GDC

Genetic Diversity Centre Zurich

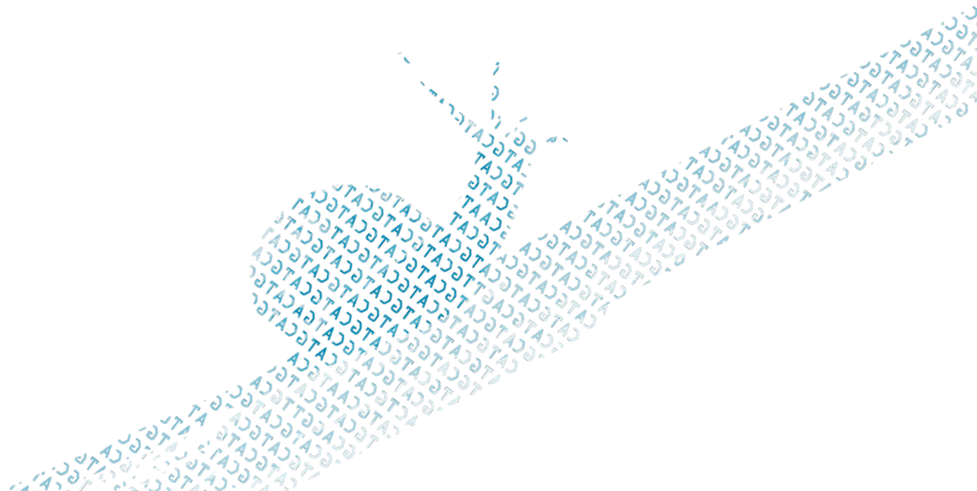
Genetic Diversity: Analysis

Terminal / Command Line

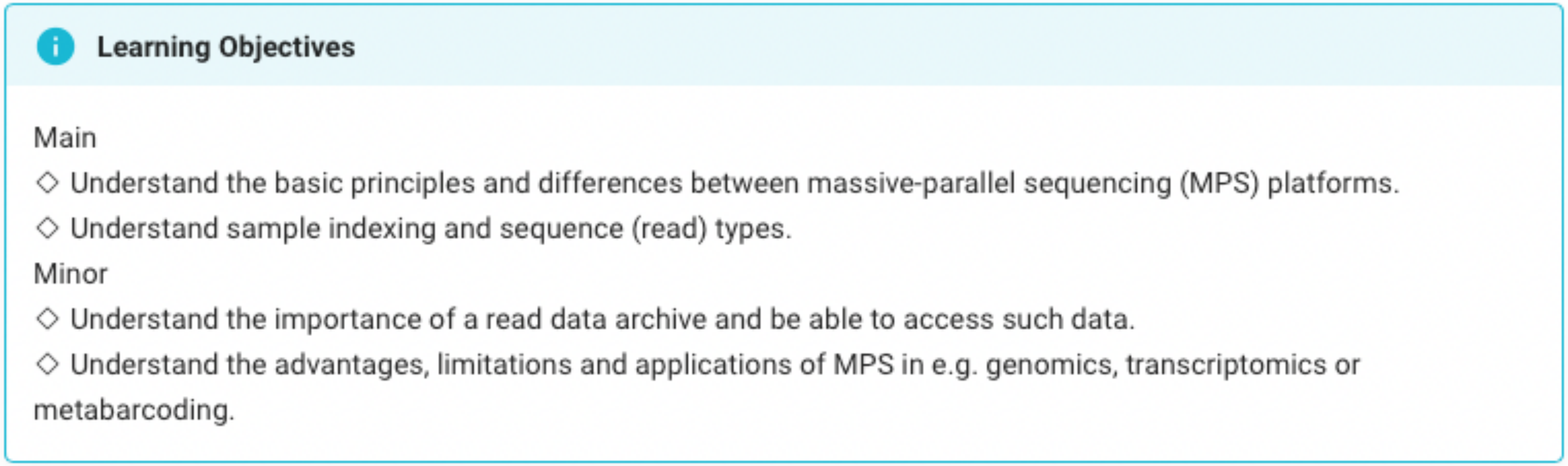
Tuesday, June 18, 2024

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Terminal ▷ Introduction



Learning Objectives

Main

◇ Understand the basic principles and differences between massive-parallel sequencing (MPS) platforms.

◇ Understand sample indexing and sequence (read) types.

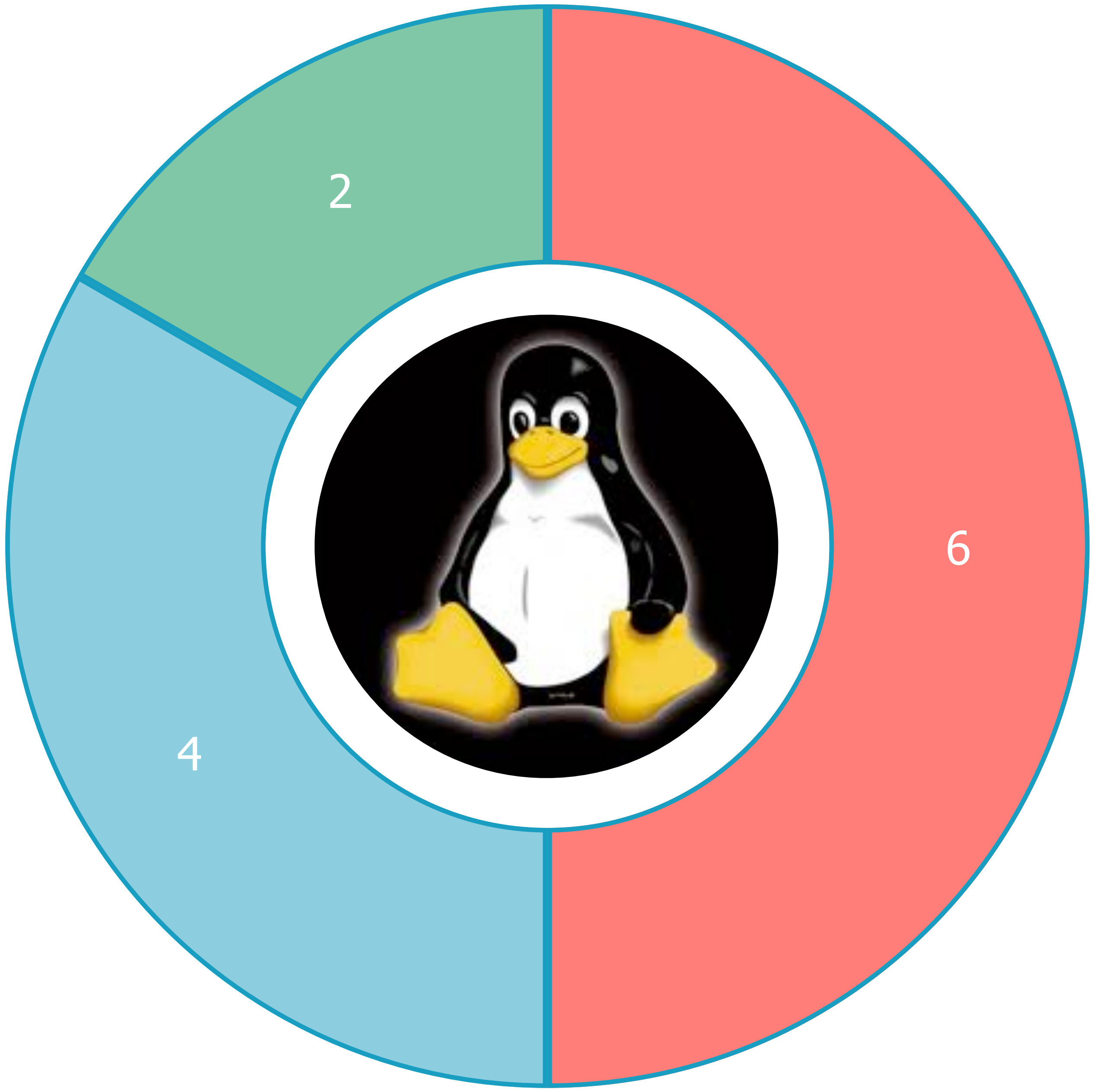
Minor

◇ 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.

Terminal ▷ Introduction

GDC
Zurich Centre Diversity Genetic



🟥 Beginner
🟦 Novice
🟩 Intermediate

Terminal ▷ Introduction

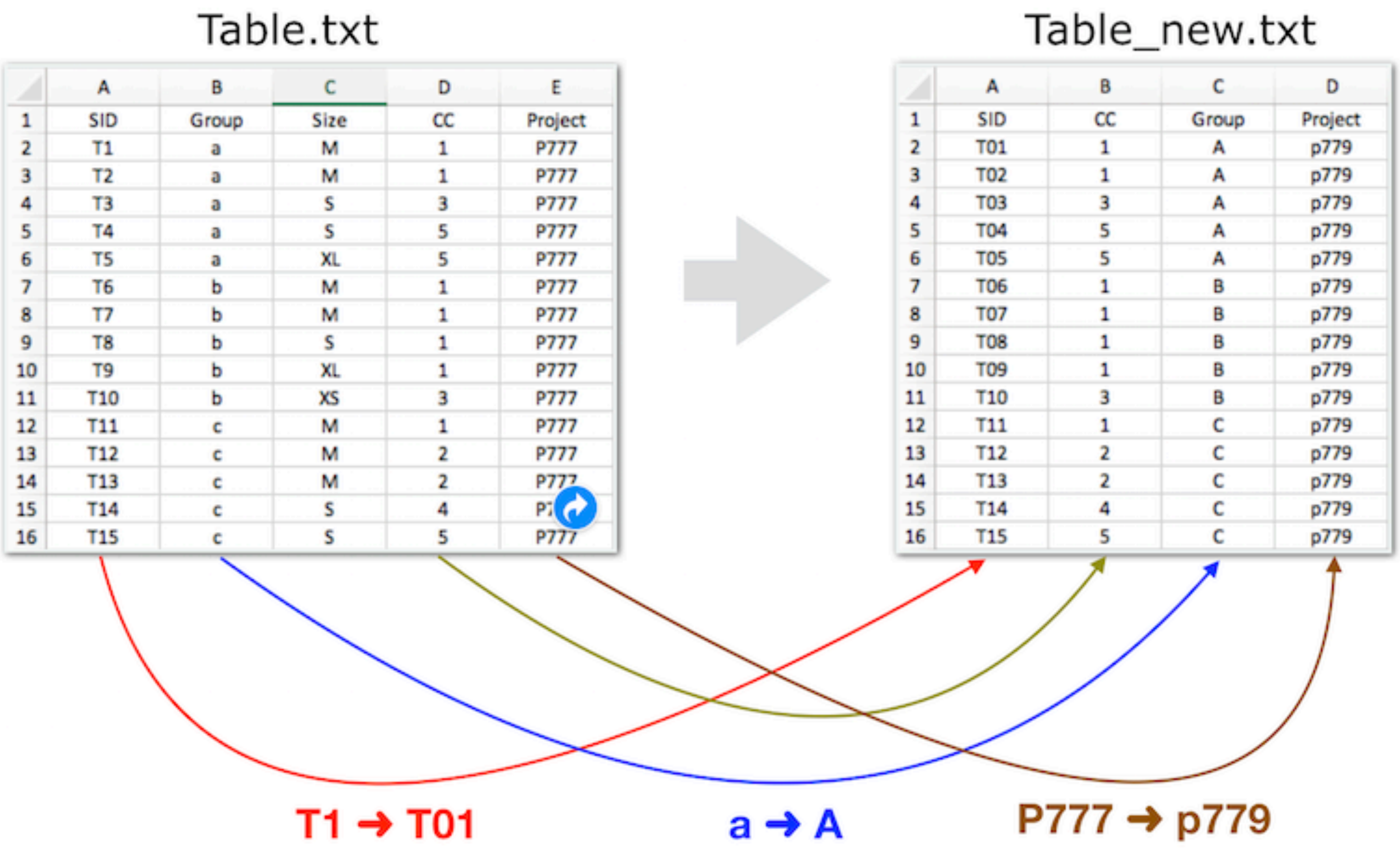


Table.txt

	A	B	C	D	E
1	SID	Group	Size	CC	Project
2	T1	a	M	1	P777
3	T2	a	M	1	P777
4	T3	a	S	3	P777
5	T4	a	S	5	P777
6	T5	a	XL	5	P777
7	T6	b	M	1	P777
8	T7	b	M	1	P777
9	T8	b	S	1	P777
10	T9	b	XL	1	P777
11	T10	b	XS	3	P777
12	T11	c	M	1	P777
13	T12	c	M	2	P777
14	T13	c	M	2	P777
15	T14	c	S	4	P7
16	T15	c	S	5	P777

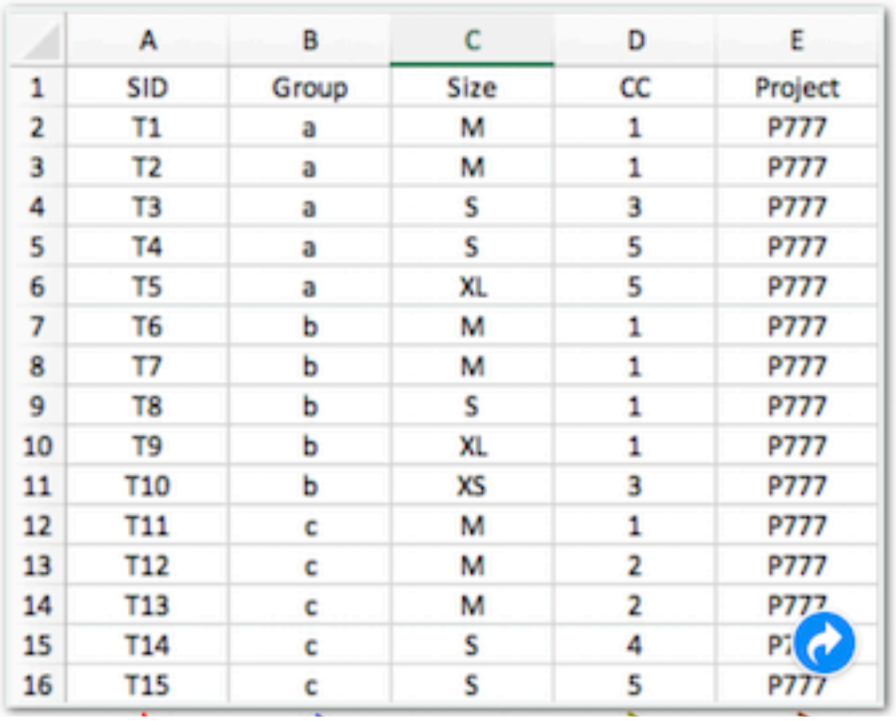
Table_new.txt

	A	B	C	D
1	SID	CC	Group	Project
2	T01	1	A	p779
3	T02	1	A	p779
4	T03	3	A	p779
5	T04	5	A	p779
6	T05	5	A	p779
7	T06	1	B	p779
8	T07	1	B	p779
9	T08	1	B	p779
10	T09	1	B	p779
11	T10	3	B	p779
12	T11	1	C	p779
13	T12	2	C	p779
14	T13	2	C	p779
15	T14	4	C	p779
16	T15	5	C	p779

Terminal ▷ Introduction

```
## Get Table
curl -O https://www.gdc-docs.ethz.ch/GeneticDiversityAnalysis/GDA/data/Table.txt

## Show Table
cat Table.txt

# SID  Group  Size   CC  Project
# T1    a    M    1   P777
# T2    a    M    1   P777
# T3    a    S    3   P777
# T4    a    S    5   P777
# T5    a    XL  5   P777
# T6    b    M    1   P777
# T7    b    M    1   P777
# T8    b    S    1   P777
# T9    b    XL  1   P777
# T10   b    XS  3   P777
# T11   c    M    1   P777
# T12   c    M    2   P777
# T13   c    M    2   P777
# T14   c    S    4   P777
# T15   c    S    5   P777
```

Table.txt

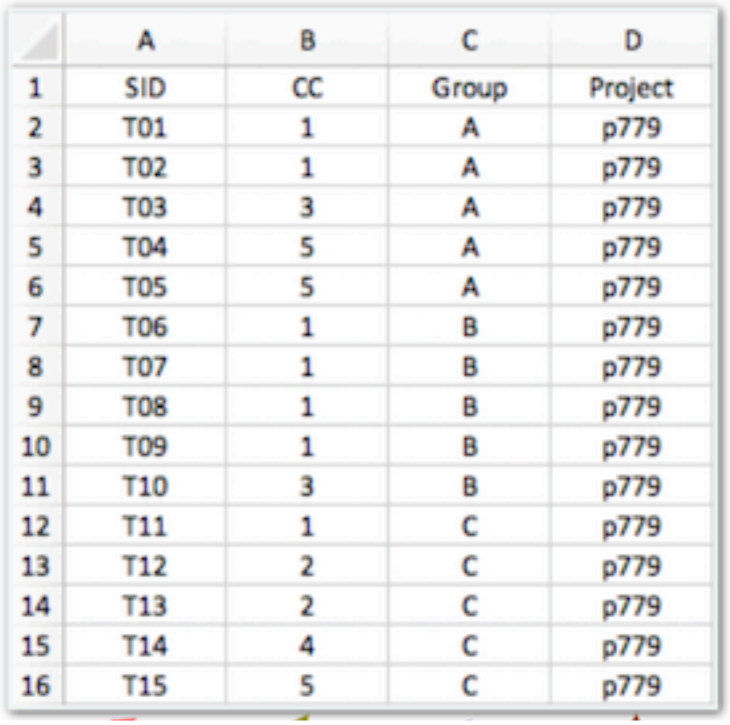


	A	B	C	D	E
1	SID	Group	Size	CC	Project
2	T1	a	M	1	P777
3	T2	a	M	1	P777
4	T3	a	S	3	P777
5	T4	a	S	5	P777
6	T5	a	XL	5	P777
7	T6	b	M	1	P777
8	T7	b	M	1	P777
9	T8	b	S	1	P777
10	T9	b	XL	1	P777
11	T10	b	XS	3	P777
12	T11	c	M	1	P777
13	T12	c	M	2	P777
14	T13	c	M	2	P777
15	T14	c	S	4	P7
16	T15	c	S	5	P777

Introduction

```
## Transform Table
awk '{
  if(NR==1) print $1,$4,$2,$5;                            # exclude header line
  else if(length($1)>2) print $1,$4,toupper($2),"p779";  # change project number
  else print "T0"substr($1,2,3),$4,toupper($2),"p779"    # rename sample < 10
}' Table.txt > Table_new.txt

## Show NEW Table
cat Table_new.txt

# SID CC Group Project
# T01 1 A p779
# T02 1 A p779
# T03 3 A p779
# T04 5 A p779
# T05 5 A p779
# T06 1 B p779
# T07 1 B p779
# T08 1 B p779
# T09 1 B p779
# T10 3 B p779
# T11 1 C p779
# T12 2 C p779
# T13 2 C p779
# T14 4 C p779
# T15 5 C p779
```

Table_new.txt

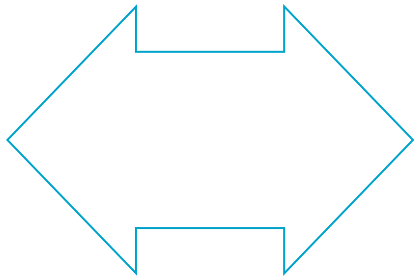


	A	B	C	D
1	SID	CC	Group	Project
2	T01	1	A	p779
3	T02	1	A	p779
4	T03	3	A	p779
5	T04	5	A	p779
6	T05	5	A	p779
7	T06	1	B	p779
8	T07	1	B	p779
9	T08	1	B	p779
10	T09	1	B	p779
11	T10	3	B	p779
12	T11	1	C	p779
13	T12	2	C	p779
14	T13	2	C	p779
15	T14	4	C	p779
16	T15	5	C	p779

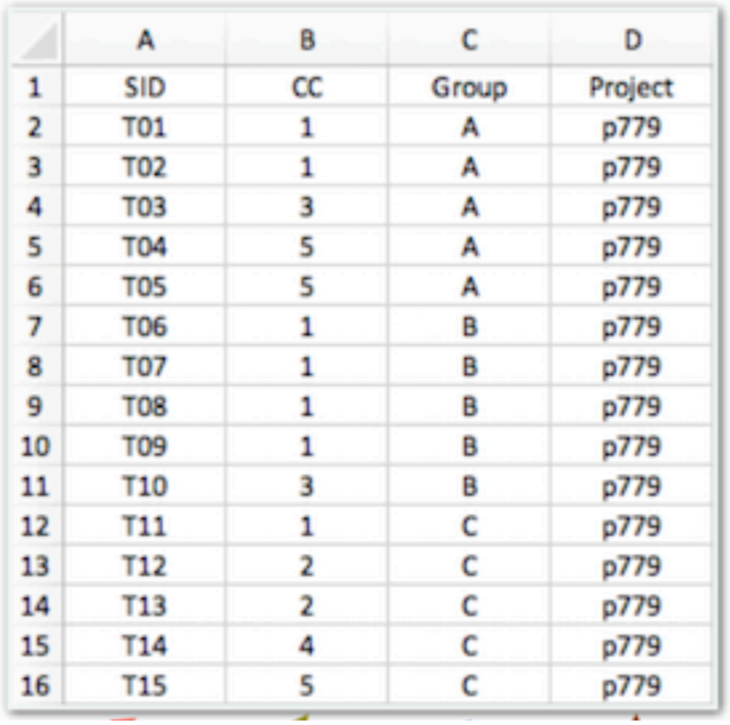
Terminal ▷ Introduction

Alternative solution providing a bash script:

```
## Get Table
curl --verbose -O https://gdc-docs.ethz.ch/GeneticDiversityAnalysis/GDA/scripts/reformat_table.sh

# Make the script executable
chmod a+x reformat_table.sh

# run the script but make sure the Table.txt file is in the same directory
./reformat_table.sh Table.txt

## Show NEW Table

cat Table_new.txt
# SID CC Group Project
# T01 1 A p779
# T02 1 A p779
# T03 3 A p779
# T04 5 A p779
# T05 5 A p779
# T06 1 B p779
# T07 1 B p779
# T08 1 B p779
# T09 1 B p779
# T10 3 B p779
# T11 1 C p779
# T12 2 C p779
# T13 2 C p779
# T14 4 C p779
# T15 5 C p779
```



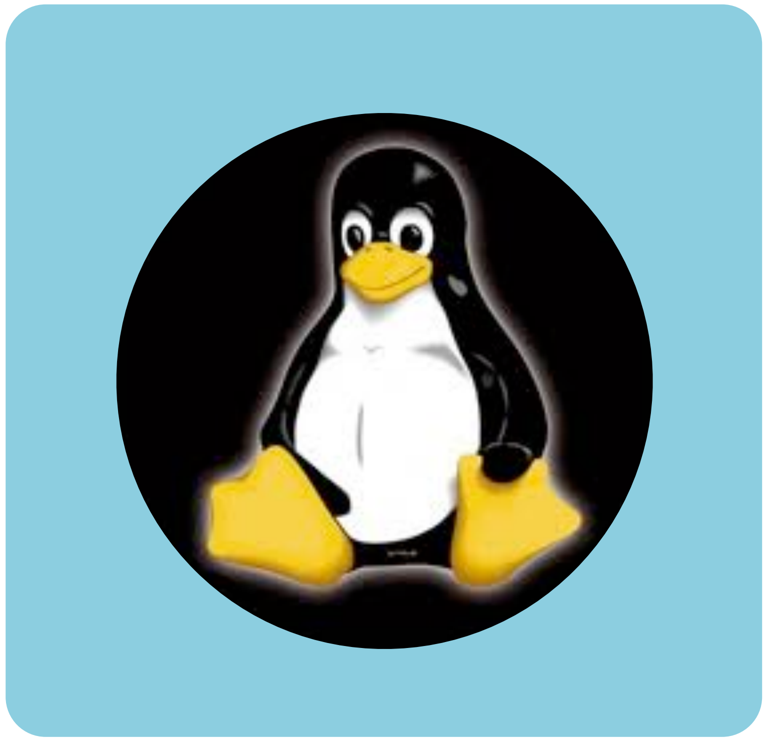
Table_new.txt



	A	B	C	D
1	SID	CC	Group	Project
2	T01	1	A	p779
3	T02	1	A	p779
4	T03	3	A	p779
5	T04	5	A	p779
6	T05	5	A	p779
7	T06	1	B	p779
8	T07	1	B	p779
9	T08	1	B	p779
10	T09	1	B	p779
11	T10	3	B	p779
12	T11	1	C	p779
13	T12	2	C	p779
14	T13	2	C	p779
15	T14	4	C	p779
16	T15	5	C	p779

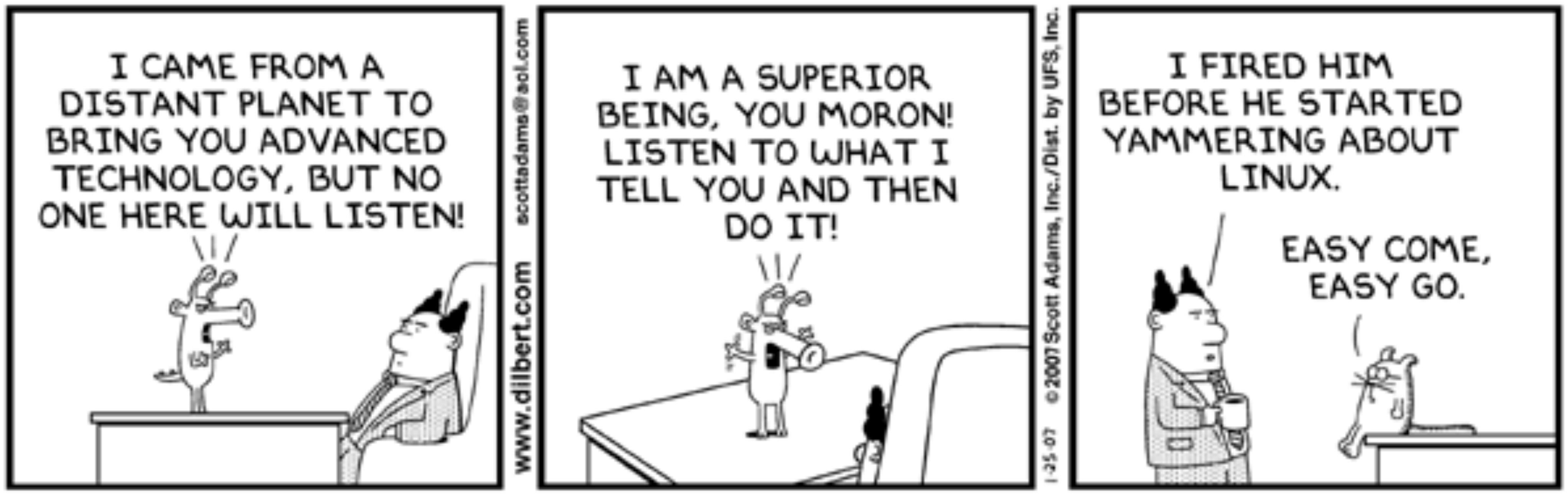
Different OSs and Versions

Mac


Linux

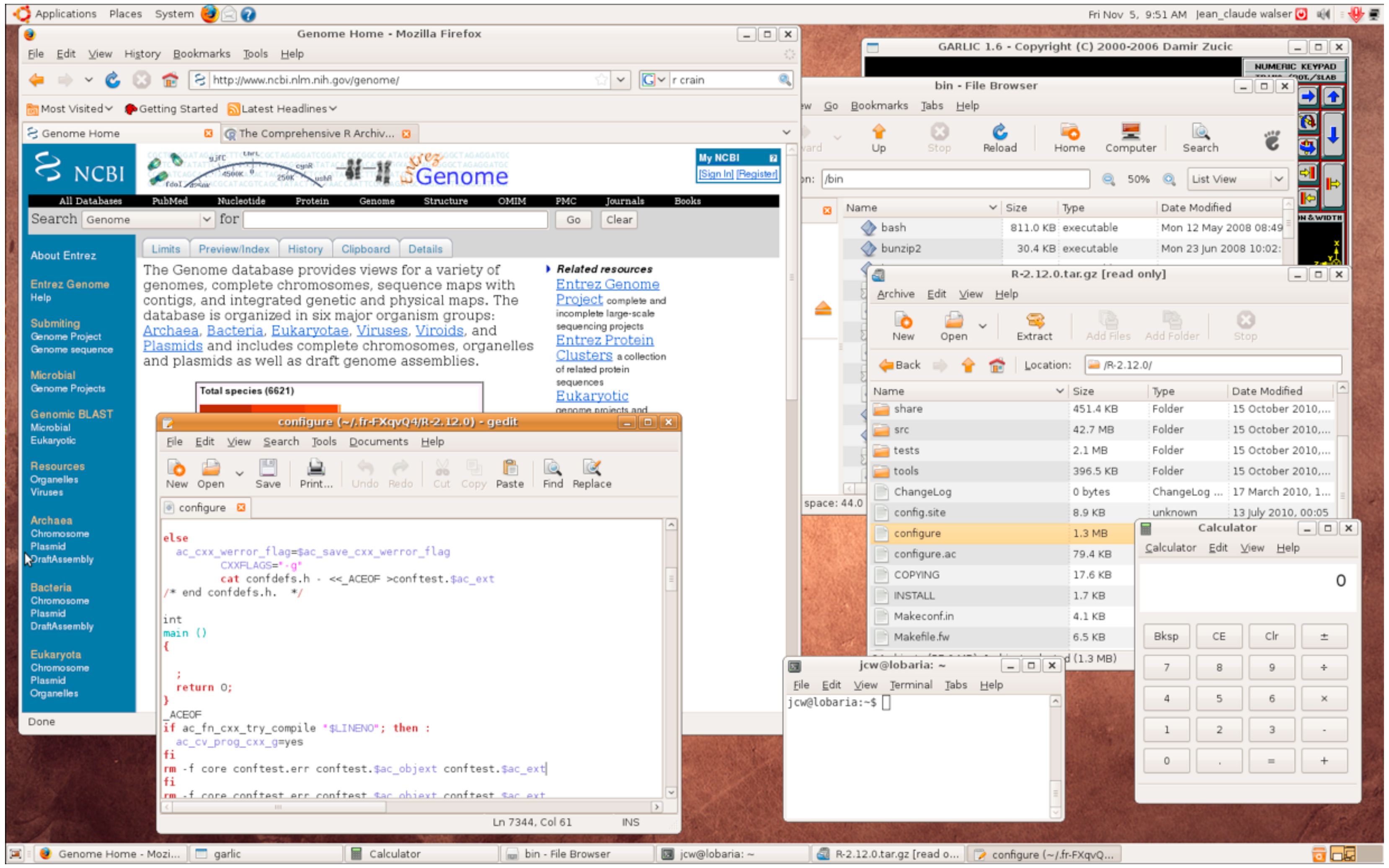
Windows

Introduction


I CAME FROM A DISTANT PLANET TO BRING YOU ADVANCED TECHNOLOGY, BUT NO ONE HERE WILL LISTEN!
scottadams@aol.com
www.dilbert.com
I AM A SUPERIOR BEING, YOU MORON! LISTEN TO WHAT I TELL YOU AND THEN DO IT!
1-25-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.
I FIRED HIM BEFORE HE STARTED YAMMERING ABOUT LINUX.
EASY COME, EASY GO.

Terminal ▷ Introduction

A **graphical user interface (GUI)** - often pronounced gooey - an interface that allows the user (you) to interact with programs in more ways than typing.



The Genome database provides views for a variety of genomes, complete chromosomes, sequence maps with contigs, and integrated genetic and physical maps. The database is organized in six major organism groups: Archaea, Bacteria, Eukaryotae, Viruses, Viroids, and Plasmids and includes complete chromosomes, organelles and plasmids as well as draft genome assemblies.

```
else
  ac_cxx_werror_flag=$ac_save_cxx_werror_flag
         CXXFLAGS="-g"
         cat confdefs.h - <<_ACEOF >conftest.$ac_ext
/* end confdefs.h.  */

int
main ()
{

  ;
  return 0;
}
_ACEOF
if ac_fn_cxx_try_compile "$LINENO"; then :
  ac_cv_prog_cxx_g=yes
fi
rm -f core conftest.err conftest.$ac_objext conftest.$ac_ext
fi
```

Name	Size	Type	Date Modified
share	451.4 KB	Folder	15 October 2010,...
src	42.7 MB	Folder	15 October 2010,...
tests	2.1 MB	Folder	15 October 2010,...
tools	396.5 KB	Folder	15 October 2010,...
ChangeLog	0 bytes	ChangeLog ...	17 March 2010, 1...
config.site	8.9 KB	unknown	13 July 2010, 00:05
configure	1.3 MB		
configure.ac	79.4 KB		
COPYING	17.6 KB		
INSTALL	1.7 KB		
Makeconf.in	4.1 KB		
Makefile.fw	6.5 KB		

```
jcw@lobaria:~$ 
```

GUIs are nice but limited. Don't be afraid to use the terminal!

Terminal ▷ Introduction

"Where there is a shell, there is a way."

GUIs were introduced in reaction to the steep learning curve of **command-line interfaces (CLI)**, which require commands to be typed on the keyboard. Since the commands available in command line interfaces can be numerous, complicated operations can be completed using a short sequence of words and symbols. **This allows for greater efficiency, productivity once many commands are learned, and better reproducibility.**

Shell - Terminal

Shell is a UNIX term for the interactive user interface with an operating system. The shell is the layer of programming that understands and executes the commands a user enters.

Bourne-Shell (sh)
Korn-Shell (ksh)
C-Shell (csh)
TC-Shell (tcsh)
Bourne-Again-Shell (bash)
Debian Almquist Shell (dash)
Z-Shell (zsh)
A-Shell (ash)
PowerShell / cmd.exe

What do I have?

```
$> echo ${SHELL}
```

Terminal ▷ Introduction

```
Bash > zcat file.gz
```

```
Zsh > zcat < file.gz
```

```
Bash > sed -i 's/d/D/2' file.txt
```

```
Zsh > sed -i"" "s/d/D/2" file.txt
```

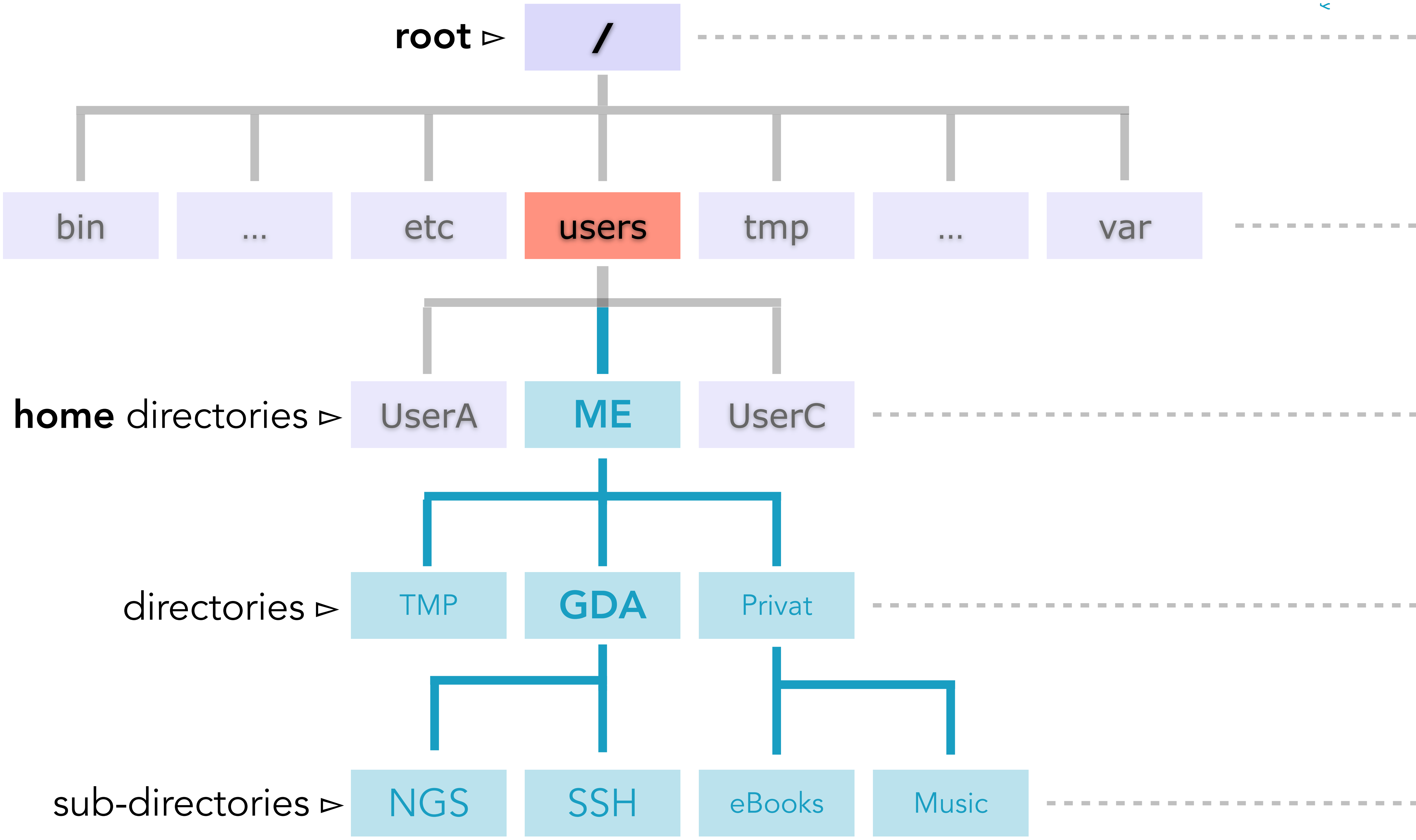
GDC
Genetic Diversity Centre Zurich

Change Shell

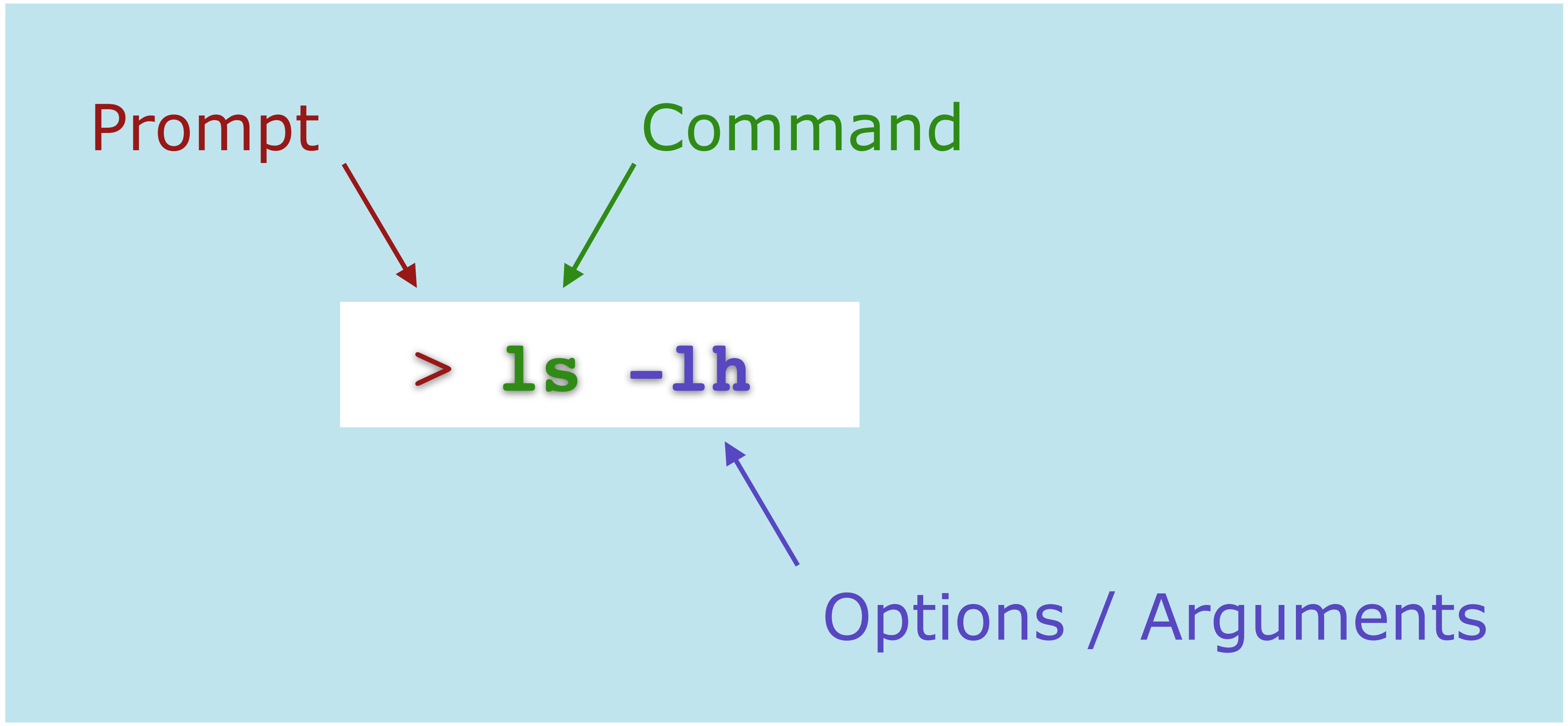
```
echo ${SHELL}
#which zsh
#which bash

## Bash > Zsh
chsh -s /bin/bash

## Zsh > Bash
chsh -s /bin/zsh
```

Which one is better? Both shells will get the job done. The bash is a bit outdated (version 3.2 - 2006). Many of the conveniences provided by zsh can be made available in bash. It seems zsh is more helpful to newer shell user.

Introduction



Command - Line



```
> ls -lh
```

Built-in Help

```
> info <command>
```

```
> info ls
```

```
> man <command>
```

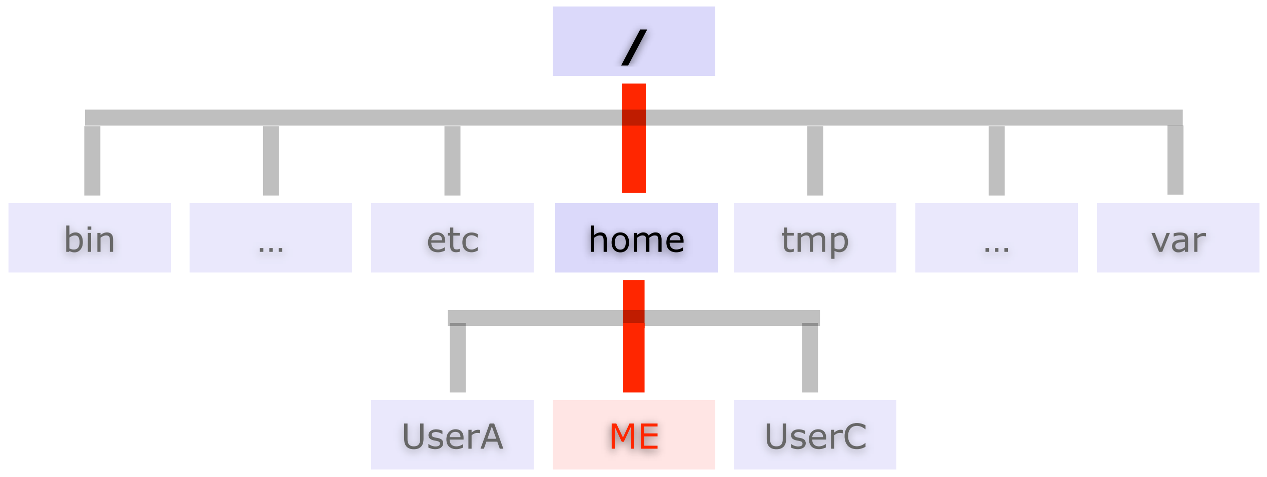
```
> man ls
```

* press Q to leave info or manual

Terminal ▷ Introduction

GDC Zurich Centre Diversity Genetic

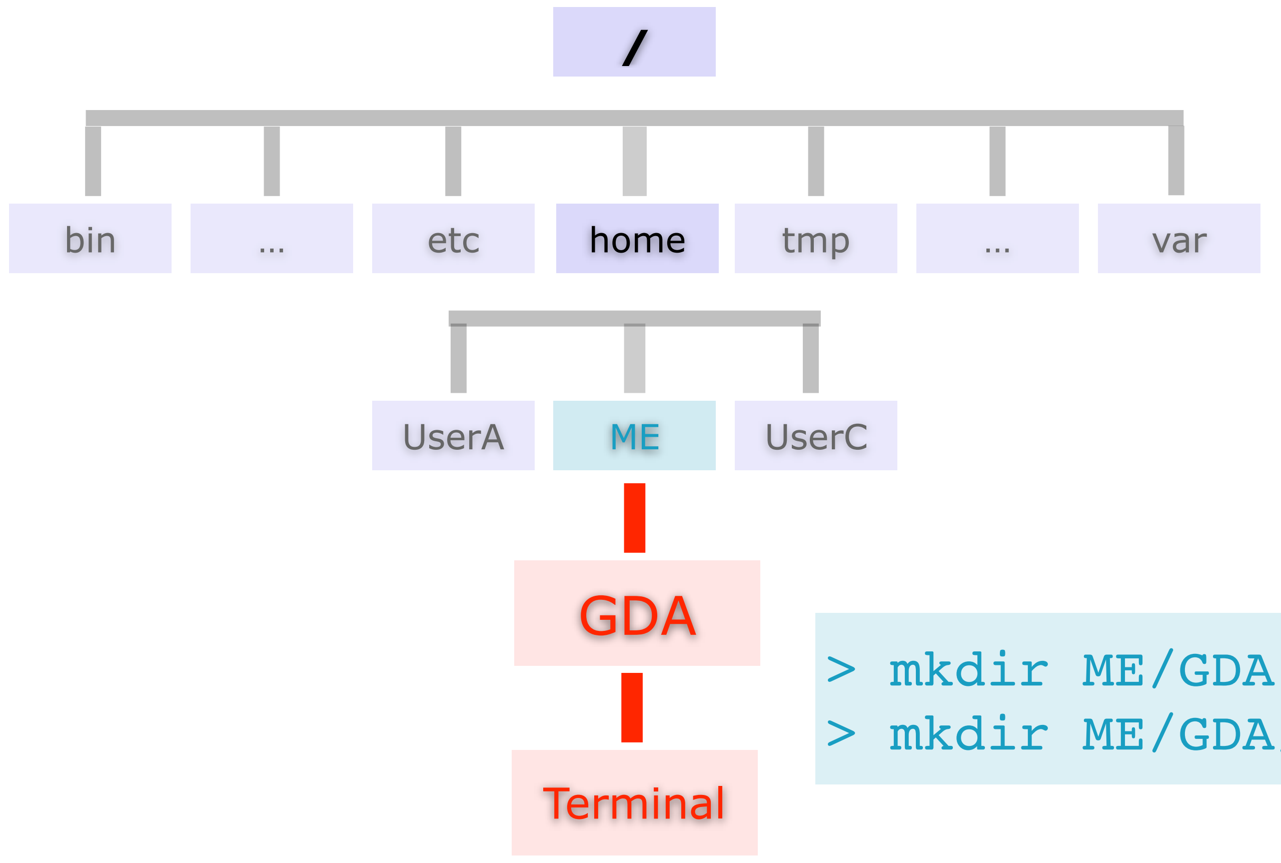
pwd - print name of current/working directory



```
> pwd
/home/ME
```

```
> pwd
/home/ME
> cd ${HOME}
> pwd
/home/ME
> cd ~
> pwd
/home/ME
> cd
/home/ME
```

mkdir - creating/making directories



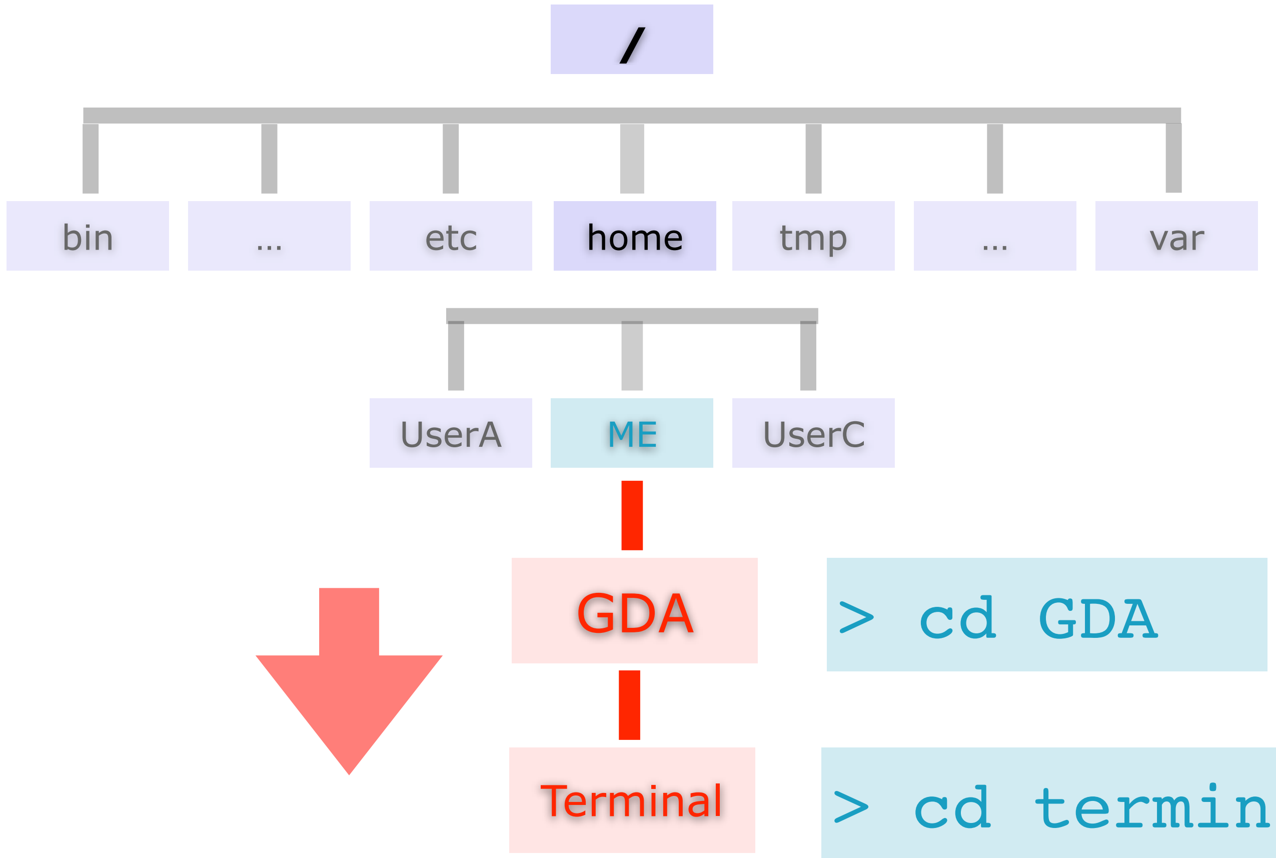
```
> mkdir ME/GDA
> mkdir ME/GDA/Terminal
```

Alternatively, use option p to create diretories and sub-directories

```
> mkdir -p ME/GDA/Terminal
```

GDC Genetic Diversity Centre Zurich

cd - change directory



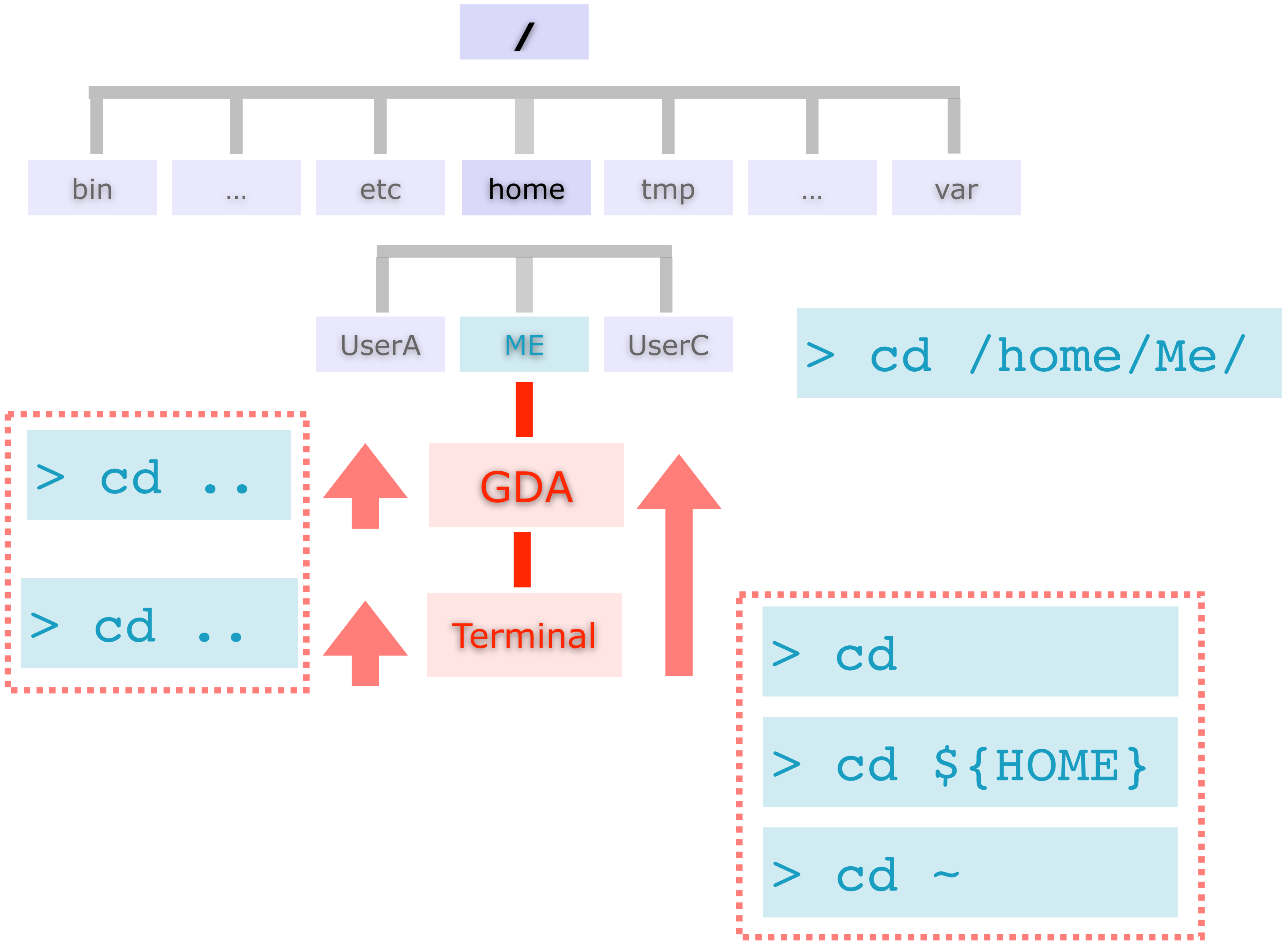
```
> cd GDA
```

Alternative

```
> cd GDA/terminal
```

```
> cd terminal
```

ways to go home



```
> cd /home/Me/
```

```
> cd ..
> cd ..
```

```
> cd
> cd ${HOME}
> cd ~
```

Terminal ▷ Introduction

Local
Software
Management

Software
Dependencies

Parallel
Versions

GDC Genetic Diversity Centre Zurich

CONDA

*Package, **dependency and environment management** for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN. Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.*

BIOCONDA®

Bioconda is a ***channel for the conda package manager*** *specialising in bioinformatics software. The conda package manager makes installing software a vastly more streamlined process. Conda is a combination of other package managers you may have encountered, such as pip, CPAN, CRAN, Bioconductor, apt-get, and homebrew. Conda is both language- and OS-agnostic, and can be used to install C/C++, Fortran, Go, R, Python, Java etc programs on Linux, Mac OSX, and Windows.*

GDC
Zurich Centre Diversity Genetic

```
python --version
# Python 2.7.15
bwa
# -bash: bwa: command not found
blast -help
# -bash: blast: command not found
```

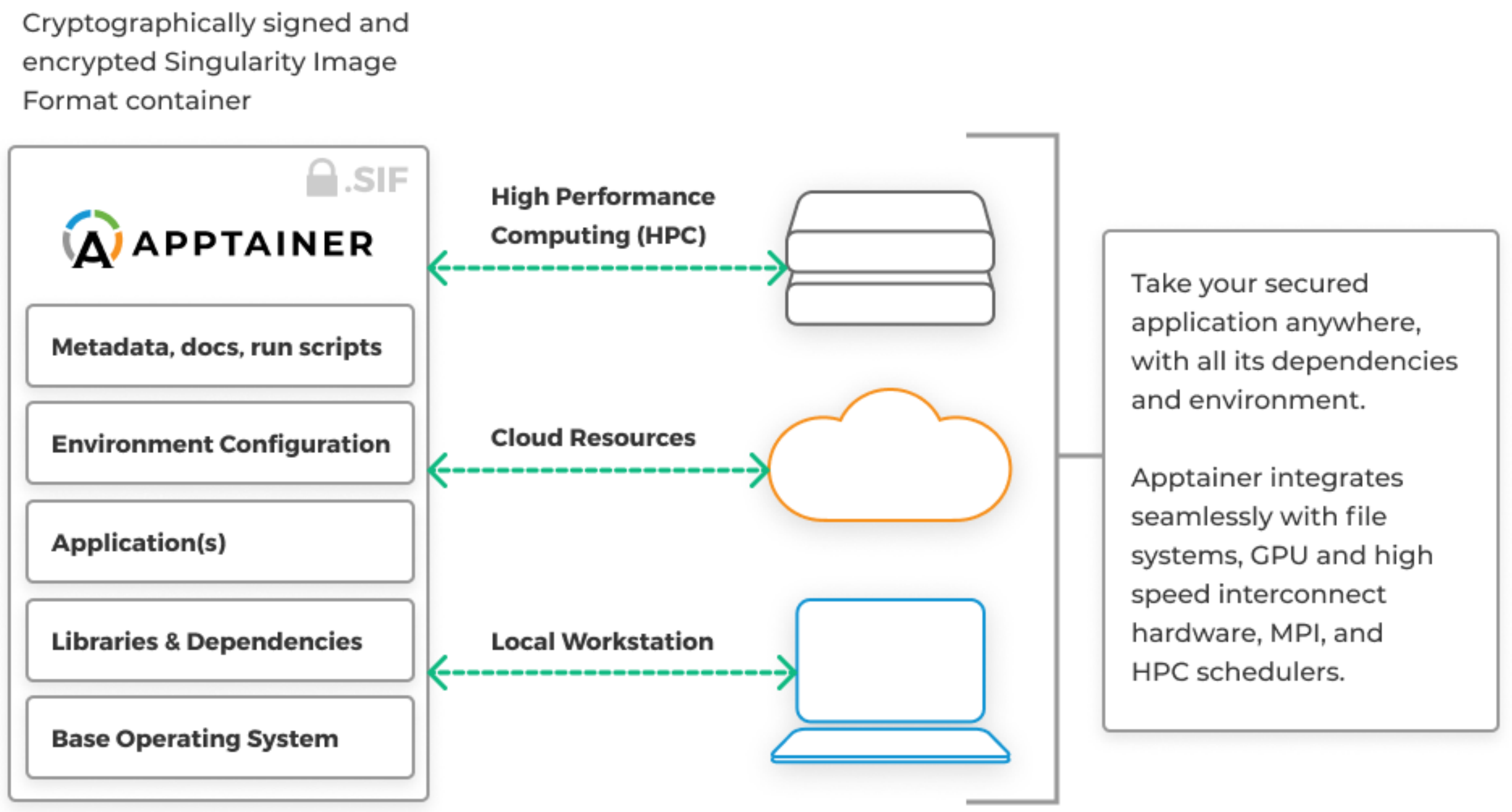
```
conda info --envs
source activate aligners
conda info --envs
python --version
# Python 3.6.7
bwa
blastn -help
```

Terminal ▷ Introduction

1. **Portability**: Containers are portable and can be deployed across different systems without modifications. They encapsulate the application and its dependencies, ensuring consistent behaviour regardless of the underlying infrastructure.
2. **Scalability**: Containers enable applications to be easily scaled horizontally by running multiple instances of the same container across a cluster of machines. This allows for efficient resource utilisation and the ability to handle increased workload demands.
3. **Efficiency**: Containers share the host operating system's kernel and only require the necessary dependencies, making them lightweight and resource-efficient compared to traditional virtual machines.
4. **Isolation**: Containers provide a level of isolation between the application and the host operating system, as well as between different containers. This isolation helps prevent conflicts between applications and improves security by limiting the impact of any potential vulnerabilities.
5. **Rapid deployment**: Containers simplify the deployment process by providing a consistent environment. They can be quickly started, stopped, and updated, allowing for fast and efficient application deployment and rollbacks.

Containerisation is a technique used in software development and deployment that **allows applications and their dependencies to be packaged together into self-contained units called containers**. Containers provide a lightweight and isolated runtime environment that can run consistently across different computing environments, such as laptops, servers, or cloud platforms.

Introduction


Cryptographically signed and encrypted Singularity Image Format container
.SIF
APPTAINER
Metadata, docs, run scripts
Environment Configuration
Application(s)
Libraries & Dependencies
Base Operating System
High Performance Computing (HPC)
Cloud Resources
Local Workstation
Take your secured application anywhere, with all its dependencies and environment.
Apptainer integrates seamlessly with file systems, GPU and high speed interconnect hardware, MPI, and HPC schedulers.

1. **Reproducibility**: Singularity enables researchers to create and distribute containers that encapsulate their entire software stack, including the application, libraries, and dependencies. This ensures that the environment remains consistent and reproducible across different systems.
2. **Compatibility** with HPC: Singularity is designed to seamlessly integrate with HPC clusters and resource managers commonly used in scientific computing. It allows users to leverage specialised hardware resources, such as GPUs and high-performance interconnects, while still providing isolation and security.
3. **Security** and user isolation: Singularity employs a security model that focuses on user isolation and containment. It ensures that users can run containers securely without requiring escalated privileges, making it suitable for multi-user HPC environments.
4. **Mobility**: Singularity containers are portable and can be easily shared, distributed, and run on different systems, including local workstations, clusters, and cloud platforms.
5. **Interoperability**: Singularity supports a wide range of container image formats, including Docker, making it possible to leverage existing container images and workflows.

Singularity (Apptainer) is an open-source **containerisation platform** that focuses on providing secure and reproducible environments for scientific and **high-performance computing** (HPC) workloads. It was developed specifically to address the challenges faced by researchers and scientists when working with complex software stacks and specialized computing resources.

EXTRAS

Terminal ▷ Introduction

```
#!/usr/bin/perl -w
use strict;

my $input_fasta=$ARGV[0];
open(IN,"<$input_fasta") || die ("Error opening $input_fasta $!");

my $line = <IN>;
print $line;

while ($line = <IN>)
{
 chomp $line;
 if ($line=~m/^>gi/) { print "\n",$line,"\n"; }
 else { print $line; }
}

print "\n";
```

```
perl fa2one.pl multiple.fa > one.fa
```

Terminal ▷ Introduction

```
#!/usr/bin/perl -w
use strict;

# Perl script to reformat fasta files
# Usage: perl fa2one.pl multiple.fa > one.fa

my $input_fasta=$ARGV[0];
open(IN,"<$input_fasta") || die ("Error opening $input_fasta $!");

my $line = <IN>;
print $line;

while ($line = <IN>)
{
 chomp $line;          ↙ limiting factor
 if ($line=~m/^>gi/) { print "\n",$line,"\n"; }
 else { print $line; }
}

print "\n";
```

```
perl fa2one.pl multiple.fa > one.fa
```