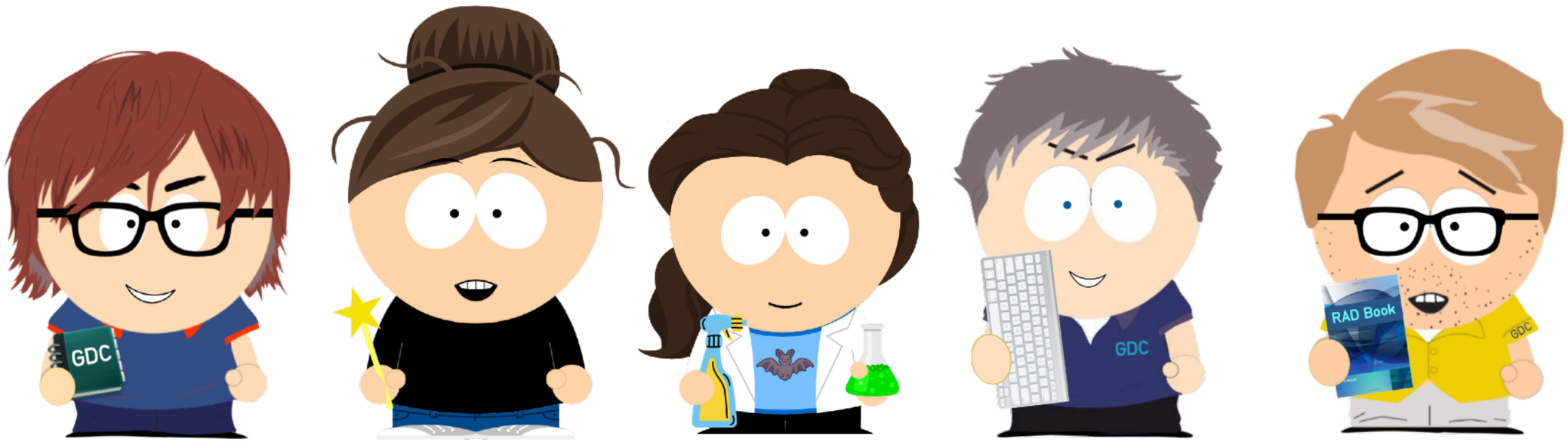
GDC Genetic Diversity Centre Zurich

Genetic Diversity: Analysis

15 June - 26 June 2026

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Genetic Diversity Centre Zurich



Aria Silvia Tamara Jean-Claude Nik

At the Genetic Diversity Centre (**GDC**), we like our DNA diverse and our organisms anything but standard. As a knowledge and technology platform within the USYS department at ETH Zurich, we support research into the genetic and genomic diversity of all kinds of life, especially the weird, wild, and wonderful non-model organisms that don't usually get the spotlight. Whether you're tracking trout genes or decoding desert microbes, we're here with the tools and expertise to help.

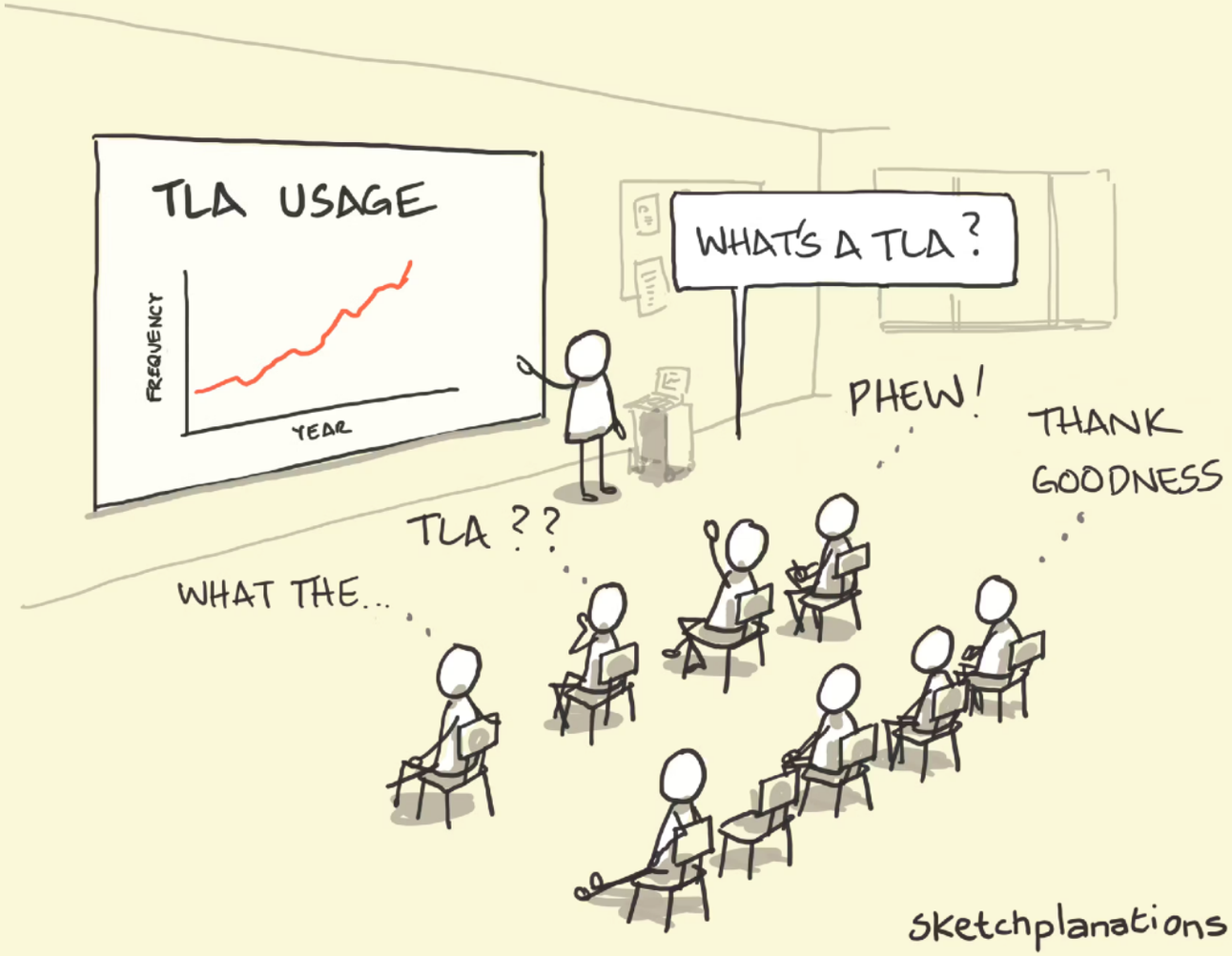
GDC Genetic Diversity Centre Zurich

Power-Disa-Point-Ability

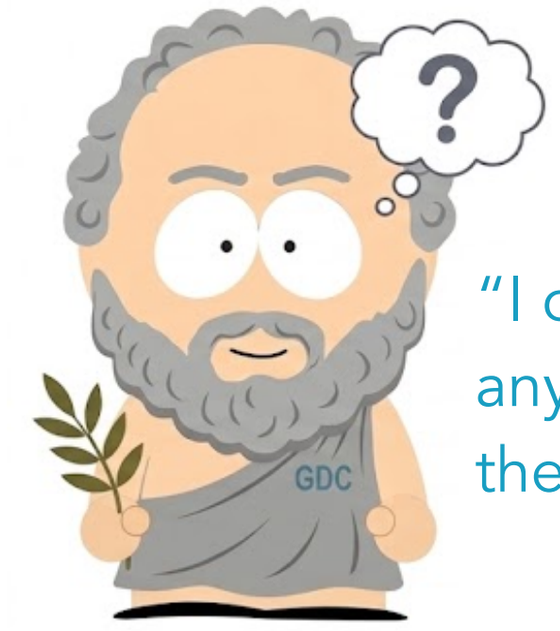
Power-Disa-Point-Ability: When a presenter confuses “more slides” with “more insight” and hides behind animations, charts, and bullet points like a kid behind the sofa during a horror movie.

ASK THE QUESTION AT TALKS

IF YOU'RE WONDERING, LIKELY OTHERS ARE TOO



Genetic Diversity Centre Zurich



"I cannot teach anybody anything, I can only make them think."

—Socrates

"Science is a way of thinking much more than it is a body of knowledge."

—Carl Sagan

What is your thinking mode?

Source: A framework popularised by behavioural scientist Jonathan Haidt, which describes how people often switch between different 'modes of thinking' - especially when they are debating, persuading or making decisions.

Humans are surprisingly flexible thinkers, but not always in the way we expect. Depending on the situation, we tend to slip into different mental modes that shape how we reason, argue, and respond to the world around us.

Scientist mode
We are curious, evidence driven, and willing to change our minds. In this mode, we seek the truth, test hypotheses, and follow the data, even when it challenges our existing beliefs. Unfortunately, we enter this mode less often than we like to imagine.

Preacher mode
We are defending values or beliefs that we hold dear and want others to share. Rather than asking questions, we provide answers and seek to persuade.

Prosecutor mode
We concentrate on identifying weaknesses in other people's arguments. Instead of testing our own assumptions, we focus on criticizing opposing views, often with the goal of winning an argument rather than discovering the truth.

Politician mode
We become concerned with gaining approval and maintaining support. We tailor our message to our audience, build alliances, and manage impressions, sometimes at the expense of accuracy.

Each of these modes can be useful in the right context. However, scientist mode is the most valuable for genuine learning and honest inquiry, and often the most difficult to sustain. The challenge is to recognise which mode you are in and to know when it is time to switch.

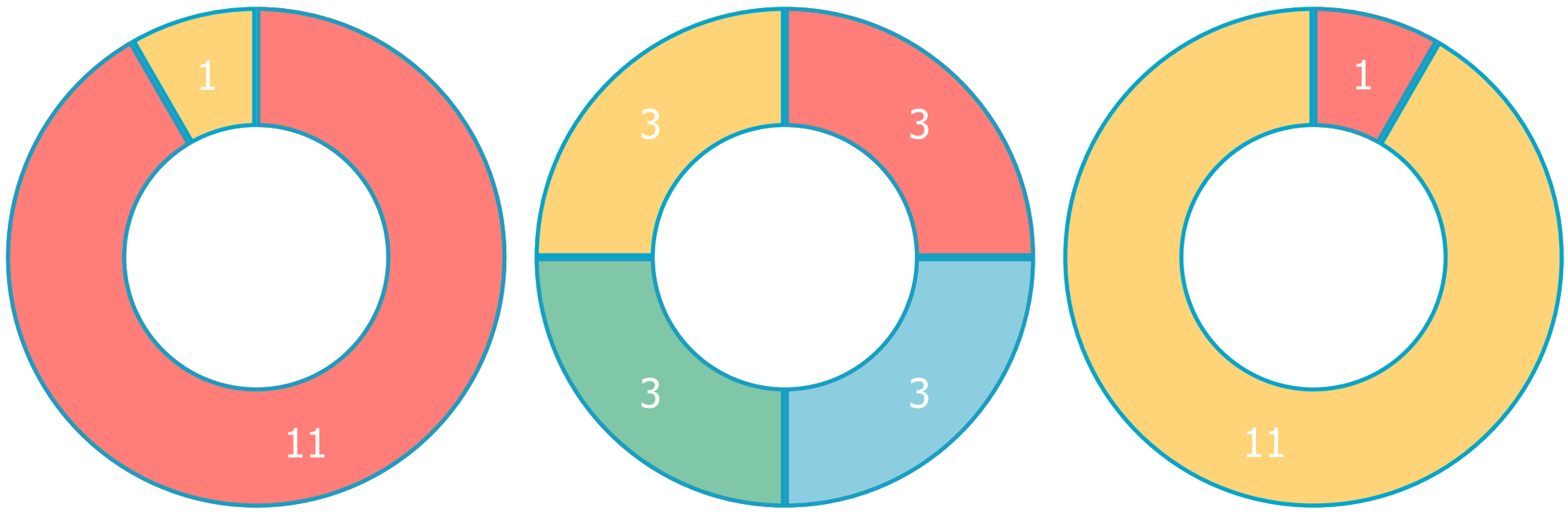
Genetic Diversity Centre Zurich

Request #1: Survey

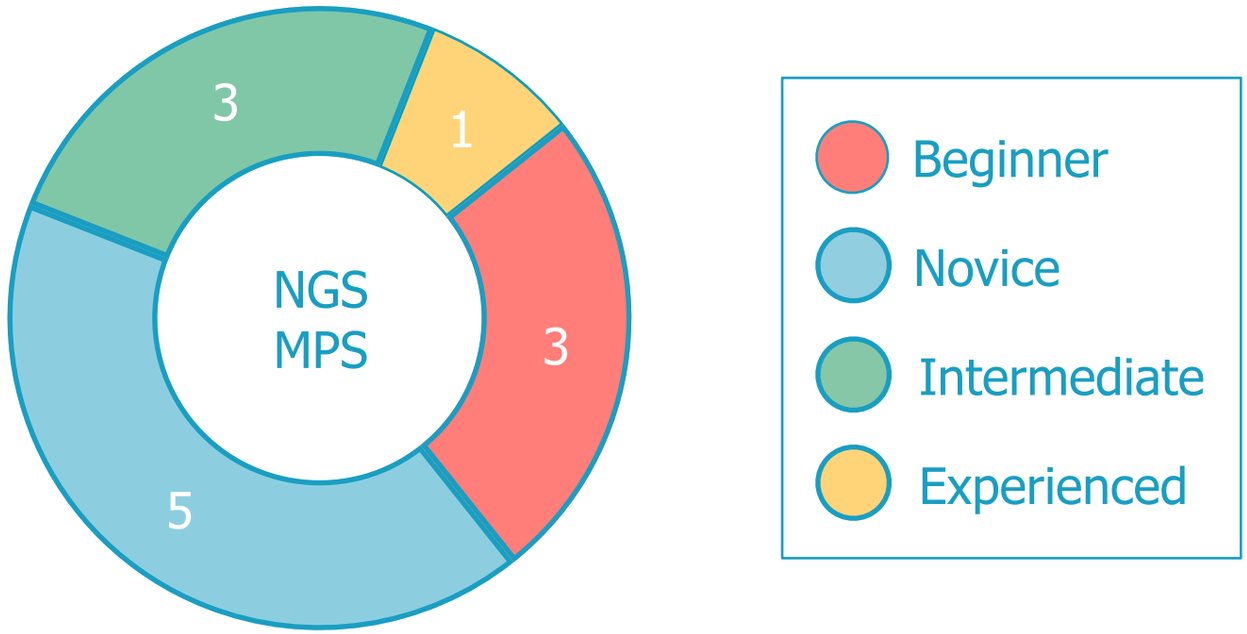
Request #2: R-Code

Request #3: Requirements

Request #1: Survey - Competency Level

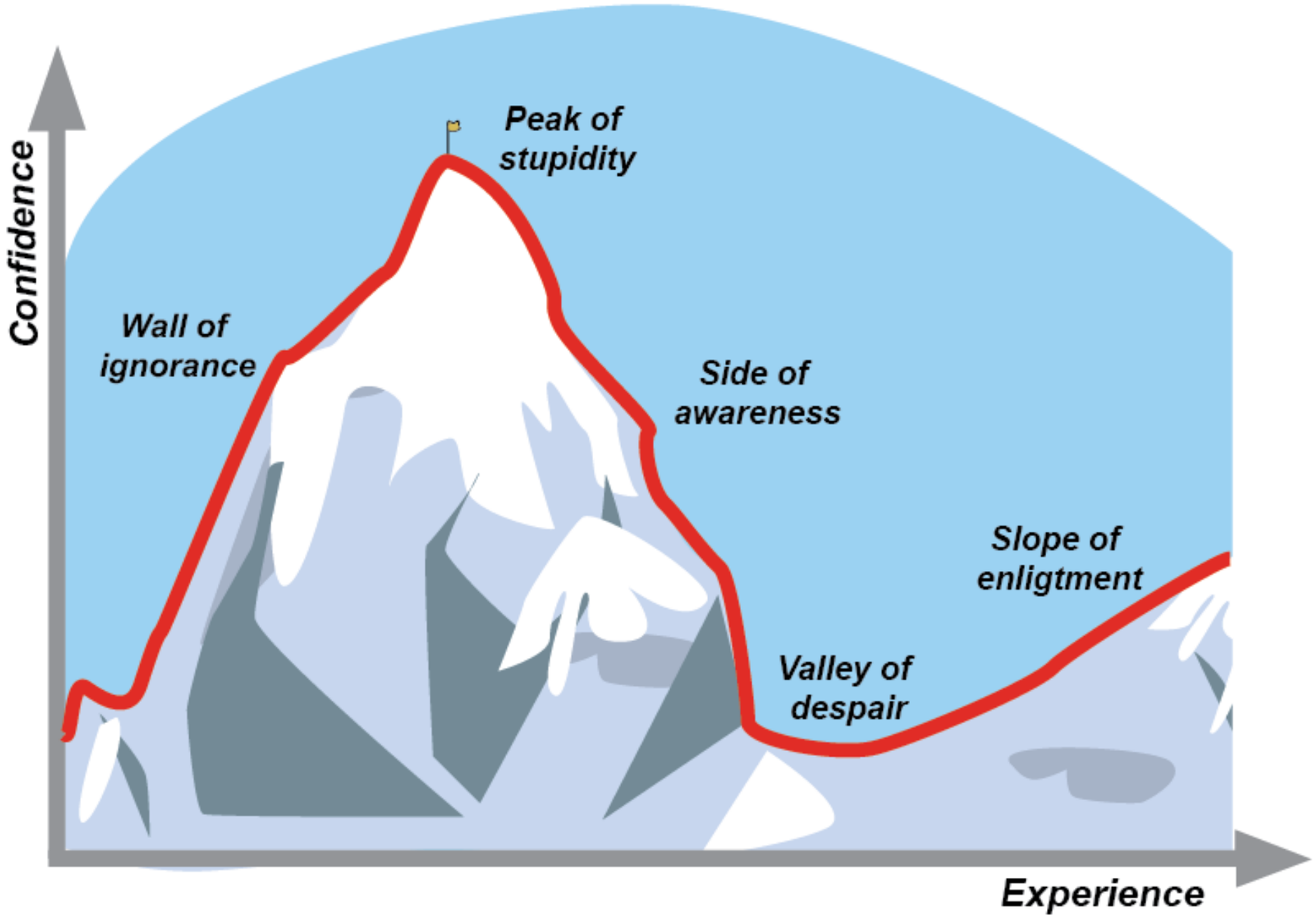


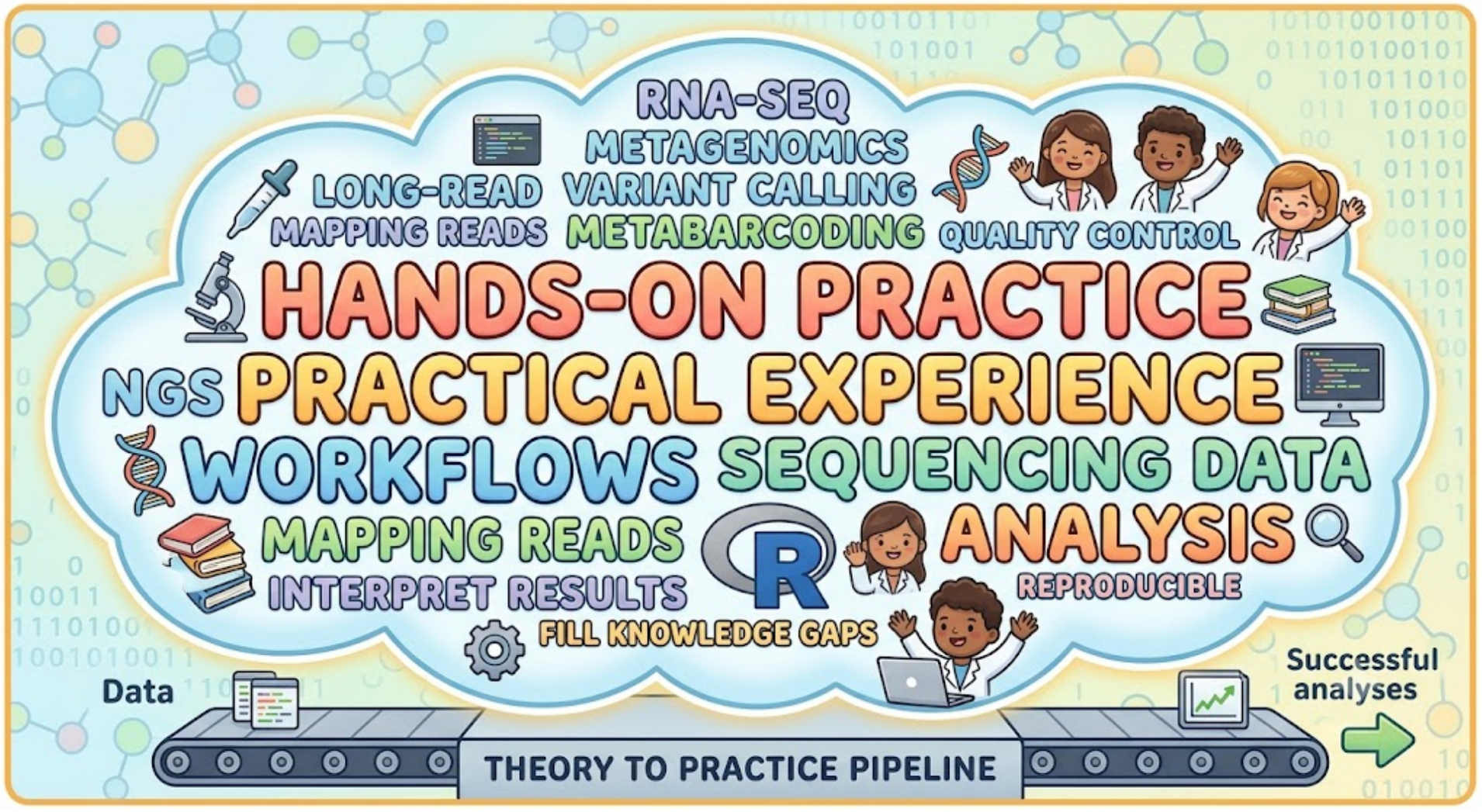
Request #1: Survey - Competency Level



A - Beginner	No idea what this is.
B - Novice	I've heard of it, maybe tried it briefly.
C - Intermediate	I understand it and have some experience with it.
D - Experienced	I know it well and could teach it to others.

Dunning-Kruger Effect Curve





1. The Core Delivery: "Less Theory, More Doing”
2. The Universal Tooling
3. Specific Methodological Clusters

Request #2: R-Code

Please complete a simple task in R: Write a script that creates a text file with the results of **10 rolls** of a **pair of dice**.

When reviewing the R scripts, we noticed a few areas that could be improved:

- **Script header**: Does the script include a clear and informative header with the author, date, and purpose?
- **Coding style**: Are formatting, indentation, and naming conventions consistent and tidy?
- **Consistency**: Are similar tasks handled in a consistent way throughout the script?
- **Typos**: Are variable names and function calls free of spelling mistakes?
- **Functionality**: Does the script run without errors and produce the expected results?
- **Reproducibility**: Can someone else run the script and achieve the same results?
- **Readability**: Are variable names meaningful, and are helpful comments included?
- **Simplicity**: Is the code as simple as possible, without unnecessary complexity?
- **Correctness**: Are the tasks implemented correctly and aligned with the assignment's goals?

```
d1 <- sample(1:6, 10, replace = TRUE)
d2 <- sample(1:6, 10, replace = TRUE)

results <- data.frame(d1, d2)

write.table(results, file = "dice_rolls.txt", )
```

```
## Turner Helix
## Rolling Dices (D6)
## Version: 1.1 (250517)

set.seed(123)
d <- sample(1:6, 20, replace = TRUE)

results <- data.frame(d1 = d[1:10], d2 = d[11:20])

write.table(
  results, file = "dice_rolls.txt",
  row.names = FALSE, quote = FALSE, sep = "\t"
)
```

```
## Turner Helix
## Rolling Dices (D6)
## Version: 1.2 (250518)

# Set the random seed > this ensures the results are reproducible
set.seed(123)

# Generate 20 random integers between 1 and 6 with replacement
d <- sample(1:6, 20, replace = TRUE)

# Create a data frame to organise the results
results <- data.frame(d1 = d[1:10], d2 = d[11:20])

# Write the data frame to a text file
write.table(
  results,
  file = "dice_rolls.txt",
  row.names = FALSE,
  quote = FALSE,
  sep = "\t"
)
```

```r
# Function to simulate rolling a pair of dice
roll_dice <- function() {
  return(sample(1:6, 2, replace = TRUE) %>% sum())
}
```

In R, **%>%** is known as the pipe operator. It is primarily associated with the `magrittr` package and is heavily used in the `tidyverse` collection of packages (like `dplyr, tidyr, ggplot2`).

What it does: It takes the output (result) of the function on the left and passes it as the first argument to the function on the right.

What is the main problem? It **requires loading external package** like `magrittr` or `dplyr`.

```r
# Function to simulate rolling a pair of dice
roll_dice <- function() {
  return(sample(1:6, 2, replace = TRUE) %>% sum())
}
```

Modern Alternative: |>

```r
# Function to simulate rolling a pair of dice
roll_dice <- function() {
  return(sample(1:6, 2, replace = TRUE) |> sum())
}
```

As of R version 4.1.0 (released in 2021), R includes a **native pipe operator |>** that works very similarly to %>% but **does not require loading an external package.**

```
## Turner Helix
## Rolling Dices (D6)
## Version: 3.0 (250521)

write.table(TeachingDemos::dice(10,2), "dice2x10.txt")
```

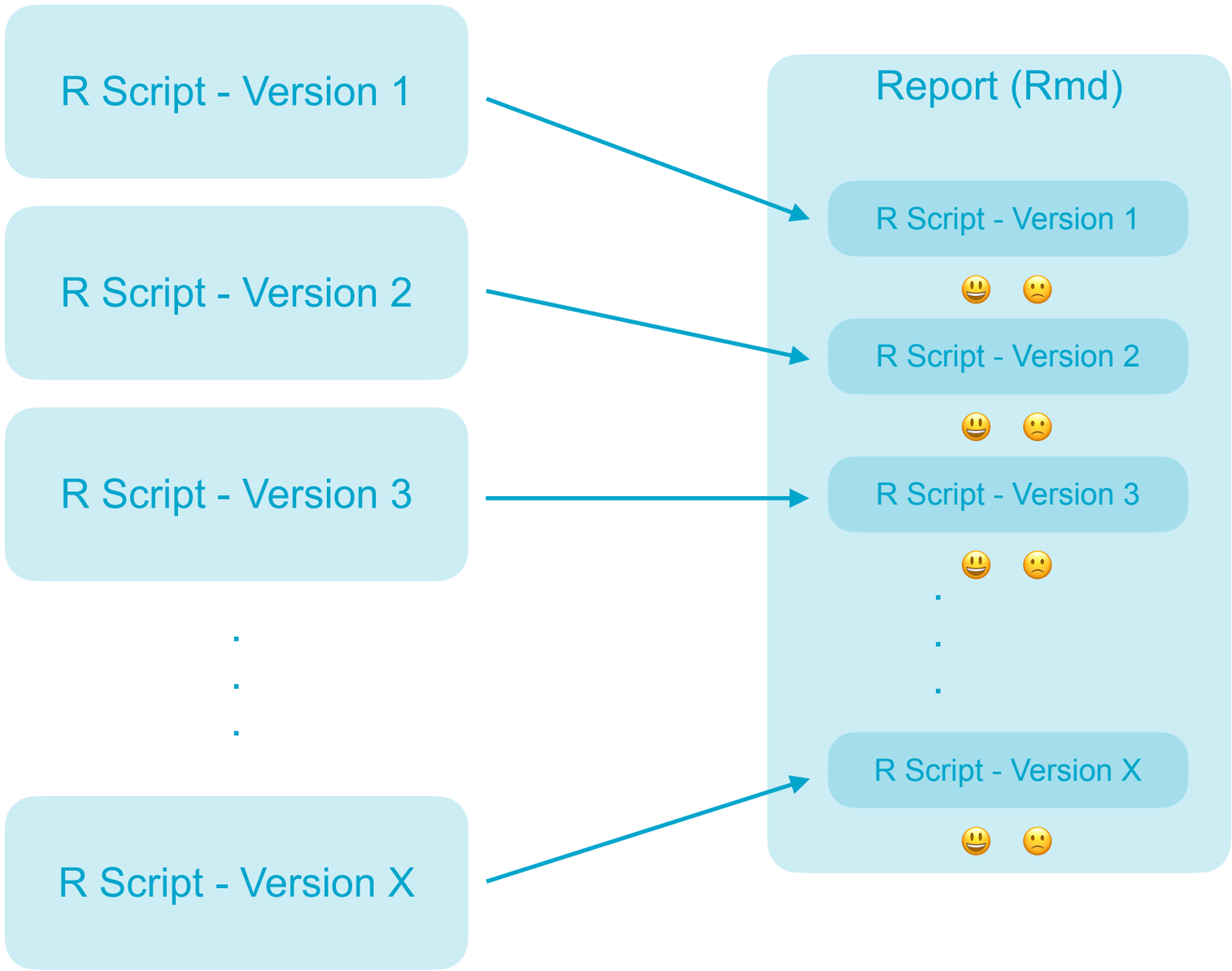

```
## Turner Helix
## Rolling Dices (D6)
## Version: 4.0(250522)

# Load the droll package
library(droll)

# Roll two six-sided dice (2d6) ten times
results <- r("2d6", times = 10)

# Save the results
write.table(results, file = "dice_rolls.txt",
            row.names = FALSE, col.names = FALSE)

# Show the results in the console
print(results)
```




```{r First Attempt, file = "dice_v1.r"}
```

```{r V2 Faster, file = "dice_v2.r"}
```

```{r V3 Flexible, file = "dice_v3.r"}
```

```{r VX Final, file = "dice_vx.r"}
```

GDA: Rolling Dice Example

Jean-Claude Walser

22/06/23

Intro

Request #2 - An important part of this course is to apply the knowledge you have learned. For this reason, we would like you to solve the following assignment using R:

"Write an R script (R code) that creates a text file containing the results of 10 rolls of a pair of dice."

The R script should be easy to use (user friendly) and flexible for extensions (e.g. more rolls, more dices, or flexible output). Think about ways to implement reproducibility, if possible? We don't need the perfect solution, partial solutions are fine. If the task is beyond your R skills, a step-by-step description of a solution would also be fine. Please send your solution (zipped) via e-mail directly to me using subject: GDA23: Request #2.

Differnt Versions of the Same

My First Attempt | My Second Attempt | My Third Attempt | **My Forth Attempt**

There is a package `TeachingDemo` and it contains a function dice.

```
# GDA: Rolling Dice Script
# Author: Jean-Claude Walser
# Date: 19.06.23

require(TeachingDemos)
```

Request #3: Requirements

- ✅ Internet access
- ✅ Linux terminal
- ✅ Code (text) editor
- ✅ MarkDown editor
- ✅ R and RStudio

Request #3: Requirements

Vescovi Anne 10 June 2026 at 15:55

GDA26: Question(s)

To: Walser Jean-Claude

Hi Jean Claude,

I have some questions regarding setting up my computer:

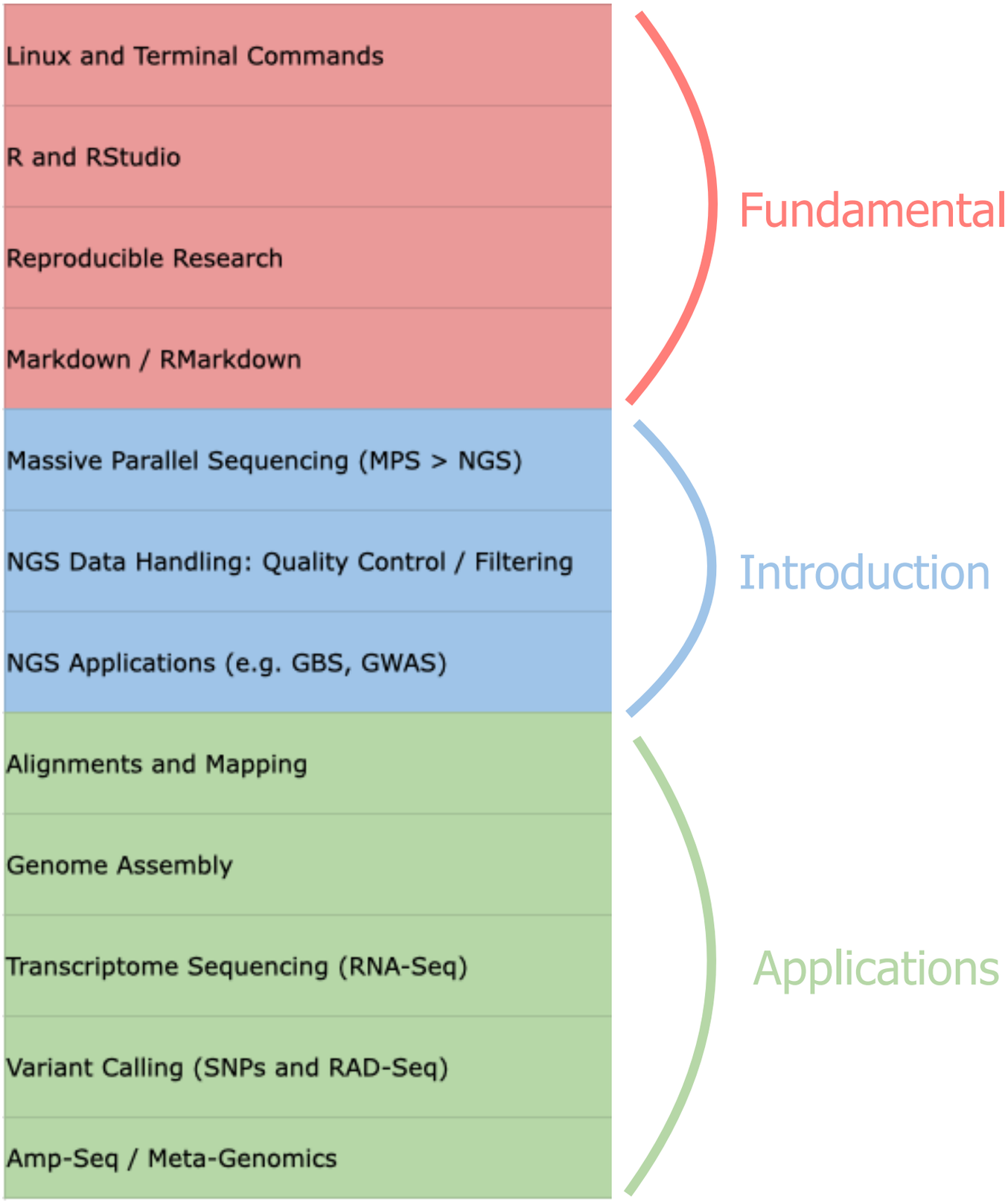
I) I downloaded **Cyberduck** but I wondered how to connect. I saw the settings SFTP.... but I am not sure which "Benutzername" / "Passwort" to use. Is this something we have to set up at the course? Or did I missed were I can found them?
II) I had problems installing the **Bioconductor** packages (**seqTools, qckitfastq**). I just updated R, is it possible that those packages are not active anymore? Should I install something else instead, or am I doing something wrong?

Best.

Anne

Genetic Diversity Centre Zurich

Course Structure


Linux and Terminal Commands
R and RStudio
Reproducible Research
Markdown / RMarkdown
Massive Parallel Sequencing (MPS > NGS)
NGS Data Handling: Quality Control / Filtering
NGS Applications (e.g. GBS, GWAS)
Alignments and Mapping
Genome Assembly
Transcriptome Sequencing (RNA-Seq)
Variant Calling (SNPs and RAD-Seq)
Amp-Seq / Meta-Genomics
Fundamentals
Introduction
Applications

Genetic Diversity: Analysis - 701-1425-00L

GDA26
Week 25+26

Welcome	Time	Topic	Lecturer
Monday, 15.06.26	09:00-10:00	First Things First	JCW / NZ

NGS	Time	Topic	Lecturer
Monday, 15.06.26	10:00-12:00	Ctrl+Alt+Genome: Starting Out in Bioinformatics	JCW
	13:00-15:00	NGS Demystified: A Beginner's Guide to Sequencing	JCW
	15:00-16:00	Discussion & Takeaways	JCW

BioComputing 1	Time	Topic	Lecturer
Tuesday, 16.06.26	09:00-12:00	Navigating the Terminal: A Beginner's Guide	JCW
	13:00-15:00	Think & Tinker Time	Self Study
	15:00-16:00	Discussion & Takeaways	JCW / NZ

BioComputing 2	Time	Topic	Lecturer
Wednesay, 17.06.26	09:00-11:00	A bit more advanced biocomputing using R	NZ
	11:00-15:00	Think & Tinker Time ->Terminal||R||HPC <-	Self Study
	15:00-16:00	Discussion & Takeaways	JCW / NZ

Quality Control / Filtering	Time	Topic	Lecturer
Thursday, 18.06.26	09:00-10:30	Checking Your Data: A Primer on Quality Control	JCW
	10:30-12:00	From Raw to Reliable: Quality Filtering Explained	NZ
	13:00-15:00	Think & Tinker Time	Self Study
	15:00-16:00	Discussion & Takeaways	JCW / NZ

Alignment-Mapping / Assemblies	Time	Topic	Lecturer
Monday, 22.06.26	09:00-09:30	Genome Jigsaw: The Art of Assembly	NZ
	09:30-10:00	Think & Tinker Time	Self Study
	10:30-11:30	Basics of Sequence Alignment and Mapping	NZ
	13:00-15:00	Think & Tinker Time	Self Study
	15:00-16:00	Discussion & Takeaways	NZ

RNA-Seq	Time	Topic	Lecturer
Tuesday, 23.06.26	09:00-12:00	From DNA to Dialogue: The World of Transcriptomics	JCW
	13:00-15:00	Think & Tinker Time	Self Study
	15:00-16:00	Discussion & Takeaways	JCW

SNP	Time	Topic	Lecturer
Wednesday, 24.06.26	09:00-10:00	Spotting Genetic Variations: Basics of SNP Calling	NZ
	10:30-15:00	Think & Tinker Time (SNPs or RAD)	Self Study
	15:00-16:00	RADseq and Discussion	NZ

Amp-Seq / Meta-Genomics	Time	Topic	Lecturer
Thursday, 25.06.26	09:00-12:00	DNA, Diversity & Data: Intro to AmpSeq & Metagenomics	JCW
	13:00-15:00	Think & Tinker Time	Self Study
	15:00-16:00	Discussion & Takeaways	JCW

Paper Discussion / Project Work	Time	Topic	Lecturer
Friday, 26.06.26	09:00-11:45	Science Unplugged	JCW / NZ
	11:45-12:00	Reflect & Respond	
	13:00-16:00	Work-On-Your-Project Slot	JCW / NZ

Genetic Diversity Centre Zurich

9:00 - 12:00 Lecture

Self-study with Challenges

Questions
Problems

15:00 - 16:00 Discussion

Genetic Diversity Centre Zurich

Genetic Diversity Centre (GDC) - Course Webpage

Search

Genetic Diversity Centre (GDC) - Course Webpage

- Start
- Lecture notes
- Requirements
- Biocomputing
- Biocomputing with R
- Biocomputing on a HPC cluster
- Reproducible Research
- MPS (NGS)
- Acknowledgement

Learning Objectives

Main
◇ Understand the basic principles and differences between massive-parallel sequencing (MPS) platforms.
◇ Understand sample indexing and sequence (read) types.
Minor
◇ Understand the importance of a read data archive and be able to access such data.
◇ Understand the advantages, limitations and applications of MPS in e.g. genomics, transcriptomics or metabarcoding.

Lecture Notes

⬇ MPS (NGS)

Massively Parallel Sequencing

Massive parallel sequencing or massively parallel sequencing is any of several high-throughput approaches to DNA sequencing using the concept of massively parallel processing.

MPS ~ NGS

I use the term massively parallel sequencing (MPS) because it is a more general term and it also includes newer sequencing technologies. Next-generation sequencing (NGS) refers to PCR-based sequencing technologies (e.g. Roche 454, Illumina), while single-molecule sequencing (e.g. PacBio or ONT) is often referred to as next-next or third-generation sequencing. I keep it simple and use MPS.

Table of contents

Topic	Webside
MPS (NGS)	MPS (NGS)
Terminal	Linux 1 - Local Terminal
	Linux 2 - Remote Terminal
	Biocomputing

https://www.gdc-docs.ethz.ch/GeneticDiversityAnalysis/GDA/site/

- ➜ Infos
- ➜ Challenges
- ➜ Links
- ➜ Handouts

Good to know!

E-Mail	jean-claude.walser(at)usys.ethz.ch niklaus.zemp(at)usys.ethz.ch

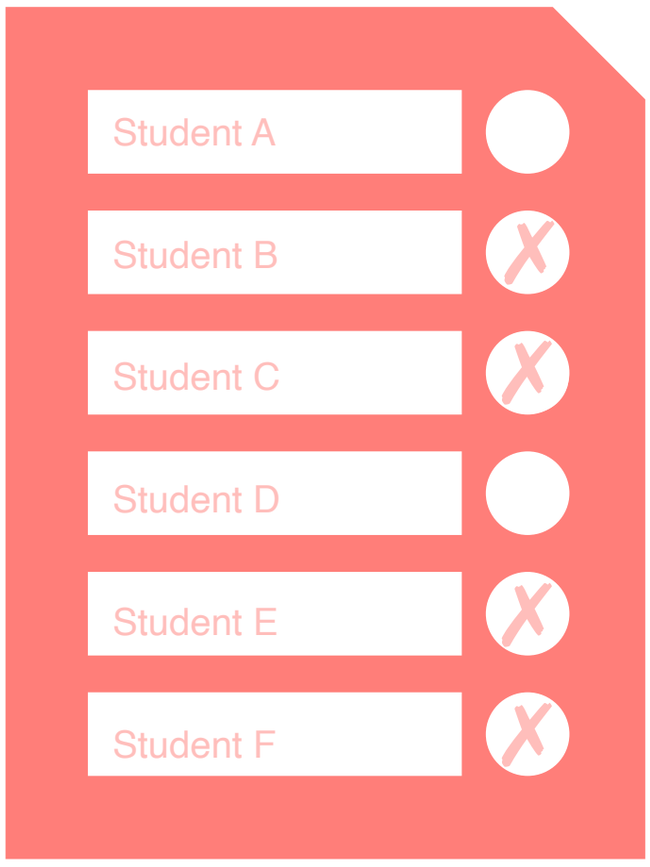
Subject **GDA26: Question**

WWW	https://www.gdc-docs.ethz.ch/GeneticDiversityAnalysis/GDA/site/

SSH	ssh guest01@gdc-vserver.ethz.ch

Genetic Diversity Centre Zurich

GDC Genetic Diversity Centre Zurich



- it is your course
- it is your choice
- it is your decision

- pick a project
- hand in a report (md/html)
- hand it in before deadline
- feedback if you wish

Feedback

helpful information or criticism that is given to someone to say what can be done to improve a performance.