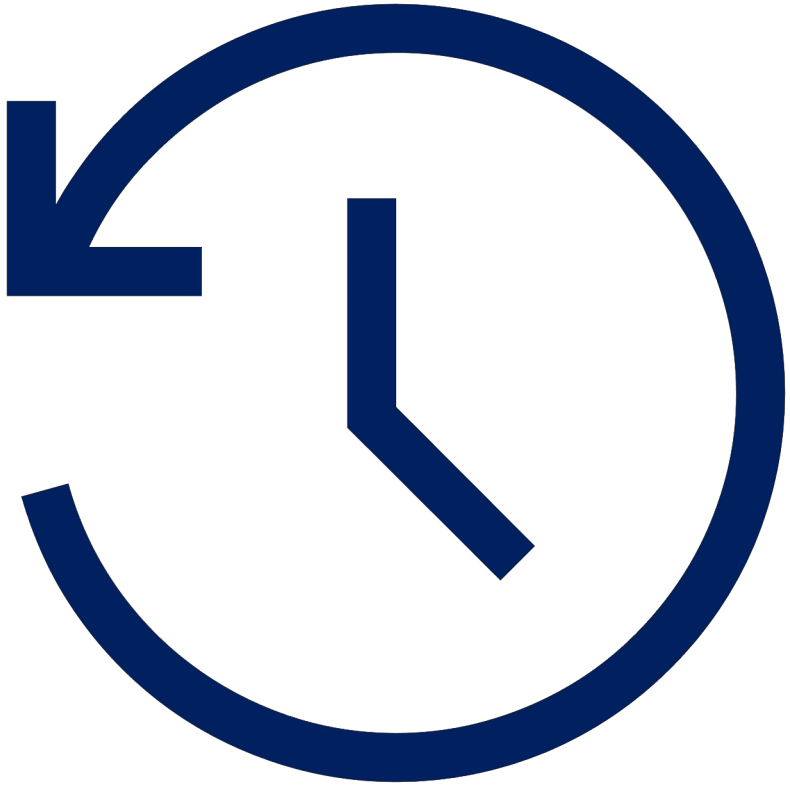
Course Recap

February 14, 2023



Overview for today's lecture

- Update in Course Grading
- Unit 1 recap
- Final project recap
- Unit 2
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - Conda Download
  - Data management

Overview for today's lecture

- **Update in Course Grading**
- Unit 1 recap
- Final project recap
- Unit 2
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - Conda Download
  - Data management

Change in Final Grade

Homework	40%
Final Project	30%
Primary article review/reflection	20%
Class Participation & Attendance	10%
Total	100%

Change in Final Grade

Homework	50%
Final Project	30%
Primary article review/reflection	10%
Class Participation & Attendance	10%
Total	100%

April 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

www.a-printable-calendar.com

Primary literature summary #1 due

scRNA-Seq

scRNA-Seq primary literature review #2 due

Final Project presentation #1

April 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

www.a-printable-calendar.com


Primary literature summary #1 due


scRNA-Seq ⟶ 1 homework or 2 “mini” homework


Final Project presentation #1

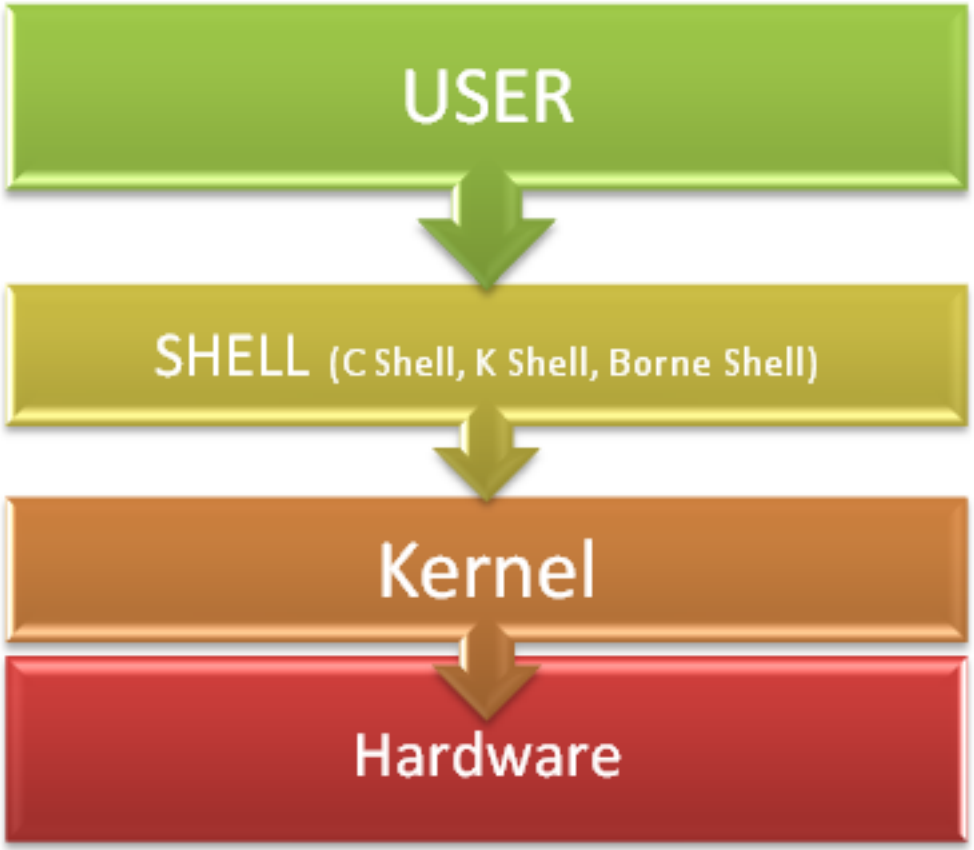
Overview for today's lecture

- Update in Course Grading
- **Unit 1 recap**
- Final project recap
- Unit 2
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - Conda Download
  - Data management

Shell

- A shell is a command line interpreter, used to launch software in Linux
- There are many different shells available:
  - BASH
  - CSH
  - ZSH etc.
- Most software will work the same in all shells
- Some functions and automation are different between shells

- We have been using the most popular shell, BASH



Operating system

Just like Windows, MacOS, Linux is an operating system

VACC is built on a LINUX server built on a LINUX operating system

Running programs

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

Running programs

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

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

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

The structure of a unix command

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

But what happens if you need or want to install a program on the VACC?

- You can do this by following the instructions, however, make sure the program you are installing is correct for the operating system we are using!

01. Downloading SRA Toolkit

Andrew Klymenko edited this page on Dec 12, 2022 · 28 revisions

NCBI SRA Toolkit

Below are the latest releases of various tools and release checksum file.

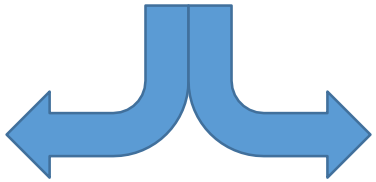
SRA Toolkit

Compiled binaries/install scripts of December 12, 2022, version 3.0.2:

- CentOS Linux 64 bit architecture - non-sudo tar archive
- Ubuntu Linux 64 bit architecture - non-sudo tar archive
- Cloud - apt-get install script - for Debian and Ubuntu - requires sudo permissions
- Cloud - yum install script - for CentOS - requires sudo permissions
- MacOS 64 bit architecture
- MS Windows 64 bit architecture
- Docker image repository
- md5 checksums


UNIX®
00011110 00011110 00011110 00011110 00011110 00011110 00011110 00011110


Linux™




debian
ubuntu


redhat
CentOS
AlmaLinux
Rocky Linux™

Overview for today's lecture

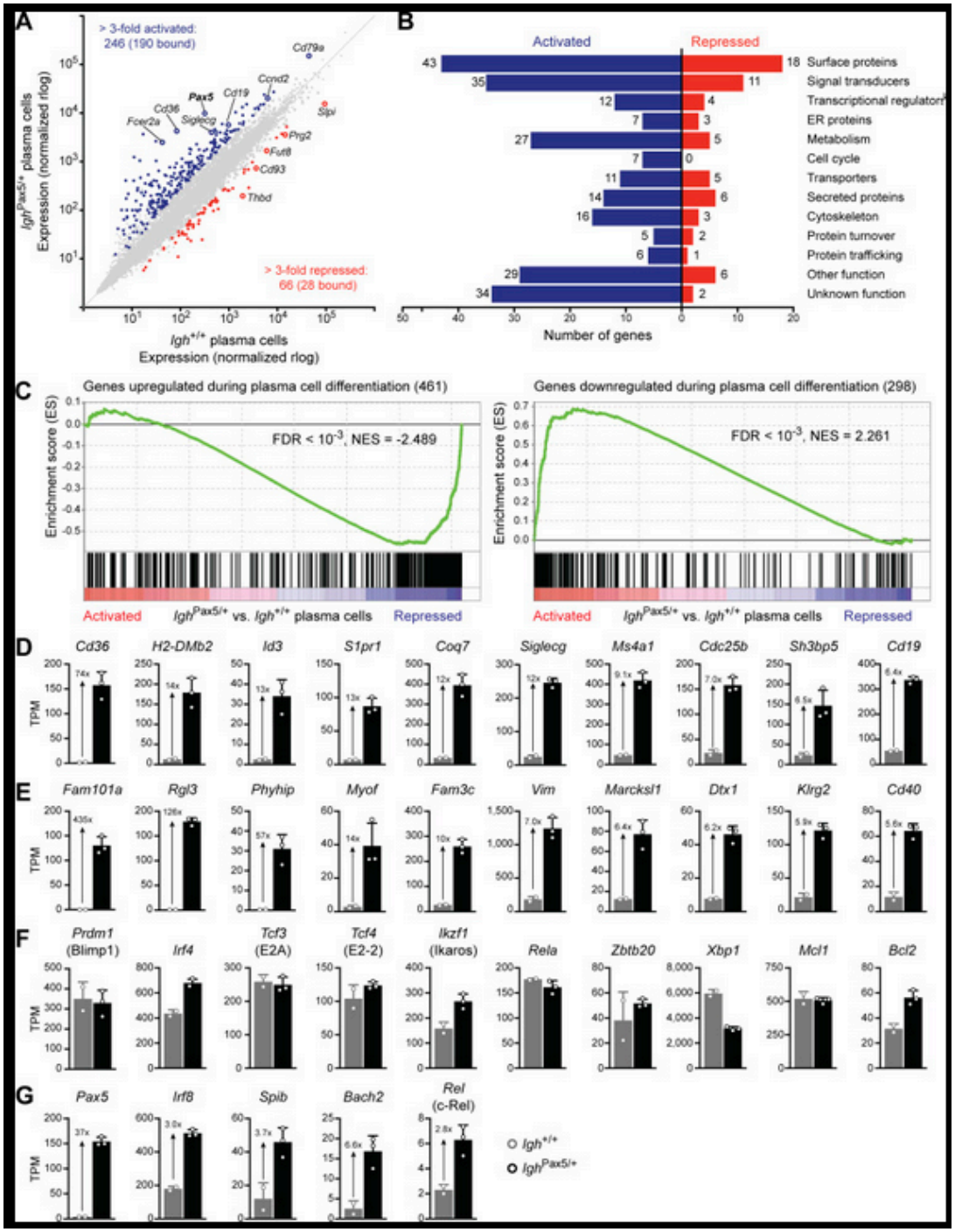
- Update in Course Grading
- Unit 1 recap
- **Final project recap**
- Unit 2
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - Conda Download
  - Data management

Application of bioinformatics


Green Mountain Trail

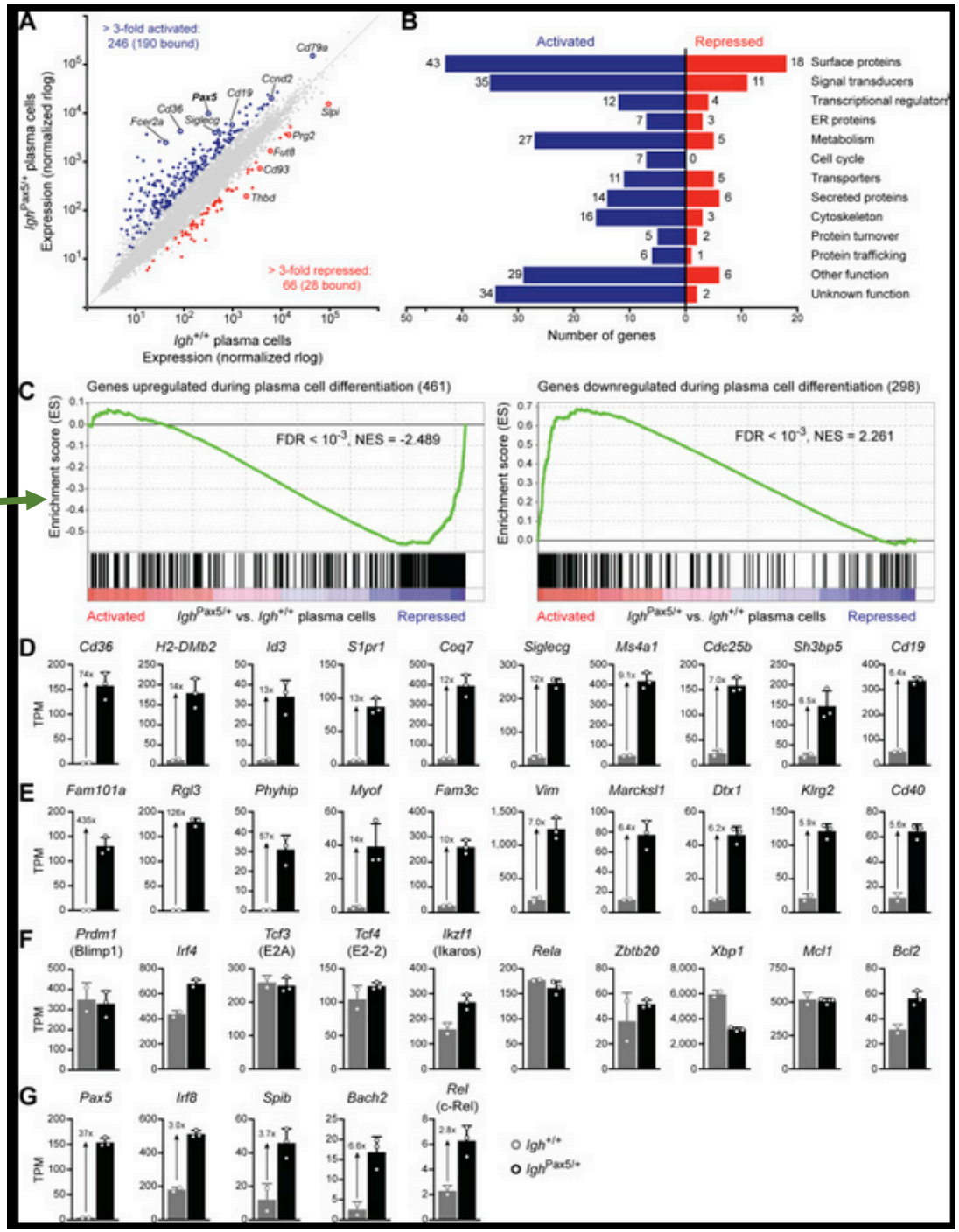
Green Mountain Trail

Your overall goal should be to replicate a Figure or Figure panel (A-C) from a peer-reviewed article



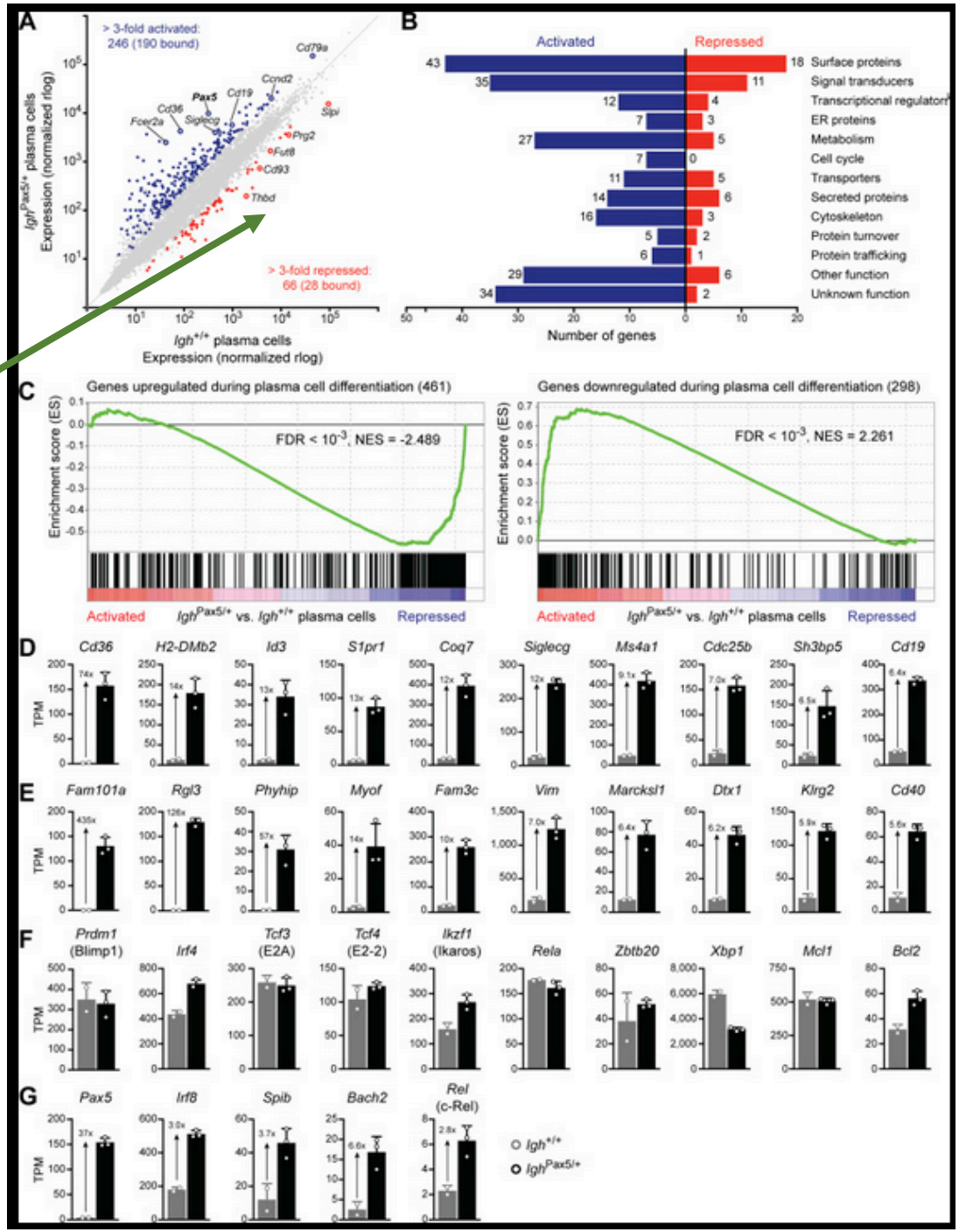
Green Mountain Trail

Perhaps you never truly understood what an Enrichment Score (ES) is?



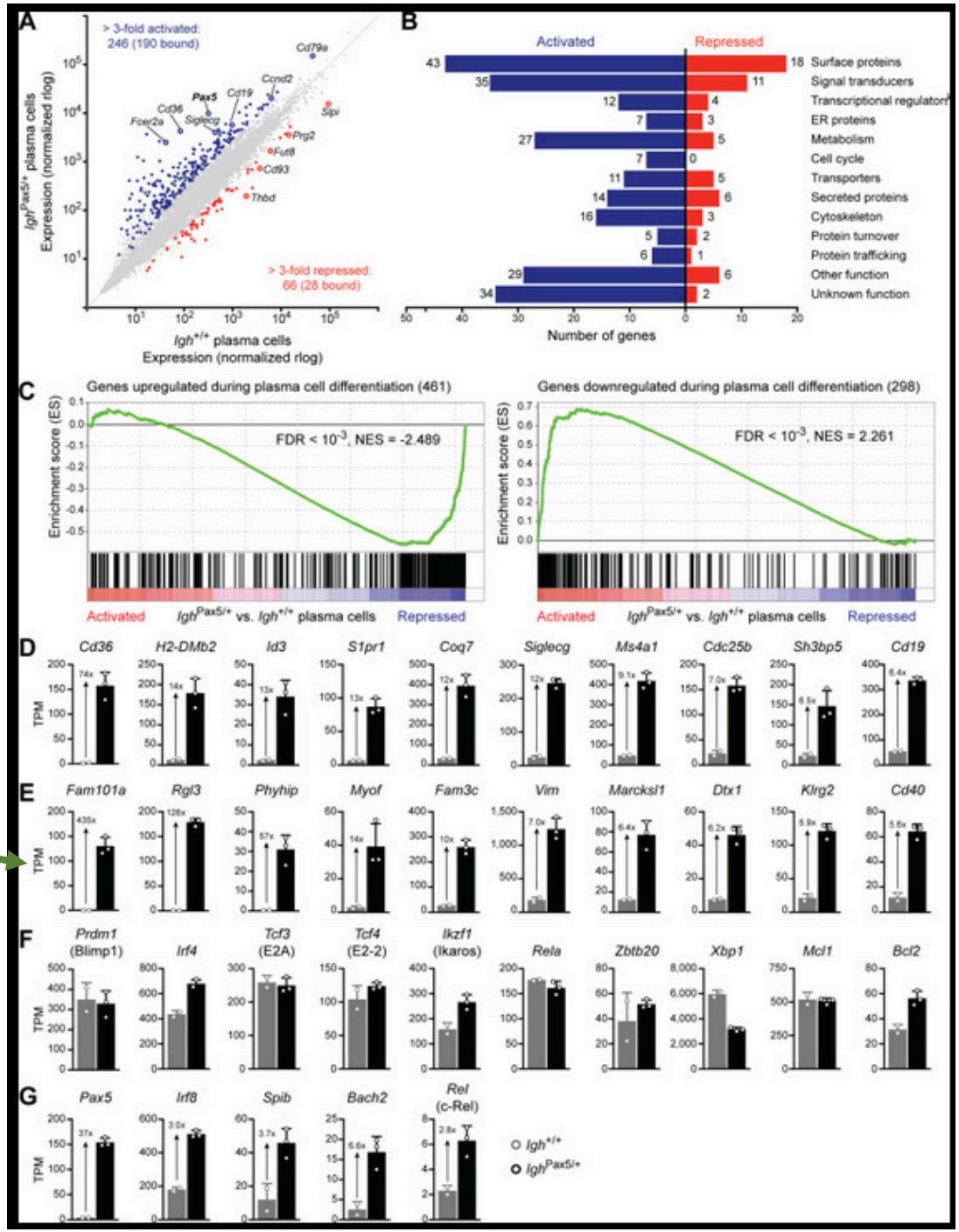
Green Mountain Trail

How would you label this MA plot?



Green Mountain Trail

What does TPM mean




Blue Sky Trail

The bioinformatic pipeline we will learn in class is:

MMG232	What it does…
FASTQC	Quality control FASTQC files
Trimmomatic	Trim adaptors and low quality reads
HISAT2 STAR	Alignment to Genome
SAMtools	SAM to BAM
HTSeq-count	Create counts files

RNA-Seq and bioinformatics analysis

Total RNA prepared was extracted using TRIzol Reagent (Invitrogen, Carlsbad, CA, USA) and submitted to Shanghai Personal Biotechnology, where RNA intergrity was confirmed using the Illumina HiSeq X ten system at 150 bp pair-ended. Double-strand cDNA libraries were prepared and constructed using the TruSeq RNA Sample Prep Kit (Illumina, San Diego, CA). Two replicates of the RNA-Seq experiments were performed. RNA-Seq reads were quality controlled and trimmed for adapter sequences using Trim Galore. Filtered reads were aligned to hg38 using HISAT2. Read counts for each gene were carried out using HT-Seq using the hg38 refSeq refFlat GTF file accessed on July 2015. Differentially expressed genes (DEGs) were analysed using the DESeq2 package ($|\text{fold change}| \geq 1.5$, $P < 0.05$).

FASTQC

Trimmomatic

HISAT2

STAR

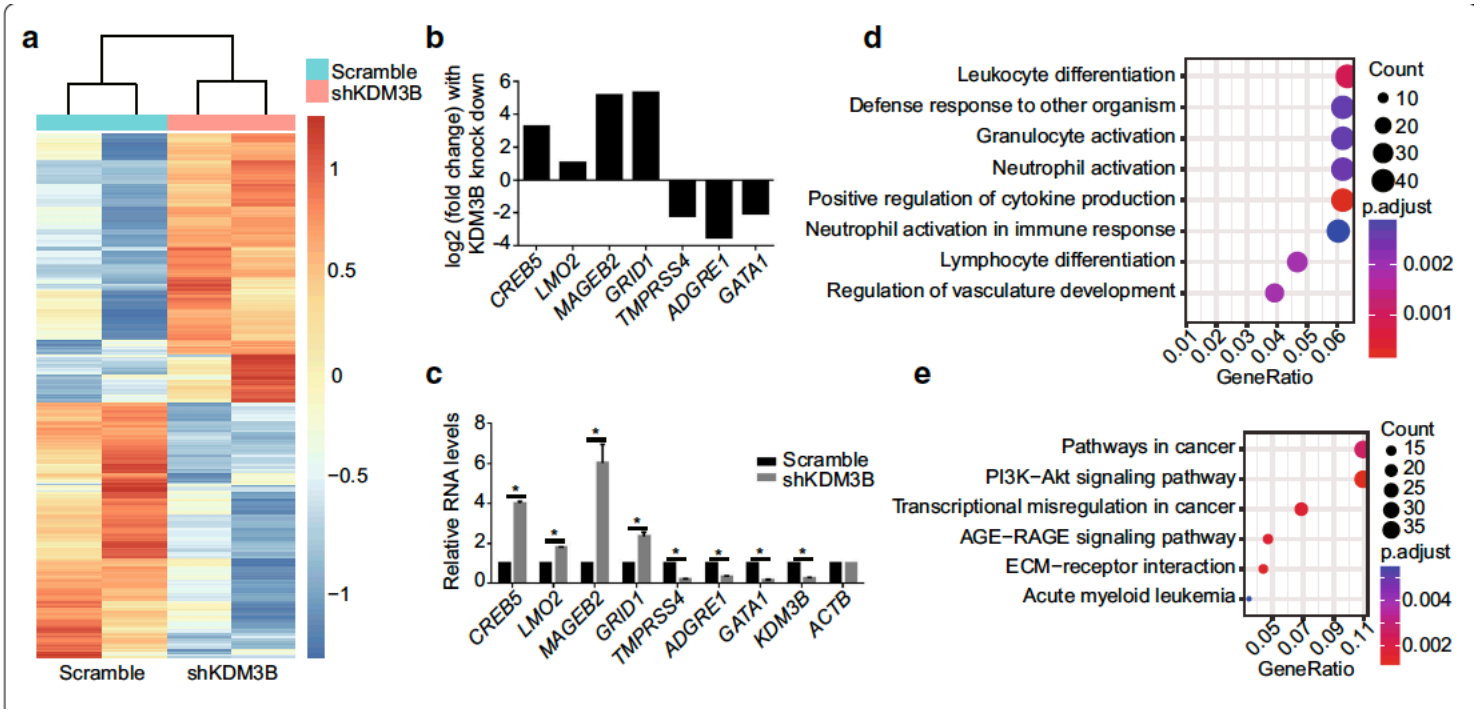
SAMtools

HTSeq-count

DeSeq2

# of DEGs

VERSUS



PAPER #2

RNA sequencing. MDMs (5×10^5/well in 24-well plates) were infected with *A. fumigatus* conidia at a 1:2 effector-to-target ratio, and pooled replicates from three different individuals were collected after 2 and 6 h. Uninfected MDMs were cultured in parallel as controls. Sample processing, sequencing, and analysis were performed at IMGM Laboratories GmbH (Germany). Briefly, the total RNA was isolated using the RNeasy Mini Kit (Qiagen) according to the manufacturer's instructions, including on-column DNase digestion. The total RNA was eluted in 30 µL of RNase-free water. The quality of the total RNA was analyzed with the 2100 Bioanalyzer using RNA 6000 Nano and Pico LabChip kits (Agilent Technologies). Library preparation was performed using the TruSeq® Stranded mRNA HT technology, according to the manufacturer's protocol. All single libraries were pooled into a final sequencing library with an equal DNA amount per sample. The final sequencing library generated by pooling was quantified using the highly sensitive fluorescent dye-based Qubit® dsDNA HS Assay kit (Invitrogen) before sequencing at a final concentration of 1.8 pM and with a 1% PhiX v3 control library spike-in (Illumina) on the NextSeq500 sequencing system (Illumina). For the clustering and sequencing of samples, a high-output single-end 75 cycles (1×75 bp SE) run was performed under the control of the NextSeq Control Software (NCS, Illumina). Quality control was carried out using NCS and Real Time Analysis 2.4.11 softwares applying the *FastQ only* pipeline. Read data were imported into the CLC Genomics Workbench (CLC bio, Qiagen), and reads were mapped against the human reference genome (GRCh37.p13) with subsequent counting and distribution of reads across genes and transcripts. The expression values were then processed to reads per kilobase million (RPKM), a normalized measure of relative abundance of transcripts[68], followed by analysis using the EdgeR Bioconductor package[69] to identify differentially expressed genes with a fold change value ≥2 or ≤ −2 with a false discovery rate (FDR)-corrected p-value < 0.05. Heatmaps were created for the most significantly represented genes of a specific functional class using the Morpheus tool (Broad Institute; https://software.broadinstitute.org/morpheus/). For pathway analysis, the annotated hallmark gene sets from the Molecular Signatures Database (MSigDB)[33] were used and enrichment analysis was performed using the Gene Ontology Biological Processes category in the Gene Ontology Consortium software (http://www.geneontology.org/).

Very vague methods

PAPER #2

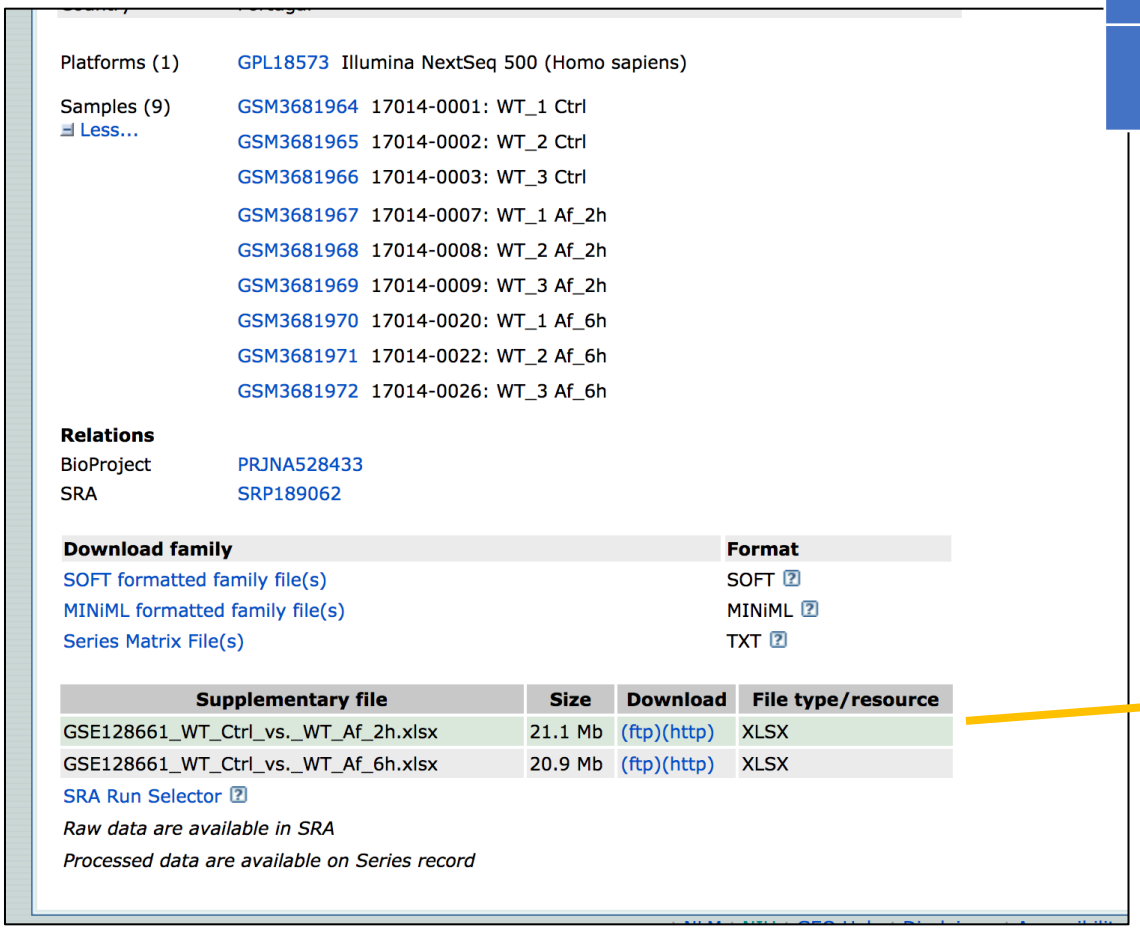
FASTQC

Trimmomatic

HISAT2

SAMtools

HTSeq-count
DeSeq2



Platforms (1) GPL18573 Illumina NextSeq 500 (Homo sapiens)

Samples (9)
Less...

GSM3681964 17014-0001: WT_1 Ctrl
GSM3681965 17014-0002: WT_2 Ctrl
GSM3681966 17014-0003: WT_3 Ctrl
GSM3681967 17014-0007: WT_1 Af_2h
GSM3681968 17014-0008: WT_2 Af_2h
GSM3681969 17014-0009: WT_3 Af_2h
GSM3681970 17014-0020: WT_1 Af_6h
GSM3681971 17014-0022: WT_2 Af_6h
GSM3681972 17014-0026: WT_3 Af_6h

Relations

BioProject PRJNA528433
SRA SRP189062

Download family	Format
SOFT formatted family file(s)	SOFT
MINiML formatted family file(s)	MINiML
Series Matrix File(s)	TXT

Supplementary file	Size	Download	File type/resource
GSE128661_WT_Ctrl_vs._WT_Af_2h.xlsx	21.1 Mb	(ftp)(http)	XLSX
GSE128661_WT_Ctrl_vs._WT_Af_6h.xlsx	20.9 Mb	(ftp)(http)	XLSX

SRA Run Selector

Raw data are available in SRA

Processed data are available on Series record

VERSUS

# of DEGs


Black Diamond Trail

The approach is different

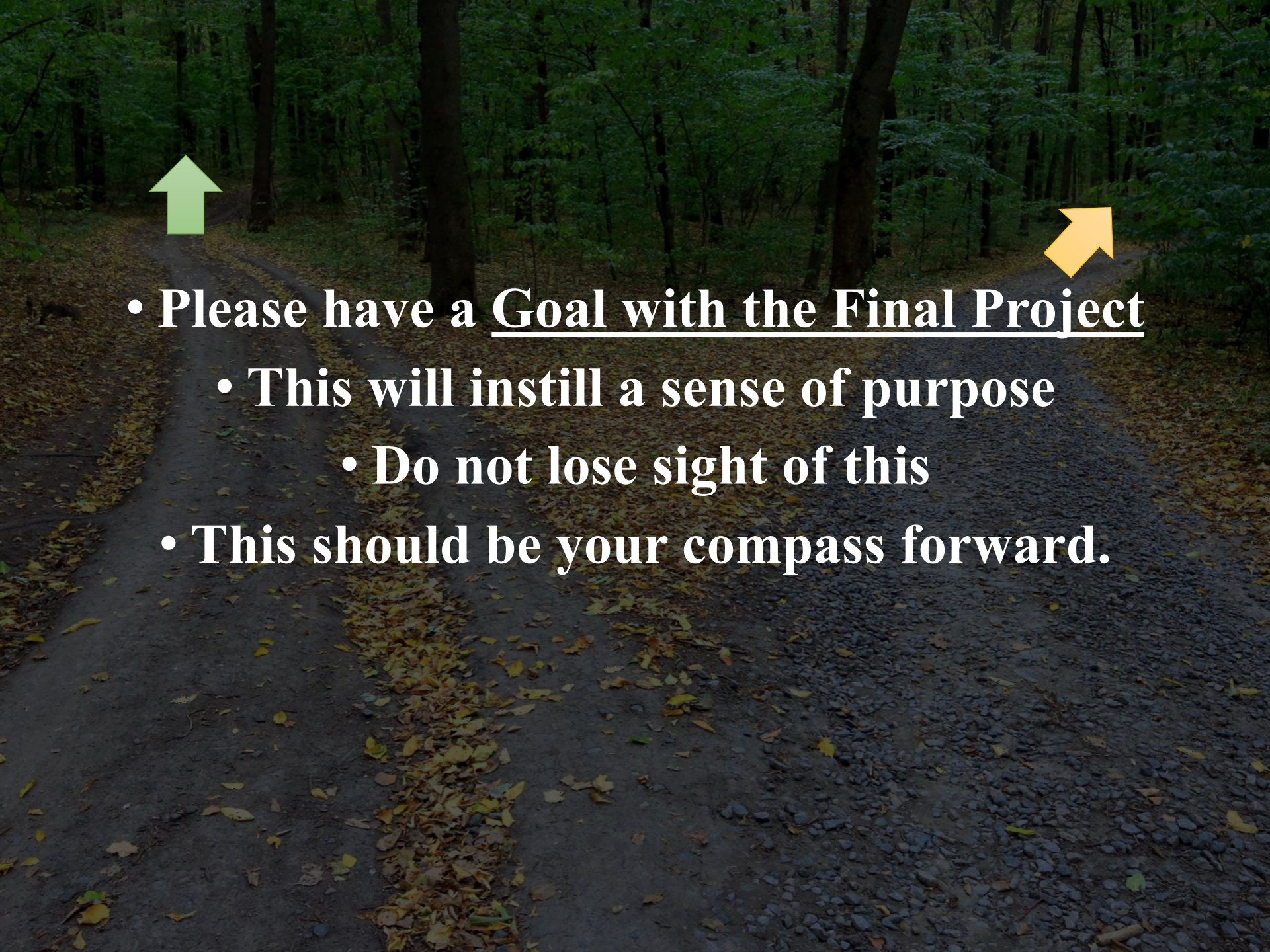
You are going in with a question and using the dataset to answer this question!

"I am interested in studying the difference in metabolic pathway regulation in macrophage versus dendritic cell activation."

"I am interested in understanding how inhibition of GATA1 impacts the ability of naïve B cells to differentiate into plasma cells."

"I am interested in studying the difference in metabolic pathway regulation in macrophage versus dendritic cell activation."

Dataset	# of replicates
Macrophages (control)	3
Dendritic cells (control)	3
Macrophages + LPS	3
Dendritic cells + LPS	3
Macrophages + Zeb1 KO	3
Dendritic cells + Zeb1 KO	3
Macrophages + Zeb1 KO + LPS	3
Dendritic cells + Zeb1 KO + LPS	3


• Please have a Goal with the Final Project
• This will instill a sense of purpose
• Do not lose sight of this
• This should be your compass forward.

Overview for today's lecture

- Update in Course Grading
- Unit 1 recap
- Final project recap
- **Unit 2**
  - **Download dataset from GEO**
  - **Steps in Job submission – SLURM**
    - **Commands overview**
    - **Expectation/Output**
  - Conda Download
  - Data management

Downloading data from GEO:

1. Collect SRR Numbers using RUN SELECTOR
2. Download SRA-ToolKit and set up
3. Run sra_fqdump.sh script
   - Use output from step #1
   - To run this script, we need first need to understand how to submit a job using SLURM

Submission of jobs

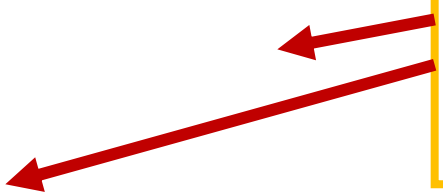
- Submitting a job to an HPC machine (such as Bluemoon or DeepGreen) is done using the batch system.
- The batch system allows users to submit jobs requesting the resources (nodes, processors, memory, GPUs) that they need.
- The jobs are queued and then run as resources become available.
- The batch system is called SLURM

Basic Steps to Job Submission:

1. Log in to VACC
2. Write job script
3. Submit Job
4. Monitor job and wait for it to run
5. Retrieve your output

Basic Steps to Job Submission:

1. Log in to VACC
2. **Write job script**
3. **Submit Job**
4. Monitor job and wait for it to run!
5. Retrieve your output



Requires an understanding of SLURM directives

What are SLURM Directives?

- At the top of the job script will always be several lines that start with **#SBATCH**.
- The SLURM directives provide the job setup information used by Slurm.
- This information is then followed by the commands to be executed in the script.

Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```

Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```

shebang = used to tell the LINUX OS which interpreter to use

Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```

partition = default is bluemoon if not specified

Other partitions are available.
This is important to know as some jobs will take longer to run!

Other Partitions:

Partition	Intended Use	Max Runtime
bluemoon	General computing – default partition	30 hours
short	General computing with short runtime	3 hours
week	General computing with longer runtime	7 days
bigmem	Large memory requirements computing	30 hours
bigmemwk	Large memory requirements with longer runtime	7 days

You can check partition usage using the following command:

```
sinfo -p partition_name
sinfo -p bluemoon
```

Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```

Node: A "node" is a server in the cluster. Each node has is configured with a certain number of cores (CPUs).

Task: A "task" is a process sent to a core. By default, 1 core is assigned per 1 task

Recommend that you begin with 1 node and 2 processes

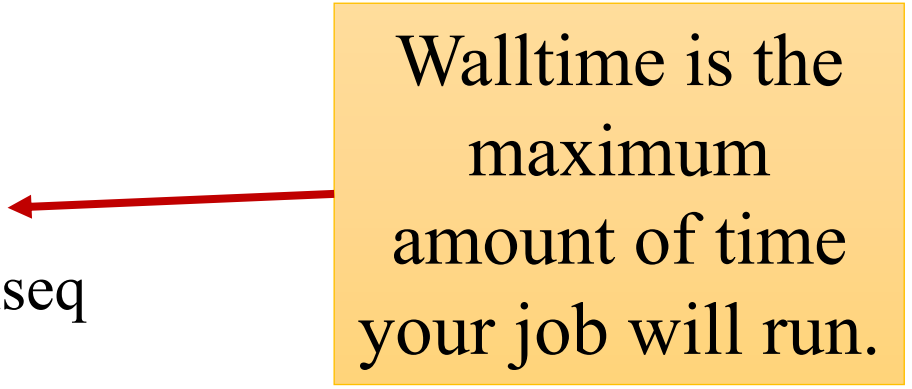
Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```

If your job requires more than 1G of memory – need to specify this

Common SLURM Directives

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=RNAseq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out
```



Downloading data from GEO:

1. Collect SRR Numbers using RUN SELECTOR
2. Download SRA-ToolKit and set up

   Note: Once this is done, you do not need to do this again!
3. Run sra_fqdump.sh script using output from step #1

Scripts used to download SRR files from GEO:

Location on VACC:

```
cp -r /gpfs1/cl/mmg232/
course_materials/download_from_SRA .
```

Look inside sra_fqdump.sh

The first script is a loop, which goes through your list of SRR's, and calls a second script at each iteration, passing it an SRR number in the list.

```
nano sra_fqdump.sh
```

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=fastq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out

#while there are lines in the list of SRRs file
while read p
do
#call the bash script that does the fastq dump, passing it the SRR number next $
sbatch inner_script.sh $p
done <list_of_SRRs.txt
```

Make unique-add userid or some other identifier

inner_script.sh

The script that is called inside the loop (inner_script.sh) is the one that takes the given SRR number and runs fastq-dump on it:

```
nano inner_script.sh
```

```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=fastq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out

#for single end reads only
fastq-dump --gzip $1
```

To run script use sbatch command:

```
sbatch sra_fqdump.sh
```

When you submit your job, Slurm will respond with the job ID. For example, where the job ID Slurm assigns is “123456,” Slurm will respond:

```
Submitted batch job 123456
```

Other important commands:

Command	What It Does
sbatch	Submits a job, e.g., sbatch myjob
scontrol show job	Detailed information about a particular job, e.g., scontrol show job 123456
squeue	Checks status of all jobs in scheduling queue
squeue -u	Checks status of all jobs belonging to the named user, e.g., squeue -u usr1234
squeue –start -j	Estimates earliest start time of a particular job, e.g., squeue –start -j 123456
squeue –start -u	Estimates earliest start time of all jobs belonging to the named user, e.g., squeue –start -u usr1234
scancel -u	Deletes/cancels all jobs belonging to the named user, e.g., scancel -u usr1234
scancel	Deletes/cancels a particular job, e.g., scancel 123456

Recording the output of programs

- Three data streams exist for all Linux programs
  - STDIN (Standard Input - a way to send data into the program)
  - STDOUT (Standard Output - a way to send expected data out of the program)
  - STDERR (Standard Error - a way to send errors or warnings out of the program)

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

These will be redirected to .out files

```
[pdrodrig@vacc-user1 GSE164713_Tcf1]$ ls
fastq_6008331.out  fastq_6426350.out     SRR13422709.fastq.gz
fastq_6366635.out  fastq_6426351.out     SRR13422710.fastq.gz
fastq_6366636.out  inner_script.sh       SRR13422711.fastq.gz
fastq_6366637.out  list_of_SRRs.txt      SRR13422712.fastq.gz
fastq_6366638.out  list.txt              SRR13422713.fastq.gz
fastq_6366639.out  sra_download.sh       SRR13423162.fastq.gz
fastq_6366640.out  sra_fqdump.sh         SRR13423163.fastq.gz
fastq_6366641.out  SRR13416485.fastq.gz  SRR13423164.fastq.gz
fastq_6426340.out  SRR13416486.fastq.gz  SRR13423165.fastq.gz
fastq_6426341.out  SRR13422702.fastq.gz  SRR13423166.fastq.gz
fastq_6426342.out  SRR13422703.fastq.gz  SRR13423167.fastq.gz
fastq_6426343.out  SRR13422704.fastq.gz  SRR17379677.fastq.gz
fastq_6426344.out  SRR13422705.fastq.gz  SRR17379678.fastq.gz
fastq_6426347.out  SRR13422706.fastq.gz  SRR17379679.fastq.gz
fastq_6426348.out  SRR13422707.fastq.gz  SRR17379680.fastq.gz
fastq_6426349.out  SRR13422708.fastq.gz
```

Any errors will be recorded in these .out files

```
hg38_bam/917_EC90_2hr_R1B_S8_unmapped_hg38_sorted.bam
Traceback (most recent call last):
  File "/users/p/d/pdrodrig/anaconda3/bin/htseq-count", line 5, in <module>
    HTSeq.scripts.count.main()
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    args = _parse_sanitize_cmdline_arguments()
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    _check_sam_files(args.samfilenames)
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    with pysam.AlignmentFile(sam_filename, "r") as sf:
  File "pysam/libcalignmentfile.pyx", line 742, in pysam.libcalignmentfile.AlignmentFile._$
  File "pysam/libcalignmentfile.pyx", line 947, in pysam.libcalignmentfile.AlignmentFile._$
ValueError: file does not contain alignment data
hg38_bam/917_EC90_2hr_R2A_S22_unmapped_hg38_sorted.bam
Traceback (most recent call last):
  File "/users/p/d/pdrodrig/anaconda3/bin/htseq-count", line 5, in <module>
    HTSeq.scripts.count.main()
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    args = _parse_sanitize_cmdline_arguments()
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    _check_sam_files(args.samfilenames)
  File "/users/p/d/pdrodrig/anaconda3/lib/python3.8/site-packages/HTSeq/scripts/count.py",$
    with pysam.AlignmentFile(sam_filename, "r") as sf:
```

Overview for today's lecture

- Update in Course Grading
- Unit 1 recap
- Final project recap
- **Unit 2**
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - **Conda Download**
  - Data management

Conda download

Install Conda

The first thing we need to do is download miniconda:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

And then run the bash command to install it:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

A few interactive things will happen during this portion:

- You will be asked to review the license agreement, hit return/enter to view it and the space bar to advance. At the end you will need to type + enter yes to accept.
- It will ask if the installation location is ok. Since we are in our home directory, hit enter to accept. If you are not in your home directory, hit ctrl+c, navigate there, and start over with the bash command.
- Finally, it will ask if you want to initialize Miniconda3. Say yes to this as well.

Now that it's installed we need to reload the current shell session with:

```
source ~/.bashrc
```

You should now see a (base) at the start of your command prompt, indicating you are in the "base" conda environment.

Install tools in your Base environment

Your base environment is your “home base”, things installed here will always be accessible to you as a user regardless of your location within the server. However, you want to avoid installing everything this way, as it is bound to lead to version or software conflicts. Instead, this can be a good place to install smaller, simple programs that get used very frequently.

One example of this kind of tool is FastQC, a lightweight tool for generating summary reports of sequencing read files. Try googling conda install FastQC. The first page should be the anaconda.org page for the tool, and searching any tool this way is a quick and easy to find the installation “recipe”.

```
conda install -c bioconda fastqc
```

Now try to execute the command (try tab complete too!)

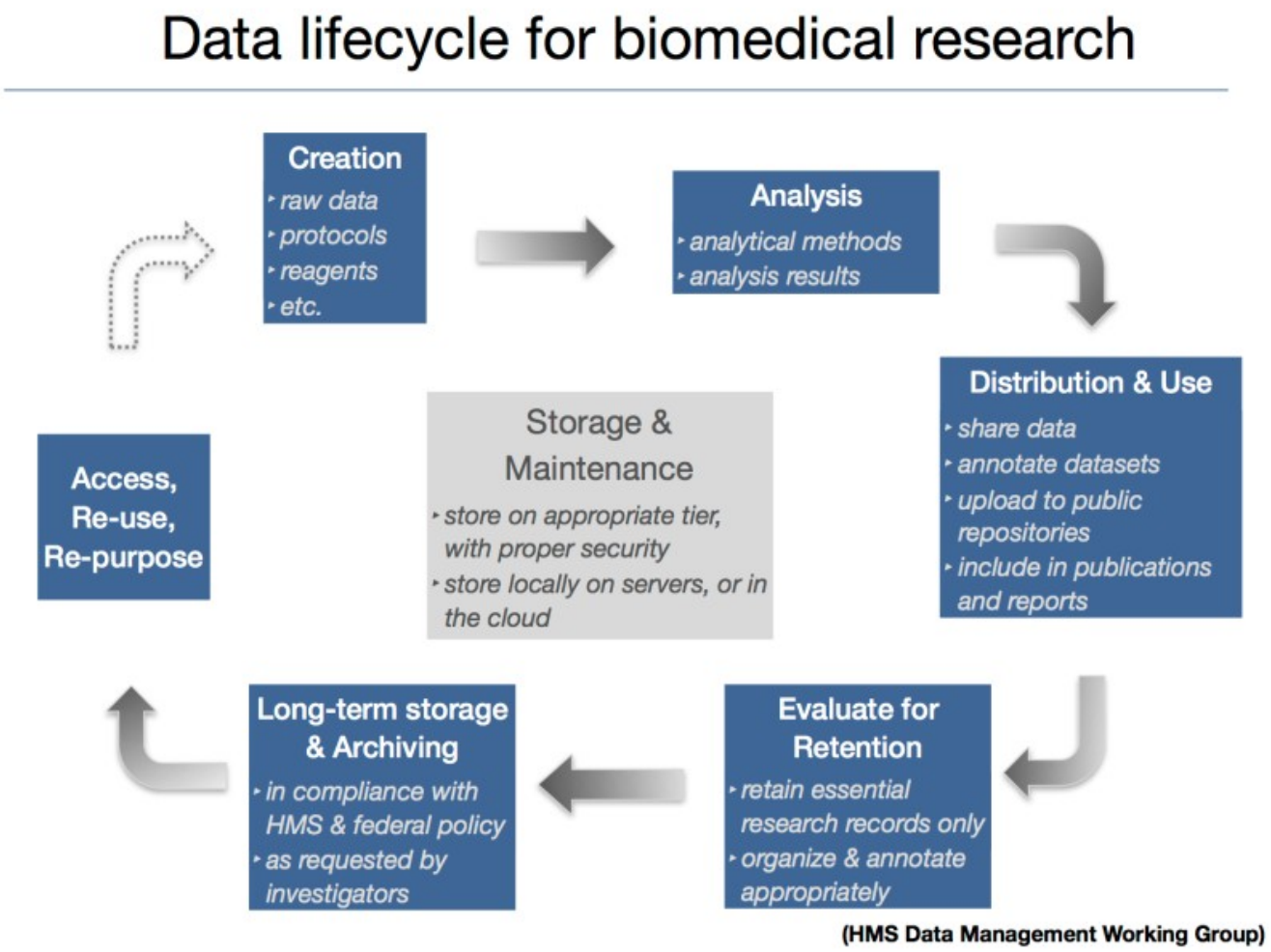
```
fastqc --help
```

The manual page should now show up on your screen if the installation was successful.

Overview for today's lecture

- Update in Course Grading
- Unit 1 recap
- Final project recap
- **Unit 2**
  - Download dataset from GEO
  - Steps in Job submission – SLURM
    - Commands overview
    - Expectation/Output
  - Conda Download
  - **Data management**
    - **Metafile generation**

Data life-cycle

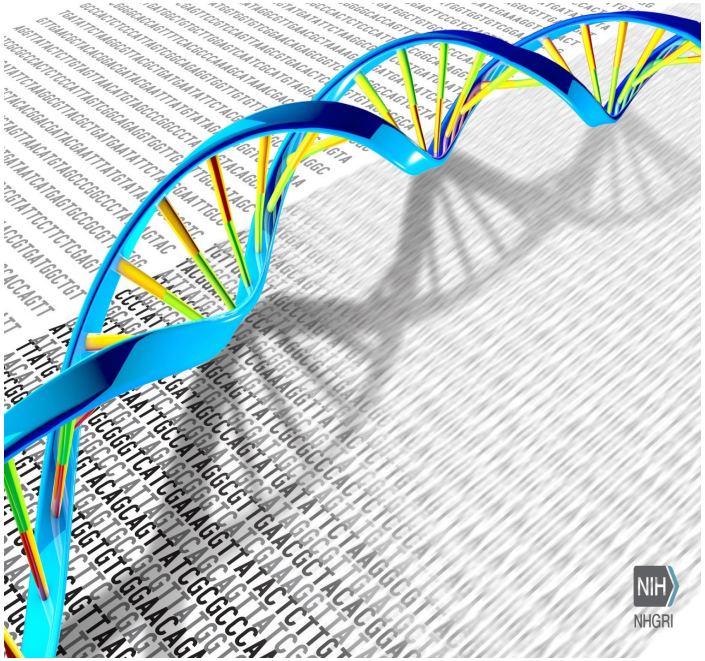


Data management planning

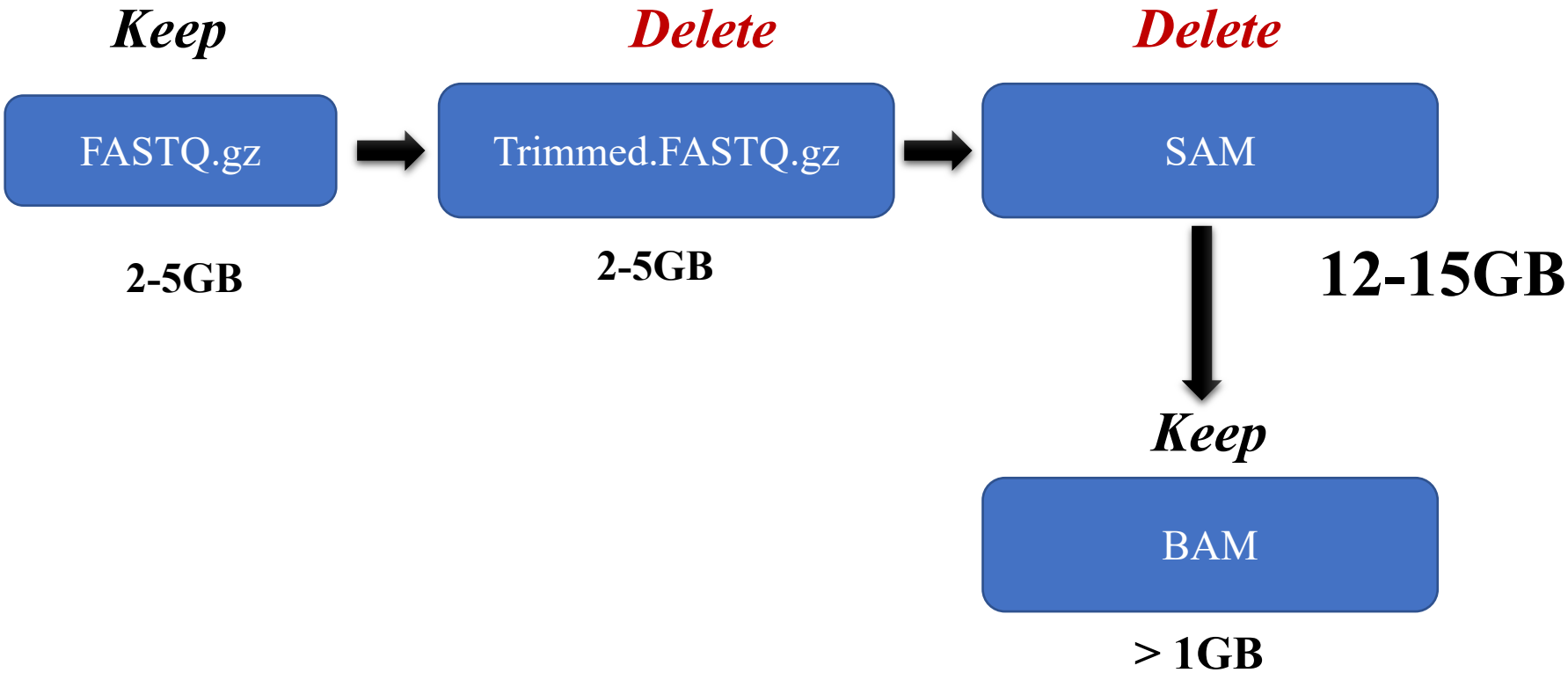
- Consider the **datatype and file sizes**. How much total raw data do you have?
- Consider the **type of analyses** you plan on performing.
- What types of intermediate data files will I generate?
- What is the best way to organize my **directory structure**?

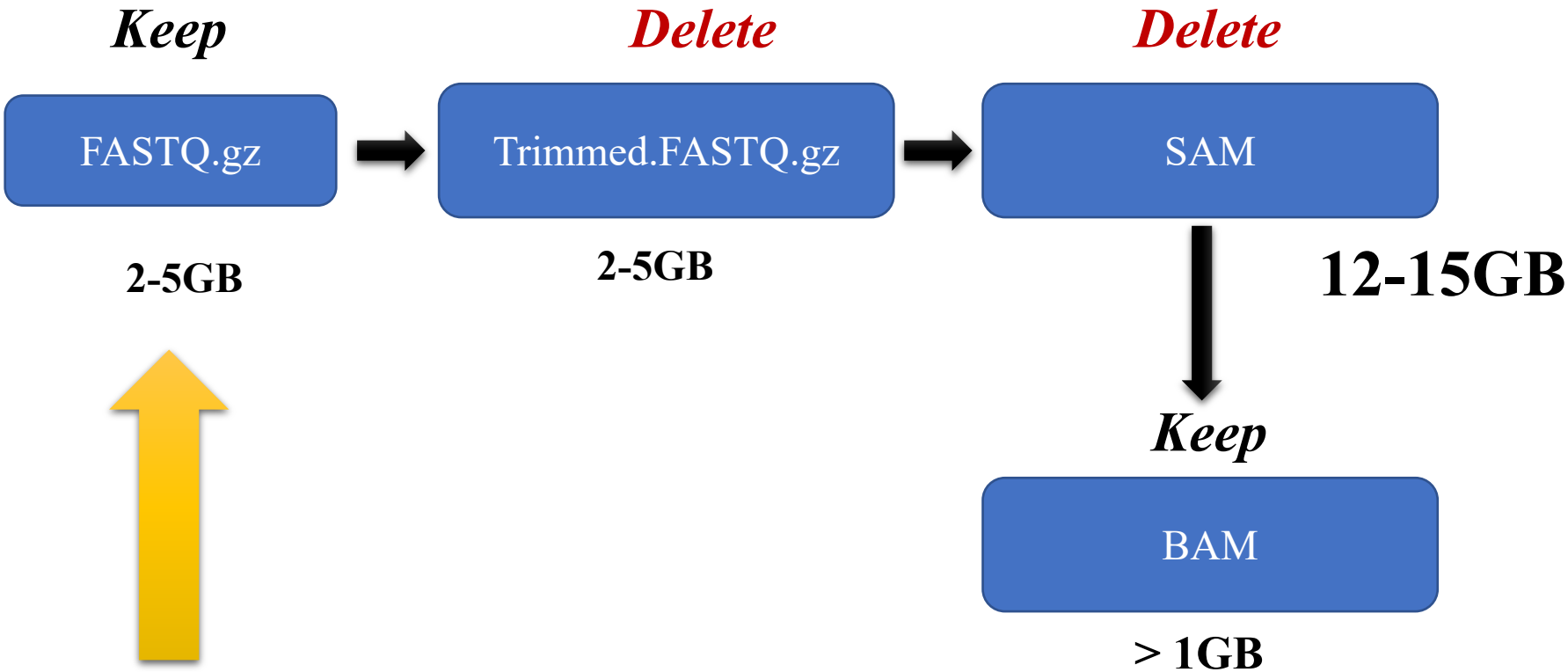
Data types: Raw data

- For NGS experiments, these are the **FASTQ** files you obtain from your sequencing facility or GEO



- This data should never be directly modified (i.e. always keep a copy of this stored somewhere, untouched from its original state)


Keep
Delete
Delete
FASTQ.gz
Trimmed.FASTQ.gz
SAM
2-5GB
2-5GB
12-15GB
Keep
BAM
> 1GB



Files you are collecting using
sratoolkit
Fastq-dump

You will need to collect information about the data you are processing

Metadata

- Metadata are data that provide information about the data you are processing
- Metadata exists to give data context
- In genomics or transcriptomics, metadata describes the sample that the DNA/RNA sequence was obtained from, the organism, the cell line, and library preparation method

Submitting HTS data to GEO

https://www.ncbi.nlm.nih.gov/geo/info/seq.html

Homework assignment #1:
HTS_GEO_submission_template.xlsx