

**Navigation & help**

<code>pwd</code>	print working directory
<code>ls -lh</code>	list files with sizes
<code>ls -lah</code>	list including hidden files
<code>cd &lt;dir&gt;</code>	change directory
<code>cd ~</code>	home directory
<code>cd ..</code>	parent directory
<code>man &lt;command&gt;</code>	manual/help page (q to quit)
<code>history   grep &lt;word&gt;</code>	search your past commands

**File & directory operations**

<code>mkdir &lt;dir&gt;</code>	make a new directory
<code>cp SRC DEST</code>	copy file
<code>cp -r DIR/ DEST/</code>	copy directory recursively
<code>mv SRC DEST</code>	move or rename
<code>rm &lt;file&gt;</code>	remove file (use <code>rm -i</code> for safety)
<code>touch &lt;file&gt;</code>	create an empty file
<code>ln -s TARGET LINK</code>	make a symbolic link

**Viewing files**

<code>less -S &lt;file&gt;</code>	view file (no wrapping)
<code>head -n 20 &lt;file&gt;</code>	first 20 lines
<code>tail -n 20 &lt;file&gt;</code>	last 20 lines
<code>zcat file.fastq.gz   head -n 8</code>	peek into a compressed FASTQ
<code>cat &lt;file&gt;</code>	print file to terminal

**Search, count, sort**

<code>grep -n "PATTERN" file</code>	find pattern with line numbers
<code>grep -i / -v / -o / -n</code>	ignore case / invert match / only match / line numbers
<code>zgrep "AGATCGGAAG" *.fastq.gz</code>	search inside compressed FASTQs
<code>wc -l file</code>	count lines
<code>sort -n file</code>	numeric sort
<code>uniq -c</code>	unique lines with counts

**Pipes & redirects**

<code>cmd1   cmd2</code>	pipe output of one command into another
<code>cmd &gt; out.txt</code>	write to file (overwrite)
<code>cmd &gt;&gt; out.txt</code>	append to file
<code>cmd 2&gt; err.txt</code>	write errors to file
<code>cmd &amp;&gt; all.txt</code>	capture both stdout and stderr
<code>cmd &lt; in.txt</code>	use file as input

**Compression & archives**

<code>gzip file / gunzip file.gz</code>	compress or decompress
<code>tar -czvf out.tar.gz DIR/</code>	archive and compress directory
<code>tar -xf out.tar.gz</code>	extract archive

---

### ***tmux essentials***

<code>tmux new -s analysis</code>	Start a new named session
<code>Ctrl-b d</code>	Detach from session
<code>tmux ls</code>	List sessions
<code>tmux attach -t analysis</code>	Reattach to session
<code>tmux kill-session -t name</code>	Kill a session
<code>Ctrl-b %</code>	Vertical split
<code>Ctrl-b "</code>	Horizontal split
<code>Ctrl-b arrow-key</code>	Move between panes
<code>Ctrl-b c</code>	Create new window
<code>Ctrl-b n / Ctrl-b p</code>	Next / previous window
<code>Ctrl-b ,</code>	Rename window

### ***Slurm essentials***

<code>squeue -u \$USER</code>	Show your jobs
<code>sinfo</code>	Show partitions/nodes
<code>sbatch job.slurm</code>	Submit batch job
<code>scancel &lt;jobid&gt;</code>	Cancel job
<code>srun -t 30:00 -a &lt;AC&gt; -p cpu -c 2 script.sh</code>	Run one-off job with resources
<code>sinteractive -t 2:00:00 -a &lt;AC&gt; -p cpu -c 4</code>	Start interactive job with resources
<code>seff &lt;jobid&gt;</code>	Show efficiency of a completed job

### ***Slurm sample job***

<code>#!/bin/bash</code>	<i># Use Bash as the job's shell</i>
<code>#SBATCH --job-name=fastqc</code>	<i># Job name shown by squeue/sacct</i>
<code>#SBATCH --account=ACCOUNT</code>	<i># Slurm account/allocation to charge</i>
<code>#SBATCH --partition=cpu</code>	<i># Queue/partition to submit to</i>
<code>#SBATCH --nodes=1</code>	<i># Number of nodes requested</i>
<code>#SBATCH --cpus-per-task=4</code>	<i># CPU threads for *each* task (here 1 task)</i>
<code>#SBATCH --output=fastqc.%j.out</code>	<i># Stdout log file (%j = JobID)</i>
<code>#SBATCH --error=fastqc.%j.err</code>	<i># Stderr log file</i>
 <code>module --force purge</code>	 <i># Start with a clean module environment</i>
<code>module load biocontainers</code>	<i># Load site module providing containers</i>
<code>module load fastqc</code>	<i># Load FastQC (module or container wrapper)</i>
 <code>for file in *fastq.gz; do</code>	 <i># Run FastQC using Slurm-provided thread count</i>
<code>fastqc -t \${SLURM_CPUS_ON_NODE} \${file};</code>	
<code>done</code>	