Introduction to Using Linux

Dr. Kitty Lo and Dr. Dario Strbenac

THE UNIVERSITY OF SYDNEY

What is UNIX

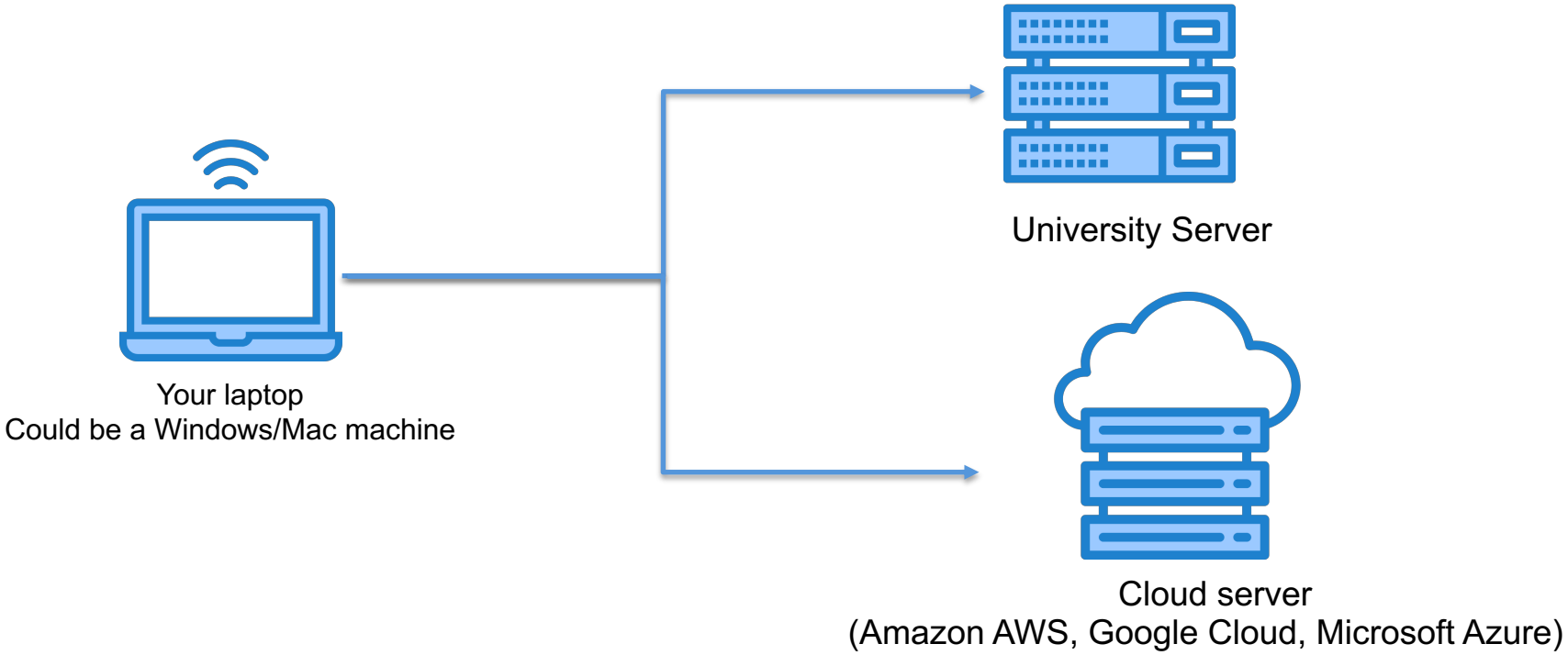
- UNIX is a family of multi-user multi-tasking operating system, orginially developed at the Bell labs in the 1970s.
- The majority of tools and software are *command-line* as opposed to *event-driven* (e.g. graphical menus and mouse clicks). Command line tools are easier and faster to create than GUIs, so most bioinformatics software is typically command-line software.
- UNIX is a common operating system on servers
- Many different variants – including Linux, even MacOS is a derivative of UNIX

Linux: What and Why

- Linux describes one of a number of free-to-obtain operating systems (e.g. Debian, Ubuntu, CentOS) first invented by a Finnish university student in 1991 and is widely used by high-performance computers.
- The official logo of Linux is the penguin, chosen in 1996 because the inventor of Linux was bitten by one† in National Zoo & Aquarium in Canberra while on holiday.

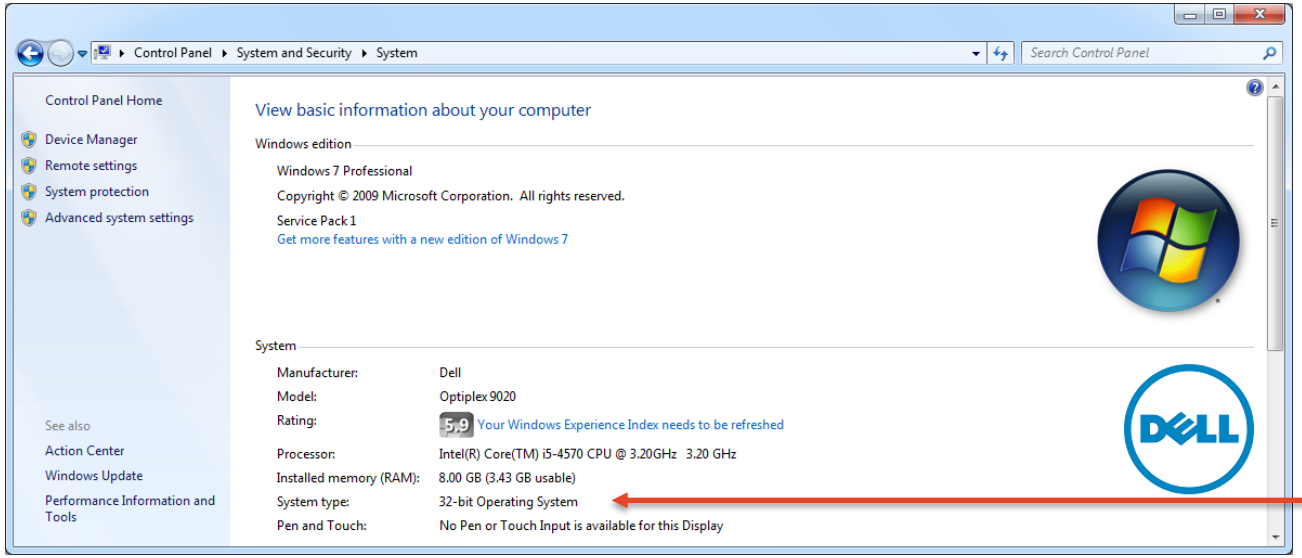
†https://en.wikipedia.org/wiki/History_of_Linux

Connecting to the server



Connecting Using SSH on Windows

1. Determine if you are running a 32-bit or 64-bit computer.



32-bit system

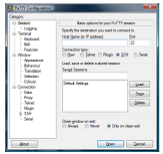
Connecting Using SSH on Windows

2. Download a SSH client. PuTTY will be used. Browse https://putty.org

https://putty.org

Search

Most Visited | School | Usyd



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY here.

Download PuTTY: latest release (0.73)

Home | FAQ | Feedback | Licence | Updates | Mirrors | Keys | Links | Team
Download: **Stable** · Snapshot | Docs | Changes | Wishlist

This page contains download links for the latest released version of PuTTY. Currently this is 0.73, released on 2019-09-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a permanent link to the 0.73 release.

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the development snapshots, to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

MSI ('Windows Installer')

32-bit:	putty-0.73-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.73-installer.msi	(or by FTP)	(signature)

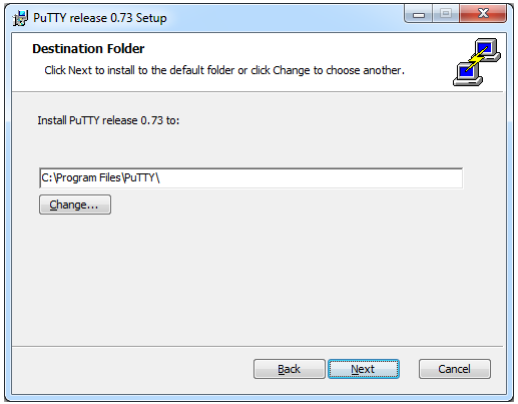
Unix source archive

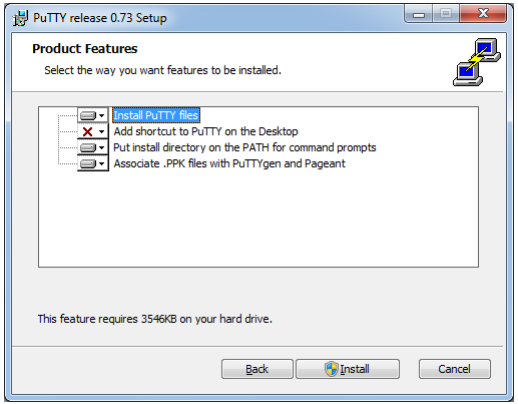
.tar.gz:	putty-0.73.tar.gz	(or by FTP)	(signature)

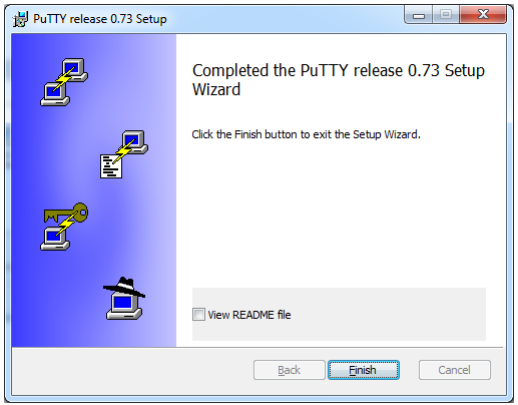
Click one

Connecting Using SSH on Windows

2. Install the software. Leaving the options at their defaults is fine.

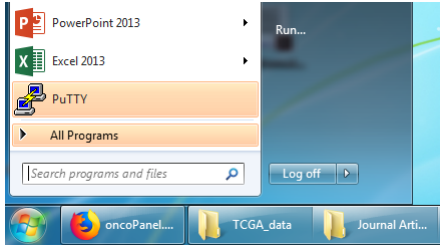


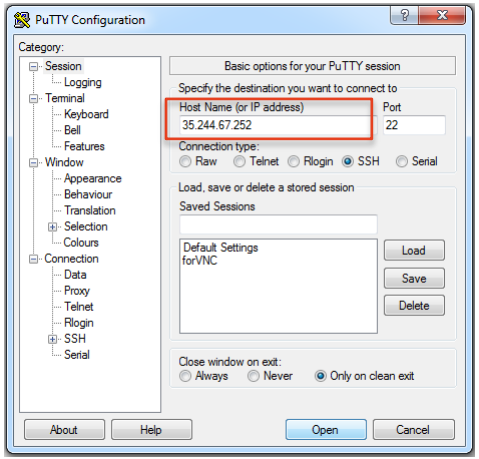




Connecting Using SSH on Windows

3. Open PuTTY. Connect to the Linux server.



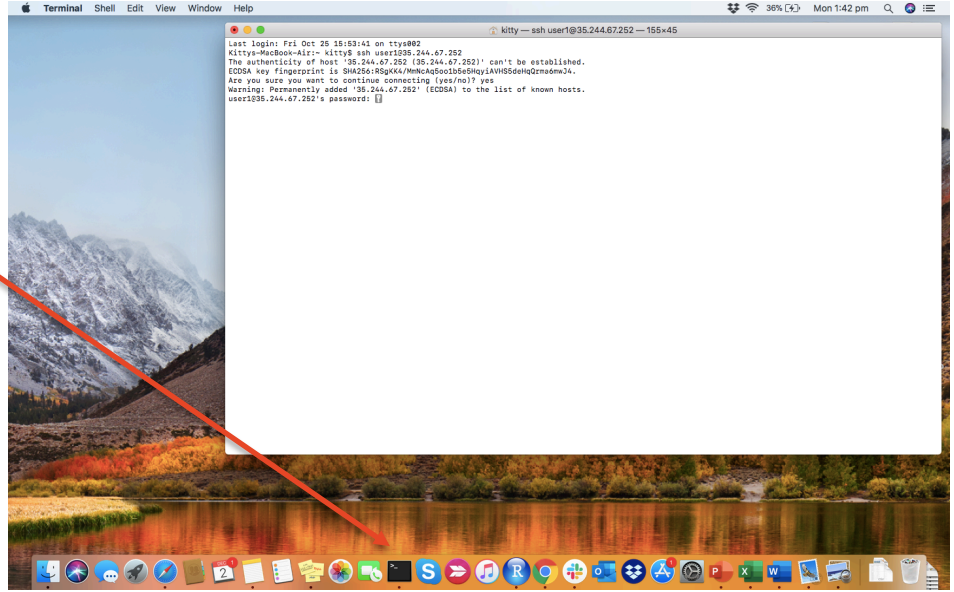


Connecting Using SSH on MacOS

1. Open the Terminal application

2. In the terminal, type in:

```
> ssh username@ipaddress
```



Where Am I?

- Once successfully logged in, you'll be in your *home directory*.

```
$ pwd
/home/trainer
```

- `pwd` is an abbreviation for **p**resent **w**orking **d**irectory.
- The first / is called the *root directory*. home is a directory in the root directory. trainer is a directory in the home directory.

Files in Directories

- The biological data files are in the directory `/home/data/GM12878/`

```
$ ls

$ ls /home/data/GM12878/
alignments.bam  geneCounts.tsv  reads.fastq.gz
```

- `ls` is short for *list*.
- All of the files and directories are shown in either the current working directory or the one you specify.

Navigating Directories

- You might want to change directory to another one to avoid typing the full path to an input file for each command.

```
$ cd /home/data/GM12878/
$ ls
alignments.bam  geneCounts.tsv  reads.fastq.gz
```

- Since you're in the same directory as the files, you don't need to specify the path of the directory to `ls`.

File Characteristics

- Often, it's important to know how big a file is or when it was last modified. The output of ls can be made to have more details for each entry.

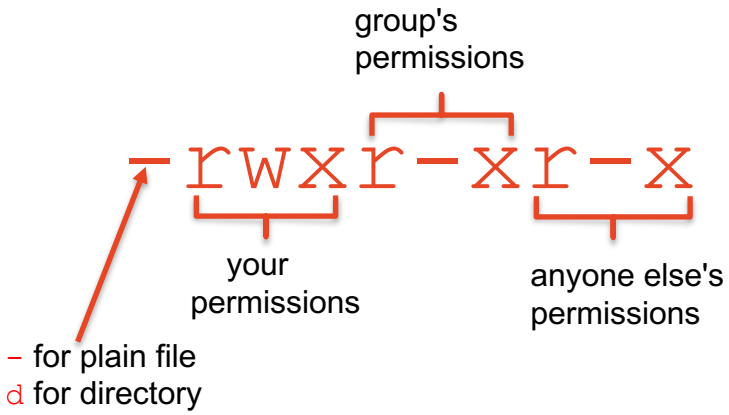
```
$ ls -l -h /home/data/GM12878/
total 30G
-rwxr-xr-x 1 trainer trainer  23G Nov 20 03:26 alignments.bam
-rwxr-xr-x 1 trainer trainer 448K Nov 18 02:05 geneCounts.tsv
-rwxr-xr-x 1 trainer trainer 7.4G Nov 20 03:28 reads.fastq.gz
```

User Group File Size Last Modified Name

- The `-l` option makes `ls` output a long listing with more details.
- All of the files and directories are shown in either the current working directory or one you specify.

File Type and Permission

- 10 characters, each representing something different.



r read the contents of the item

w overwrite the contents of the item

x execute the file or change into the directory using cd

File Deletion

- You don't have permission to delete any of the workshop files. Try to do it using the `rm` command.

```
$ rm alignments.bam
rm: remove write-protected regular file 'alignments.bam'? y
rm: cannot remove 'alignments.bam ': Permission denied
```

- After typing the first couple of letters of the file name, press 

to have the computer complete the rest of it for you.

Software Options

- Each Linux command has a manual page which describes what it does and how you can customise it.

```
$ man ls
```

```
NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX
       nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file status information); with  -l:  show  ctime
              and sort by name; otherwise: sort by ctime, newest first

       -C     list entries by columns

 Manual page ls(1) line 3 (press h for help or q to quit)
```

Resource Usage

- See how much RAM and CPU are being currently used by `top`.

```
top - 02:01:36 up  1:15,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 188 total,   1 running, 187 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
GiB Mem :   58.986 total,    0.402 free,    0.304 used,   58.280 buff/cache
GiB Swap:    0.000 total,    0.000 free,    0.000 used.   58.007 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   56864   6580   5272 S   0.0  0.0   0:01.39 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S   0.0  0.0   0:00.08 kworker/u32:0
    7 root      20   0       0      0      0 S   0.0  0.0   0:00.07 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/0
   10 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 lru-add-drain
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/0
   12 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/1
   14 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/1
   15 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/1
   16 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/1
   18 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/2
   20 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/2
   21 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/2
   22 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/2
   23 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/2:0
   24 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/2:0H
   25 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/3
   26 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/3
   27 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 migration/3
   28 root      20   0       0      0      0 S   0.0  0.0   0:00.00 ksoftirqd/3
   30 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/3:0H
   31 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/4
   32 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/4
```

%CPU: How much of one processor is used. Can be more than 100% if you use multiple processors.

%RAM: How much of the computer's memory is used.

If typing becomes delayed or you can't even connect to the server, it's probable that someone is using too much resources. Identify them from the user column.

Example RNA-seq Data

- FASTQ file of cDNA (reverse-transcribed RNA) reads of cell line GM12878.

```
TCGCAACATCTCGA
ACTGACCCTCATGC
AATAGCTATCAAGG
```

- BAM (**B**inary sequence **A**lignment **M**ap) file of alignments of reads to a genome.

```
TCGCAACATCTCGA          experimental data
...TGTCGGAACATCTCGAAG...  chromosome 17
                          reference genome
```

- TSV (**T**ab **S**eparated **V**alues) file of counts of reads to each known gene.

Gene Symbol	GM12878 Reads
TP53	6378
CD274	2494
GAPDH	486158

FASTQ File

- The type of file you would get for almost any DNA or RNA sequencing data from the facility that sequences your biological samples.

```
@DFDF8JF1:304:C24EYACXX:8:1101:1208:1936 1:N:0:ACACAC 1
GGGGAGGAAGAGGAGGAGGAAGAAGAAGGTGATGGTGAGGAAGAGGATGGAGATGAAGATGAGGAAGCTGAG 2
+ 3
@@CFDFFADHDHGGGH=FHDCH;FGIHGI*?DHGG0?DDA?;DDFF9CFG3=C@GG>AE>?EHE@6;;>ACD 4
```

1: Unique identifier for each record
2: The nucleic acid sequence the laboratory instrument determined.
3: No purpose, always is +
4. Corresponding quality scores for the DNA sequence in Line 2, so same number of characters as Line 2. Each letter or symbol corresponds to a quality score.

Compressed FASTQ File

- You'll typically see `.gz` on the end of FASTQ file names. These files have been compressed to reduce the amount of disk space they use. You can't immediately view them using a text editor.

- You also don't want to decompress the files because they'll use lots of disk space.

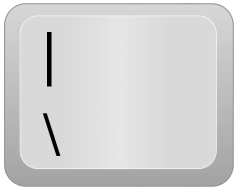
  GM12878RNAseqReads.fastq.gz: 7.4 GB

  Converted into a plain file: 24 GB !

Solution: Decompress the file in memory and pass the stream of data to the program that uses it immediately using a *pipe*.

Viewing the Beginning or End of A File

- `head` and `tail` commands show you the first and last ten lines of a file, respectively. `-n <integer>` changes the number of lines displayed.
- `zcat` will decompress a file ending in `.gz`
- `|` is the pipe which passes output of one command into another command. Hold down Shift key before pressing it.




```
$ zcat reads.fastq.gz | head -n 4
@DFDF8JF1:304:C24EYACXX:8:1101:1208:1936 1:N:0:ACACAC
GGGGAGGAAGAGGAGGAGGAAGAAGAAGGTGATGGTGAGGAAGAGGATGGAGATGAAGATGAGGAAGCTGAGTCAGCTACGGGCAAGCGGGCAGCTGAAGA
+
@@CFDFFADHDHGGGH=FHDCH;FGIHGI*?DHGG0?DDA?;DDFF9CFG3=C@GG>AE>?EHE@6;;>ACD6>A;AC>>?@B=?B<1?>B><B@9@@>>3
```

Pipes Are Efficient

Q: Apart from disk space, why don't we decompress and then use `head` separately?

A: The pipe stops the first command when the second command has enough data to finish whatever it does.

```
$ zcat reads.fastq.gz | head -n 4
```

Time: 0.005 seconds

```
$ zcat reads.fastq.gz > RNA.fastq
$ head -n 4 RNA.fastq
```

Time: 3 minutes 21 seconds

> symbol can be thought of as an arrowhead causing the command to output the results to a file instead of the interactive command console.

How Many Reads?

- Each read is 4 lines of a FASTQ file.

```
$ zcat reads.fastq.gz | wc -l
390192208
```

Takes a long time.

Tip: If a command is very similar to a previous command press  to go back to it and modify it as necessary. Reduces typing.

- `wc` command is short for *word count*. Despite its name, it can count the number of characters (`-c`), words (`-w`), or lines (`-l`) in a file.

There are 97548052 RNA-seq reads

Inspecting BAM Files

- Compressed and binary files, so need special software to view them.
- `samtools` is developed by bioinformaticians and is not a standard part of Linux. It has already been installed for your convenience.
- Can do lots of different tasks with BAM files.

SYNOPSIS

```
samtools view -bt ref_list.txt -o aln.bam aln.sam.gz
samtools sort -T /tmp/aln.sorted -o aln.sorted.bam aln.bam
samtools index aln.sorted.bam
samtools idxstats aln.sorted.bam
samtools flagstat aln.sorted.bam
samtools stats aln.sorted.bam
samtools bedcov aln.sorted.bam
samtools depth aln.sorted.bam
samtools view aln.sorted.bam chr2:20,100,000-20,200,000
samtools merge out.bam in1.bam in2.bam in3.bam
samtools tview aln.sorted.bam ref.fasta
samtools split merged.bam
samtools quickcheck in1.bam in2.cram
samtools fixmate in.namesorted.sam out.bam
samtools mpileup -C50 -f ref.fasta -r chr3:1,000-2,000 in1.bam in2.bam
```

Inspecting BAM Files

Let's look at the first alignment in the BAM file.

```
$ samtools view alignments.bam | head -n 1
```

Unique read ID

Flag

Chromosome

Start Position

101 matches

```
DFDF8JF1:304:C24EYACXX:8:1314:15477:30610       355     chr1    11212   3       101M
=       11353   242
GTGCTGTGCCAGGGCGCCCCCTGCTGGCGACTAGGGCAACTGCAGGGCTCTCTTGCTTAGAGTGGTGGCCAGCGCCCCCTGCTGGCGCCGGGGCACTGCAG
CCCFFFFFHHGHHIDHIJJIJJADHIJIIJJIIJJJGIJJDHCHHHFFFFDEEDEEDDDDDD@@BCD@A@CBBDDDDDB<BACDDBBBB>BDD>D@CCCDC
NH:i:2  HI:i:2  AS:i:200        NM:i:0  MD:Z:101
```

Number of **M**ismatches (of bases to the reference genome).

Number of **H**its (locations in the genome it matches to)

Nice graphical images are produced with software such as IGB or IGV.

SAM Flags

Summary of alignment properties. Complicated to interpret. Use web application Explain SAM Flags https://broadinstitute.github.io/picard/explain-flags.html

To decode a given SAM flag value, just enter the number in the field below. The encoded properties will be listed under Summary below, to the right.

SAM Flag: 355 Explain

	Decimal
	1
	2
	32
	64
+	256
	355

Summary:

- read paired (0x1)
- read mapped in proper pair (0x2)
- mate reverse strand (0x20)
- first in pair (0x40)
- not primary alignment (0x100)

hexadecimal numeral system (base 16)

Gene Counts

- The large alignments files are typically converted into gene-level counts files using some software (e.g. HTSeq-count, RSEM) for statistical analysis
- T.S.V. is an abbreviation for **T**ab **S**eparated **V**alues. Each column of data is separated by a tab character.

```
$ head -n 5 geneCounts.tsv
Gene Symbol     Count
TSPAN6  0
TNMD    0
DPM1    3559
SCYL3   1596
```

Searching Files

- A powerful command is `grep`
- First parameter is the search term, second parameter is the file to search.

Search for gene ZNF678.

```
$ grep ZNF678 geneCounts.tsv
ZNF678  411
```

Gene ZNF678 has 411 RNA-seq reads within its boundaries.

- Data-analysis-oriented programming languages such as Python and R have much more functionality for working with tables and numbers than the Linux command line does.