Informatics on High-throughput Sequencing Data

(Summer Course 2020)

Day 4

Agenda

- Unix-based systems.
- Why Linux!
- Let's start!
- Linux Commands for:
  - Files & Directories.
  - System.
  - Process Management.
  - Networking.
  - Compression.
  - Searching.
- Piping output.
- Wildcard character.
- Redirecting output.
- Stream Editor (Sed).
- Linux tools for text files processing.
- Shell Scripting

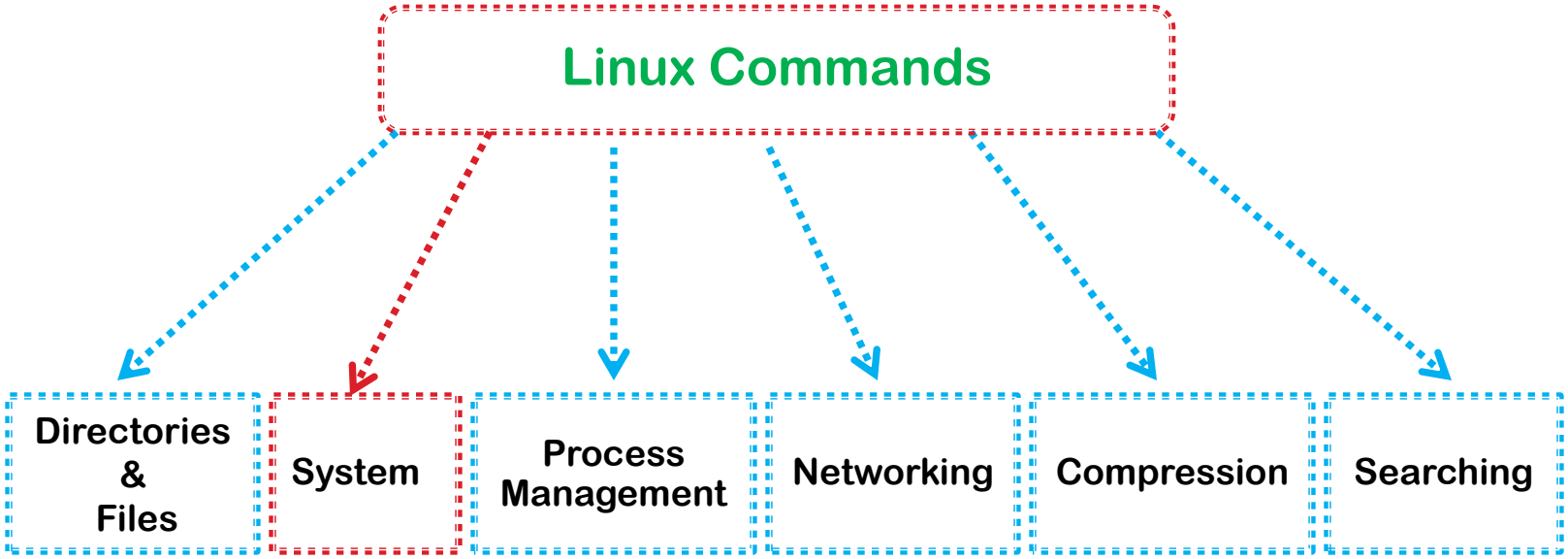
Directories and Files (Fun!)

```
alias name="command"
```

Create an alias for a command

```
alias sara="ls -l"
```

Getting Started !!



System (Users & Status)

`exit`	Exit the shell (logout).
`date`	Show the current date and time.
`whoami`	Who you are logged in as.

System (Resources)

`df -h`	Show disk usage in a human readable format.
`du -hs *`	Show directory space usage.
`free`	Show memory and swap usage.
`whereis app`	Show possible locations of app.
`which app`	Show which app will be run by default.

System (Resources)

```
toshiba@ubuntu:~/Documents/agri_training$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/loop0       14G  9.9G  2.6G  80% /
udev            3.9G  4.0K  3.9G   1% /dev
tmpfs           790M  980K  789M   1% /run
none            5.0M     0  5.0M   0% /run/lock
none            3.9G  160K  3.9G   1% /run/shm
/dev/sda4       198G   16G  182G   8% /host
/dev/sda3       196G   56G  141G  29% /media/349EE2649EE21E5C
```

df -h

```
toshiba@ubuntu:~/Documents/agri_training$ free
             total       used       free     shared    buffers     cached
Mem:       8082036    6134344    1947692          0    2302156    2860308
-/+ buffers/cache:     971880    7110156
Swap:       262140          0     262140
```

Terminal

free

System (Resources)

whereis app

```
toshiba@ubuntu:~/Documents/agri_training$ whereis perl
perl: /usr/bin/perl /etc/perl /usr/lib/perl /usr/bin/X11/perl /usr/share/perl
toshiba@ubuntu:~/Documents/agri_training$ whereis git
git: /usr/bin/git /usr/bin/X11/git /usr/share/man/man1/git.1.gz
```

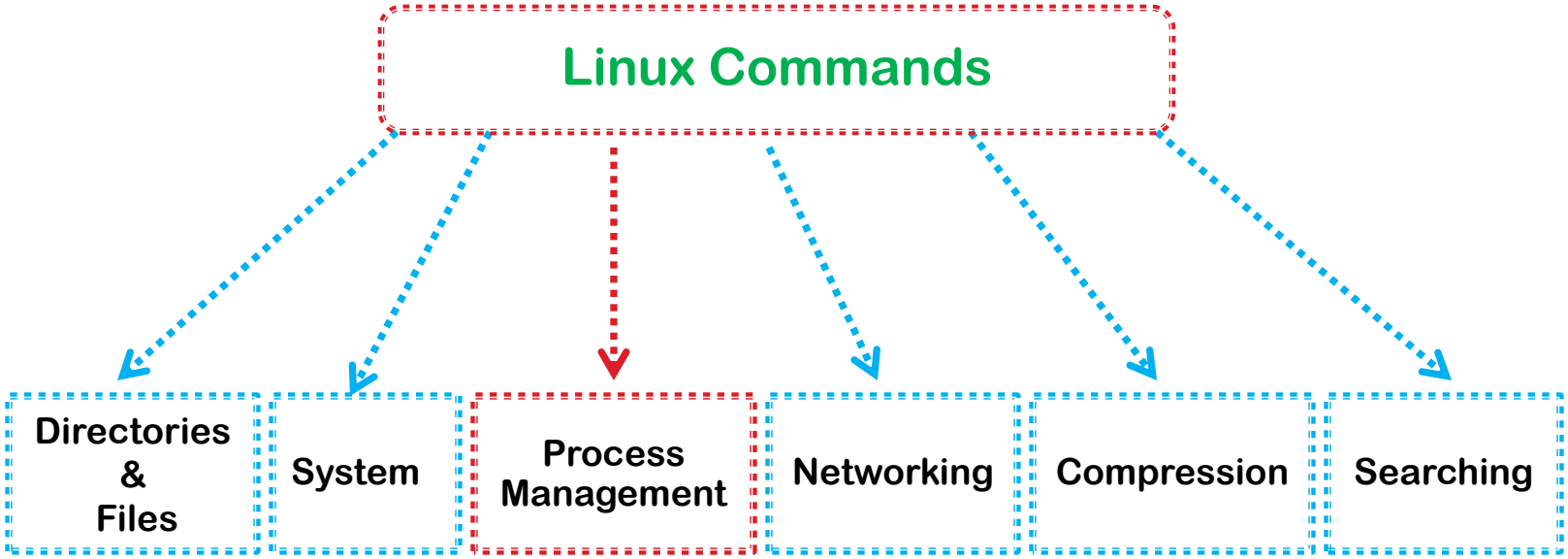
which app

```
toshiba@ubuntu:~/Documents/agri_training$ which perl
/usr/bin/perl
toshiba@ubuntu:~/Documents/agri_training$ which git
/usr/bin/git
```

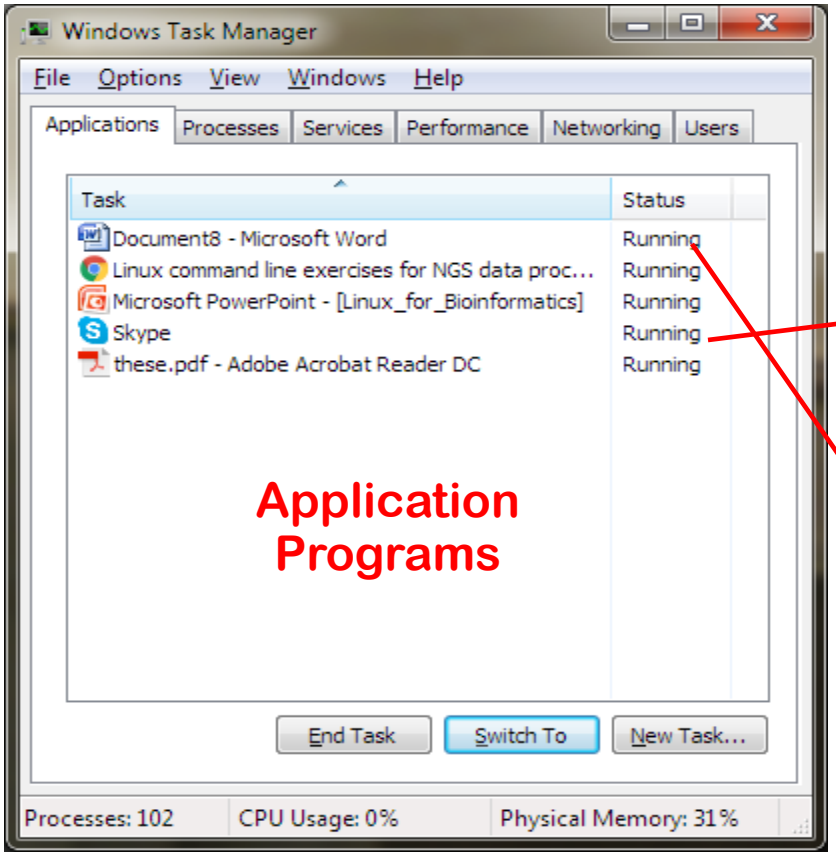
System (help!)

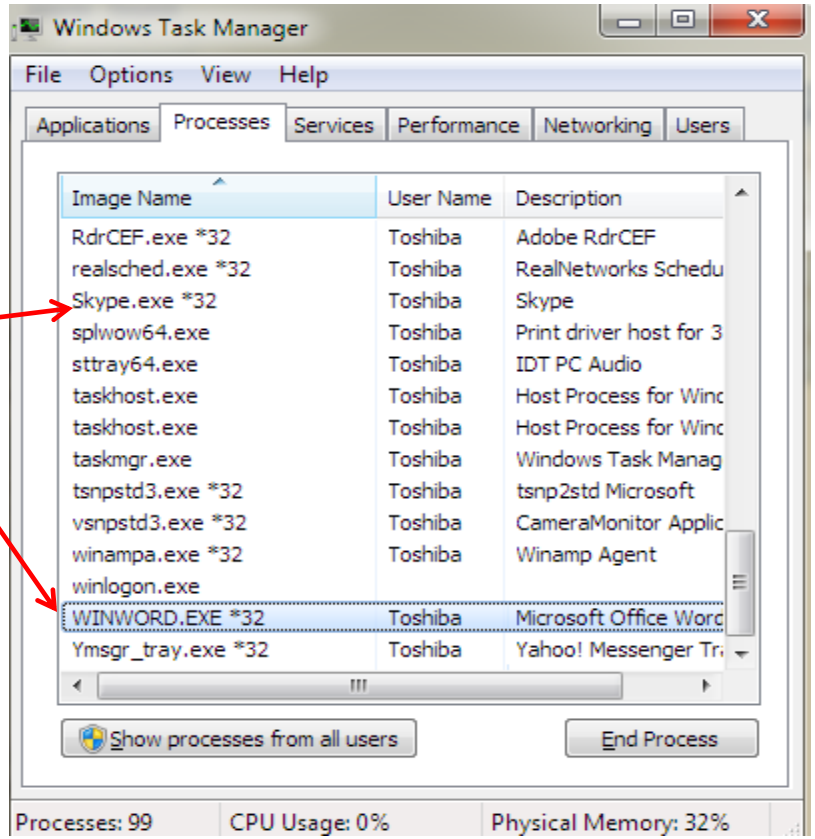
`man command`	Show the manual for command.
`command --help`	Show what options are available for command.
`history`	List all recently used commands.

Getting Started !!



Process Management



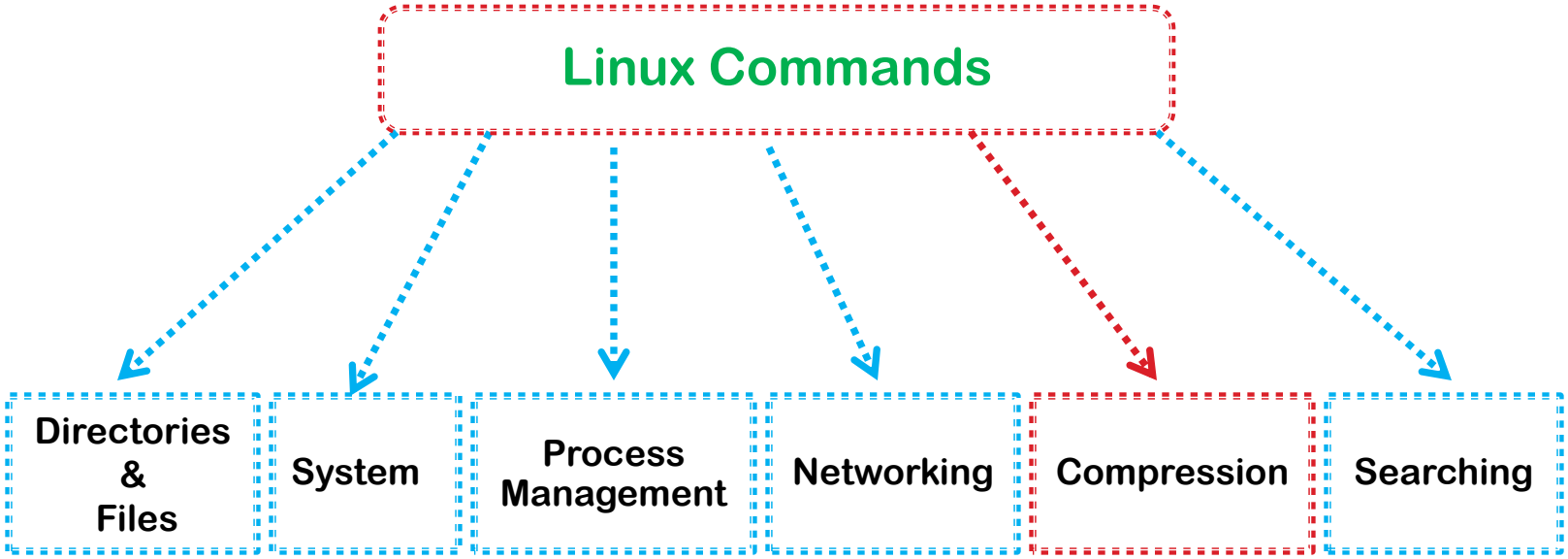


Process

Process Management

`ps`	Display your currently active processes.
`top`	Display all running processes.
`kill pid`	Kill process id pid.

Getting Started !!



Archiving and Compression

- **Archiving** is the process of collecting and storing a group of files and directories into one file. The **tar** utility performs this action.

- **Compression** is the act of shrinking the size of a file, which is quite useful in sending large files over the internet.

Archiving and Compression

- To save disk space, we can compress large files if we do not intend to use them for a while.
- Files downloaded from the web are typically compressed and sometimes need to be uncompressed before processing can take place.

Archiving and Compression

Common compressed formats and compression/decompression tools.

Format (extension)	Tool to Compress	Tool to Uncompress	Function
.gz	gzip	gzip -d	compress a single file
.bz2	bzip2	bunzip2	compress a single file
.zip	zip	unzip	make compressed archive (single file) of a directory structure; same as on Windows
.tar	tar cvf	tar xvf	make an archive (single file) of a directory structure
.tar.gz (.tgz)	tar czf	tar xzf	make a compressed archive (single file) of a directory structure

Archiving

```
tar cvf file.tar file1 file2 my_dir
```

Create a tar named file.tar containing files.

create

verbose: tar program prints comments and progress messages.

```
tar xvf file.tar
```

Extract the files from file.tar.

extract

specify the name of the archive you want to extract.

Archiving

```
tar tvf file.tar
```

list

List contents of the tar archive file.tar.

Archiving &Compression

```
tar czvf file.tgz file1 file2 my_dir
```

create a tar with gzip compression.

```
tar xzvf file.tgz
```

Extract a tar using gzip.

tar file compressed with gzip

```
tar tzvf file.tgz
```

List contents of the tar archive `file.tar.gz`.

Compression

`gzip file`	Compresses file and renames it to file.gz.
`gzip -d file.gz`	Decompresses file.gz back to file.

Compression

```
zip -r file.zip file
```

Create a zip file named file.zip.

```
zip -r file.zip file1 my_dir
```

Create a zip file named file.zip containing the file file1 and the directory my_dir with all its content.

```
unzip file.zip
```

Extract the file.zip.

Genomic Data
(A read in FASTQ)

```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
```

Note: A typical sequence run with 400,000,000 reads will generate a file containing 1.6 billion lines of data.

https://www.youtube.com/playlist?list=PL2mpR0RYFQsBiCWVJSvVAO3OJ2t7DzoHA

Base qualities

Bases and qualities line up:

```
AGCTCTGGTGACCCATGGGCAGCTGCTAGGGA
|||||||||||||||||||||||||||||||
HHHHHHHHHHHHHHHGCGC5FEFFFGHHHHHH
```

Base quality is ASCII-encoded version of $Q = -10 \log_{10} p$

Usual ASCII encoding is "Phred+33":

take Q, rounded to integer, add 33, convert to character

https://www.youtube.com/playlist?list=PL2mpR0RYFQsBiCWVJSvVAO3OJ2t7DzoHA

ASCII

Example: Q=36.7

Phred+33= 37+33=70 = F

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	/	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	fl	255	ˇ

Base qualities

Bases and qualities line up:

```
AGCTCTGGTGACCCATGGGCAGCTGCTAGGGA
|||||||||||||||||||||||||||||||
HHHHHHHHHHHHHHHGCGC5FEFFFGHHHHHH
```

40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n

72-33=39

https://www.youtube.com/playlist?list=PL2mpR0RYFQsBiCWVJSvVAO3OJ2t7DzoHA

Genomic Data
(A read in FASTQ)

PHRED Score	Probability of Incorrect Base Call	Accuracy of Base Call
0	1 in 1	0%
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%

- 10 corresponds to 10% error (1/10),
- 20 corresponds to 1% error (1/100),
- 30 corresponds to 0.1% error (1/1,000) and
- 40 corresponds to one error every 10,000 measurements (1/10,000) that is an error rate of 0.01%.

https://www.youtube.com/playlist?list=PL2mpR0RYFQsBiCWVJSvVAO3OJ2t7DzoHA

Genomic Data
(A data in FASTA)

```
>VIT_201s0011g03530.1
AATTAAGCATAAATACTCACTCTTACCCCCTTATTTTCTTATCTCTCATCACTTTTGGTGCGAAG
GACCATGAGAACAAGCTGCAATGGGTGTAGGGTTCTTCGCAAGGCATGCAGCCAAGACTGCATCA
>VIT_201s0011g03540.1
CAGGTAGCGTGAAGTTAAACCCTAGCGCTTTAGACAAACAGCTGTAGTCACCGCCCACAAACACC
AGCCTCTGAGACACCACCTCAAACCTTTCCACTTAAATACACATCCCTCACACCCTTTTCAATTC
>VIT_201s0011g03550.1
CATGCAAAGCTGAACGCGATGCTGTGATTGGTGGTAAGTGGTAGTTGAGTAAATTTGACAGTGAA
GCCGAAATGGTAAAAGACTAAGGCTAGAAGTAGAATACCACTGTTCTTCTCATCACGTGGGCCCA
```

References

- Paul Stothard, An Introduction to Linux for bioinformatics , 2016.
- Robert Bukowski, Linux for Biologists- Part 1.
- Steve Pederson, Introduction To Linux/Ubuntu & Sell Scripting, 2014.
- https://bioinformatics.uconn.edu/unix-basics/#
- https://learn.gencore.bio.nyu.edu/ngs-file-formats/quality-scores/
- https://coding4medicine.com/Members/pages/home/
- https://open.oregonstate.education/computationalbiology/chapter/patterns-regular-expressions/
- https://bioinformaticsworkbook.org/Appendix/Unix/unix-basics-3grep.html#gsc.tab=0
- https://datacarpentry.org/shell-genomics/04-redirection/

Thanks!

// | ?