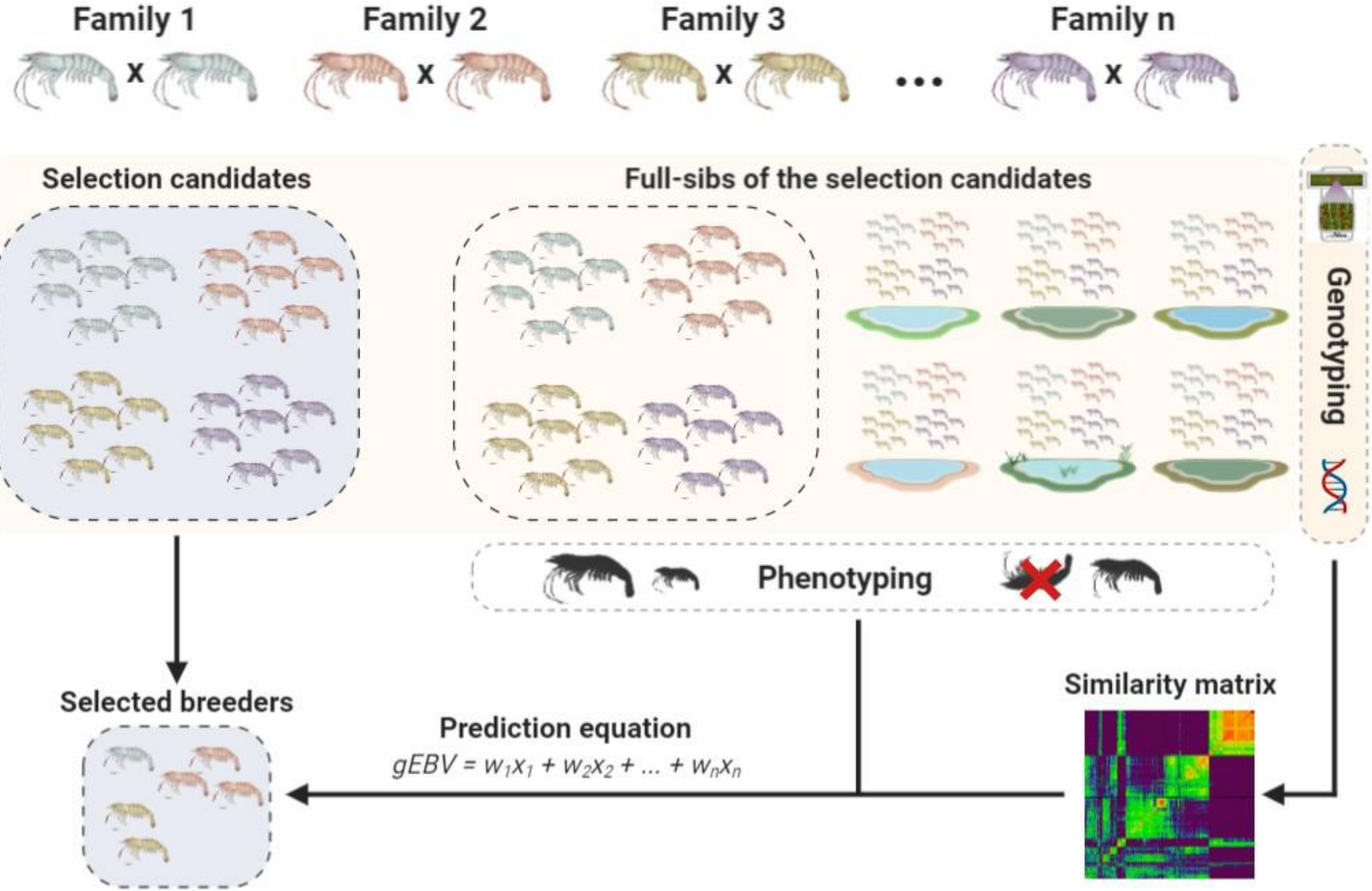


BLUP approaches




Family 1
Family 2
Family 3
Family n
Selection candidates
Selected breeders


Family 1
Family 2
Family 3
Family n
Selection candidates
Full-sibs of the selection candidates
Phenotyping
Selected breeders


Family 1
Family 2
Family 3
Family n
Selection candidates
Full-sibs of the selection candidates
Genotyping
Phenotyping
Selected breeders
Prediction equation
gEBV = w1x1 + w2x2 + ... + wnxn
Similarity matrix

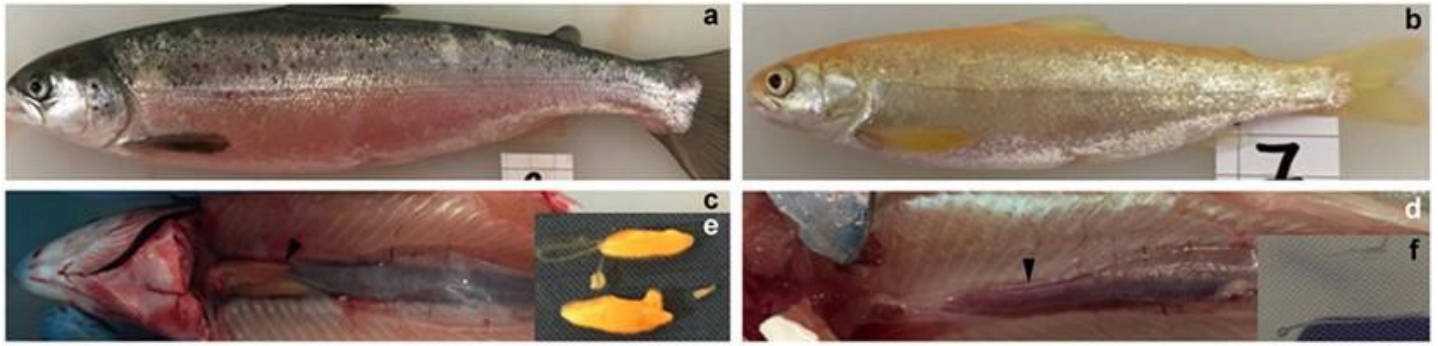
ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

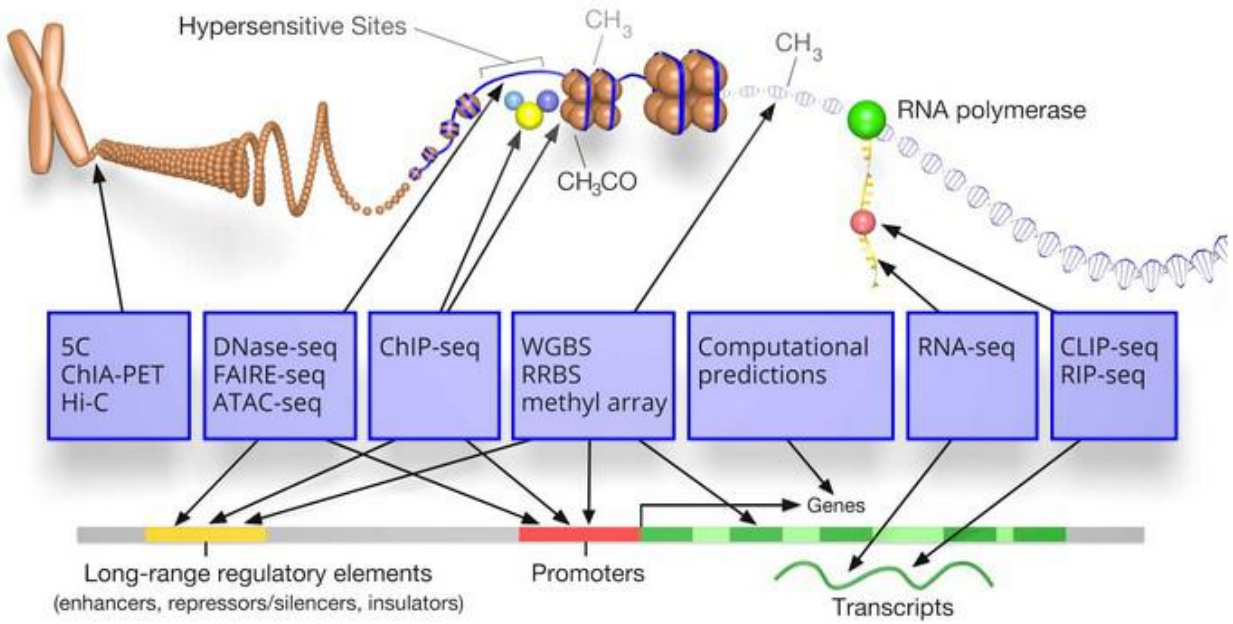
Genome editing

CRISPR/Cas9 successfully applied to salmonids, carp, tilapia, oyster, etc.

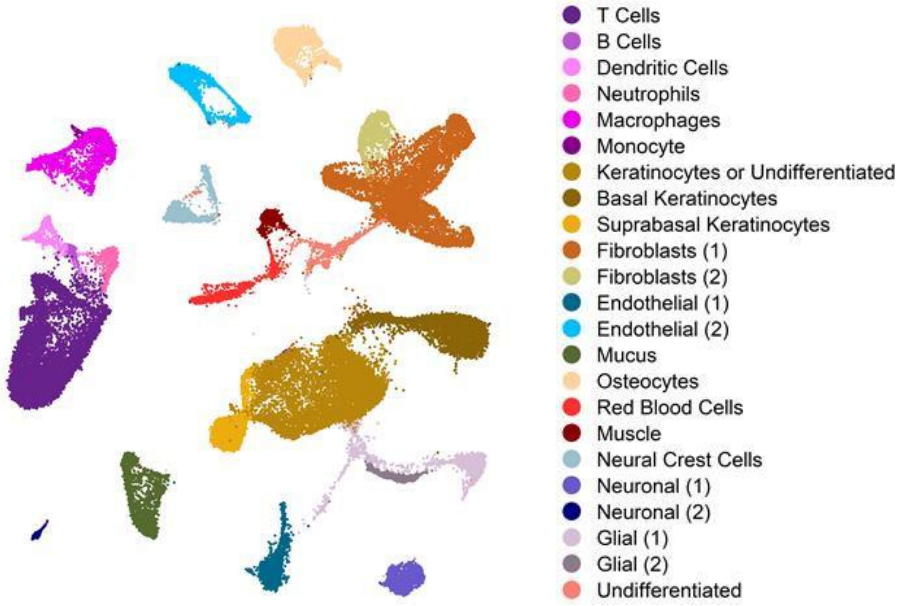


Knockout of dnd gene to induce sterility in salmon. Concurrent knockout of slc45a2 to produce albinos as a 'tracer' edit (*Wargelius et al. 2016*)

Functional Genomics



Atlantic salmon skin

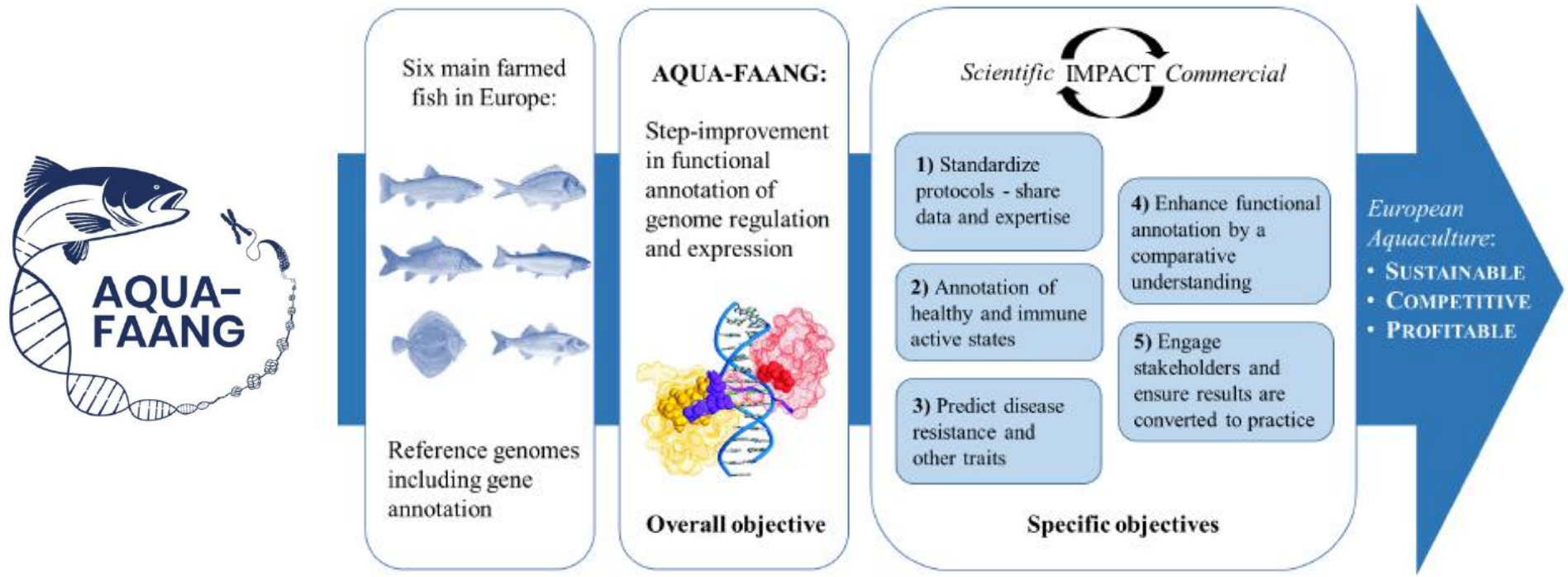


ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Functional Genomics

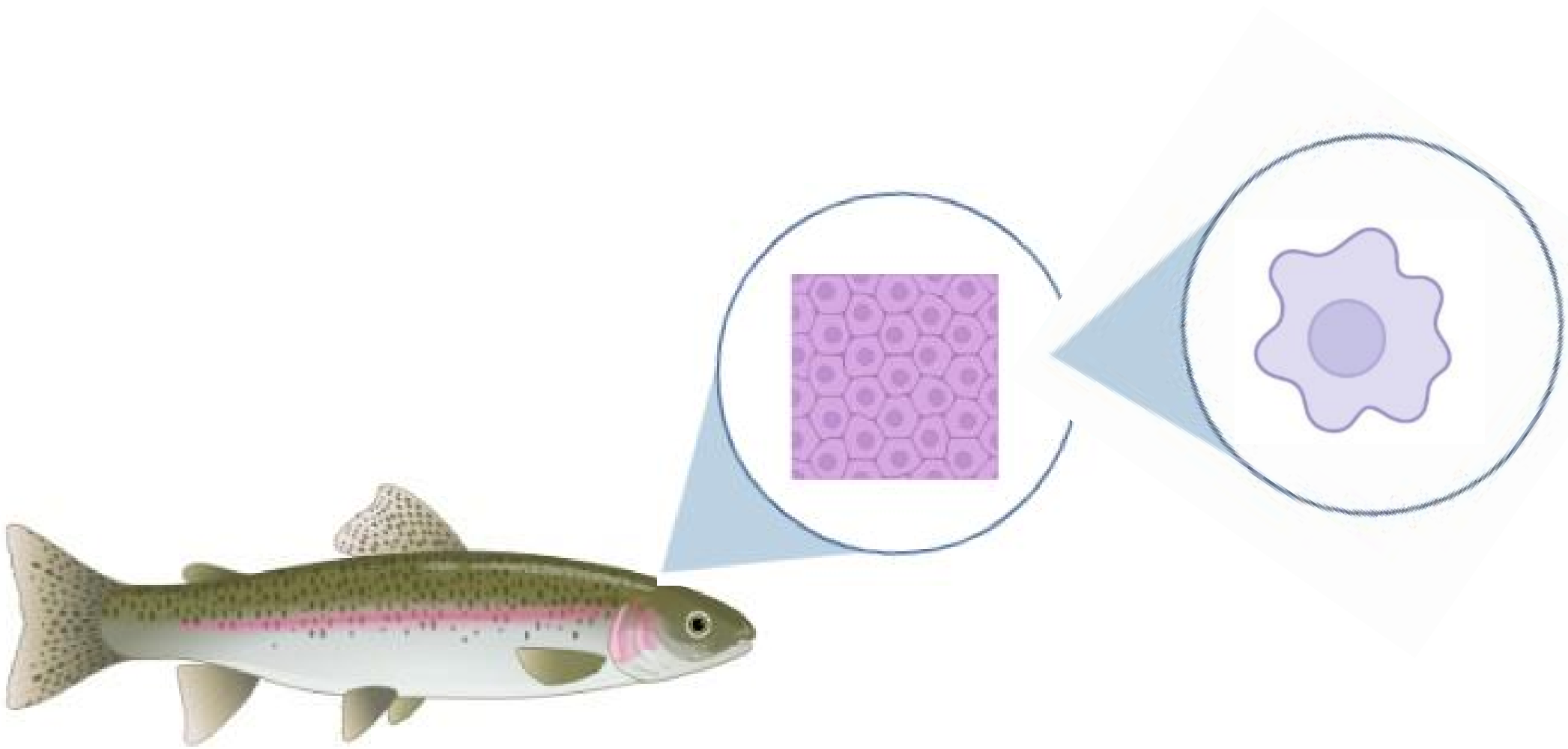


ROSLIN

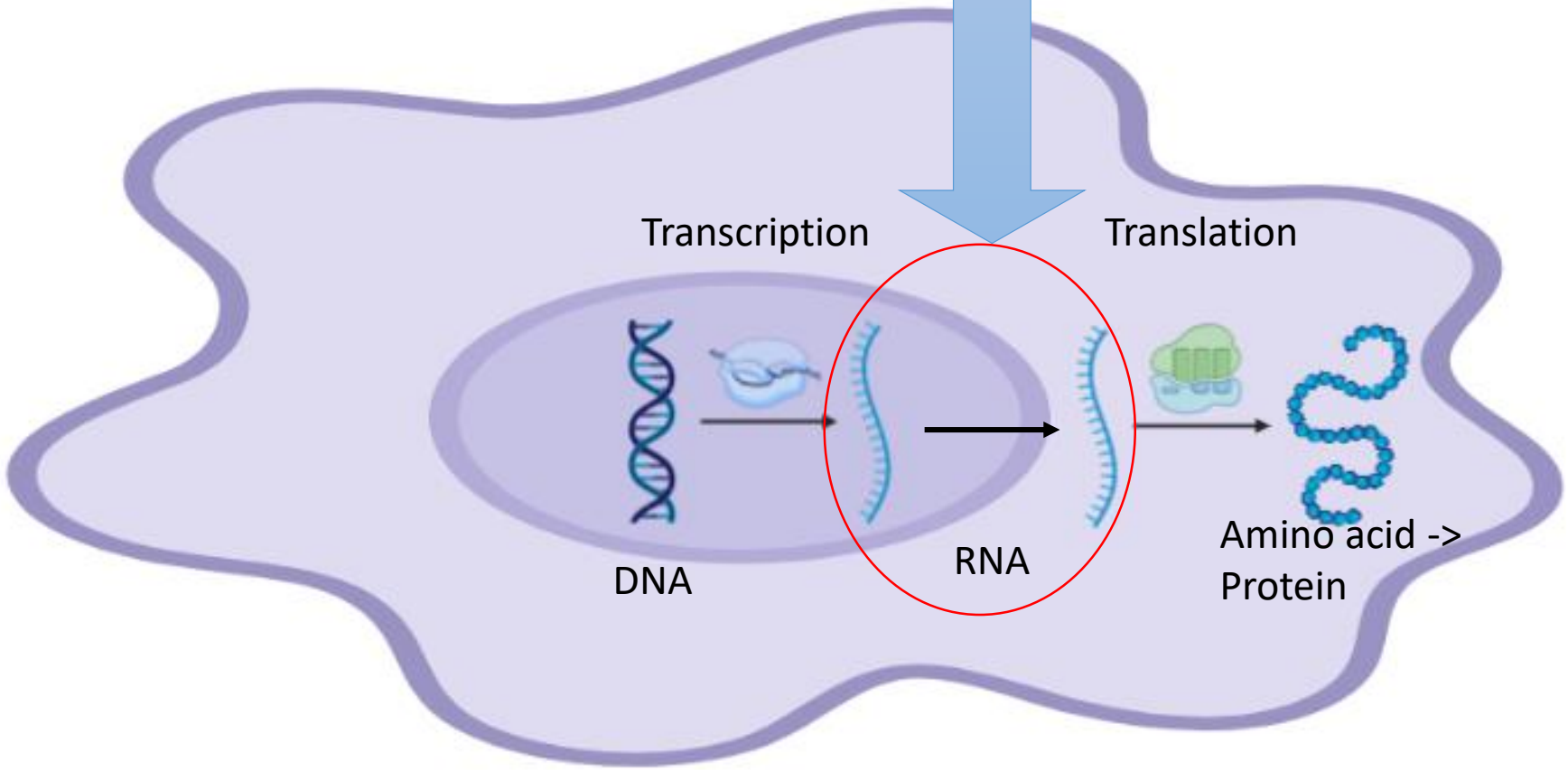
THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

RNA-Seq Definition & Technologies

Porto, 16th & 17th March 2023



- Reveals molecular processes of a cell/tissue
- Identifies functional elements of a genome
- Enables understanding of development, disease etc.

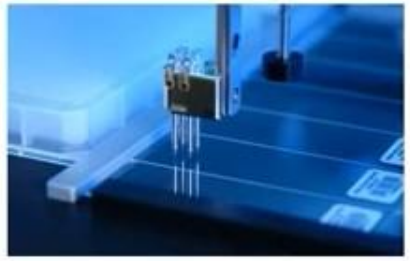


Transcription Analysis Methods

Sanger sequencing

1970s

SAGE: serial analysis of gene expression



Microarray

1970s

complementary probe hybridization

Next generation sequencing

2006

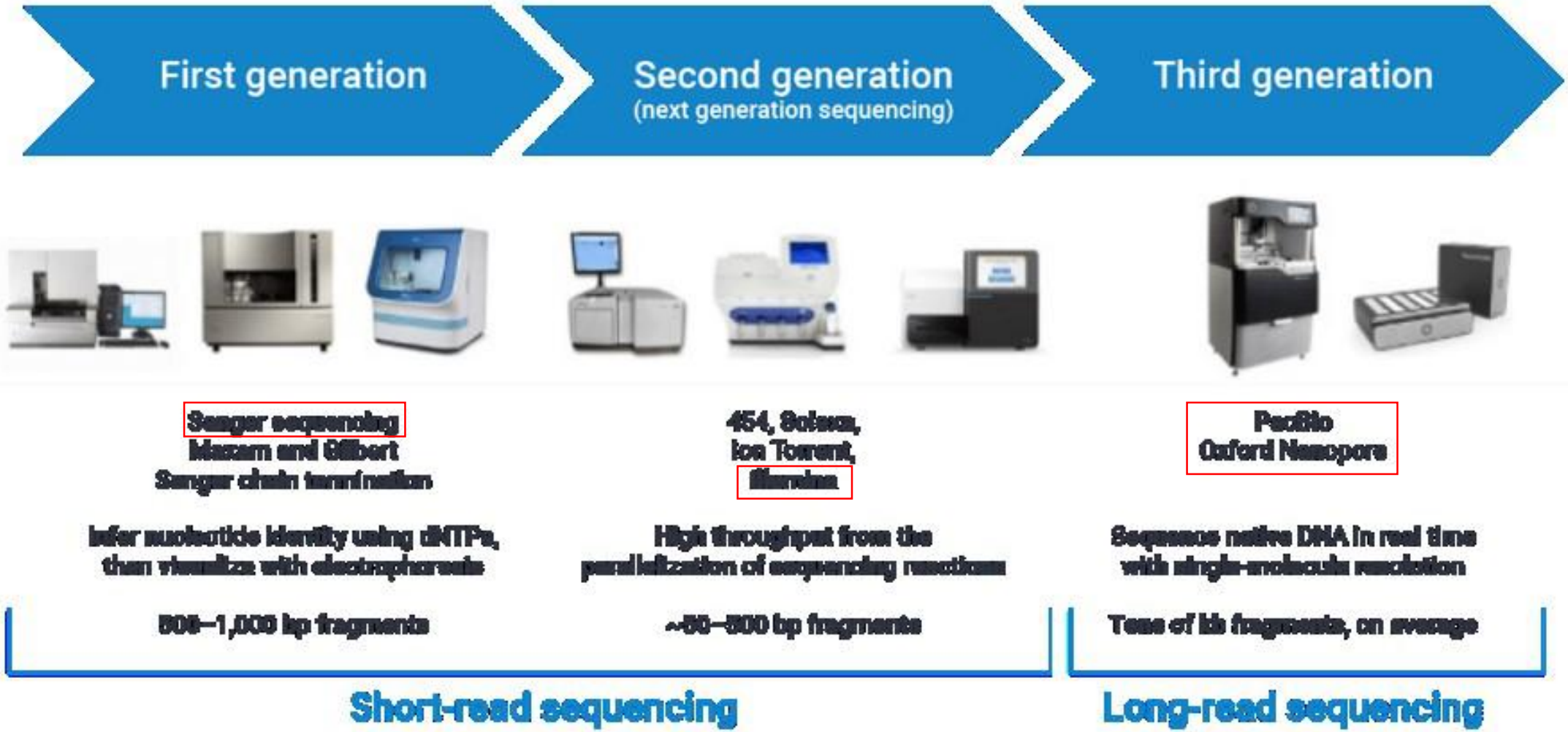
Illumina
Roche 454
Solid

Third generation sequencing

2010

PacBio SMRT
Oxford Nanopore

Current capabilities



What have we used RNA-Seq for?

- Describing evolutionary adaptation
- Cross-species comparisons
- Identifying genetic basis for disease resistance

ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

What have we used RNA-Seq for?

- Describing evolutionary adaptation
- Cross-species comparisons
- Identifying genetic basis for disease resistance

Marine Genomics
Volume 51, June 2020, 100734

Method paper

De novo transcriptome assembly of the Qatari pearl oyster *Pinctada imbricata radiata*

Tim P. Bean [a], Zenaba Khatir [b], Brett P. Lyons [c], Ronny van Aerle [c], Diana Minardi [c], John P. Bignell [c], David Smyth [b d], Bruno Welter Giraldes [b], Alexandra Leitão [b]

Table 2. Transcriptome statistics.

Descriptive statistic	Summary
Number of transcripts	179,599
Number of genes	24,676[a] / 30,739[b]
Total length (bp)	201,029,654
Shortest transcript length (bp)	201
Mean transcript length (bp)	1119.30
Longest Transcript length (bp)	16,371
N50 (bp)	2013

Table 3. Top five (or all) categories for gene set enrichment analysis from each tissue.

Tissue	GO ID	GO Name	GO Category	Nominal p-val	FDR q-val
Digestive Gland	GO:0004866	endopeptidase inhibitor activity	Molecular Function	0	0
	GO:0061135	endopeptidase regulator activity	Molecular Function	0	0
	GO:0061134	peptidase regulator activity	Molecular Function	0	0
	GO:0004857	enzyme inhibitor activity	Molecular Function	0	0
Adductor muscle	GO:0015629	actin cytoskeleton	Cellular Component	0	0
	GO:0016459	myosin complex	Cellular Component	0	0
	GO:0043292	contractile fiber	Cellular Component	0	0
	GO:0030016	myofibril	Cellular Component	0	0
	GO:0030017	sarcomere	Cellular Component	0	0

- Close evolutionary distance to *Pinctada imbricata fucata*
- Highlights differences in immune genes and distinctive transposon families, suggesting recent adaptive divergence.

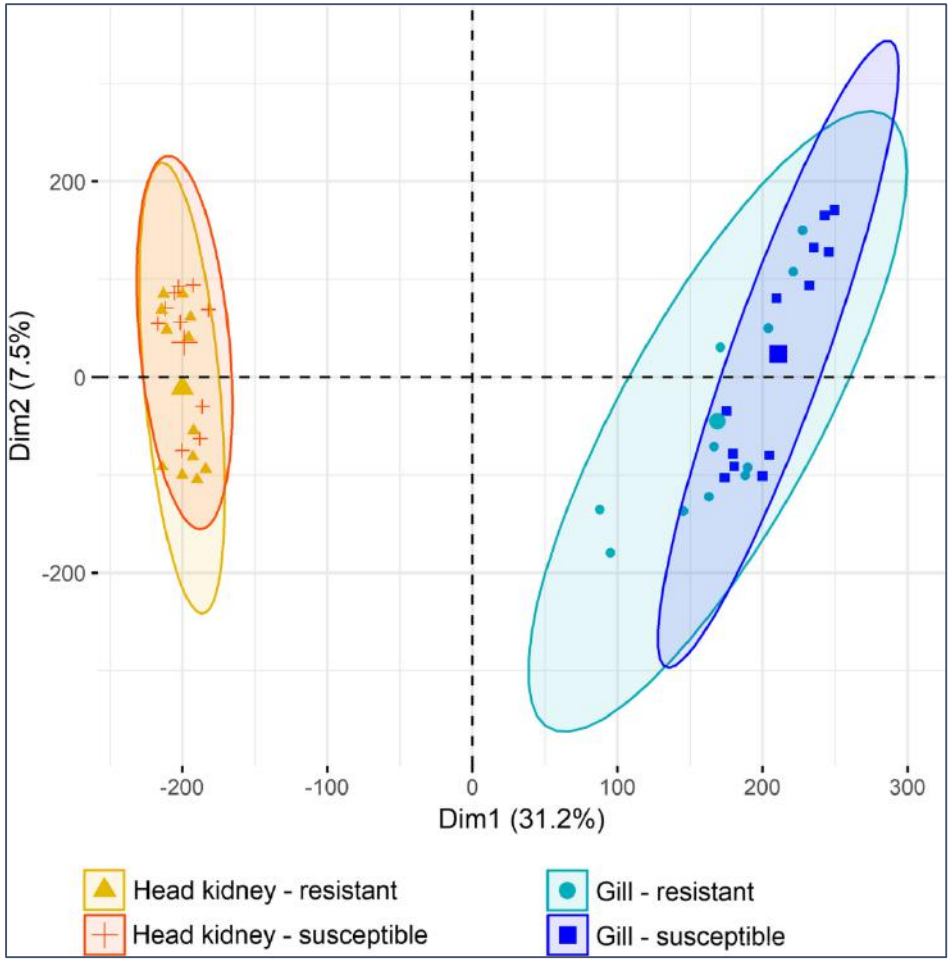
Research article | Open Access | Published: 30 March 2020

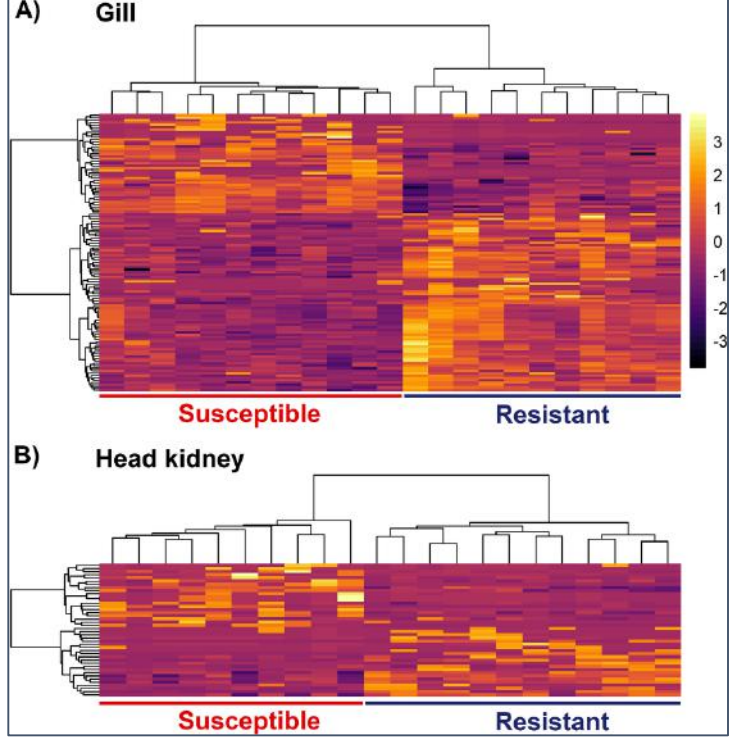
Characterising the mechanisms underlying genetic resistance to amoebic gill disease in Atlantic salmon using RNA sequencing

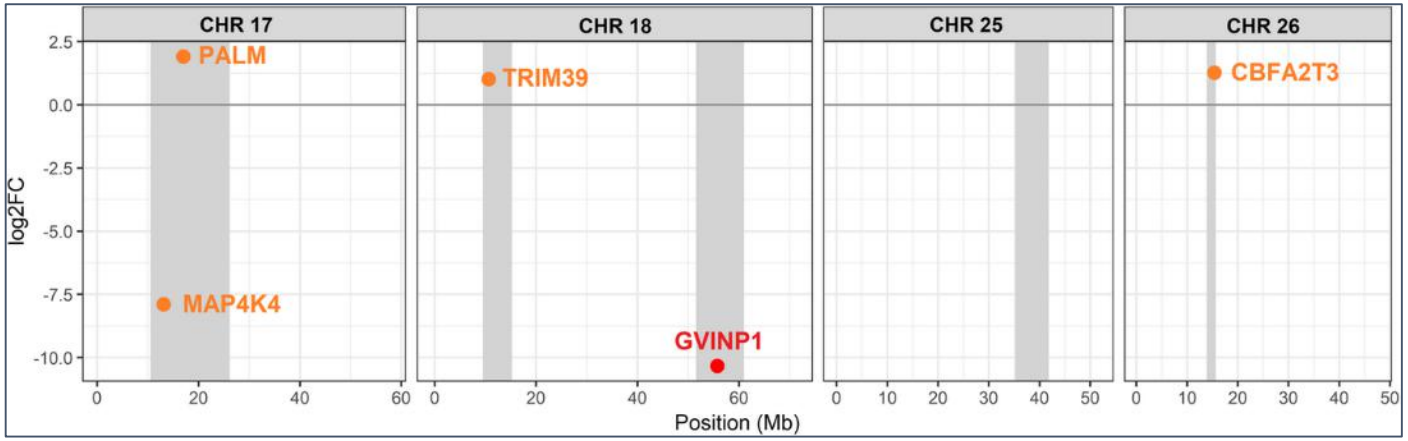
Diego Robledo, Alastair Hamilton, Alejandro P. Gutiérrez, James E. Bron & Ross D. Houston

BMC Genomics **21**, Article number: 271 (2020) | Cite this article

2776 Accesses | **16** Citations | **16** Altmetric | Metrics







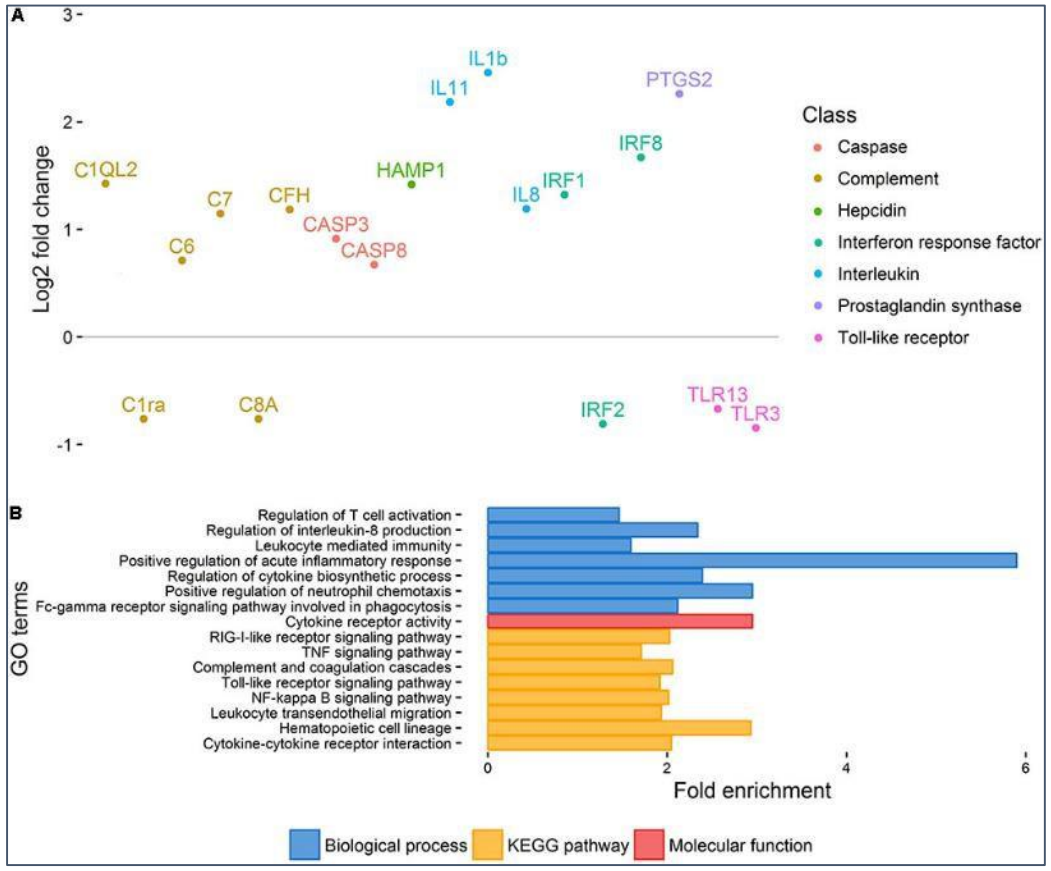
- Potential candidate genes of interest

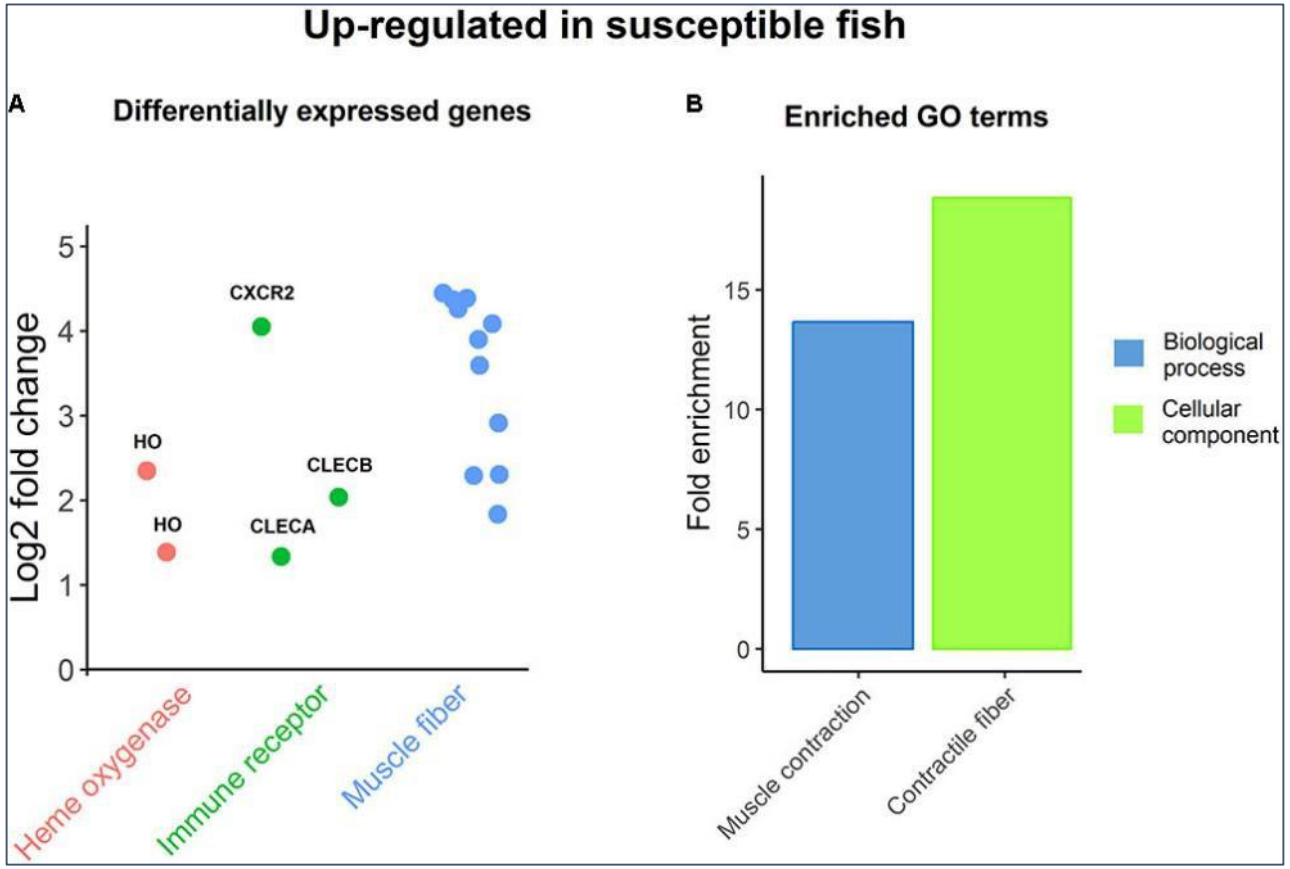
Front. Genet., 03 August 2018
Sec. Livestock Genomics
Volume 9 - 2018 | https://doi.org/10.3389/fgene.2018.00287

This article is part of the Research Topic
Animal Genetics and Diseases: Advances in Farming and Livestock Systems
View all 16 Articles ›

Gene Expression Response to Sea Lice in Atlantic Salmon Skin: RNA Sequencing Comparison Between Resistant and Susceptible Animals

Diego Robledo[1*], Alejandro P. Gutiérrez[1], Agustín Barría[2], José M. Yáñez[2,3†] and Ross D. Houston[1*†]





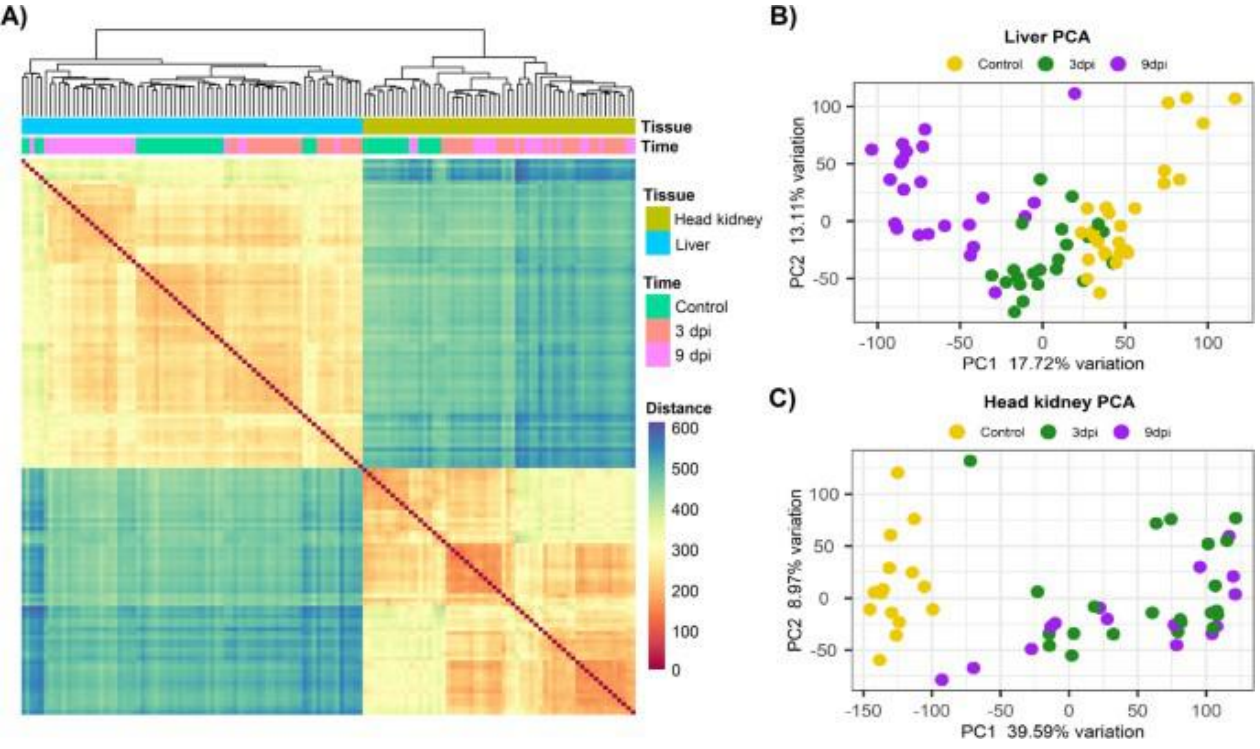
- Differences in immune response and pattern recognition genes, also myogenic and iron availability factors.

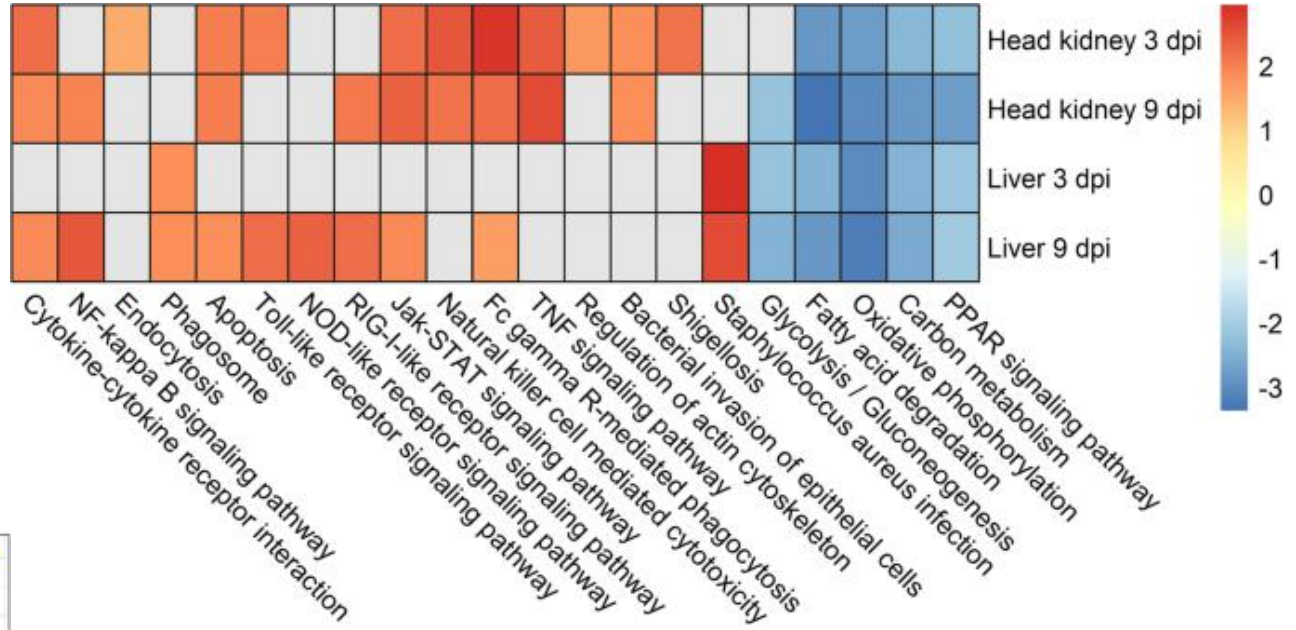
Research article | Open Access | Published: 06 March 2021

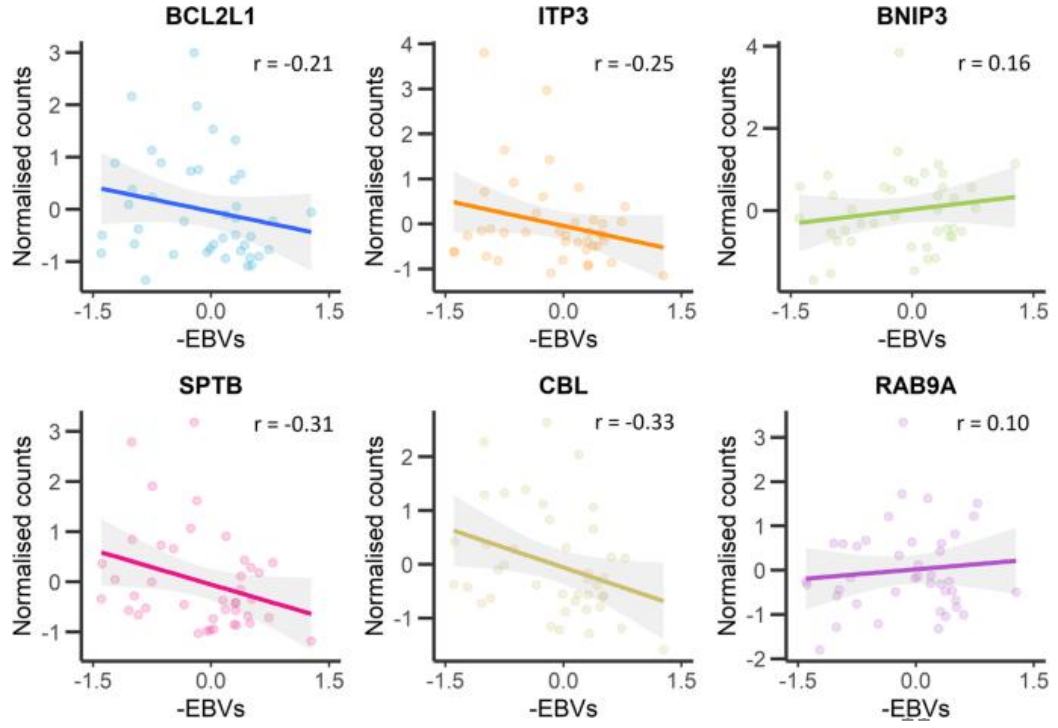
Investigating mechanisms underlying genetic resistance to Salmon Rickettsial Syndrome in Atlantic salmon using RNA sequencing

Carolina P. Moraleda, Diego Robledo, Alejandro P. Gutiérrez, Jorge del-Pozo, José M. Yáñez ✉ & Ross D. Houston ✉

BMC Genomics **22**, Article number: 156 (2021) | Cite this article







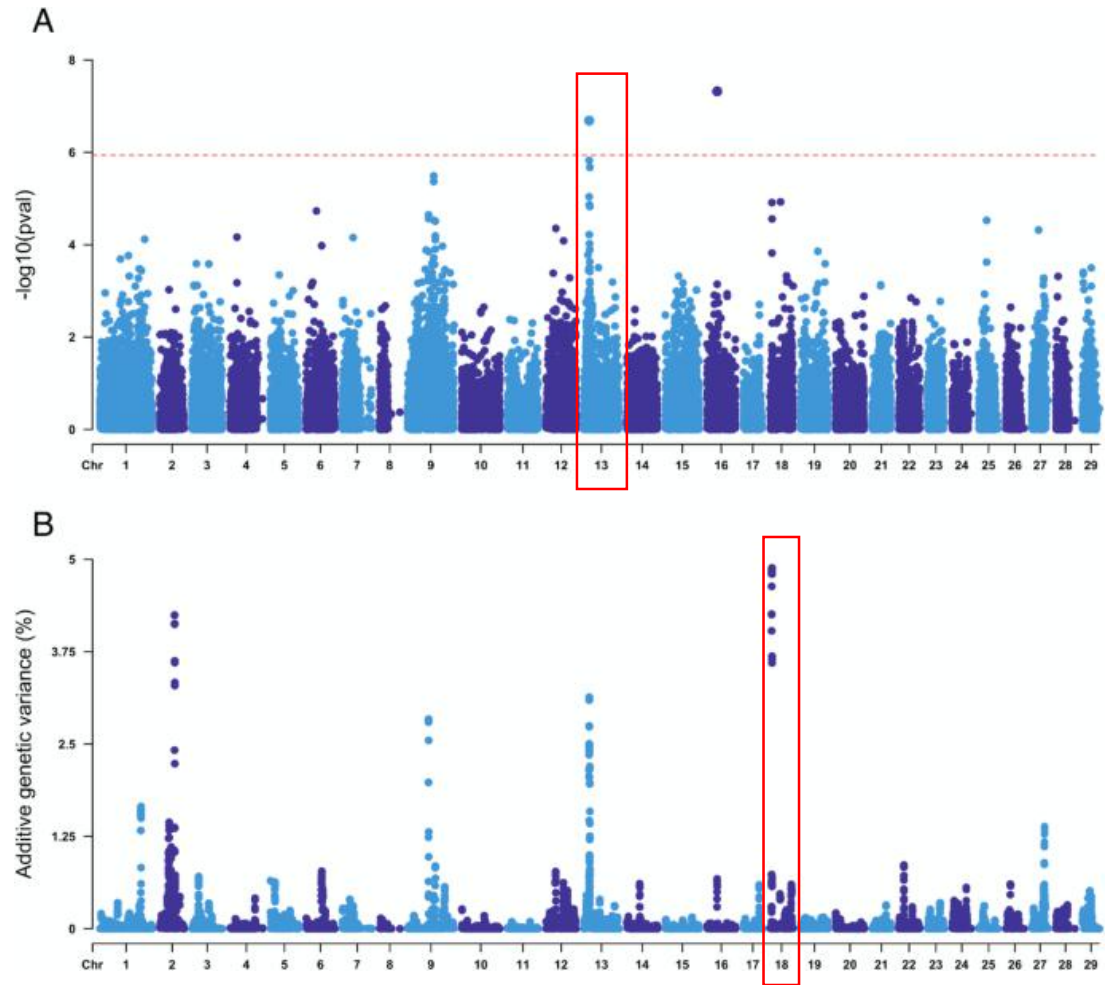
- Several networks correlated with SRS resistance EBVs : Apoptosis, cytoskeletal organisation, and the inflammasome.

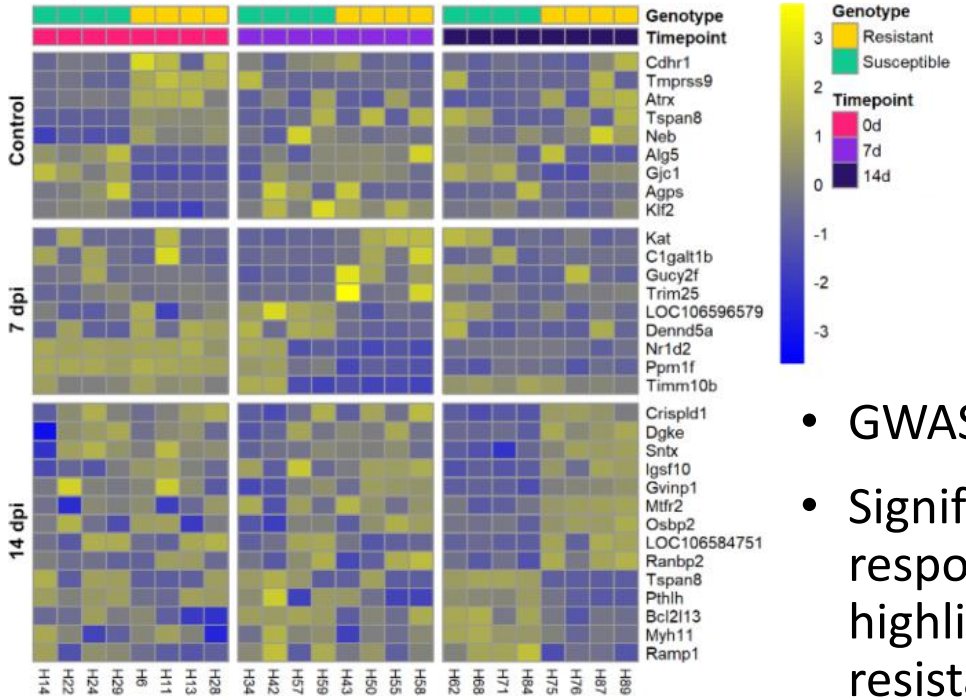
Research | Open Access | Published: 13 May 2021

Exploring genetic resistance to infectious salmon anaemia virus in Atlantic salmon by genome-wide association and RNA sequencing

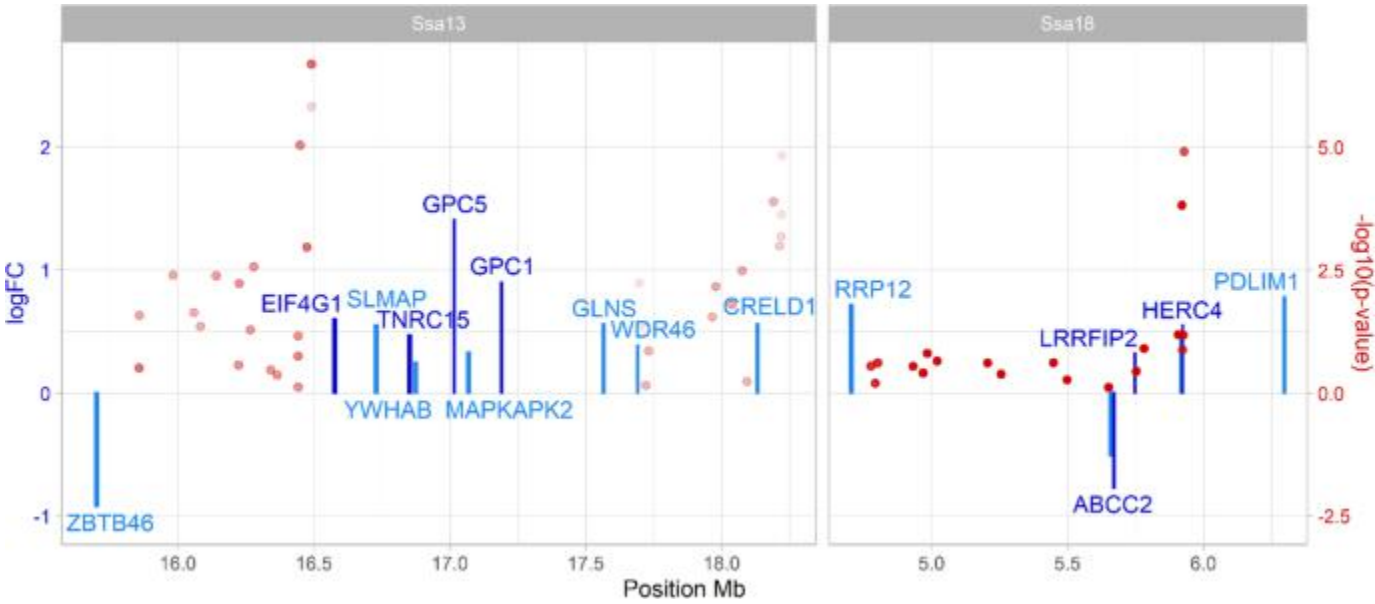
O. Gervais, A. Barria, A. Papadopoulou, R. L. Gratacap, B. Hillestad, A. E. Tinch, S. A. M. Martin, D. Robledo & R. D. Houston

BMC Genomics **22**, Article number: 345 (2021) | Cite this article





- GWAS vs RNA-Seq hits
- Significance of IFN response elements highlighted for resistance



ROSLIN

grinnaqua

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Sample types, RNA extraction and library preparation

Porto, 16th & 17th March 2023

- Start with highest quality freshest material possible
- Use TRIzol, column or bead based extraction system
- DNase treat the sample
- Check for contamination
  - Clean up any contamination (e.g. guanidinium or phenol) with secondary column or bead purification
- Accurately quantify RNA
- Assess quality of RNA

- Congratulate yourself... The easy bit is done

RNA...

- rRNA ~ 80% ribosomal mass in a cell
- mRNA ~ 1-5% ribosomal mass in a cell

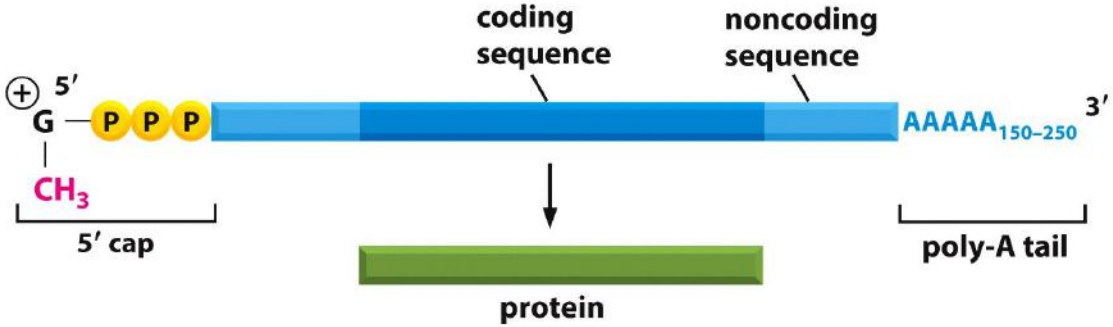


Figure 7-10 *Essential Cell Biology* (© Garland Science 2010)

mRNA is short lived

Cell	Cell generation time	Average mRNA half life	Range of mRNA half lives
Escherichia coli	20 – 60 min	3 – 5 min	2 – 10 min
Sacchromyces cerevisiae	3 hr	22 min	4 – 40 min
Cultured mammalian cells	16 – 24 hr	10 hr	<30 min - 24 hr

RNA degradation – magic and mystery

Physical / chemical...

- RNA can degrade in solution through hydrolysis, a process enhanced by heat and high pH.

Ribonucleases

- RNAses are enzymes that directly and rapidly degrade RNA.
- Remain active at -20 °C.
- Exist virtually everywhere.

RNAses are a very common justification for degraded RNA...

Degredation only usually occurs when good practice is missing **(no magic or mystery involved!).**

Practice + Being aware of your sample and processes will prevent most RNAse issues.

RNA protection – magic and mystery

- The majority of RNases that will cause you issues exist within the sample... They are a normal part of cellular function.
  - Preserve you sample as soon as possible.
- Other RNases exist on you and in the lab...
  - Always operate good practice in the lab.
  - Keep things clean and cold.
  - Work quickly.
- Don’t freeze-thaw more than you need to!

Use RNase away / RNase zap as a **last resort.**

It destroys enzymes indiscriminately and is thus not beneficial for the majority of lab work!

Sample Collection

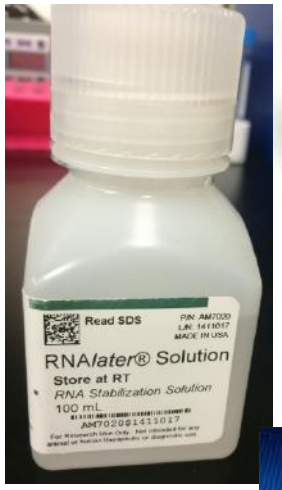
Sample collection methods should be optimised and tested to suit your sample type, size and lab situation

- Clean equipment, scalpels, forceps etc with ethanol.

- Excise tissue – if contaminated (e.g. blood) wash with RNAse free water

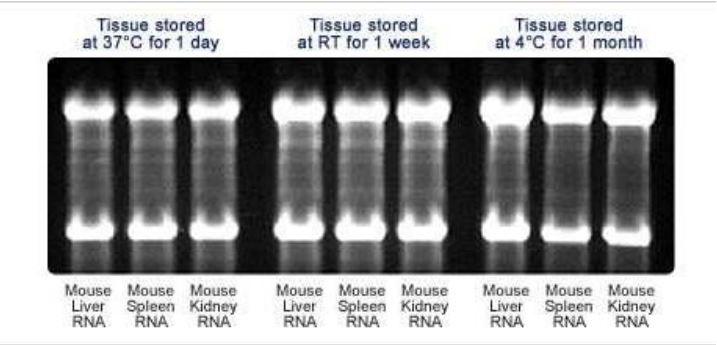
    - Flash freezing in liquid nitrogen. (greatest flexibility)
    - Flash freeze in supercooled isopentane (allows for greater control and freezing of larger samples).
    - Flash Freeze tissue in tube on dry ice.
    - Fix and Store in RNAlater. (safest)
        - Allow the sample to fix overnight.
    - Fix and Store in TRIzol (limits later options)
        - Sample should be homogenised for effective fixation.

Quantities required are hugely dependent on type of tissue and experiment, but generally speaking keep pieces of tissue to less than $3mm^3$. Always take extra samples as backup and for testing methods.



Sample storage

- Directly Frozen tissue
  - Always store tissue at -70 °C or lower
  - Transfer tubes on dry ice or LN2.
  - -20 °C very limited scope for storage – samples will degrade rapidly.
  - Cannot go through freeze-thaw cycles without issue



- RNAlater
  - Fixative – requires time to fix samples.
  - Functions for 1 day at room temp, 1 month in a fridge, 1 year at -20, a long time at -80.
  - Freeze thaw cycles are possible but eventually reduce integrity and varied between tissues.
  - RNAlater also good for storing DNA...

RNA extraction

- Multiple options for RNA extraction.
- Most important is to check the process on mock samples before starting on actual samples.

Stabilize > **Lyse** > Collect RNA > *Remove DNA* > Clean RNA > Elute RNA> re purify as required

- Trizol /TriReagent/ Tripure - **phenol** and guanidinium isothiocyanate
- Column based extraction / cleanup
- Bead based extraction / cleanup
- Automated (Robot) extraction (generally just automated column or bead)





RNA extraction

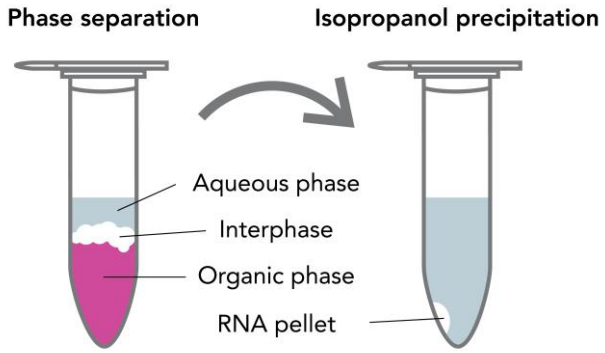
Effective homogenisation and cell lysis is crucial. LOOK at the samples

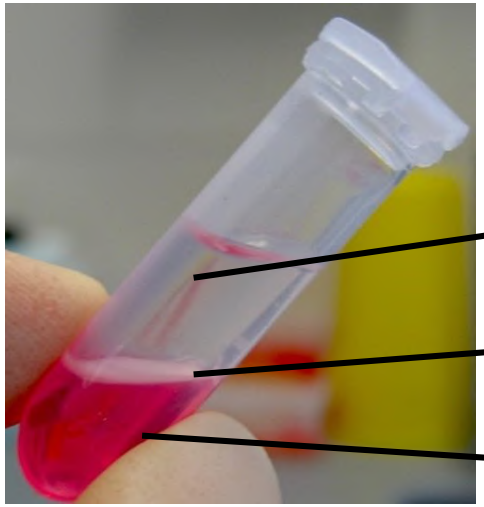
- MpBio fastprep is very useful. Easy, scalable, quick, effective.
- Use liquid nitrogen pestle and mortar for large samples.
- Pellet pestle for medium small samples.
- Pass through syringe to remove clumps (if necessary)

- For tricky samples some material will never homogenise (pearls / scales)
- Spin down homogenate in lysis buffer and separate supernatant from cell debris.

Remove DNA with DNase step during extraction.

TRIzol





- Effective on most tissues.
- Large RNA yields

➢ Time consuming.
➢ Dangerous Chemicals (phenol will burn skin and eyes).

TIPS...

USE TEST SAMPLES FIRST to check required quantities.

The most common mistake is using too much starting material.

- Always look at the tube! Use your (goggle protected) eyes.
- Too much material = messy interphase.
- Use less, clean the aqueous again (take aqueous phase, add more chloroform (or BCP), mix, spin and collect).

Only use clean aqueous phase... less is more.

Work with smaller numbers of samples and work quickly.

Extraction kits (Qiagen / Promega etc etc)

100s of Kits available for specific cell / tissue types.

- Same principles apply...
  - Lysis protocols / reagents in these kits are designed for classic tissues (mouse liver, spleen, heart etc)
  - Good quality lysis is crucial. Practice, observe and
  - Less material is generally better than more material.
  - Can often be automated.

- TIPS
  - Most common problem is too much starting material... Remember to optimise.
  - If lysis looks messy dilute samples in more lysis buffer and use half.
  - OR... Spin lysate and only use the supernatant.
  - Tissue lysed in RLT (or equivalent buffer) is safe to be stored at -80°C.
  - Work with small numbers of samples and keep things cold.

Simple kits will almost always work when you have good lysate.
Don't waste money on specialist extraction kits.
Spend time optimising lysis.

Other comments...

- Columns and beads can be used for total extraction, to purify RNA that has come contamination, to clean up the aqueous phase from a TRIzol extraction.

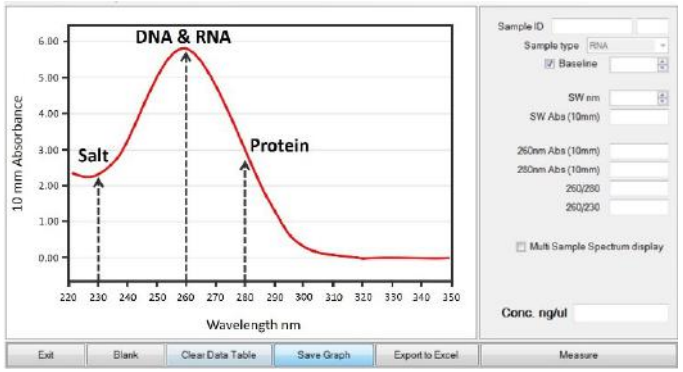
- RNA can be cleaned again.

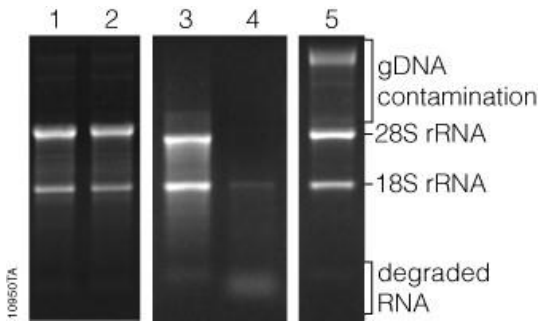
- Be consistent. Use the same protocol for a whole experiment. For example different methods can result in differential size selection which may prove important.

- Always remove DNA with DNase step during or after extraction > many methods available > choose the most appropriate.

Quality control

- Check **quantity and purity with Nanodrop** or other spectrophotometer.
- Confirm quantity with fluorescent methods (Qubit, Quantifluor)
- Check RNA integrity with capillary electrophoresis (Bioanalyzer, Tapestation)
- Consistency across an experiment is usually the most important factor.
  - Discard or take note of samples which do not behave as expected.





260/280 ~2.0
260/230 ~2.0 – 2.2

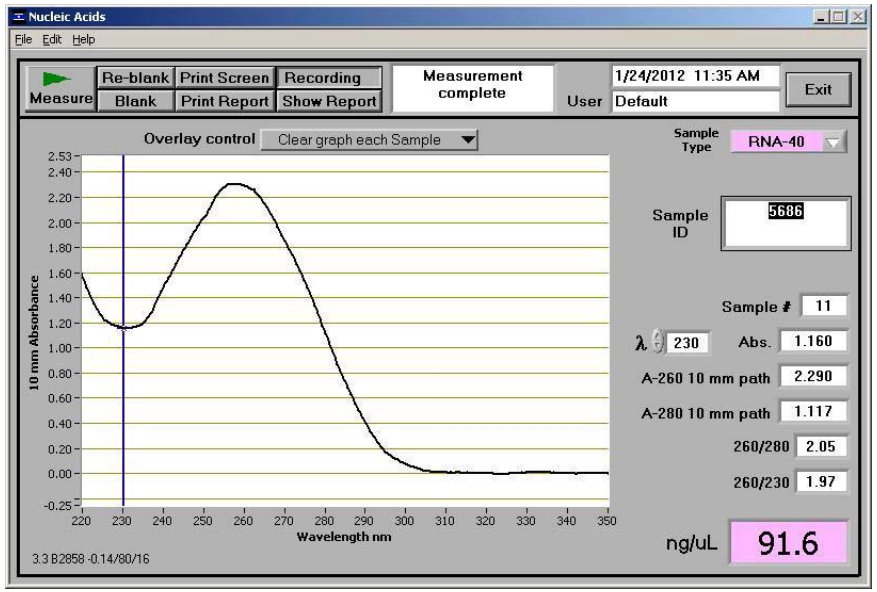
For Stranded mRNA-Seq RIN>8

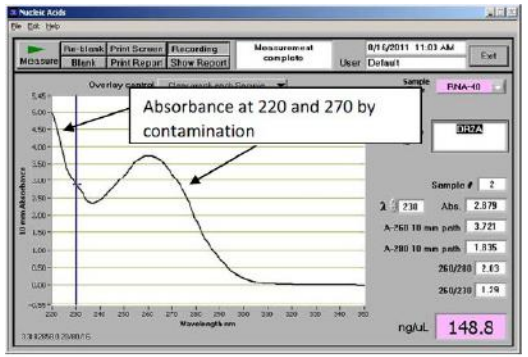
500 ng to 1ug total RNA required per sample.

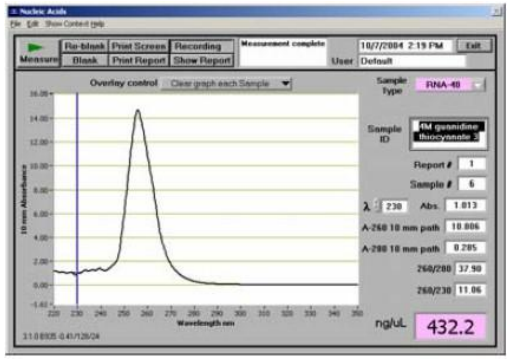
NOTE – Some species (drosophila, bivalves) only show one RNA band

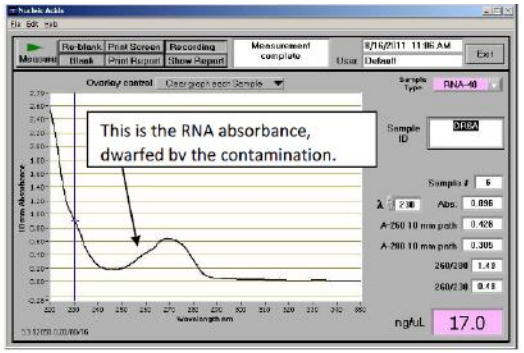
Nanodrop (Spectrophotometer)

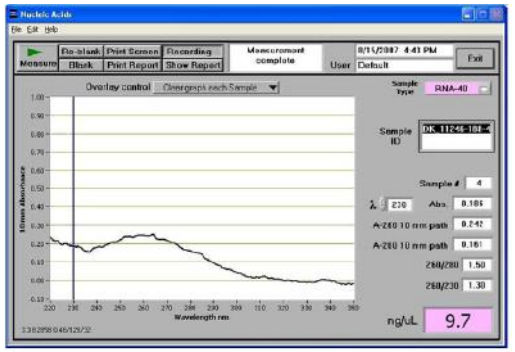
- Nanodrop measures absorption of light at different wavelengths from 220 to 350 nm.
- Nucleic acids absorb light at 260 nm (**all nucleic acids... Not just RNA**).
- Used for quantification and to check for contamination.
- To quantify nucleic acid take the absorption at 260 x 50 for DNA or Ab260 x 40 for RNA.











- Use all the information available to you... Not just the number in the bottom right!

Dye based... Invitrogen Qubit / Promega Quantus / RiboGreen

- Very accurate quantification.
- Distinguishes between nucleic acid types.
- Measures against a standard (make up fresh standards for important experiments!)
- Cannot spot contamination

- Qubit measures fluorescence of dyes which bind to specific nucleic acids
  dsDNA Broad Range (BR)
  dsDNA High Sensitivity (HS)
  ssDNA Assay Kit
  RNA High Sensitivity (HS)
  RNA Broad Range (BR)
  RNA Extended Range (XR)
  MicroRNA kit

- Quantification more accurate that nanodrop or tapestation
- Differences between qubit and nanodrop can be down to contamination and vary **a lot** depending on the extraction method used.

(Promega Quantus is very similar and usually cheaper.)

Bioanalyzer / Tapestation (Capillary electrophoresis)

- Generally use the “high Sensitivity RNA Protocol” to assess RNA Integrity Number (RIN).
- RIN is calculated by the ratio of large to small RNAs, measuring degredation.
- Most sequencing providers are delighted with a RIN>8 but can work with much lower if required
- Sequencing companies will usually insist on running this QC for you.

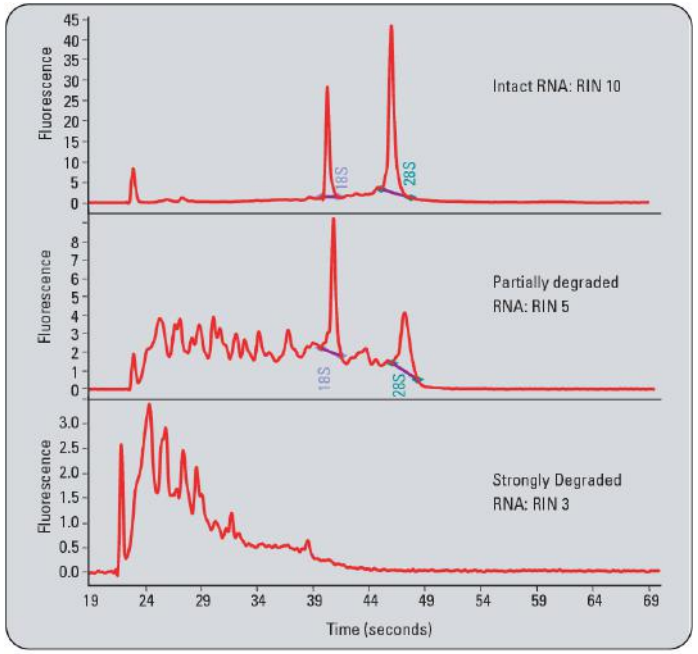
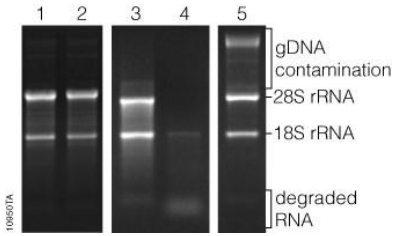
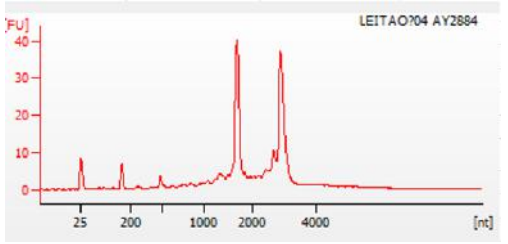
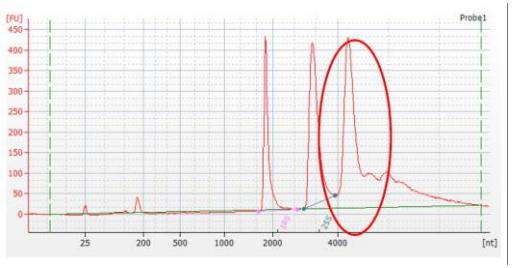


Figure 6

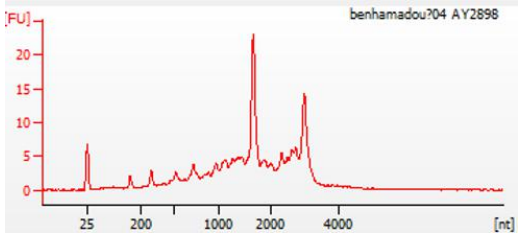




Good quality (normal) trace



DNA contamination



Less RNA with some degradation

*Note... Some invertebrates (e.g. bivalves) only show one RNA band... The 28S rRNA breaks into two that are the same size as the 18S rRNA. The Tapestation will still calculate a RIN as normal but it is not visible on a gel / trace.

Novogene

Library Type	Sample Type	Amount	Volume	Concentration	RNA Integrity Number (Agilent 2100)	Purity (NanoDrop™)
Eukaryotic mRNA library (poly A enrichment)	Total RNA (Animal)	⩾ 200 ng	⩾ 10 μL	⩾ 20 ng/μL	⩾ 4.0, with flat base line	OD260/280 ⩾ 2.0; OD260/230 ⩾ 2.0; no degradation, no contamination
	Total RNA (Plant and Fungus)	⩾ 200 ng	⩾ 10 μL	⩾ 20 ng/μL	⩾ 4.0, with flat base line	
	Total RNA (Blood)	⩾ 400 ng	⩾ 20 μL	⩾ 20 ng/μL	⩾ 5.8, with flat base line	
	Amplified cDNA (double-strand)	⩾ 100ng	⩾ 10 μL	⩾ 10 ng/μL	Fragments distributing between 400bp-5000bp, with the main peak at ~2000bp;	OD260/230 ⩾ 2.0; no degradation, no contamination
Eukaryotic Directional mRNA library (poly A enrichment)	Total RNA (Animal)	⩾ 400 ng	⩾ 20 μL	⩾ 20 ng/μL	⩾ 5.8, with flat base line	OD260/280 ⩾ 2.0; OD260/230 ⩾ 2.0; no degradation, no contamination
	Total RNA (Plant and Fungus)	⩾ 400 ng	⩾ 20 μL	⩾ 20 ng/μL	⩾ 5.8, with flat base line	
	Total RNA (Blood)	⩾ 400 ng	⩾ 20 μL	⩾ 20 ng/μL	⩾ 5.8, with flat base line	OD260/280 ⩾ 2.0, OD260/230 ⩾ 2.0, no degradation, no contamination
Prokaryotic RNA library	Total RNA	⩾ 500 ng	⩾ 10 μL	⩾ 50 ng/μL	⩾ 6.0, with flat base line	OD260/280 ⩾ 2.0; OD260/230 ⩾ 2.0; no degradation, no contamination
Meta-transcriptome library	Total RNA	⩾ 1 μg	⩾ 20 μL	⩾ 50ng/μL	⩾ 6.5, with flat base line	OD260/280 ⩾ 2.0; OD260/230 ⩾ 2.0; no degradation, no contamination
Dual RNA library	Total RNA	⩾ 1 μg	⩾ 20 μL	⩾ 50ng/μL	⩾ 6.5, with flat base line	OD260/280 ⩾ 2.0; OD260/230 ⩾ 2.0; no degradation, no contamination

BGI

Sample Requirements

- Human/Animal: Total RNA ≥200ng, Concentration ≥20ng/μl. RIN≥7.0 28S/18S≥1.0
- Fungi: Total RNA ≥1μg, Concentration ≥40ng/μ. RIN≥6.5 28S/18S≥1.0
- Plant: Total RNA ≥1μg, Concentration ≥40ng/μl. RIN≥6.5 28S/18S≥1.0
- Insects: Total RNA ≥1μg, Concentration ≥40ng/μl.
- Prokaryotes: Total RNA ≥1μg, Concentration ≥40ng/μl. RIN≥7.0 23S/16S≥1.0

Library Preparation – Choices...

- It is possible to prepare your own libraries and send for sequencing.
- However, it is easier, cheaper and more reliable to use a service provider.

1. Aim of sequencing (transcriptome, quantitative gene expression, lcRNA, micro RNA)
2. Total or mRNA sequencing
    - Type of rRNA depletion
3. Strandedness
4. Poly A selection or 3 prime biased sequencing.

Library Preparation – rRNA

The majority of RNA in a cell is rRNA... This is of little to no interest for most RNAseq experiments.

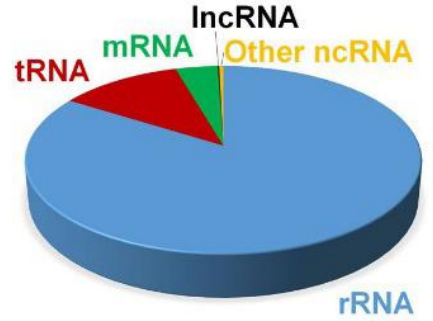
Two ways to deal with this...

1. rRNA depletion – removal of rRNA from the sample **<u>total RNA-seq.</u>**

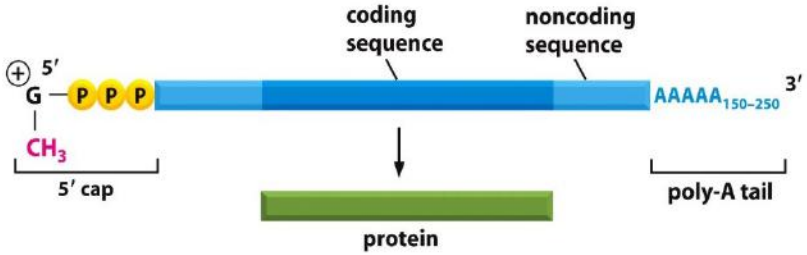
    - Illumina: TruSeq
        - Probes hybridize rRNA on magnetic beads
        - RNA of interest remains in supernatant

    - RiboErase (or equivalent)
        - Probes hybridize rRNA in solution
        - Hybrids are digested with RNase H
        - Probes digested with DNase I
        - NOTE – efficiency of rRNA removal may vary between species / samples

2. mRNA enrichment – **<u>mRNA-seq</u>**
    - Selective reverse transcription of mRNA using either OligodT primed RT OR polyA selection



RNA by mass



Library Preparation – rRNA

For differential gene expression analysis, it is best to enrich for Poly(A)+ (unless you are aiming to obtain information about long non-coding RNAs), but ribosomal RNA depletion is also very common.

Noting... Poly A selection also depletes...

- Ribosomal/Transfer RNA
- Histone mRNA
- Long-noncoding RNA
- Nascent intron containing transcripts
- Micro RNA
- Degraded RNA
- Many viral transcripts
- Most prokaryote transcripts

Library Preparation

For differential gene expression analysis, it is best to enrich for Poly(A)+, unless you are aiming to obtain information about long non-coding RNAs, then do a

Test your sample storage and extraction method prior to any large-scale experiment.

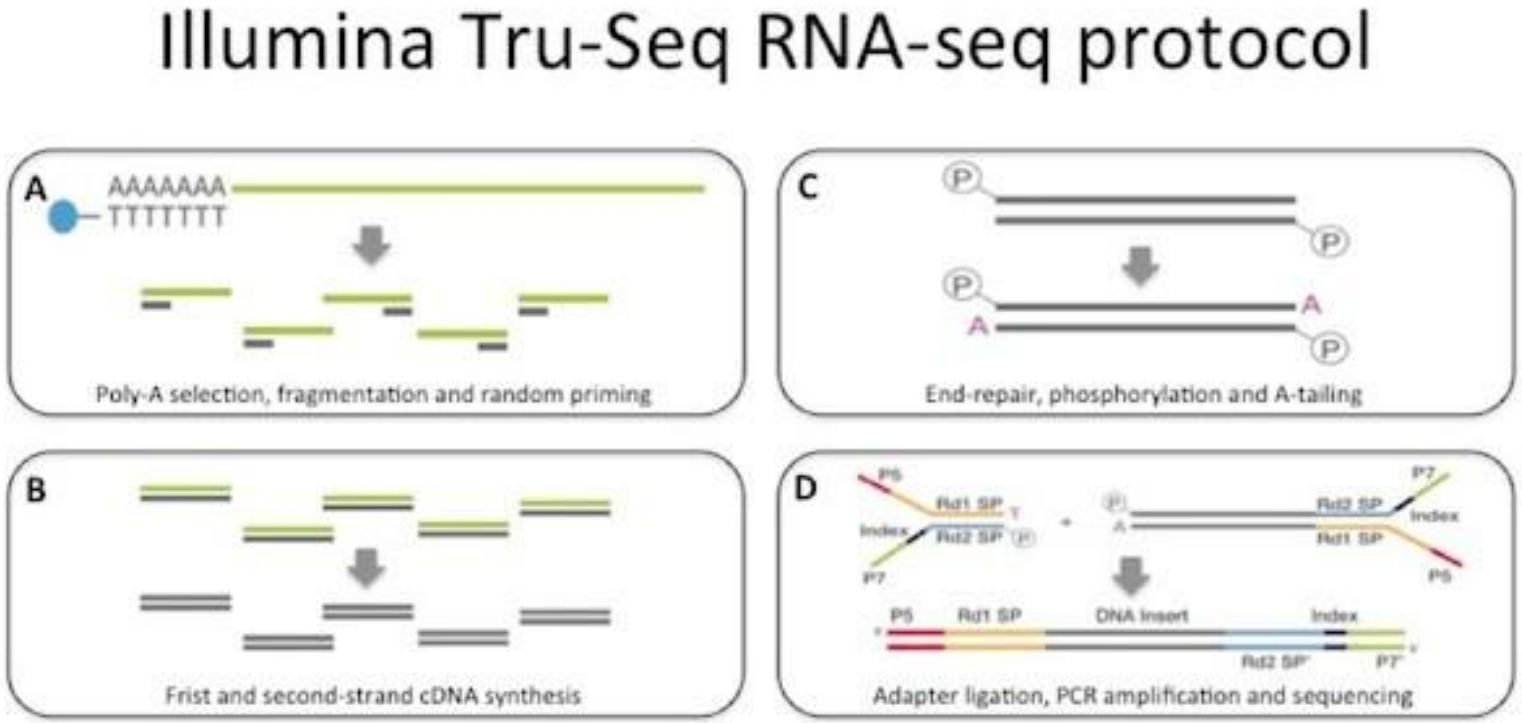
Determine the quantity and quality of your sample.

Note special requirements for experiment

Decide if you are performing mRNA-seq or rRNA depletion

- Degraded RNA
- Many viral transcripts
- Most prokaryote transcripts

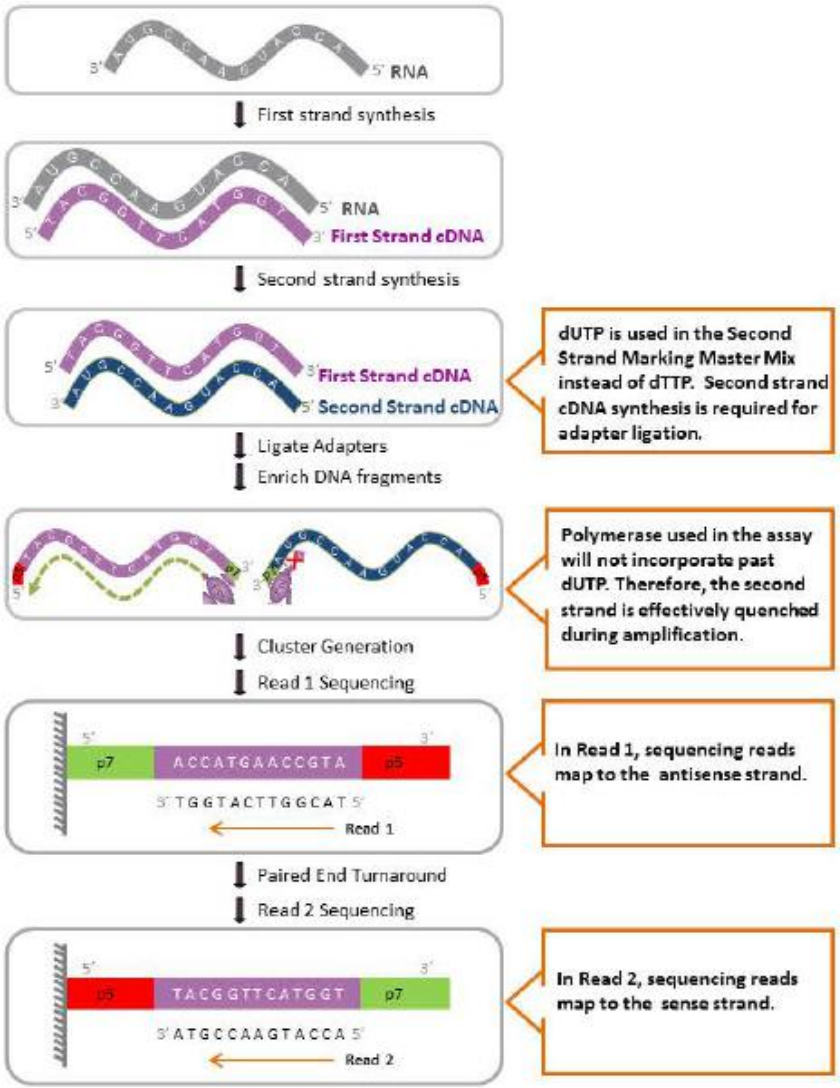
Library Preparation – Illumina TruSeq...

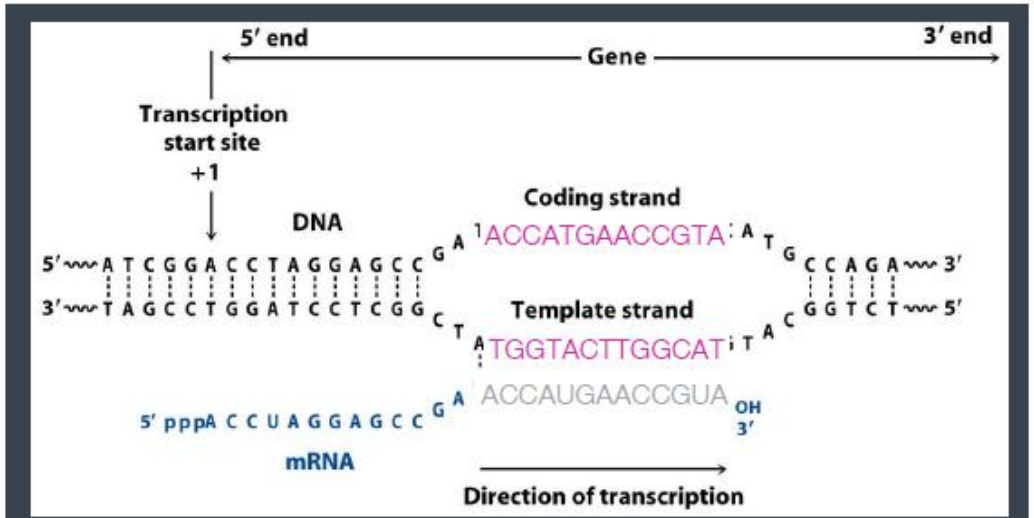


Library prep begins from 100ng-1ug of Total RNA which is poly-A selected (A) with magnetic beads. Double-stranded cDNA (B) is phosphorylated and A-tailed (C) ready for adapter ligation. The library is PCR amplified (D) ready for clustering and sequencing.

Tru-Seq is poly-A selection. 3’ biased sequencing is similar... But primes off poly A rather than with random primer.

Library Strandedness

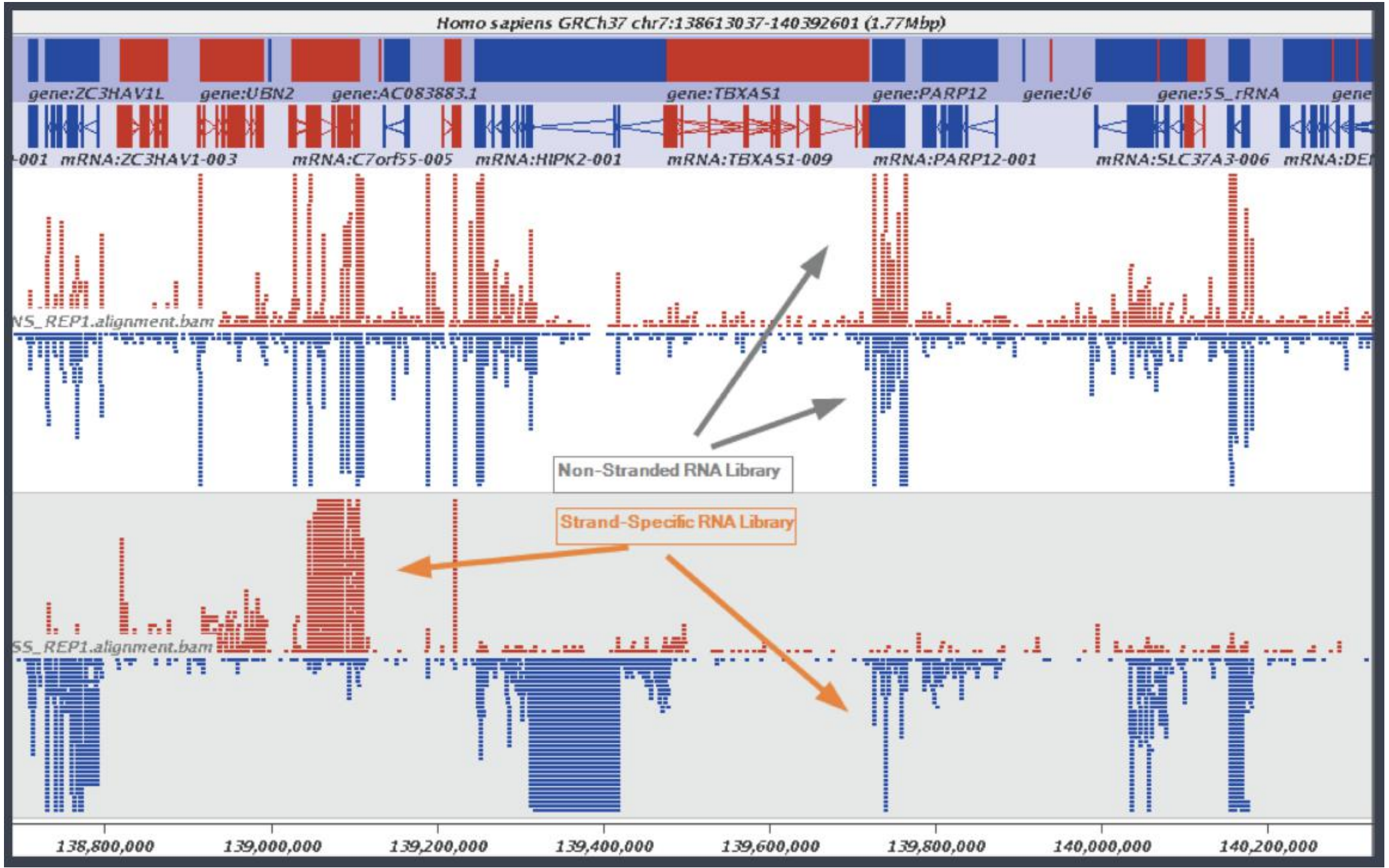




- Read alignment depends on direction of transcription
- "sense" strand of transcript can be on either the sense or antisense strand of the DNA

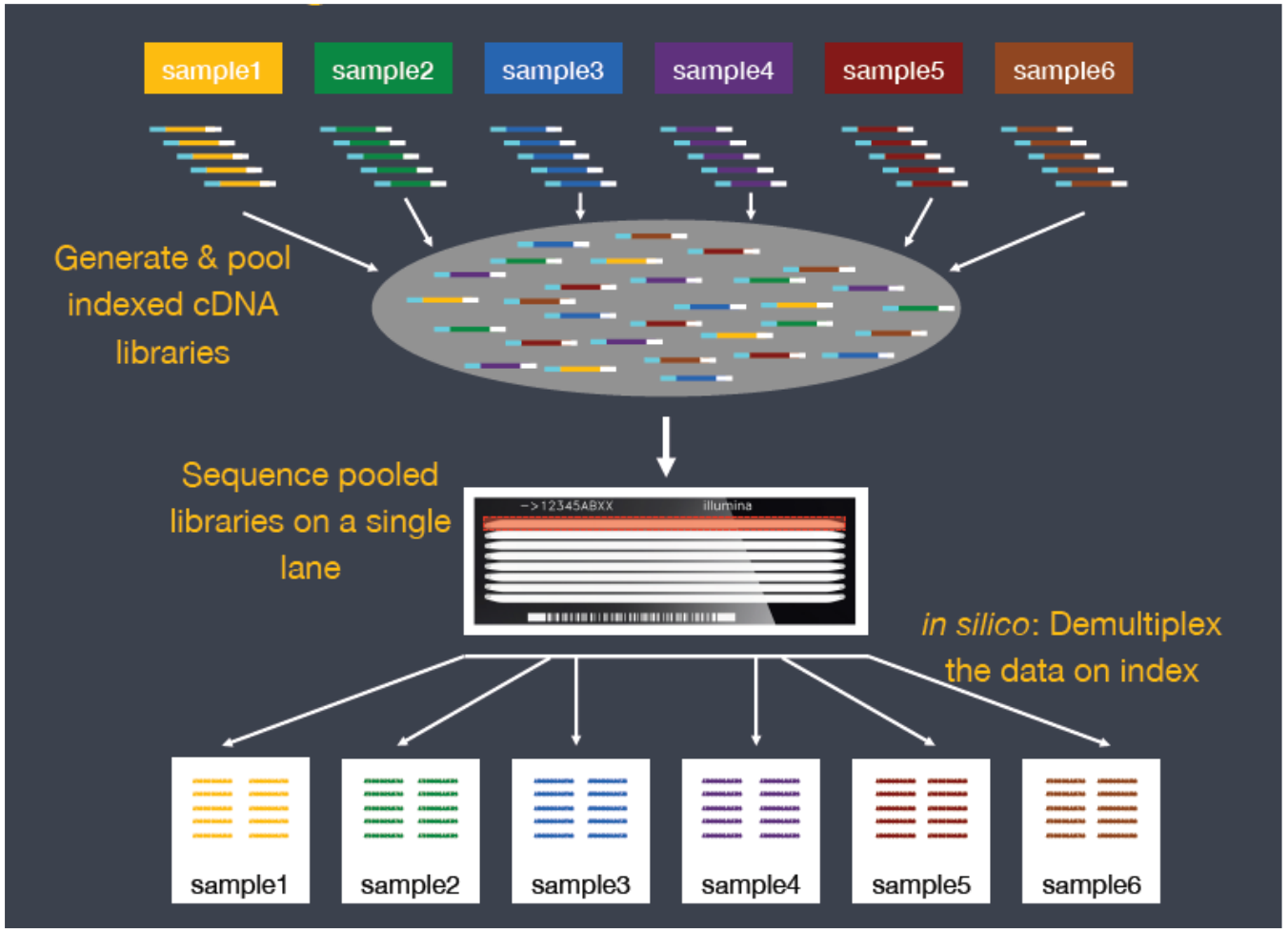
Can be useful for certain applications ( ) but costs more.

Library Strandedness



Library Preparation – Barcoding

Now a standard part of library prep – allows multiplexing of large numbers of samples.



Library Preparation – Final QC

On to the sequencing (the easy bit)...

Title

Other thoughts...

- Logistics of posting RNA or samples...
  - Frozen Samples can be sent on dry ice or dry shippe
  - RNAlater samples can be sent frozen, chilled, or if required at room temp for short trips. Where possible avoid freeze thaw cycles. Always have test samples.
  - Purified RNA needs to be sent either on dry ice, or using a vapour (dry) shipper
  - If shipping overseas... World courier are expensive but by far the safest

- Long term storage of RNA requires different techniques
  - RNA can be stored at -80 for several years but will eventually degrade.
  - Can be stored in liquid nitrogen for longer
  - For long term storage, precipitate RNA (with Acetate and ethanol) and store RNA pellet on ethanol at -80

World Courier®
AmerisourceBergen

RNA gel loading buffer

Reagent	Quantity (for 10 mL of 1.5×)	Final concentration
Formamide, ultrapure	9.5 mL	95%
Bromophenol blue (2.5%, w/v)	100 µL	0.025%
Xylene cyanol FF (2.5%, w/v)	100 µL	0.025%
EDTA (0.25 M, pH 8.0)	200 µL	5 mM

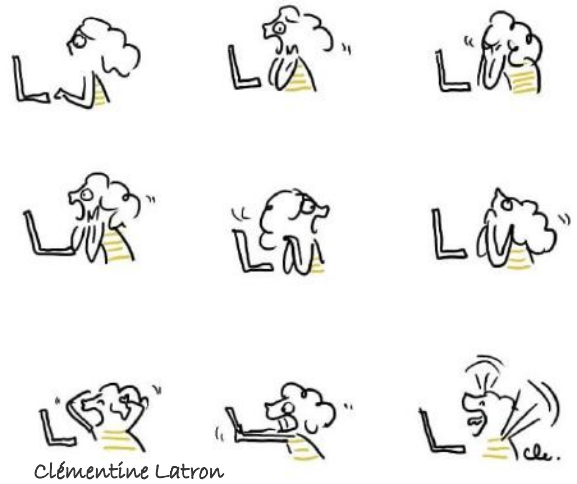
Use 5 µL for a 2.5-µL sample. Purchase a distilled, deionized preparation of formamide and the above loading dyes. Store in small (1-mL) aliquots for up to 1 yr at −20°C. This solution is available commercially (Ambion) and is recommended over homemade.

ROSLIN

grinnaqua
green innovation strategies for animal health management: towards sustainable aquaculture

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Introduction to UNIX



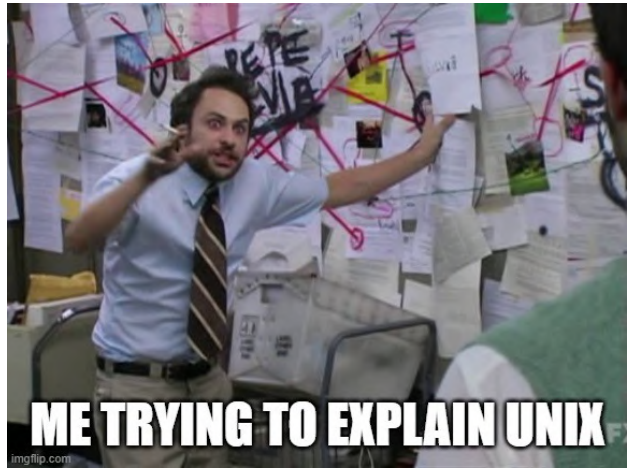
Useful links and websites
Google
Stack overflow
GitHub
https://www.unixtutorial.org/commands

clemence.fraslin@roslin.ed.ac.uk
@FraslinClemence

UKRI Biotechnology and Biological Sciences Research Council

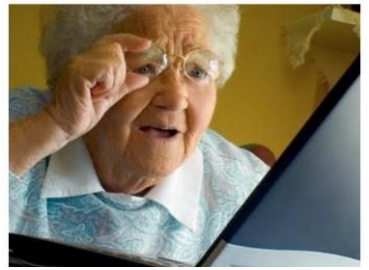
Outline

- What is UNIX and why is it useful?
- How to navigate in a UNIX environment?
- How to open/read files
- How to create, copy, move, delete a directory or a file
- How to manipulate big and complicated files in a easy way
- How edit a file
- How to submit a job
- Tips and Cheat-sheets

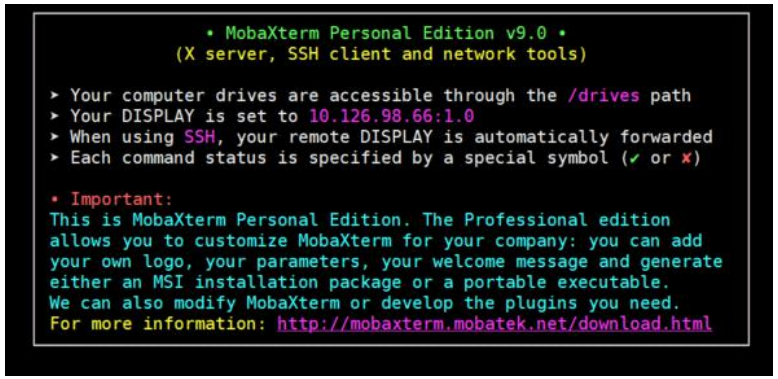


ROSLIN

What is UNIX?



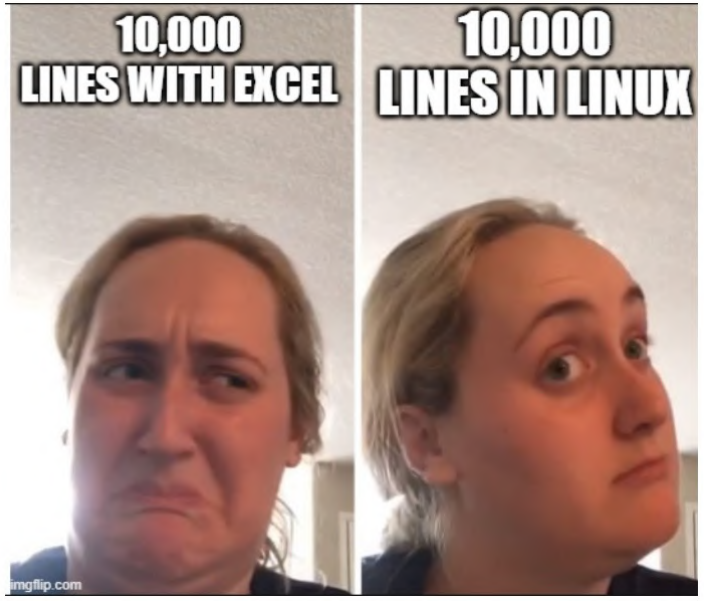
- **Operating system** = suite of programs which make a computer work (DOS / Windows) supporting multiple software and multiple users
- Created in the 1960's is coded in C language
- Can have a graphical user interface (GUI) similar to Microsoft Windows but usually we use it in a **terminal** (interface) with **operations and command lines** interpreted by the **shell**
- Through a console you can access a server (your own computer or a big computer (Eddie in Edinburgh)) using a **SSH** (**secure Shell**)
- Several version of UNIX (Linux, Solaris, MacOs X …) 
- You can use client to access shell (MobaXterm / WinSCP / PuTTY /KiTTY …)

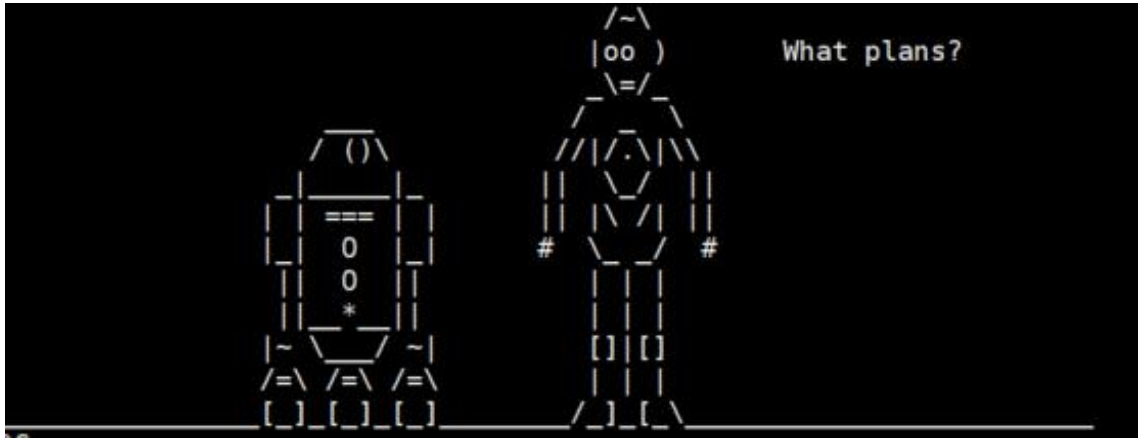


ROSLIN

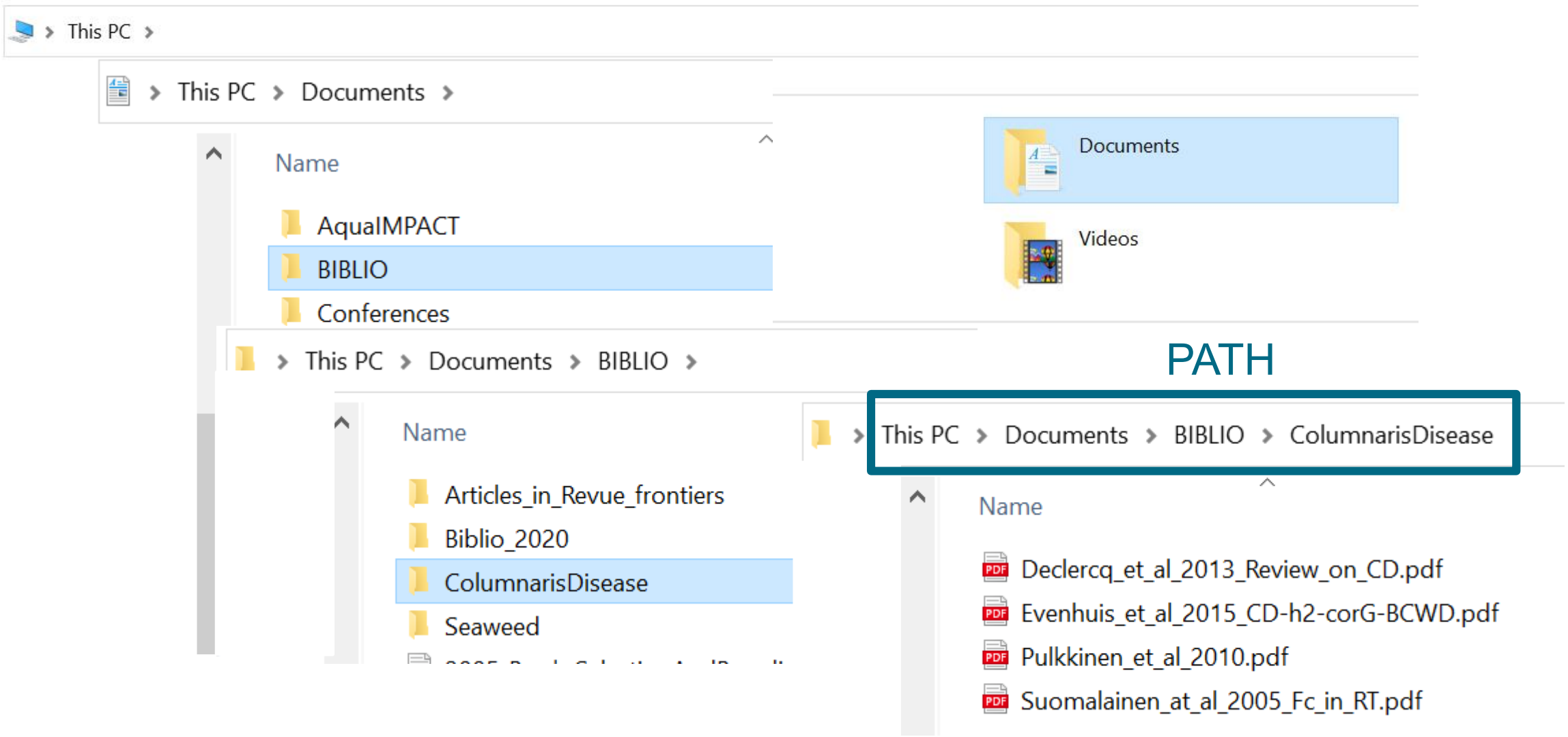
Why do we use UNIX for?

- Allows you to run multiple software/programs that are not available / have low performances on windows interface
- Gives you direct control of the computer
- Powerful to manage big data (thousands of lines that would make Excel crash)

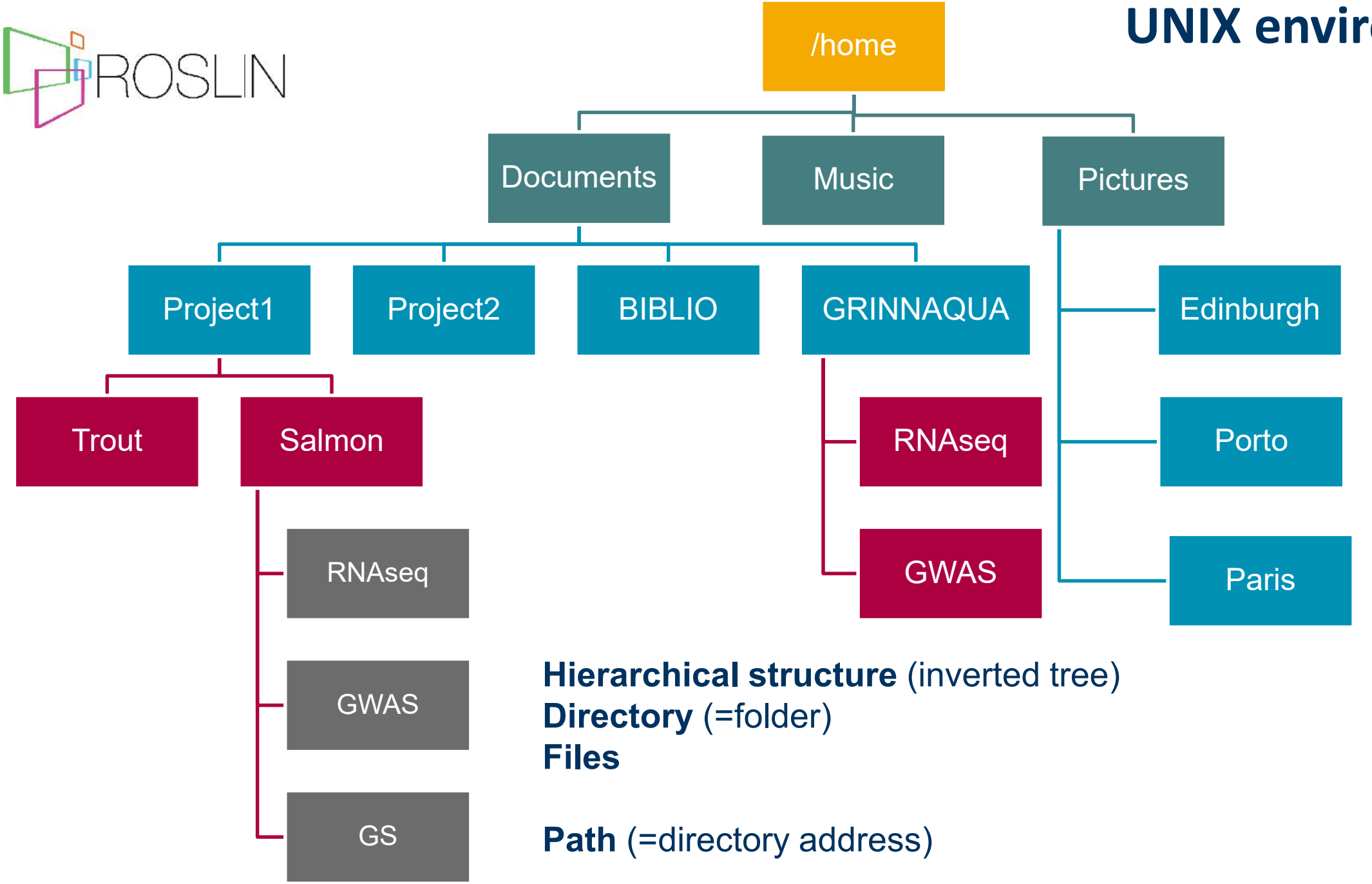




Window's environment

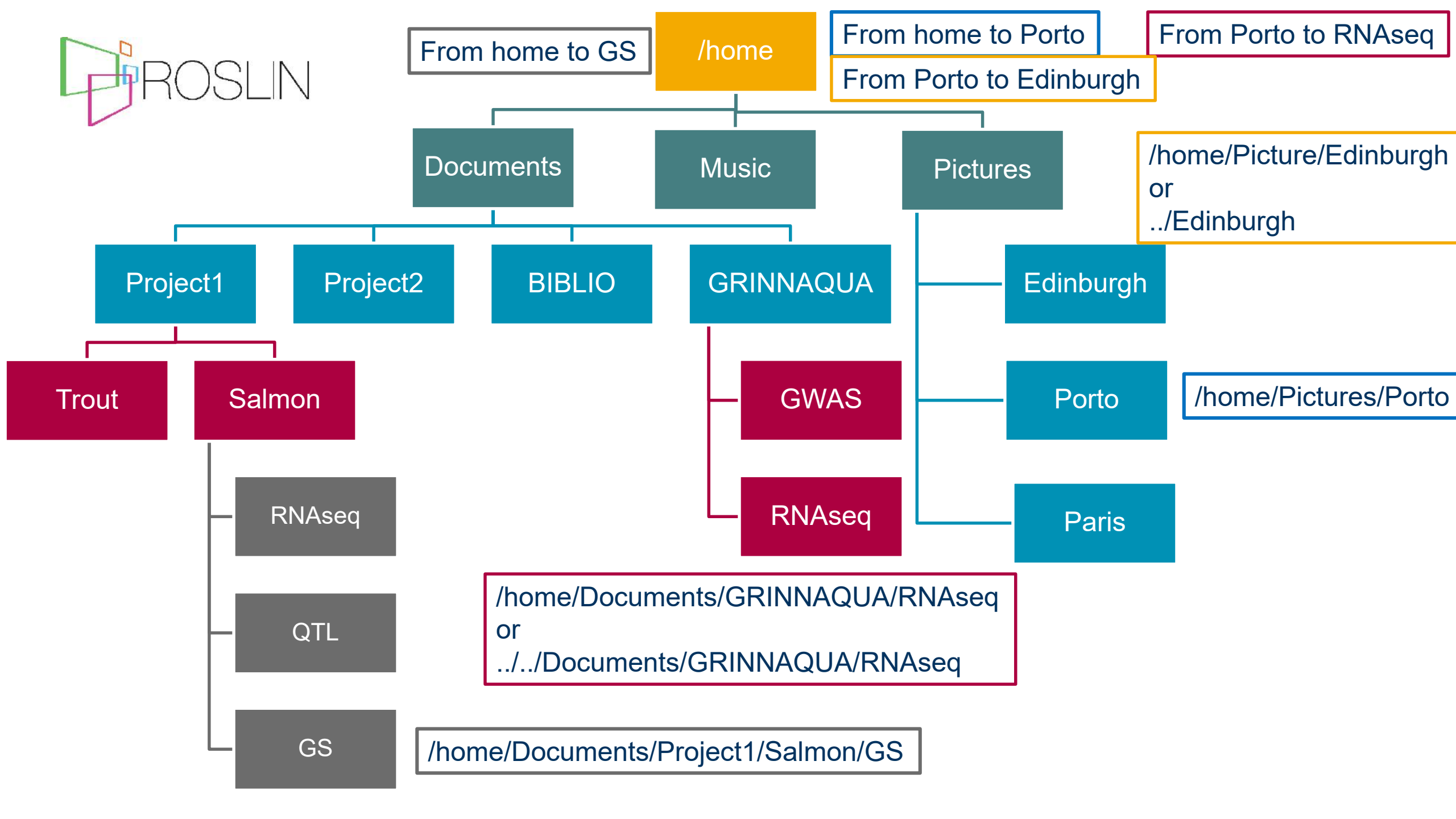


UNIX environment



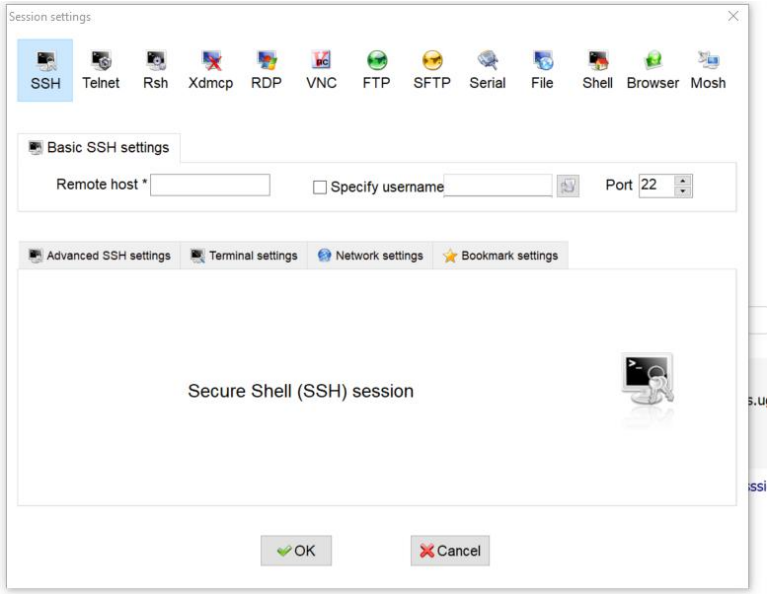
Hierarchical structure (inverted tree)
Directory (=folder)
Files

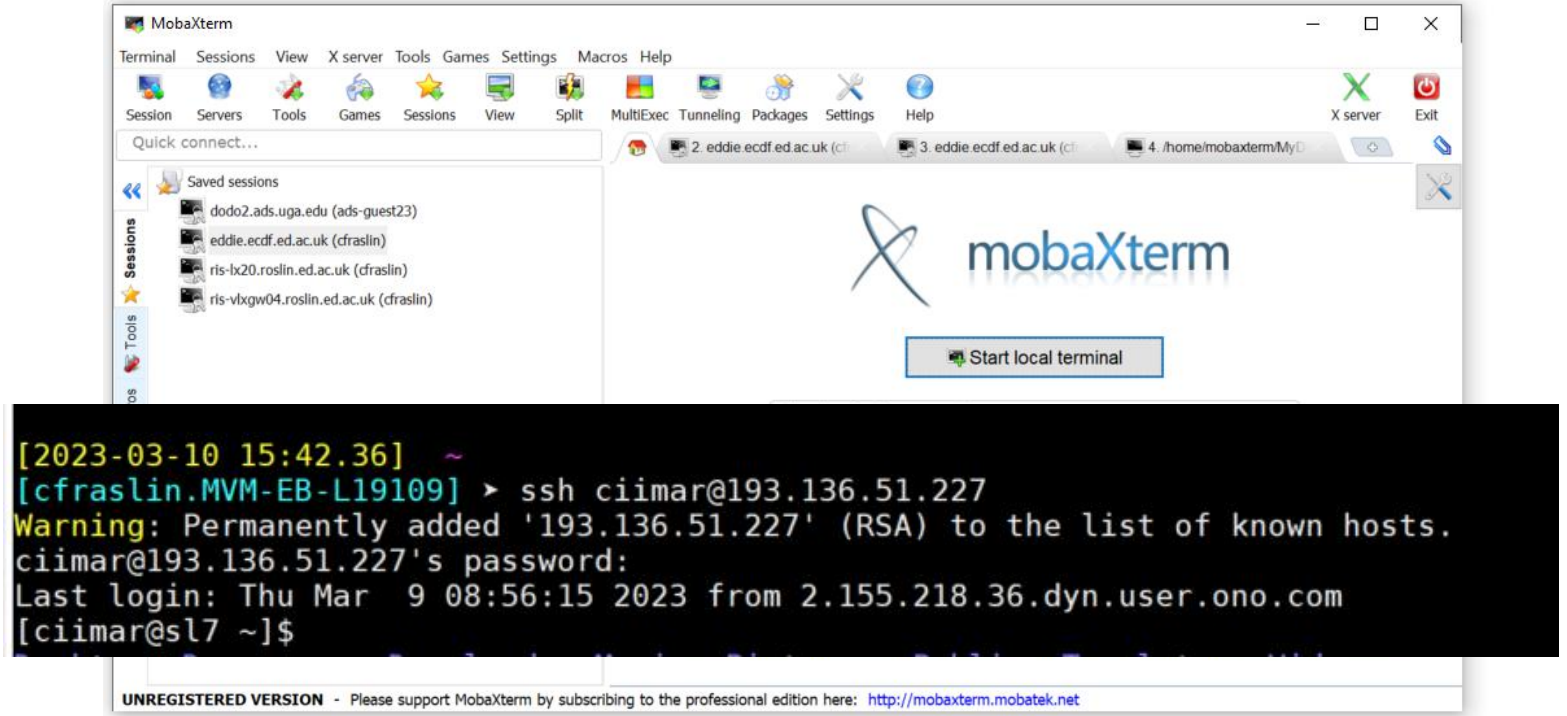
Path (=directory address)


ROSLIN
From home to GS
/home
From home to Porto
From Porto to Edinburgh
From Porto to RNAseq
Documents
Music
Pictures
/home/Picture/Edinburgh
or
../Edinburgh
Project1
Project2
BIBLIO
GRINNAQUA
Edinburgh
Trout
Salmon
GWAS
Porto
/home/Pictures/Porto
RNAseq
RNAseq
Paris
/home/Documents/GRINNAQUA/RNAseq
or
../../Documents/GRINNAQUA/RNAseq
QTL
GS
/home/Documents/Project1/Salmon/GS

Starting a UNIX terminal

- From a client you open a terminal
  - To your computer
  - To a server through an SSH connexion
  - Type command lines
  - help



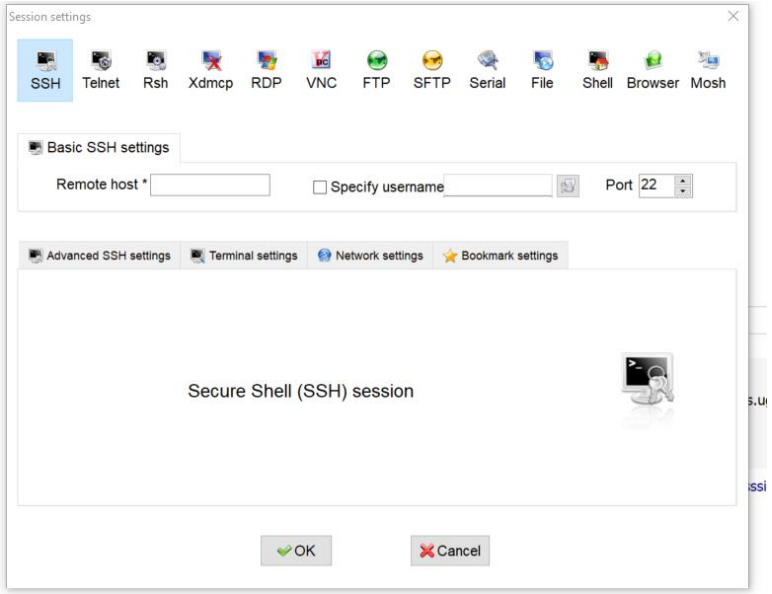


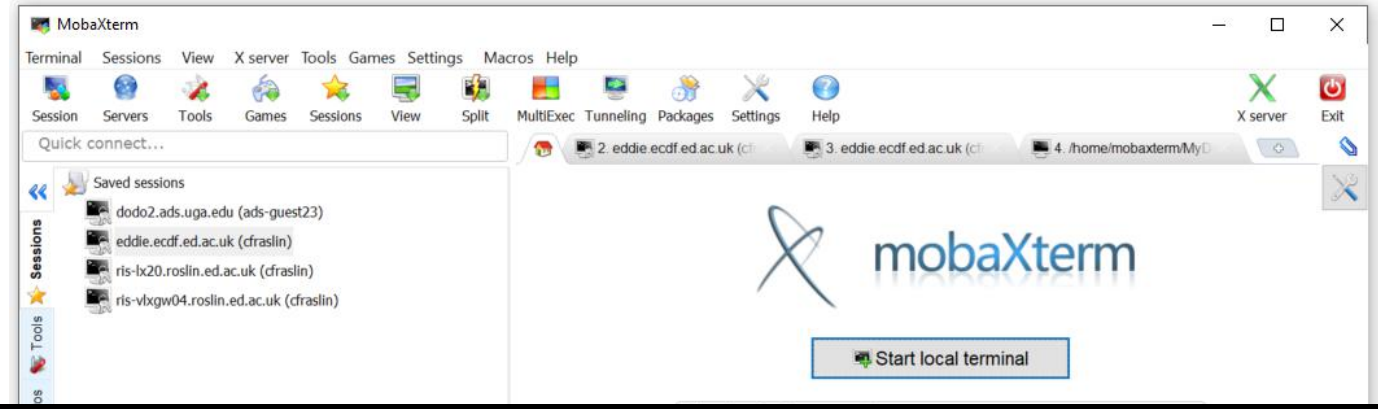

```
[ciimar@sl7 ~]$ help
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

 job_spec [&]                                                    history [-c] [-d offset] [n] or history -anrw [filename] or history>
 (( expression ))                                                if COMMANDS; then COMMANDS; [ elif COMMANDS; then COMMANDS; ]... [ >
 . filename [arguments]                                          jobs [-lnprs] [jobspec ...] or jobs -x command [args]
 :                                                               kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill >
 [ arg... ]                                                      let arg [arg ...]
 [[ expression ]]                                                local [option] name[=value] ...
 alias [-p] [name[=value] ... ]                                  logout [n]
 bg [job_spec ...]                                               mapfile [-n count] [-O origin] [-s count] [-t] [-u fd] [-C callback>
 bind [-lpvsPVS] [-m keymap] [-f filename] [-q name] [-u name] [-r ke>  popd [-n] [+N | -N]
 break [n]                                                       printf [-v var] format [arguments]
 builtin [shell-builtin [arg ...]]                               pushd [-n] [+N | -N | dir]
 caller [expr]                                                   pwd [-LP]
 case WORD in [PATTERN [| PATTERN]...) COMMANDS ;;]... esac       read [-ers] [-a array] [-d delim] [-i text] [-n nchars] [-N nchars]>
 cd [-L|[-P [-e]]] [dir]                                         readarray [-n count] [-O origin] [-s count] [-t] [-u fd] [-C callba>
 command [-pVv] command [arg ...]                                readonly [-aAf] [name[=value] ...] or readonly -p
 compgen [-abcdefgjksuv] [-o option]  [-A action] [-G globpat] [-W wo>  return [n]
 complete [-abcdefgjksuv] [-pr] [-DE] [-o option] [-A action] [-G glo>  select NAME [in WORDS ... ;] do COMMANDS; done
 compopt [-o|+o option] [-DE] [name ...]                         set [-abefhkmnptuvxBCHP] [-o option-name] [--] [arg ...]
 continue [n]                                                    shift [n]
 coproc [NAME] command [redirections]                            shopt [-pqsu] [-o] [optname ...]
 declare [-aAfFgilrtux] [-p] [name[=value] ...]
 dirs [-clpv] [+N] [-N]
 disown [-h] [-ar] [jobspec ...]
 echo [-neE] [arg ...]
 enable [-a] [-dnps] [-f filename] [name ...]
 eval [arg ...]
 exec [-cl] [-a name] [command [arguments ...]] [redirection
 exit [n]
 export [-fn] [name[=value] ...] or export -p
 false
 fc [-e ename] [-lnr] [first] [last] or fc -s [pat=rep] [comma
 fg [job_spec]
 for NAME [in WORDS ... ] ; do COMMANDS; done
 for (( exp1; exp2; exp3 )); do COMMANDS; done
 function name { COMMANDS ; } or name () { COMMANDS ; }
 getopts optstring name [arg]
 hash [-lr] [-p pathname] [-dt] [name ...]
 help [-dms] [pattern ...]
```

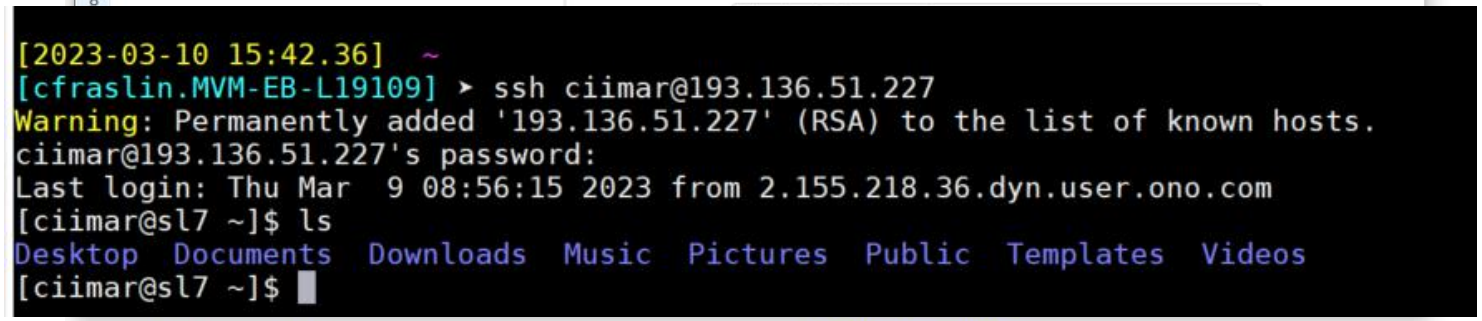
```
[ciimar@sl7 ~]$ help pwd
pwd: pwd [-LP]
    Print the name of the current working directory.

    Options:
      -L        print the value of $PWD if it names the current working
        directory
      -P        print the physical directory, without any symbolic links

    By default, `pwd' behaves as if `-L' were specified.

    { COMMANDS ; }
```

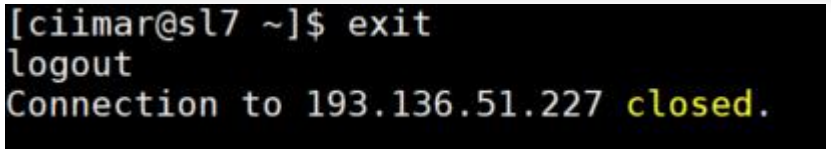
Starting a UNIX terminal

- From a client you open a terminal
  - To your computer
  - To a server through an SSH connexion
  - Type command lines
  - **help**
  - exit









How to navigate in a UNIX environment

pwd → print the path to the current directory
/home/username/DirectoryName

```
[cfraslin@login02(eddie) GRINNAQUA]$ pwd
/home/cfraslin/GRINNAQUA
```

ls → list what is in the directory you are in

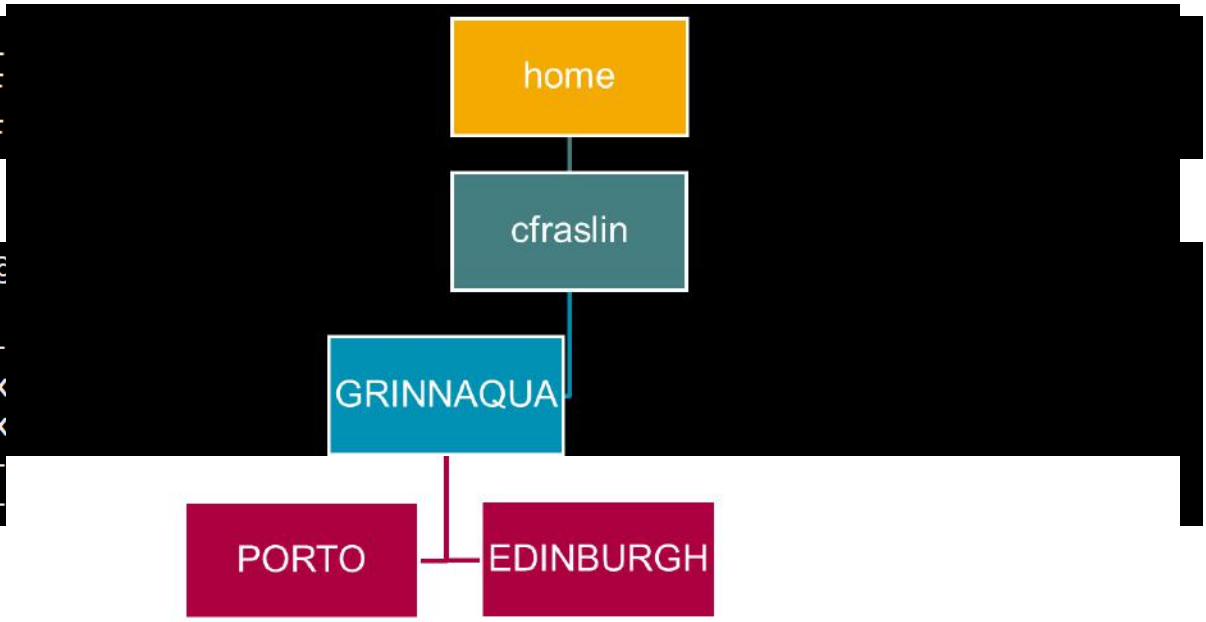
```
[cfraslin@login02(eddie) GRINNAQUA]$ ls
Confidential.txt  EDINBURGH  PORTO  ReadMe.txt  Script.sh
```

ls /PATH/ → list what is in the directory at the end of the path

```
[cfrasli
File1  F
File2  F
```

ls -lh → list + information on permissions, who created the file, the size of the files (in human readable format) and date of creation/modification

```
[cfraslin@
total 0
-rw-------
drwxr-xr-x
drwxr-xr-x
-rw-r--r--
-rwxr-xr--
```



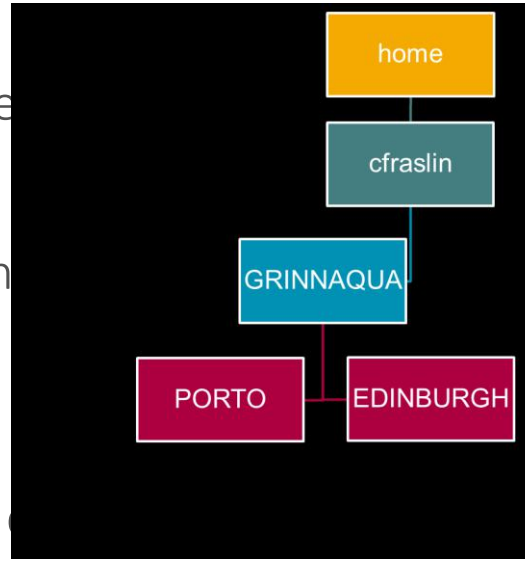
How to navigate in a UNIX environment

cd → to move betwee

cd ../ → go back to th
directory (**cd ../../../**)

cd ~ → equivalent to

cd path/ → move to path




```
[cfraslin@login02(eddie) ~]$ cd GRINNAQUA/PORTO/
[cfraslin@login02(eddie) PORTO]$ ls
File1  File3  Fish_common_latin  Fish_latin  Fish_name  Other_Fish
```

```
[cfraslin@login02(eddie) PORTO]$ cd ../
[cfraslin@login02(eddie) GRINNAQUA]$
```

```
[cfraslin@login02(eddie) PORTO]$ cd ~
[cfraslin@login02(eddie) ~]$ pwd
/home/cfraslin
```

```
[cfraslin@login02(eddie) GRINNAQUA]$ cd EDINBURGH/
[cfraslin@login02(eddie) EDINBURGH]$ cd ~/GRINNAQUA/PORTO/
[cfraslin@login02(eddie) PORTO]$
```

Auto completion using tab for file and directories names

```
[cfraslin@login02(eddie) GRINNAQUA]$ cd [tab]
EDINBURGH/ PORTO/
[cfraslin@login02(eddie) GRINNAQUA]$ cd P [tab]
PARIS/ PORTO/
```

Useful command to read files

head File1 → print the first 10 lines of File1

head -20 File1 → print the first 20 lines of File1

tail File1 → print the last 10 lines of File1

```
[cfraslin@login02(eddie) PORTO]$ ls
File1  File2  File3  Fish_common_latin  Fish_latin  Fish_name
[cfraslin@login01(eddie) PORTO]$ head File1
a b c
1 2 3
d e f
4 5 6
g h i
7 8 9
l m n
10 11 12
o p q
13 14 15
[cfraslin@login01(eddie) PORTO]$ tail -5  File1
16 17 18
u v w
19 20 21
x y z
22 23 24
```

wc File1 → counts the number of lines / words / bytes of File1

wc –l File1→ count the number of lines in File1

```
[cfraslin@login01(eddie) PORTO]$ wc File1
 16  48 112 File1
[cfraslin@login01(eddie) PORTO]$ wc -l File1
16 File1
```

Useful command to read files

less File1 → Shows on your screen the content of File1 page-by-page. To escape a less command type **q**

```
[cfraslin@login01(eddie) PORTO]$ less File1
```

```
a b c
1 2 3
d e f
4 5 6
g h i
7 8 9
l m n
10 11 12
o p q
13 14 15
r s t
16 17 18
u v w
19 20 21
x y z
22 23 24
File1 (END)
```

[q]

less -S long.txt → Shows on your screen the content of File1 line-by-line. Very useful for long lines (genotype files for eg.)

```
[cfraslin@login02(eddie) PARIS]$ less StarWars.txt
```

```
EPISODE_1 The Phantom Menace:   Turmoil has engulfed the Galactic Republic. The taxation of t
rade routes to outlying star systems is in dispute. Hoping to resolve the matter with a block
ade of deadly battleships, the greedy Trade Federation has stopped all shipping to the small
planet of Naboo. While the Congress of the Republic endlessly debates this alarming chain of
events, the Supreme Chancellor has secretly dispatched two Jedi Knights, the guardians of pea
ce and justice in the galaxy, to settle the conflict....
EPISODE_2 Attack of the clones: There is unrest in the Galactic Senate. Several thousand sola
r systems have declared their intentions to leave the Republic. This separatist movement, und
er the leadership of the mysterious Count Dooku, has made it difficult for the limited number
 of Jedi Knights to maintain peace and order in the galaxy. Senator Amidala, the former Queen
 of Naboo, is returning to the Galactic Senate to vote on the critical issue of creating an A
RMY OF THE REPUBLIC to assist the overwhelmed Jedi....
EPISODE_3 Revenge of the sith:  War! The Republic is crumbling under attacks by the ruthless
Sith Lord, Count Dooku. There are heroes on both sides. Evil is everywhere. In a stunning mov
e, the fiendish droid leader, General Grievous, has swept into the Republic capital and kidna
pped Chancellor Palpatine, leader of the Galactic Senate. As the Separatist Droid Army attemp
ts to flee the besieged capital with their valuable hostage, two Jedi Knights lead a desperat
e mission to rescue the captive Chancellor....
EPISODE_4 A new hope:   It is a period of civil war. Rebel spaceships, striking from a hidden
 base, have won their first victory against the evil Galactic Empire. During the battle, Rebe
l spies managed to steal secret plans to the Empire's ultimate weapon, the DEATH STAR, an arm
ored space station with enough power to destroy an entire planet. Pursued by the Empire's sin
ister agents, Princess Leia races home aboard her starship, custodian of the stolen plans tha
t can save her people and restore freedom to the galaxy....
EPISODE_5 The Empire strikes back:      It is a dark time for the Rebellion. Although the Dea
```

Useful command to read files

less File1 → Shows on your screen the content of File1 page-by-page. To escape a less command type **q**

```
[cfraslin@login01(eddie) PORTO]$ less File1
```

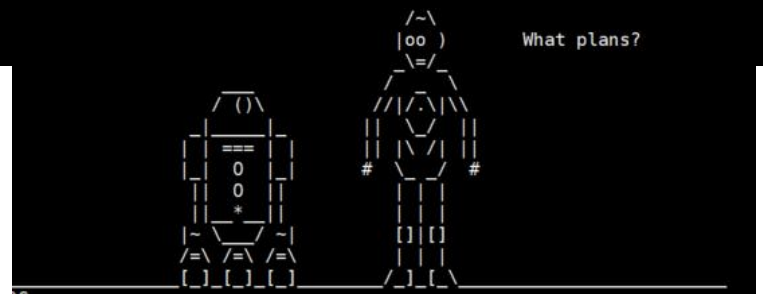
```
a b c
1 2 3
d e f
4 5 6
g h i
7 8 9
l m n
10 11 12
o p q
13 14 15
r s t
16 17 18
u v w
19 20 21
x y z
22 23 24
File1 (END)
```

[q]

less -S long.txt → Shows on your screen the content of File1 line-by-line. Very useful for long lines (genotype files for eg.)

```
[cfraslin@login02(eddie) PARIS]$ less -S StarWars.txt
```

```
EPISODE_1 The Phantom Menace:   Turmoil has engulfed the Galactic Republic. The taxat
EPISODE_2 Attack of the clones: There is unrest in the Galactic Senate. Several thous
EPISODE_3 Revenge of the sith:  War! The Republic is crumbling under attacks by the r
EPISODE_4 A new hope:   It is a period of civil war. Rebel spaceships, striking from
EPISODE_5 The Empire strikes back:      It is a dark time for the Rebellion. Although
EPISODE_6 Return of the Jedi:   Luke Skywalker has returned to his home planet of Tat
EPISODE_7 The Force Awakens:    Luke Skywalker has vanished. In his absence, the sini
EPISODE_8 The Last Jedi:        The FIRST ORDER reigns. Having decimated the peaceful
EPISODE_9 The rise of Skywalker:The dead speak! The galaxy has heard a mysterious bro
StarWars.txt (END)
```



Useful command to read files

less File1 → Shows on your screen the content of File1 page-by-page. To escape a less command type **q**

less -S File1 → Shows on your screen the content of File1 line-by-line. Very useful for long lines (genotype files for eg.)

```
[cfraslin@login01(eddie) PORTO]$ less File1
```

```
a b c
1 2 3
d e f
4 5 6
g h i
7 8 9
l m n
10 11 12
o p q
13 14 15
r s t
16 17 18
u v w
19 20 21
x y z
22 23 24
File1 (END)
```

[q]

head Fish_* → print the first 10 lines of all files starting in Fish_

```
[cfraslin@login01(eddie) PORTO]$ head Fish_*
==> Fish_common_latin <==
trout O. mykiss
salmon S. salar
tilapia O. niloticus
seabass D. labrax
seabream S. aurata
carp C. carpio
cod G. morhua

==> Fish_latin <==
C. carpio
S. salar
O. niloticus
C. gigas
P. monodon

==> Fish_name <==
Common Carp
Atlantic Salmon
Nile Tilapia
Pacific Oyster
Tiger Shrimp
```

Copy, create, remove files and directories

cp to copy and paste

cp → copy files or **cp –r** → copy directory

(-r recursive: apply to everything after the command)

Copy with the same name in a new folder

cp path/FileName NewPath/.

Good practice: Copy and "rename"

cp path/FileName NewPath/FileName2

mv → **move** a file or directory (cut and paste)

mv Directory /path/Directory

mv → rename files/directory

mv FileName1 NewFileName

```
[cfraslin@login02(eddie) GRINNAQUA]$ ls EDINBURGH/
File1  File2
[cfraslin@login02(eddie) GRINNAQUA]$ ls PORTO/
File1  File4  Fish_common_latin  Fish_latin  Fish_name
```

```
[cfraslin@login02(eddie) GRINNAQUA]$ cp EDINBURGH/File2 PORTO/.
[cfraslin@login02(eddie) GRINNAQUA]$ cp EDINBURGH/File1 PORTO/File
File1  File2  File4
[cfraslin@login02(eddie) GRINNAQUA]$ cp EDINBURGH/File1 PORTO/File3
```

```
[cfraslin@login02(eddie) GRINNAQUA]$ ls PORTO/
File1  File2  File3  File4  Fishes  Fish_latin  Fish_name
```

```
[cfraslin@login02(eddie) PORTO]$ ls
File1  File2  File3  File4  Fishes  Fish_latin  Fish_name  More_fish  Other_Fish
[cfraslin@login02(eddie) PORTO]$ mv Fishes Fish_common_latin
[cfraslin@login02(eddie) PORTO]$ ls
File1  File2  File3  File4  Fish_common_latin  Fish_latin  Fish_name  More_fish  Other_Fish
```

Copy, create, remove files and directories

rm → delete files or directory

rm FileName

rm –r DirectoryName (**-r** recursive this directory and all the subdirectories)

rm File* (delete all files starting by File)

rm *.pdf (delete all the files ending in pdf)

→ **Use with caution the rm ***

```
[cfraslin@login01(eddie) EDINBURGH]$ ls
File1  File2  ReadMe.txt
[cfraslin@login01(eddie) EDINBURGH]$ rm ReadMe.txt
[cfraslin@login01(eddie) EDINBURGH]$ ls
File1  File2
```

mkdir → make new directory

mkdir NewDirectoryName

```
[cfraslin@login01(eddie) GRINNAQUA]$ ls
Confidential.txt  EDINBURGH  PORTO  ReadMe.txt  Script.sh
[cfraslin@login01(eddie) GRINNAQUA]$ mkdir PARIS
[cfraslin@login01(eddie) GRINNAQUA]$ ls
Confidential.txt  EDINBURGH  PARIS  PORTO  ReadMe.txt  Script.sh
```

rmdir → remove directory

rmdir DirectoryName

```
[cfraslin@login01(eddie) GRINNAQUA]$ rm PARIS/
rm: cannot remove 'PARIS/': Is a directory
[cfraslin@login01(eddie) GRINNAQUA]$ rm -r PARIS/
[cfraslin@login01(eddie) GRINNAQUA]$ ls
Confidential.txt  EDINBURGH  PORTO  ReadMe.txt  Script.sh
```

Text processing tools

paste → paste lines of two files

```
[cfraslin@login01(eddie) PORTO]$ paste File3 File4 > Other_Fish
```

```
==> File3 <==
trout
salmon
tilapia
Seabass
Oyster
Salmon
Carp
trout

==> File4 <==
O. mykiss
S. salar
O. niloticus
D. labrax
C. gigas
O. keta
C. carpio
S. trutta
```

```
[cfraslin@login01(eddie) PORTO]$ head Other_Fish
trout   O. mykiss
salmon  S. salar
tilapia O. niloticus
Seabass D. labrax
Oyster  C. gigas
Salmon  O. keta
Carp    C. carpio
trout   S. trutta
```

```
[cfraslin@login02(eddie) PORTO]$ wc -l File3
8 File3
[cfraslin@login02(eddie) PORTO]$ wc -l File4
8 File4
[cfraslin@login02(eddie) PORTO]$
```

Text processing tools

cat File1 File2 > File3 → concatenates File2 after File1 in a new file File3

```
[cfraslin@login02(eddie) PORTO]$ head Fish_common_latin
trout O. mykiss
salmon S. salar
tilapia O. niloticus
seabass D. labrax
seabream S. aurata
carp C. carpio
cod G. morhua
[cfraslin@login02(eddie) PORTO]$ head File2
Oyster C. gigas
Salmon S. salar
Carp C. carpio
```

```
[cfraslin@login02(eddie) PORTO]$ cat File2 Fish_common_latin > More_fish
[cfraslin@login02(eddie) PORTO]$ head -20 More_fish
Oyster C. gigas
Salmon S. salar
Carp C. carpio
trout O. mykiss
salmon S. salar
tilapia O. niloticus
seabass D. labrax
seabream S. aurata
carp C. carpio
cod G. morhua
```

Text processing tools

cat File1 File2 > File3 → concatenates File2 after File1 in a new file File3

cat File4 >> File3 → Add the content of File4 in File3

`>` → create and if exist replace `>>` → add at the end

```
[cfraslin@login02(eddie) PORTO]$ cat Other_Fish >> More_fish
[cfraslin@login02(eddie) PORTO]$ tail More_fish
carp C. carpio
cod G. morhua
trout   O. mykiss
salmon  S. salar
tilapia O. niloticus
Seabass D. labrax
Oyster  C. gigas
Salmon  O. keta
Carp    C. carpio
trout   S. trutta
```

```
[cfraslin@login02(eddie) PORTO]$ head Other_Fish
trout   O. mykiss
salmon  S. salar
tilapia O. niloticus
Seabass D. labrax
Oyster  C. gigas
Salmon  O. keta
Carp    C. carpio
trout   S. trutta
```

```
[cfraslin@login02(eddie) PORTO]$ head -20 More_fish
Oyster C. gigas
Salmon S. salar
Carp C. carpio
trout O. mykiss
salmon S. salar
tilapia O. niloticus
seabass D. labrax
seabream S. aurata
carp C. carpio
cod G. morhua
```

Text processing tools

grep pattern File → finds lines that contains the pattern in File. (**zgrep** for gzip files)

-n for the line number

-i for case insensitive

-c to count the number of lines

-e to find a pattern with space

-v all lines that do NOT match the pattern

```
[cfraslin@login02(eddie) EDINBURGH]$ head File5
trout O. mykiss
salmon S. salar
tilapia O. niloticus
Seabass D. labrax
Oyster C. gigas
Salmon S. salar
Carp C. carpio
trout O. mykiss
[cfraslin@login02(eddie) EDINBURGH]$ grep sal File5
salmon S. salar
Salmon S. salar
[cfraslin@login02(eddie) EDINBURGH]$ grep -n sal File5
2:salmon S. salar
7:Salmon S. salar
[cfraslin@login02(eddie) EDINBURGH]$ grep -i sal File5
salmon S. salar
Salmon S. salar
[cfraslin@login02(eddie) EDINBURGH]$ grep -c sal File5
2
[cfraslin@login02(eddie) EDINBURGH]$ grep -e "S. salar" File5
salmon S. salar
Salmon S. salar
```

Text processing tools

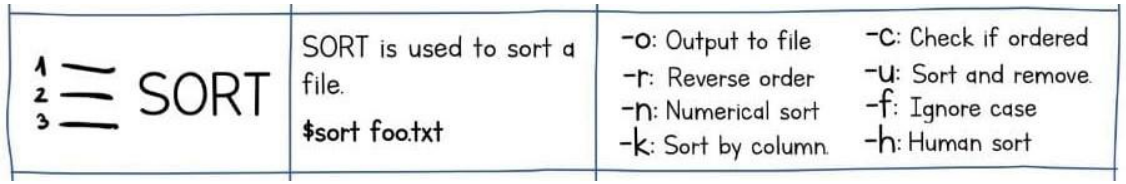
sort → sort a file

```
[cfraslin@login02(eddie) PORTO]$ sort More_fish > More_fish_Sorted
```

Alphabetically, on column 1

```
Oyster C. gigas
Salmon S. salar
Carp C. carpio
trout O. mykiss
salmon S. salar
tilapia O. niloticus
seabass D. labrax
seabream S. aurata
carp C. carpio
cod G. morhua
trout O. mykiss
salmon S. salar
tilapia O. niloticus
Seabass D. labrax
Oyster C. gigas
Salmon O. keta
Carp C. carpio
trout S. trutta
More_fish (END)
```

```
carp C. carpio
Carp C. carpio
Carp C. carpio
cod G. morhua
Oyster C. gigas
Oyster C. gigas
Salmon O. keta
salmon S. salar
salmon S. salar
Salmon S. salar
seabass D. labrax
Seabass D. labrax
seabream S. aurata
tilapia O. niloticus
tilapia O. niloticus
trout O. mykiss
trout O. mykiss
trout S. trutta
More_fish_Sorted (END)
```




```
[cfraslin@login02(eddie) PORTO]$ sort -k 2 More_fish
carp C. carpio
Carp C. carpio
Carp C. carpio
Oyster C. gigas
Oyster C. gigas
seabass D. labrax
Seabass D. labrax
cod G. morhua
Salmon O. keta
trout O. mykiss
trout O. mykiss
tilapia O. niloticus
tilapia O. niloticus
seabream S. aurata
salmon S. salar
salmon S. salar
Salmon S. salar
trout S. trutta
```

Text processing tools

uniq → retains unique lines on a **sorted** file

```
[cfraslin@login02(eddie) PORTO]$ uniq More_fish_Sorted > Uniq_fish
```

Case sensitive

```
[cfraslin@login02(eddie) PORTO]$ uniq -i More_fish_Sorted > Uniq_fish_caseINsensitive
```

```
carp C. carpio
Carp C. carpio
Carp C. carpio
cod G. morhua
Oyster C. gigas
Oyster C. gigas
Salmon O. keta
salmon S. salar
salmon S. salar
Salmon S. salar
seabass D. labrax
Seabass D. labrax
seabream S. aurata
tilapia O. niloticus
tilapia O. niloticus
trout O. mykiss
trout O. mykiss
trout S. trutta
More_fish_Sorted (END)
```

```
carp C. carpio
Carp C. carpio
cod G. morhua
Oyster C. gigas
Salmon O. keta
salmon S. salar
Salmon S. salar
seabass D. labrax
Seabass D. labrax
seabream S. aurata
tilapia O. niloticus
trout O. mykiss
trout S. trutta
Uniq_fish (END)
```

```
carp C. carpio
cod G. morhua
Oyster C. gigas
Salmon O. keta
salmon S. salar
seabass D. labrax
seabream S. aurata
tilapia O. niloticus
trout O. mykiss
trout S. trutta
Uniq_fish_caseINsensitive (END)
```

```
[cfraslin@login02(eddie) PORTO]$ wc -l More_fish_Sorted
18 More_fish_Sorted
[cfraslin@login02(eddie) PORTO]$ wc -l Uniq_fish*
 13 Uniq_fish
 10 Uniq_fish_caseINsensitive
 23 total
```

Create and edit a file

Create a file:

- touch filename
- echo “start writing” > filename

```
[cfraslin@login02(eddie) PARIS]$ touch Eiffel
[cfraslin@login02(eddie) PARIS]$ ls
Eiffel  StarWars.txt
```

```
[cfraslin@login02(eddie) PARIS]$ echo "It's just a big church" > NotreDame
```

```
[cfraslin@login02(eddie) PARIS]$ ls
Eiffel  NotreDame  StarWars.txt
```

```
[cfraslin@login02(eddie) PARIS]$ head NotreDame
It's just a big church
[cfraslin@login02(eddie) PARIS]$ head Eiffel
[cfraslin@login02(eddie) PARIS]$
```

Read and edit a file

nano

geany

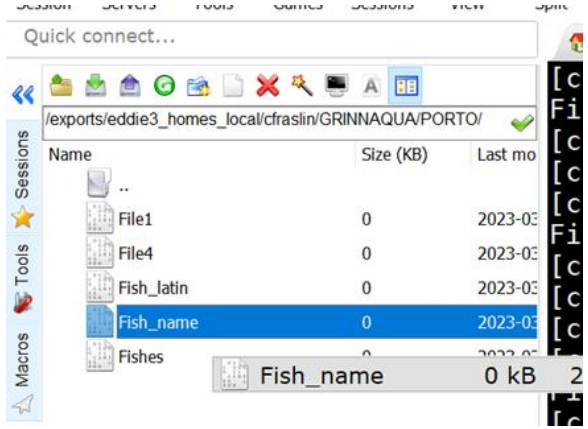
nano

Vi

xedit

nedit

Using your client (MobaXterm)



https://www.nano-editor.org/docs.php

Nano Shortcuts

Files	
Ctrl-R	Read file
Ctrl-O	Save file
Ctrl-X	Close file
Cut and Paste	
ALT-A	Start marking text
CTRL-K	Cut marked text or line
CTRL-U	Paste text
Navigate File	
ALT-/	End of file
CTRL-A	Beginning of line
CTRL-E	End of line
CTRL-C	Show line number
CTRL-_	Go to line number
Search File	
CTRL-W	Find
ALT-W	Find next
CTRL-\	Search and replace

More nano info at:
http://www.nano-editor.org/docs.php

VI Editing Commands

Command	Description
i	Insert at cursor (goes into insert mode)
a	Write after cursor (goes into insert mode)
A	Write at the end of line (goes into insert mode)
ESC	Terminate insert mode
u	Undo last change
U	Undo all changes to the entire line
o	Open a new line (goes into insert mode)
dd	Delete line
3dd	Delete 3 lines
D	Delete contents of line after the cursor
C	Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
dw	Delete word
4dw	Delete 4 words
cw	Change word
x	Delete character at the cursor
r	Replace character
R	Overwrite characters from cursor onward
s	Substitute one character under cursor continue to insert
S	Substitute entire line and begin to insert at the beginning of the line
~	Change case of individual character

Submit jobs

In an interactive environment you can submit using ./

./job_script.sh

The **qsub** command scan you script.sh file line by line and execute the commands inside

qsub job_script.sh

See the status of the jobs you submitted in a qsub

qstat

```
[cfraslin@login02(eddie) PORTO]$ qstat
job-ID     prior   name       user         state submit/start at     queue
             jclass                          slots ja-task-ID
-----------------------------------------------------------------------------
-------------------------------------------------------
  28674235 0.00128 Happlotype cfraslin     r     03/15/2023 11:05:58 eddie@node3
cdf.ed.ac.uk                                    1
  28674239 0.00127 Happlotype cfraslin     r     03/15/2023 11:07:29 eddie@node3
cdf.ed.ac.uk                                    1 1
  28674239 0.00127 Happlotype cfraslin     r     03/15/2023 11:08:18 eddie@node3
cdf.ed.ac.uk                                    1 2
  28674239 0.00127 Happlotype cfraslin     r     03/15/2023 11:09:55 eddie@node3
```

Delete a job

qdel job_id

Tips: if you run a job in an interactive session and you want to keep a record of the ouptut use the command **tee** to print what appears on the screen in a file. | is called a pipe

./job_script.sh | tee job.lst

To exit a program

- **exit** → exit a session and many programs
- **quit** or **q** → quit many programs
- **crtl d** or **crtl x** → end of file
- **crtl c** → interrupt a program
- **crtl z** → suspend a program

Tips and other useful commands

Auto completion → [TAB]

history → show history of previous command

Previous commands → **top arrow**

df → present statistics (including free disk space) on all file

du –k → Shows space used in the current directory and its subdirectories

dos2unix → format a file for unix (if created on windows)

expand → replace TAB with spaces

clear → clears the terminal (does not clear the history)

chmod → change permission of the file.

Tips and other useful commands

File permissions

chmod → change permission of the file.

chmod u+x file to give permission to execute a file *

User
Group's member
Other (all)

r = read
w = write
x = execute

```
-rw-------  1 cfraslin eddie users   8 Feb 17 14:03 Confidential.txt
-rw-rw-rw-  1 cfraslin eddie_users   8 Feb 17 11:47 ReadMe.txt
-rw-r-xr--  1 cfraslin eddie_users   7 Feb 17 11:47 Script.sh
```

```
-rw-r-xr--  1 cfraslin eddie_users   7 Feb 17 11:47 Script.sh
[cfraslin@login02(eddie) GRINNAQUA]$ chmod u+x Script.sh
[cfraslin@login02(eddie) GRINNAQUA]$ ls -lh Script.sh
-rwxr-xr--  1 cfraslin eddie_users 7 Feb 17 11:47 Script.sh
```

*Change permission using numbers
4 read
2 write
1 execute
chmod 777 file → give all permissions to the file

Cheat sheets

https://cheatography.com/davechild/cheat-sheets/linux-command-line/

Directory Operations

pwd	Show current directory
mkdir *dir*	Make directory *dir*
cd *dir*	Change directory to *dir*
cd ..	Go up a directory
ls	List files

ls Options

-a	Show all (including hidden)
-R	Recursive list
-r	Reverse order
-t	Sort by last modified
-S	Sort by file size
-l	Long listing format
-1	One file per line
-m	Comma-separated output
-Q	Quoted output

Search Files

grep *pattern files*	Search for *pattern* in *files*
grep -i	Case insensitive search
grep -r	Recursive search
grep -v	Inverted search
grep -o	Show matched part of file only
find */dir/* -name *name**	Find files starting with *name* in *dir*
find */dir/* -user *name*	Find files owned by *name* in *dir*
find */dir/* -mmin *num*	Find files modifed less than *num* minutes ago in *dir*
whereis *command*	Find binary / source / manual for *command*
locate *file*	Find *file* (quick search of system index)

File Operations

touch *file1*

Create *file1*

cat *file1 file2*

Concatenate files and output

less *file1*

View and paginate *file1*

file *file1*

Get type of *file1*

cp *file1 file2*

Copy *file1* to *file2*

mv *file1 file2*

Move *file1* to *file2*

rm *file1*

Delete *file1*

head *file1*

Show first 10 lines of *file1*

tail *file1*

Show last 10 lines of *file1*

tail -F *file1*

Output last lines of *file1* as it changes

Cheat sheets

https://cheatography.com/davechild/cheat-sheets/linux-command-line/

https://www.guru99.com/linux-commands-cheat-sheet.html

Google

Linux Text Processing Tools

GREP	GREP allows you to search patterns in files. ZGREP for GZIP files. **$grep <pattern> file.log**	-n: Number of lines that matches -i: Case insensitive -v: Invert matches -E: Extended regex -c: Count number of matches -l: Find filenames that matches the pattern
NGREP	NGREP is used for analyzing network packets. **$ngrep -I file.pcap**	-d: Specify network interface -i: Case insensitive. -x: Print in alternate hexdump -t: Print timestamp -I: Read pcap file
CUT	The CUT command is used to parse fields from delimited logs. **$cut -d ":" -f 2 file.log**	-d: Use the field delimiter -f: The field numbers -c: Specifies characters position
SED	SED (Stream Editor) is used to replace strings in a file. **$sed s/regex/replace/g**	s: Search g: Replace d: Delete w: Append to file -e: Execute command -n: Suppress output
SORT	SORT is used to sort a file. **$sort foo.txt**	-o: Output to file -r: Reverse order -n: Numerical sort -k: Sort by column. -c: Check if ordered -u: Sort and remove. -f: Ignore case -h: Human sort
UNIQ	UNIQ is used to extract uniq occurrences. **$uniq foo.txt**	-c: Count the number of duplicates -d: Print duplicates -i: Case insensitive
DIFF	DIFF is used to display differences in files by comparing line by line. **$diff foo.log bar.log**	How to read output? a: Add c: Change d: Delete #: Line numbers <: File 1 >: File 2
AWK	AWK is a programming language use to manipulate data. **$awk {print $2} foo.log**	Print first column with separator ":" **$awk -F: '{print $1}' /etc/passwd** Extract uniq value from two files: awk 'FNR==NR {a[$0]++; next} !($0 in a)' f1.txt f2.txt

ROSLIN

grinnaqua
green innovation strategies for animal health management: towards sustainable aquaculture

Thank you!

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

UKRI Biotechnology and Biological Sciences Research Council

Commands and pipes

Pipes

cmd1 | *cmd2*

stdout of *cmd1* to *cmd2*

cmd1 |& *cmd2*

stderr of *cmd1* to *cmd2*

Command Lists

cmd1 ; *cmd2*

Run *cmd1* then *cmd2*

cmd1 && *cmd2*

Run *cmd2* if *cmd1* is successful

cmd1 || *cmd2*

Run *cmd2* if *cmd1* is not successful

cmd &

Run *cmd* in a subshell

Linux Command Cheat Sheet

Share This Cheat Sheet

Basic commands

|	Pipe (redirect) output
sudo [command]	run < command> in superuser mode
nohup [command]	run < command> immune to hangup signal
man [command]	display help pages of < command>
[command] &	run < command> and send task to background
>> [fileA]	append to fileA, preserving existing contents
> [fileA]	output to fileA, overwriting contents
echo -n	display a line of text
xargs	build command line from previous output
1>2&	Redirect stdout to stderr
fg %N	go to task N
jobs	list task
ctrl-z	suspend current task

File permission

chmod -c -R	chmod file read, write and executable permission
touch -a -t	modify (or create) file timestamp
chown -c -R	change file ownership
chgrp -c -R	change file group permission
touch -a -t	modify (or create) file timestamp

Network

netstat -r -v	print network information, routing and connections
telnet	user interface to the TELNET protocol
tcpdump	dump network traffic
ssh -i	openSSH client
ping -c	print routing packet trace to host network

File management

find	search for a file
ls -a -C -h	list content of directory
rm -r -f	remove files and directory
locate -i	find file, using updatedb(8) database
cp -a -R -i	copy files or directory
du -s	disk usage
file -b -i	identify the file type
mv -f -i	move files or directory
grep, egrep, fgrep -i -v	print lines matching pattern

File compression

tar xvfz	create or extract .tar or .tgz files
gzip, gunzip, zcat	create, extract or view .gz files
uuencode, uudecode	create or extract .Z files
zip, unzip -v	create or extract .ZIP files
rpm	create or extract .rpm files
bzip2, bunzip2	create or extract .bz2 files
rar	create or extract .rar files

File Editor

ex	basic editor
vi	visual editor
nano	pico clone
view	view file only
emacs	extensible, customizable editor
sublime	yet another text editor
sed	stream editor
pico	simple editor

Directory Utilities

mkdir	create a directory
rmdir	remove a directory

File Utilities

tr -d	translate or delete character
uniq -c -u	report or omit repeated lines
split -l	split file into pieces
wc -w	print newline, word, and byte counts for each file
head -n	output the first part of files
cut -s	remove section from file
diff -q	file compare, line by line
join -i	join lines of two files on a common field
more, less	view file content, one page at a time
sort -n	sort lines in text file
comm -3	compare two sorted files, line by line
cat -s	concatenate files to the standard output
tail -f	output last part of the file

Scripting

awk, gawk	pattern scanning
tsh	tiny shell
" "	anything within double quotes is unchanged except \ and $
' '	anything within single quote is unchanged
python	"object-oriented programming language"
bash	GNU bourne-again SHell
ksh	korn shell
php	general-purpose scripting language
csh, tcsh	C shell
perl	Practical Extraction and Report Language
source [file]	load any functions file into the current shell, requires the file to be executable

Memory & Processes

free -m	display free and used system memory
killall	stop all process by name
sensors	CPU temperature
top	display current processes, real time monitoring
kill -1 -9	send signal to process
service [start|stop|restart]	manage or run sysV init script
ps aux	display current processes, snapshot
dmesg -k	display system messages

Disk Utilities

df -h, -i	File system usage
mkfs -t -V	create file system
resize2fs	update a filesystem, after lvextend*
fsck -A -N	file system check & repair
pvcreate	create physical volume
mount -a -t	mount a filesystem
fdisk -l	edit disk partition
lvcreate	create a logical volume
umount -f -v	umount a filesystem

Misc Commands

pwd -P	print current working directory
bc	high precision calculator
expr	evaluate expression
cal	print calender
export	assign or remove environment variable
` [command]	backquote, execute command
date -d	print formatted date
${variable}	if set, access the variable

Read the Blog Post »
bit.ly/Linux-Commands

Compiled by Alvin Khoo

Create a file

- touch filename
- echo “start writing” > filename

```
[cfraslin@login02(eddie) PARIS]$ touch Eiffel
[cfraslin@login02(eddie) PARIS]$ ls
Eiffel  StarWars.txt
```

```
[cfraslin@login02(eddie) PARIS]$ echo "It's just a big church" > NotreDame
```

```
[cfraslin@login02(eddie) PARIS]$ ls
Eiffel  NotreDame  StarWars.txt
```

```
[cfraslin@login02(eddie) PARIS]$ head NotreDame
It's just a big church
[cfraslin@login02(eddie) PARIS]$ head Eiffel
[cfraslin@login02(eddie) PARIS]$
```

Introduction to R

Jenny C Nascimento-Schulze
March 2023, Porto

- What is R?
- Why use R?
- Basics (Data types, importing datasets, subsetting, …)

What is R?

- A computational language and an environment (a place in your where you can run application software/programs with a user interface)
- Powerful tool for computational statistics and data visualisation (graphics) wide number of statistical functions (linear and nonlinear modelling, classical statistical tests, …) and models can be visualised easily
- **Multiple data (genomics/transcriptomics/maps/growth/survival….)**
- It compiles and runs on UNIX platforms and similar systems (FreeBSD and Linux), Windows and MacOS

Relates to other computational languages

It’s free!
(GNU project)

Advanced statistical language

Produces high quality plots suitable for publication - all you need is to reference it!

Why

?

Platform independent
(no need to modify your code)

Flexible tools – use your creativity and code

Vast community –Don’t be scared of the errors; google is your best friend!
You can contribute to develop packages etc..

Importing Data in R Script

Difficulty Level : Basic • Last Updated : 25 Nov, 2021

In this article, we are going to see how to **Import data in R Programming Language.**

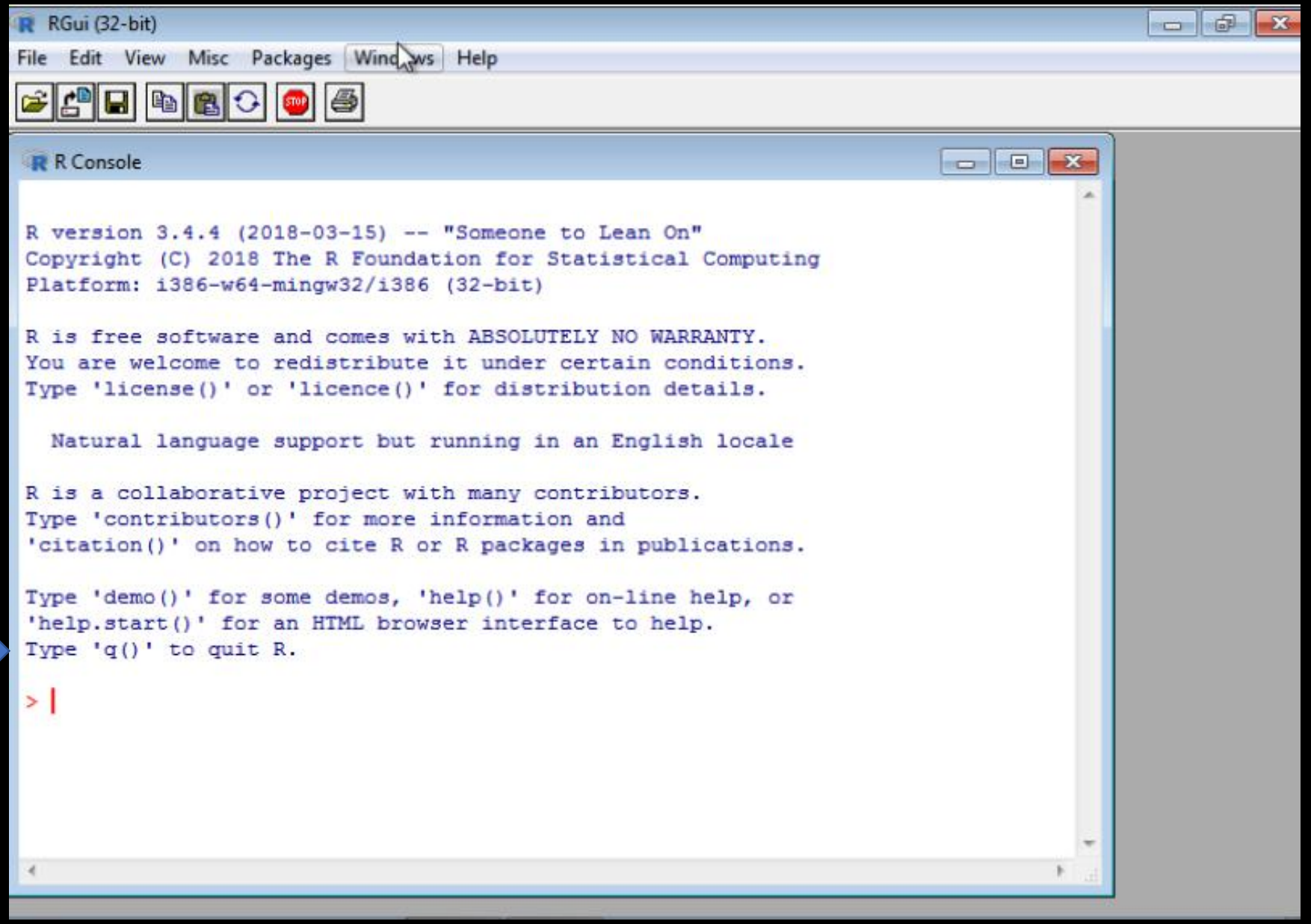
Importing Data in R

First, let's consider a data-set which we can use for the demonstration. For this demonstration, we will use two examples of a single dataset, one in .csv form and another .txt

R console and Rstudio

- R console = command line application
- Rstudio – Intuitive user-friendly interface for R

R console

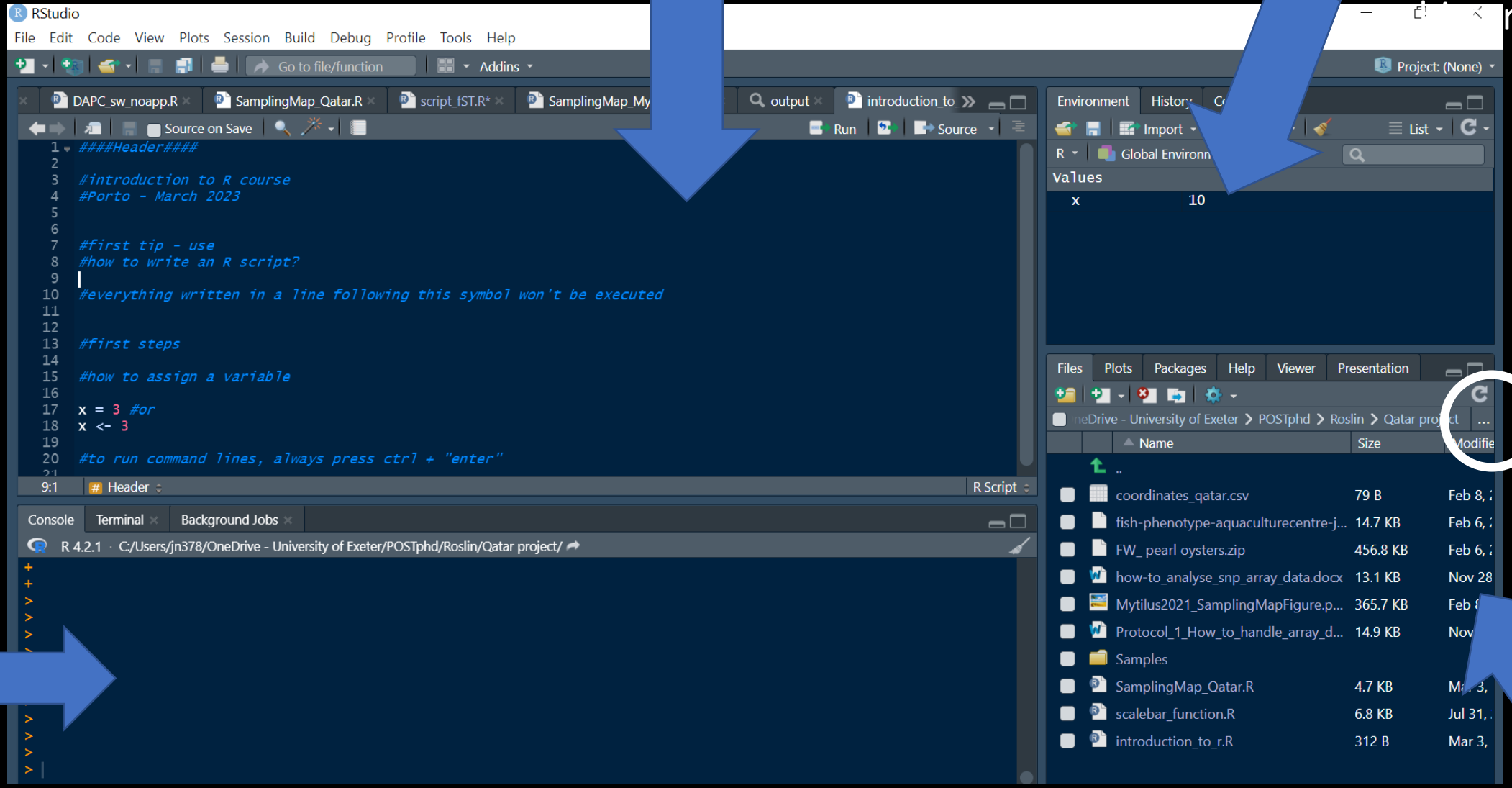


Enter your command after the > prompt

Rstudio

Script

Your environment/ history



Console

Multi task pane (output/ working directory)

Data types

Variables

- Name given to a memory locations, which store values in R
- These variables can store: numbers, words, functions …
- Creating a variable:

```
> x=3
> x
[1] 3
> 
```

Variables

- To create a variable (e.g. “name” and “age”) , you must assign it a value
- values are assign following the “<-” or “=“

```
#how to assign a variable

name <- "Jenny"
name #output Jenny
age <- 65
age #output 65
```

- You can multiple (*), add(+), substract (-), square root (sqrt()) … your variables
- R can do loops

Data types - Vectors

- Vectors consist on list of variables, those can be a collection of numbers, arithmetic expressions, logical values or character strings …
- Vectors are normally created using function: **c()**

```
> #creating numeric vector
> numvec <- c(1,2,3,4,5)
> numvec
[1] 1 2 3 4 5
> class(numvec)
[1] "numeric"
>
```

- Numeric (variable with a decimal value assigned to it), integer (non-fractional)

Data types - Vectors

- Logical vectors

```
> logical <- c(TRUE, FALSE, TRUE) #logical
> logical
[1]  TRUE FALSE  TRUE
> class(logical)
[1] "logical"
```

- Character: vector representing string values (require quotations “ “)

```
> vector1 <- c("this is a string")
> vector1
[1] "this is a string"
>
> vector2 <- c("this", "is", "a", "string")
> vector2
[1] "this"   "is"     "a"      "string"
> class(vector2)
[1] "character"
```

Data types - Vectors

- Basic functions – calculate mean, variance and sort the data within a vector

```
> x <- c(1,2,3,4,5,6,7,8,9)
> mean(x)
[1] 5
> mode(x)
[1] "numeric"
> var(x)
[1] 7.5
> sort(x)
[1] 1 2 3 4 5 6 7 8 9
>
```

Importing datasets

Importing datasets

- Datasets in multiple formats can be imported into R
- Comma delimited files (csv) and tab delimited files

```
# Read "comma separated value" files (".csv")
# first row contains variable names, comma is separator
# assign the variable id to row names
  > # note the / instead of \ on mswindows systems
df <- read.csv(file, header = TRUE, sep = ",", dec = ".", ...)
# Or use read.csv2: variant used in countries that
# use a comma as decimal point and a semicolon as field separator.
df <- read.csv2(file, header = TRUE, sep = ";", dec = ",", ...)
# Read TAB delimited files
df <- read.delim(file, header = TRUE, sep = "\t", dec = ".", ...)
df <- read.delim2(file, header = TRUE, sep = "\t", dec = ",", ...)
```

Importing datasets

- Excel files

```
# read in the first worksheet from the workbook myexcel.xlsx
# first row contains variable names
library(xlsx)
mydata <- read.xlsx("c:/myexcel.xlsx", 1)

# read in the worksheet named mysheet
mydata <- read.xlsx("c:/myexcel.xlsx", sheetName = "mysheet")
```

Checking data structure and changing data format

- After importing your dataset into R, you can check the structure of your dataset

```
# Dataframe containing lat and long of points
coord_all = read.csv("coordinates_file_SW-2021-finalpopss.csv", head=TRUE)
str(coord_all)
```

	Region	Coast	Site	Code	Lat	Lon
1	Cornwall N	Exposed	Newquay	NEWQ	50.40971	-5.121955
2	Cornwall N	Exposed	Godrevy	GDV	50.23546	-5.392014
3	Cornwall N	Exposed	Mousehole	MHO	50.08508	-5.535512
4	Cornwall N	Estuary	Padstow	PAD	50.52	
5	Cornwall N	Exposed	Porthcothan Bay	PCT	50.50	
6	Cornwall N	Exposed	Porthmeor	PMEO	50.17	
7	Cornwall N	Exposed	Port gaverne	PORTGAV	50.59	
8	Cornwall N	Exposed	Tintagel	TIG	50.66	
9	Cornwall N	Exposed	Trevone Bay	TRE	50.54	
10	Cornwall S	Estuary	Feock	FEO	50.20	

```
> str(coord_all)
'data.frame':   26 obs. of  6 variables:
 $ Region: chr  "Cornwall N" "Cornwall N" "Cornwall N" "Cornwall N" ...
 $ Coast : chr  "Exposed" "Exposed" "Exposed" "Estuary" ...
 $ Site  : chr  "Newquay" "Godrevy" "Mousehole" "Padstow" ...
 $ Code  : chr  "NEWQ" "GDV" "MHO" "PAD" ...
 $ Lat   : num  50.4 50.2 50.1 50.5 50.5 ...
 $ Lon   : num  -5.12 -5.39 -5.54 -4.9 -5.03 ...
```

Checking data structure and changing data format

- After importing your dataset, you can check the structure of your dataset

```
# Dataframe containing lat and long of points
coord_all = read.csv("coordinates_file_SW-2021-finalpopss.csv", head=TRUE)
str(coord_all)
```

- To change from character to numeric use command as.numeric
  e.g. coord_all$Region <- as.numeric(coord_all$Region)

- You can also transform your variables into factors (as.factor) and character (as.character)

- These transformations are important when applying statistical tests where you have to identify your sample info and the factors used in an experiment

Subsetting

- Used to select and exclude observations

- Remove nonrelevant parts of your dataset or focus on specific sections of it

```
# How to subset

new_dataframe <- old_dataframe[rows, columns]
```

Subsetting - example

- Dataset counts1 → RNA expression of *Physcomitrela patens* (rows: 32,926) in different time points (columns: 33) under salt stress conditions

	s1	s2	s3	s4	s5	s6	s7	s8
Pp3c1_20	652	569	693	783	625	589	714	662
Pp3c1_40	977	808	1001	775	803	677	834	681
Pp3c1_50	0	0	0	0	0	0	0	0
Pp3c1_60	621	526	601	667	597	598	740	648
Pp3c1_70	989	1060	1081	811	791	703	741	797
Pp3c1_80	23	17	19	21	4	27	13	9
Pp3c1_100	1521	1498	1434	3240	2733	2412	2450	2392
Pp3c1_110	0	0	0	1	0	0	1	0
Pp3c1_120	2834	2681	2500	2521	2464	2265	2477	2452
Pp3c1_140	4013	3583	3586	2649	2435	2293	2252	2269
Pp3c1_145	0	0	1	0	2	0	0	0
Pp3c1_170	781	692	783	505	522	426	604	557
Pp3c1_190	260	246	232	255	209	190	222	205
Pp3c1_200	895	790	741	764	655	538	685	704

Subsetting - example

- Select rows from/to specific values :
  new_dataframe = old_dataframe[1:15,] **(selecting rows)**

- Select columns from/to specific values :
  new_dataframe = old_dataframe[,1:15] **(selecting columns)**

- Select specific columns in a random order by creating a vector:
  new_dataframe = old_dataframe[,c(1,2,3,10)] **(selecting vector of columns)**

- Remove specific columns/rows by using the "-" sign:
  new_dataframe = old_dataframe[,c(-1,-2,-3,-10)] **(selecting vector of columns)**

Subsetting - example

- Subset columns 1 to 3 in counts 1

```
#subset columns 1 to 3
counts1_columns <- counts1[,1:3]
head(counts1_columns)
```

- Subset specific columns – creating a vector

```
count1_specific_columns <- counts1[,c(1,3,5)]
head(counts1_columns)
```

- Output

	s1	s2	s3
Pp3c1_20	652	569	693
Pp3c1_40	977	808	1001
Pp3c1_50	0	0	0
Pp3c1_60	621	526	601
Pp3c1_70	989	1060	1081
Pp3c1_80	23	17	19
Pp3c1_100	1521	1498	1434
Pp3c1_110	0	0	0
Pp3c1_120	2834	2681	2500
Pp3c1_140	4013	3583	3586
Pp3c1_145	0	0	1
Pp3c1_170	781	692	783
Pp3c1_190	260	246	232
Pp3c1_200	895	790	741

Subsetting - example

- Same for rows

```
counts1_rows <- counts1[1:3,]
```

- Output

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13
Pp3c1_20	652	569	693	783	625	589	714	662	720	696	691	712	718
Pp3c1_40	977	808	1001	775	803	677	834	681	831	747	825	866	958
Pp3c1_50	0	0	0	0	0	0	0	0	0	0	0	0	0

Subsetting - example

- Subset by row/column value

```
#filter by value or name
counts2 <- counts1[counts1$s1 == 652,]

#if filtering for a specific name
counts2 <- counts1[counts1$s1 == "M",]
```

	s1	s2	s3
Pp3c1_20	652	569	693
Pp3c1_13420	652	563	646
Pp3c1_20710	652	607	616
Pp3c2_25000	652	532	671
Pp3c5_8160	652	559	636
Pp3c5_10280	652	595	736
Pp3c5_26190	652	649	594
Pp3c8_21100	652	593	512
Pp3c11_5760	652	633	673
Pp3c14_9830	652	504	580
Pp3c15_23520	652	630	588
Pp3c16_1979	652	652	852
Pp3c16_18860	652	583	687
Pp3c18_20060	652	618	550

Packages and libraries

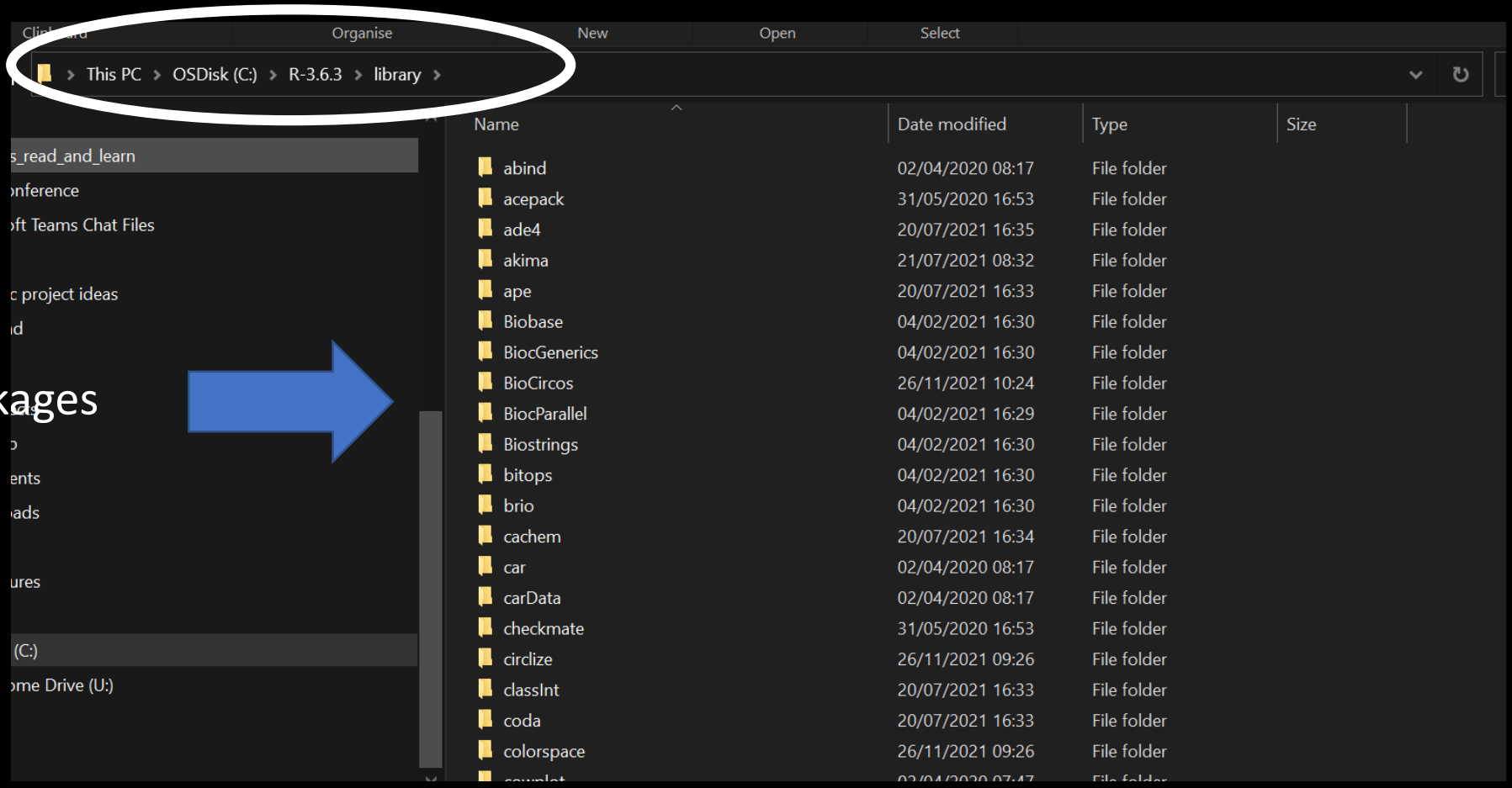
- Packages are a collection of R functions, formed by pre complied code and sample data (users can import the data to use as examples)

- Different packages have different functions

→ **xlxs** imports excel files
→ **dyplr** imports a number of functions for subsetting datasets
→ **ggplot2** main package for producing plots with numerous functions for enhancing aesthetics

Packages and libraries

- Library is the directory in which packages are installed, and stored, in your computer



My packages

Installing and calling packages

- To install packages use function (directly from CRAN network of ftp and servers for R codes and documentations)

install.packages(“yourpackage”)

```
install.packages("ggplot2")
```

- To load your package, use function library(ggplot2)

```
library(ggplot2)
```

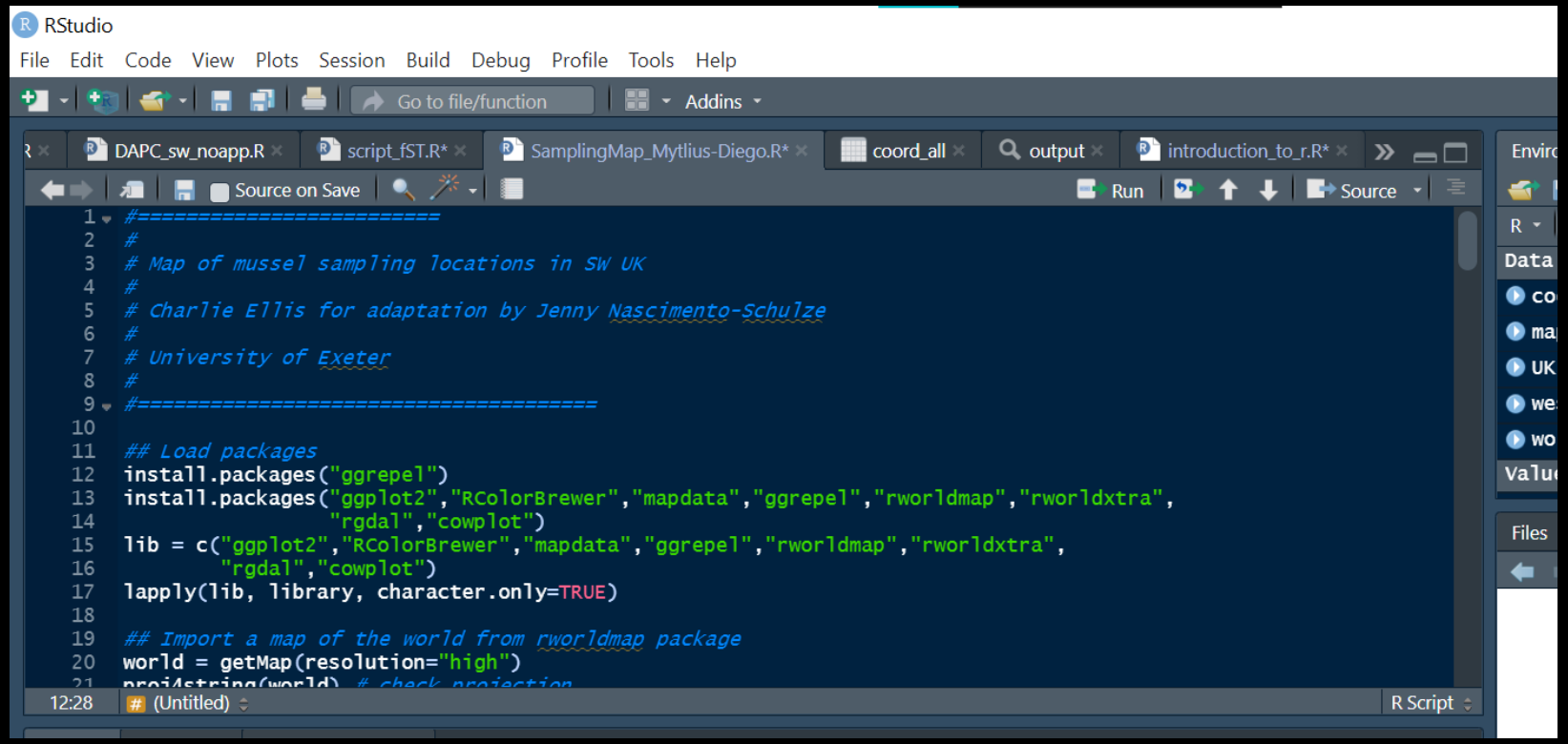
Scripts

- Scripts are a text file saved with an .R extension with the code containing the necessary commands for analysing your dataset

- If you run commands directly in the console those will not be saved and you will need to repeat all the coding over again – avoid headaches and write a script!

Scripts - Tips

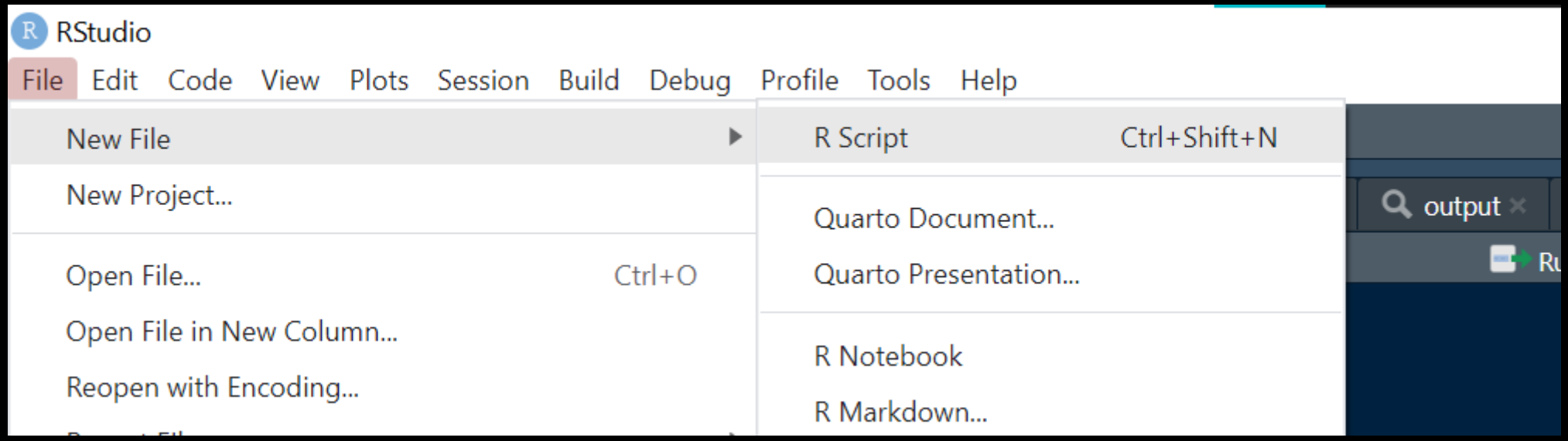
- Be organised
- Leave yourself comments with the '#' at any place in a line– they will help you in the future



Scripts

- To create a new script in Rstudio click on the upper left side of window File>> New file>> R Script

  a blank page will open in the script window and you can start writing your commands



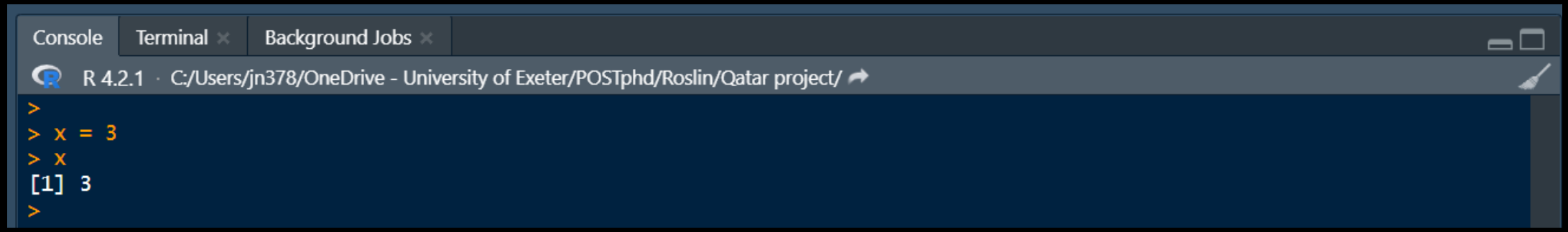
Call R from server

```
(base) [jn378@login02 ~]$ module load R/4.0.0-foss-2020a
(base) [jn378@login02 ~]$ R

R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> quit()
Save workspace image? [y/n/c]: y
(base) [jn378@login02 ~]$
```

Coding: Reminders and Tips

Code writing: Important reminders

- R is ready for input when the “>” prompt is appearing in the console
  Run commands using CTRL + ENTER



- Case sensitive language

- Name your variables with an easy/meaningful name – avoid long/uppercase complicated names – avoid mistakes & saves time

- Does not deal well with spaces – never use spaces in file names or when naming a variable

Important reminders

- Values can be assigned with either “<-” or “=“

```
> b = 3
> b
[1] 3
> c <- 4
> c
[1] 4
```

- Set working directory manually (Rstudio) or command line

```
> setwd("C:/Users/jn378")
> 
```

- Remove rows NA’s with **na.omit(yourdata)**

```
df <- na.omit(df)
```

Thank you & have fun using R!

grinnaqua

```
for (x in 1:10) {
  print(x)
}
```

- output

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

- In this case, “()” defines the function and “{}” the expression

ROSLIN

grinnaqua

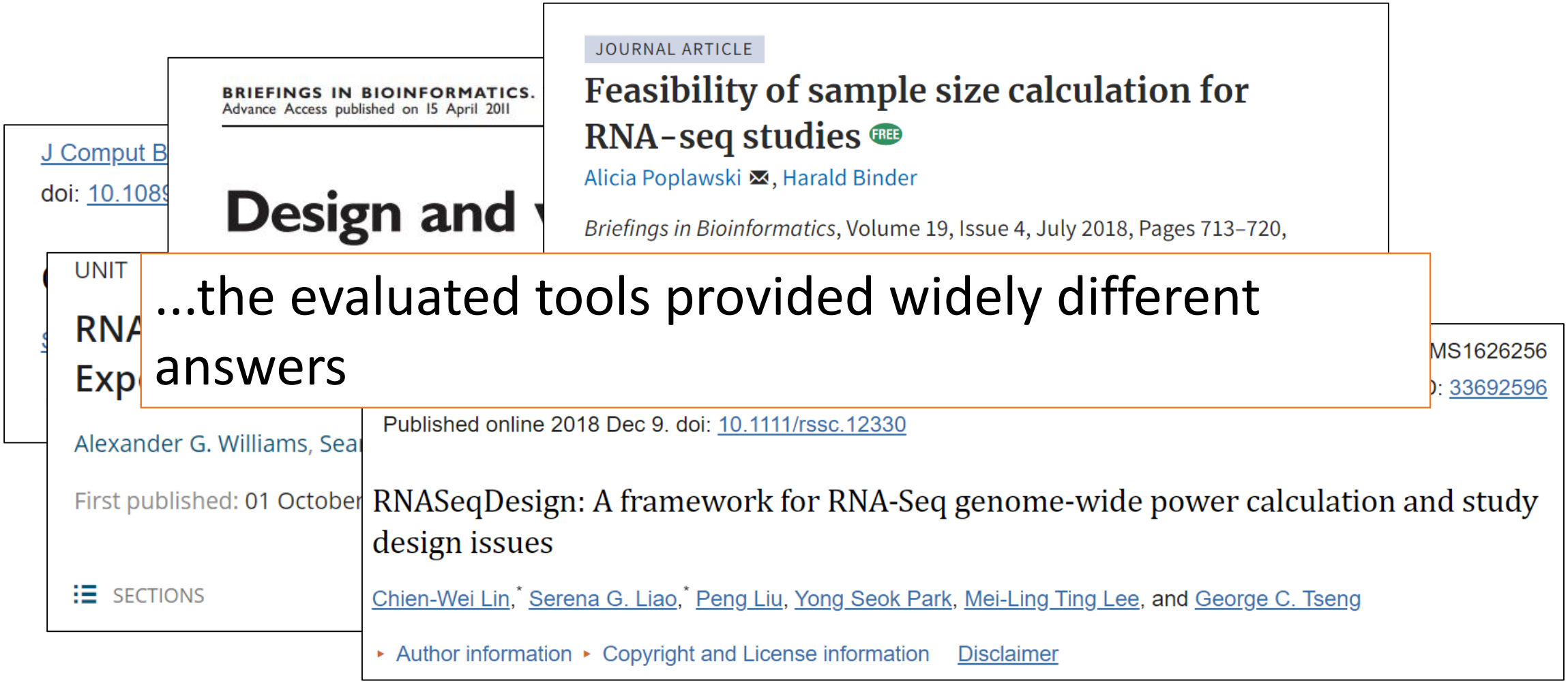
THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Experimental design

Porto, 16th & 17th March 2023

ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies



ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Number of replicates per experiment

- No easy answer...
- Hugely dependent on type of experiment, environmental control, animal background, (remember the genetic variation experiments from yesterday)
- Sex of animals often the most significant aspect of difference to expression profiles (take into account or increase your samples numbers)
- Read everything.
- As a last resort you may need pilot data.

However...

Our recommendations

3-5 biological replicates for highly controlled lab experiments (cell culture)
5-6 biological replicates for controlled animal experiments
8-12 biological replicates for non-controlled studies (e.g. studying wild animals in different field environments)

ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

Sequencing Depth (reads per sample)

Genewiz recommendations

5-10 million reads per sample for small genomes (e.g. bacteria) and 20-30 million reads per sample for large genomes (e.g. human, mouse)

Genome sizes...

Human genome = 3 Gb
Mouse genome = 2.5 Gb

Bacterial (Vibrio) genome = 4Mb

Salmon genome = 3 Gb
Pacific oyster genome = 650 Mb

Our recommendations...

- Aim for 20 million reads per sample for most Eukaryotic genomes.
- Can reduce to 15 million for smaller genomes... With experience
- Absolute **bare minimum** of 10 million with a test runs first.
  - You will lose interesting genes and (worst case scenario) incur the wrath of reviewer 2

ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies