Introduction to UNIX command-line

UPLB

January 2016

Noe Fernandez

Sol Genomics Network

Class Content

- Terminal file system navigation
- Wildcards, shortcuts and special characters
- File permissions
- Compression UNIX commands
- Networking UNIX commands
- Basic NGS file formats
- Text files manipulation commands
- Command-line pipelines

Why use command-line?

- Most software for biological data analysis is used through UNIX command-line terminal
- Most of the servers for biological data analysis use Linux as operative system
- Data analysis on calculation servers are much faster since we can use more CPUs and RAM than in a PC (e.g.: Boyce server has 64 cores and 1TB RAM)
- Large NGS data files can not be opened or loaded in most of GUI-based software and web sites
- Compression commands are useful, since NGS large data files usually are stored and shared as compressed files

What is a virtual machine?

File system navigation

- File system commands

Download the cheat sheet from:

ftp://ftp.solgenomics.net/bioinfo_class/UPLB/sgn_unix_commands_cheat_sheet_2015.pdf

http://www.slideshare.net/solgenomics/sgn-unix-commandline-cheat-sheet-2015

UNIX Command-Line Cheat Sheet

BTI-SGN Bioinformatics Course 2014

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory
locate *file_name*	searches a file
man *command*	shows commands manual
top	shows process activity
df -h	shows disk space info

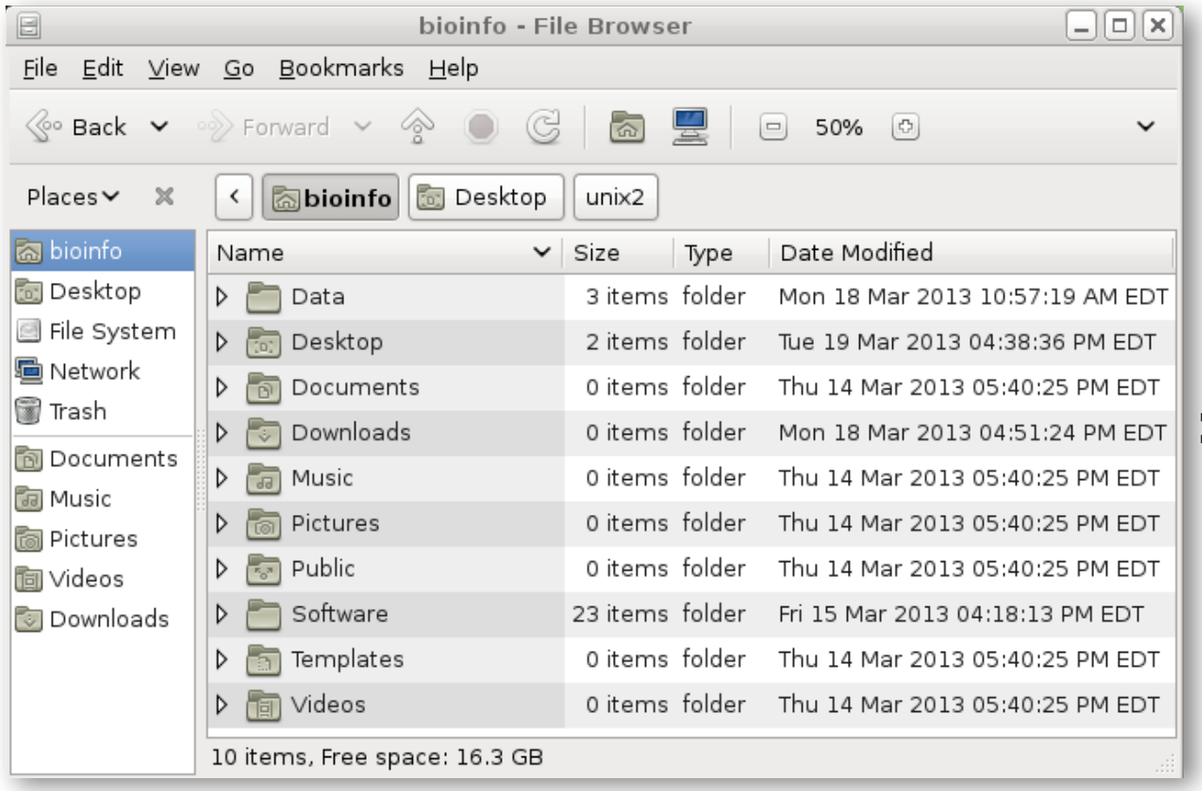
Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2 > file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern' file*	Prints lines matching a pattern
grep -c '*pattern' file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*
sort -u *file*	sorts and return unique lines
uniq -c *file*	filters adjacent repeated lines
wc *file*	counts lines, words and bytes
paste *file1 file2*	concatenates the lines of input files
paste -d ","	concatenates the lines of input files by commas
sed	transforms text

Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

Networking Commands	
wget *URL*	download a file from an URL
ssh *user@server*	connects to a server
scp	copy files between computers
apt-get install	installs applications in linux

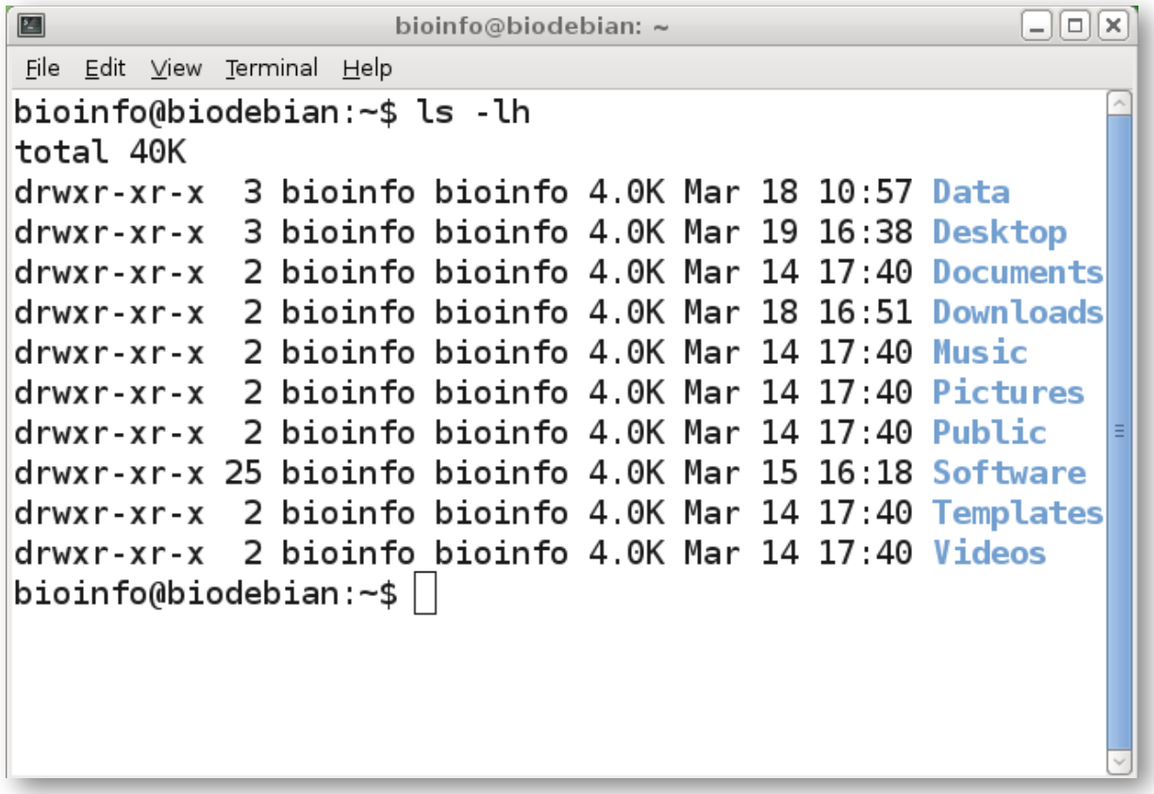
File system navigation

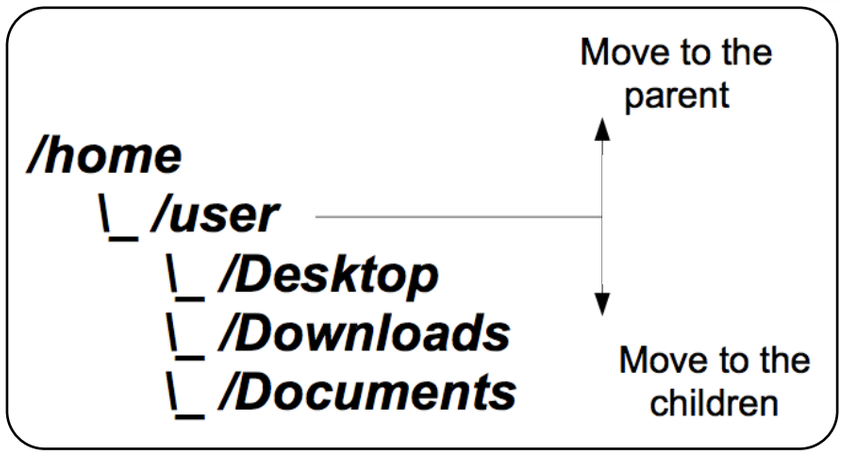
File Browser



=

Terminal





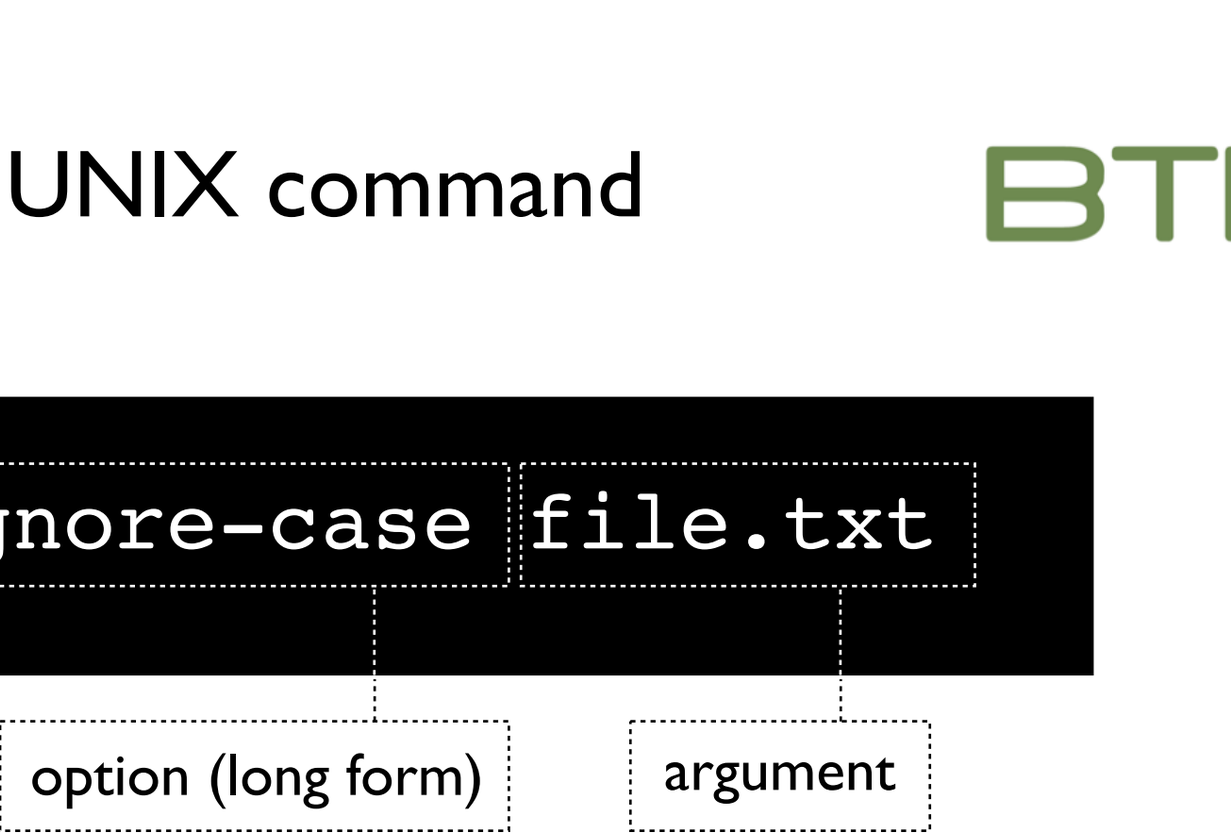
Anatomy of a UNIX command

```
grep -c -A 3 --ignore-case file.txt
```

- `grep` — command
- `-c` — Simple option flag (short form)
- `-A 3` — option with argument
- `--ignore-case` — option (long form)
- `file.txt` — argument

print grep manual

```
man grep
```



ls, cd and **pwd** to navigate the file system

- where am I? pwd
- how to change current directory cd
- what files and directories are in my current directory? ls

return current work directory

```
pwd
```

ls lists directories and files

list directories and files in current directory

list all directories and files, including hidden files

```
ls
ls -a
ls -l -h
ls -l -h -t
ls -lhS
```

list in long format

human readable

size sorted

time sorted

```
Noes-MacBook-Pro:~ Noe$ ls -lht
total 0
drwx------+ 29 Noe  staff   986B May 31 11:24 Desktop
drwx------@  8 Noe  staff   272B May 31 08:26 Dropbox
drwx------+ 54 Noe  staff   1.8K May 30 16:01 Downloads
drwx------+  8 Noe  staff   272B May 28 21:06 Pictures
drwxr-xr-x  18 Noe  staff   612B May 17 11:12 BTI
drwxr-xr-x   5 Noe  staff   170B May  8 11:44 programs
drwx------+ 15 Noe  staff   510B Apr 10 08:33 Documents
drwxr-xr-x   6 Noe  staff   204B Mar 18 09:22 VirtualBox VMs
drwxr-xr-x   8 Noe  staff   272B Mar 14 19:26 py_devel
drwx------@ 51 Noe  staff   1.7K Mar 11 15:08 Library
drwxr-xr-x   6 Noe  staff   204B Nov 28  2012 PTA
drwx------+  4 Noe  staff   136B Sep 26  2012 Music
drwx------+  3 Noe  staff   102B Sep 26  2012 Movies
drwxr-xr-x+  4 Noe  staff   136B Sep 26  2012 Public
Noes-MacBook-Pro:~ Noe$
```

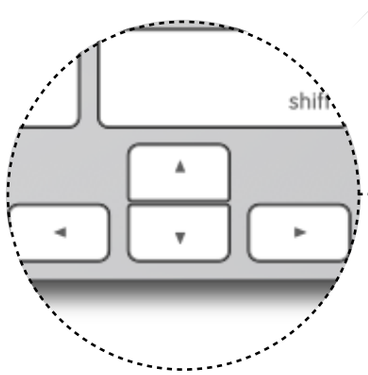
Wildcards, history and some shortcuts

```
ls *txt
```

list all txt files in current directory

```
ls D*
```

list files starting with D
e.g.: Desktop, Data, Downloads ...



Use up and down arrows to navigate the command history

`ctrl-c`	stop process
`ctrl-a`	go to begin of line
`ctrl-e`	go to end of line
`ctrl-r`	search in command history

Escaping special characters

! @ $ ^ & * ~ ? . | / [ ] < > \ ` " ;# ( )

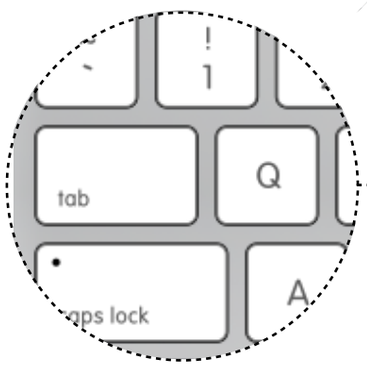
```
ls my_folder
```

list a folder

```
ls my\ folder
```

list a folder containing a space

Tip: file names in lower case and with underscores instead of spaces



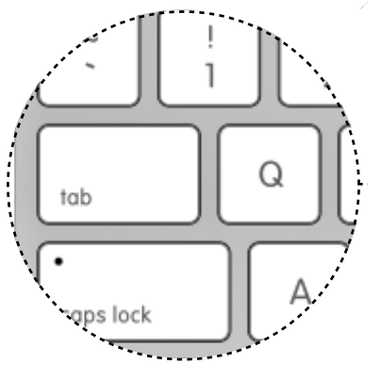
Use tab key to autocomplete names

cd changes directory

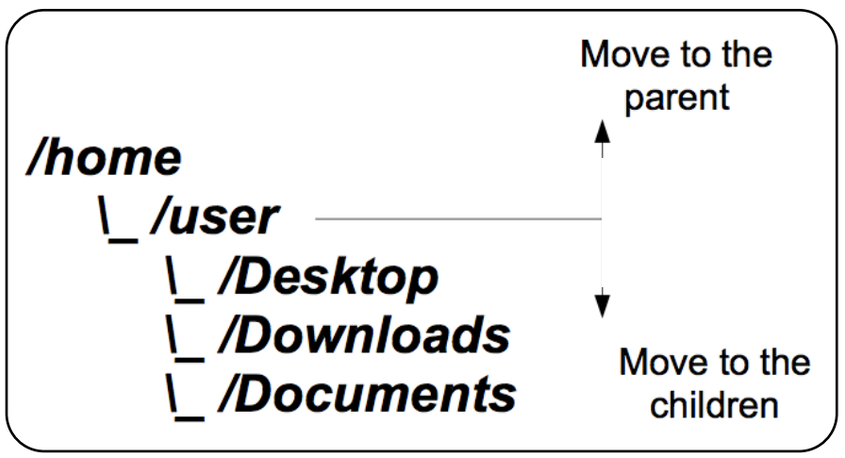
changes directory to Desktop

goes to parent directory

```
cd Desktop
cd ..
cd /        goes to root directory
cd          goes to home directory
cd -        goes to previous directory
```



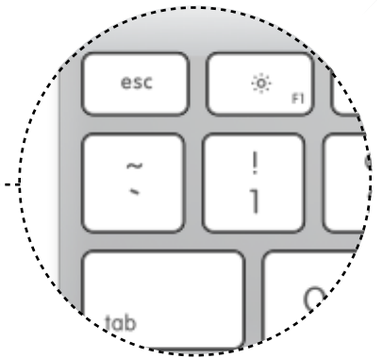
Use tab key to autocomplete names



Absolute and relative paths

list files in Desktop using an absolute path

```
ls /home/user/Desktop
ls Desktop/
ls ~/Desktop
```



list files in Desktop using your home as a reference

list files in Documents using a relative path (from your home: /home/bioinfo)

Absolute and relative paths

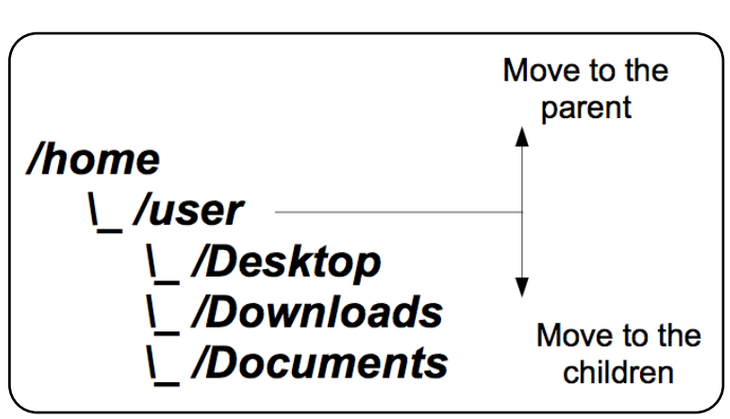
Absolute and relative paths

goes to *Desktop* from when you are in your home (/home/bioinfo)

```
cd Desktop/

ls ../Documents
```

list files from *Documents* when you are in *Desktop*



Create, copy, move and delete files

creates an empty file called tmp_file.txt

copies tmp_file.txt in file_copy.txt

```
touch tmp_file.txt
cp tmp_file.txt file_copy.txt
mv file1.txt file2.txt
rm file.txt
```

moves or rename a file

deletes file.txt

Tip: file names in lower case and with underscores instead of spaces

Create, copy and delete directories

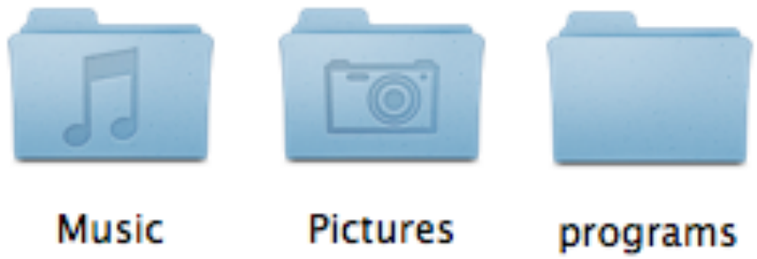
creates an empty directory called *dir_name*

deletes *dir_name* directory if it is empty

```
mkdir dir_name
rmdir dir_name
rm -r dir_name
cp -r dir_name dir_copy
```

delete *dir_name* and its files

copy *dir_name* and its files in a new folder



Compression commands

Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

groups and compress files

```
tar -zcvf file.tar.gz f1 f2
```

```
tar -zxvf file.tar.gz
```

decompress and ungroup a tar.gz file

files, directories or wildcards

UNIX Command-Line Cheat Sheet

BTI-SGN Bioinformatics Course 2014

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory
locate *file_name*	searches a file
man *command*	shows commands manual
top	shows process activity
df -h	shows disk space info

Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2* > *file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern*' *file*	Prints lines matching a pattern
grep -c '*pattern*' *file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*
sort -u *file*	sorts and return unique lines
uniq -c *file*	filters adjacent repeated lines
wc *file*	counts lines, words and bytes
paste *file1 file2*	concatenates the lines of input files
paste -d ","	concatenates the lines of input files by commas
sed	transforms text

Networking Commands	
wget *URL*	download a file from an URL
ssh *user@server*	connects to a server
scp	copy files between computers
apt-get install	installs applications in linux

Compression commands

compress file f1.txt in f1.txt.gz

```
gzip f1.txt
```

compress files f1 and f2 in file.zip

```
zip file.zip f1 f2
```

```
unzip file.zip
```

decompress file.zip

```
gunzip file.gz
```

decompress file.gz

Networking Commands

- Networking commands

UNIX Command-Line Cheat Sheet

BTI-SGN Bioinformatics Course 2014

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory
locate *file_name*	searches a file
man *command*	shows commands manual
top	shows process activity
df -h	shows disk space info

Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2* > *file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern*' *file*	Prints lines matching a pattern
grep -c '*pattern*' *file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*
sort -u *file*	sorts and return unique lines
uniq -c *file*	filters adjacent repeated lines
wc *file*	counts lines, words and bytes
paste *file1 file2*	concatenates the lines of input files
paste -d ","	concatenates the lines of input files by commas
sed	transforms text

Networking Commands	
wget *URL*	download a file from an URL
ssh *user@server*	connects to a server
scp	copy files between computers
apt-get install	installs applications in linux

Networking Commands

download a file from a URL

```
wget URL
```

```
wget ftp://ftp.solgenomics.net/bioinfo_class/UPLB/sgn_unix_commands_cheat_sheet_2015.pdf
```

download the UNIX command line cheat sheet PDF file

Commands to install software

```
aptitude search blast
sudo aptitude install blast2
sudo apt-get install pbzip2
```

installs *pbzip2* in your computer

call the command with super user permissions

Text Handling Commands

- Text Handling Commands

UNIX Command-Line Cheat Sheet

BTI-SGN Bioinformatics Course 2014

File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to home directory
touch *file_name*	creates en empty file
cp *file file_copy*	copy a file
cp -r	copy files contained in directories
rm *file*	deletes a file
rm -r *dir*	deletes a directory and its files
mv *file1 file2*	moves or renames a file
mkdir *dir_name*	creates a directory
rmdir *dir_name*	deletes a directory
locate *file_name*	searches a file
man *command*	shows commands manual
top	shows process activity
df -h	shows disk space info

Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2 > file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints first lines from a file
head -n 5 *file*	prints first five lines from a file
tail *file*	prints last lines from a file
tail -n 5 *file*	prints last five lines from a file
less *file*	view a file
less -N *file*	includes line numbers
less -S *file*	wraps long lines
grep '*pattern' file*	Prints lines matching a pattern
grep -c '*pattern' file*	counts lines matching a pattern
cut -f 1,3 *file*	retrieves data from selected columns in a tab-delimited file
sort *file*	sorts lines from a *file*
sort -u *file*	sorts and return unique lines
uniq -c *file*	filters adjacent repeated lines
wc *file*	counts lines, words and bytes
paste *file1 file2*	concatenates the lines of input files
paste -d ","	concatenates the lines of input files by commas
sed	transforms text

Networking Commands	
wget *URL*	download a file from an URL
ssh *user@server*	connects to a server
scp	copy files between computers
apt-get install	installs applications in linux

FASTA format

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol at the beginning.

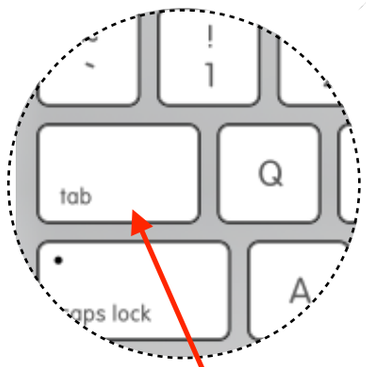
http://www.ncbi.nlm.nih.gov/

description line

sequence data

```
>sequence_ID1 description
ATGCGCGCGCGCGCGCGGGTAGCAGATGACGACACAGAGCGAGGATGCGCTGAGAGTA
GTGTGACGACGATGACGGAAAATCAGATGGACCCGATGACAGCATGACGATGGGACGGGA
AAGATTGGACCAGGACAGGACCAGGACCAGGACCAGGGATTAGA
>sequence_ID2 description
ATGGGGGGGACGACGATGGACACAGAGACAGAGACGACGACAGCAGACAGATTTACCTTA
GACGAGATAGGAGAGACGACAGATATATATATATAGCAGACAGACAGACATTTAGACGAG
ACGACGATAGACGATaaaaataa
```

Tab-delimited text files



Tab-delimited files are a very common format in scientific data. They consist in columns of text separated by tabs. Other file formats could have different delimiters.

Query	Subject	id %	length	mismatch	gaps	qstart	qend	sstart	send	evalue	score
ATCG00500.1	PACid:23047568	64.88	299	64	2	220	477	112	410	5e-131	388
ATCG00500.1	PACid:23052247	58.88	321	69	3	220	477	381	701	3e-117	361
ATCG00890.1	PACid:16418828	90.60	117	11	0	18	134	1	117	1e-71	220
ATCG00890.1	PACid:16412855	90.48	147	14	2	41	387	27	173	1e-68	214
ATCG00280.1	PACid:24129717	95.99	474	19	0	1	474	1	474	0.0	847
ATCG00280.1	PACid:24095593	95.36	474	22	0	1	474	1	474	0.0	840
ATCG00280.1	PACid:20871697	94.94	474	24	0	1	474	1	474	0.0	837

Tabular blast output example

Blast, SAM (mapping), BED, VCF (SNPs), GTF, GFF ...

BIOINFORMATICIAN

What is the best option to explore the content of a file of 2Gb?

A: MS Word

B: Less

1

C: Internet Explorer

D: Cat

WHO WANTS TO BE A

BIOINFORMATICIAN

What is the best option to explore the content of a file of 2Gb?

A: MS Word

B: Less

1

C: Internet Explorer

D: Cat

less to view large files

↓ ↑ ← →	scroll through the file
< or g	go to file beginning
> or G	go to file end
space bar	page down
b	page up

/pattern	search pattern
n	find next
N	find previous
q	quit *less*

view file *blast_sample.txt*

view file *blast_sample.txt* without wrapping long lines

```
less blast_sample.txt
less -S blast_sample.txt
less -N blast_sample.txt
```

view file *blast_sample.txt* showing line numbers

cat concatenates and prints files

prints file *sample1.fasta* on the screen

prints file *sample1.fasta* on the screen

```
cat sample1.fasta
cat /home/bioinfo/Desktop/unix_data/sample1.fasta
cat sample1.fasta sample2.fasta > new_file.fasta
```

concatenates files *sample1.fasta and sample2.fasta* and saves them in the file new_file.fasta

redirects output to a file

cat concatenates and prints files

concatenates all FASTA files in the current directory and saves them in the file *all_samples.fasta*

redirect output to a file

```
cat *fasta > all_samples.fasta

cat sample3.fasta >> new_file.fasta
```

appends *sample3.fasta* file to *new_file.fasta*

head displays first lines of a file

print first lines from *blast_sample.txt* file (10 by default) and save them in blast10.txt

```
head blast_sample.txt > blast10.txt
```

```
head -n 5 blast_sample.txt
```

print first five lines from *blast_sample.txt* file

tail displays the last part of a file

print last 10 lines from *blast_sample.txt* file

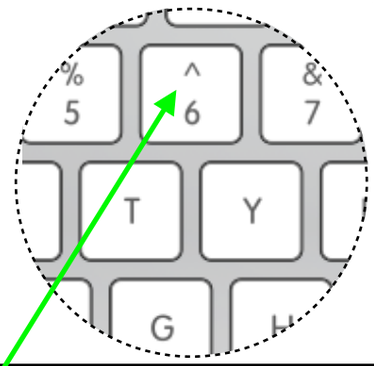
```
tail blast_sample.txt
```

```
tail -n 5 blast_sample.txt
```

print last five lines from *blast_sample.txt* file

grep searches patterns in files

prints lines starting with a “>”, i.e., prints description lines from FASTA files



counts lines starting with a “>”, i.e.,
it counts the number of sequences from a FASTA file

```
grep '^>' sample1.fasta
grep -c '^>' sample1.fasta
grep -c '^+$' *fastq
```

search pattern at line start

search pattern at line end

counts lines formed only by “+”, i.e., it counts the number of sequences from all FASTQ files in the current directory

cut gets columns from a tab-delimited file

prints columns 1 and 2 from *blast10.txt*

```
cut -f 1,2 blast10.txt
```

```
cut -c 1-4,17-21 blast_sample.txt > tmp.txt
```

prints characters from 1 to 4 and from 17 to 21 for each line in *blast_sample.txt* and save them in *tmp.txt*

sort sorts lines from a file

sort lines from file *tmp.txt* and save them in *tmp2.txt*

sort lines from file *tmp.txt* and remove the repeated ones

```
sort tmp.txt > tmp2.txt
sort -u tmp.txt
uniq -c tmp2.txt
```

removes repeated lines from *tmp.txt and counts how many times they were repeated.* Lines have to be sorted since only adjacent lines are compared

wc counts lines, words and characters

counts lines, words and characters in *blast10.txt*

counts lines in *blast10.txt*

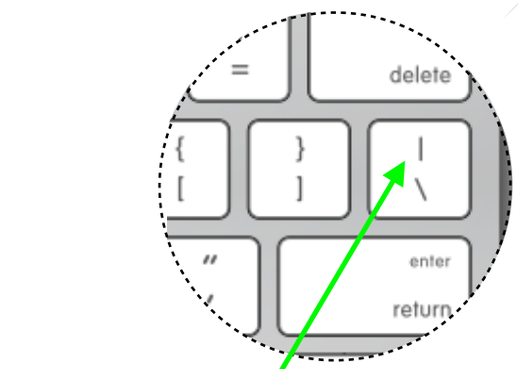
```
wc blast10.txt
wc -l blast10.txt
wc -w blast10.txt
wc -c blast10.txt
```

counts bytes in *blast_sample.txt* (including the line return)

counts words in *blast10.txt*

Pipelines

Pipelines consists in concatenate several commands by using the output of the first command as the input of the next one.
Two commands are connected placing the sign "|" between them.




```
ls | wc -l
```

counts files in current directory

Pipelines

counts sequences in all fasta files from current directory

prints sequence description line for all fasta files from current directory

```
cat *fasta | grep -c "^>"
cat *fasta | grep "^>" | sed 's/>//'
cut -f 1 blast_sample.txt | sort -u | wc -l
cut -f 1 blast_sample.txt | sort | uniq -c
```

counts different query ids in a blast tabular file

counts the appearance of each query id in a blast tabular file

shell script (bash) example

- All commands and programs we run in the terminal could be included in a text file with extension .sh
- This file will execute the commands in the order they were written, from top to bottom.

mapping.sh — Desktop

mapping.sh

```
1  #!/bin/bash
2
3  # indexing the reference
4  bowtie2-build -f nitab38.fasta ../bowtie2/mydb/mydb_indexed
5
6  # mapping of the reads over the reference and translate the sam format output to bam
7  bowtie2 --threads 42 -X 8000 --rf -x bowtie2/mydb/mydb_indexed -q -1 my_reads.pair1.fq -2 my_reads.pair2.fq | samtools view -S -b -h - -o alignment_output.bam
8
9  # sort the mapping output
10 java -jar picard-tools-1.87/SortSam.jar INPUT="alignment_output.bam" OUTPUT="align_sorted.bam" SORT_ORDER=coordinate VALIDATION_STRINGENCY=LENIENT
11
12 # get stats from the mapping output
13 java -jar picard-tools-1.87/CollectAlignmentSummaryMetrics.jar INPUT="align_sorted.bam" OUTPUT="picard_stats.txt"
14 java -jar picard-tools-1.87/CollectInsertSizeMetrics.jar HISTOGRAM_FILE="insert_histogram.txt" INPUT="align_sorted.bam" OUTPUT="insert_stats.txt"
15
```

Line: 15 | Shell Script (Bash) | Tab Size: 4

head of bash scripts

comment line

command or program line execution

EXERCISES

EPISODE I

Exercises

1. Decompress pineapple_data.tar.gz
2. Create a gff3 file for the pineapple LG1 from the pineapple.gff3 file
3. Count how many genes are in each chromosome from the pineapple genome

Extra Exercises

1. Merge all fasta files, in the order *sample3.fasta*, *sample1.fasta* and *sample2.fasta*, and save them in a new file called *all_samples.fasta*
2. Merge all fastq files (*sample1.fastq, sample2.fastq and sample3.fastq*) using wildcards, and save them in a new file called *all_samples.fastq*
3. Save in a file called *blast100.txt* the first 100 lines from *blast_sample.txt*
4. Save in a file called *blast200.txt* the last 200 lines from *blast_sample.txt*
5. How many sequences are in *all_samples.fasta*?
6. How many sequences are in *all_sample.fastq*?
7. Create a file with the subject ids and their scores for the 15 first lines from *blast_sample.txt*
8. How many different queries ids are in *blast_sample.txt*?
9. How many different subjects ids are in *blast_sample.txt*?
10. Change all '|' in *blast_sample.txt* by '_' and save the new file in Desktop as *tmp.txt*.
11. Count how many genes are in each *Arabidopsis thaliana* chromosome, chloroplast and mitochondria based on the next file:

    ftp://ftp.arabidopsis.org/home/tair/Sequences/blast_datasets/TAIR10_blastsets/TAIR10_pep_20110103_representative_gene_model_updated

Solutions

1. `cat sample3.fasta sample1.fasta sample2.fasta > all_samples.fasta`
2. `cat *fastq > all_samples.fastq`
3. `head -n 100 blast_sample.txt > blast100.txt`
4. `tail -n 200 blast_sample.txt > blast200.txt`
5. `grep -c "^>" all_samples.fasta` = 12
6. `grep -c "^+$" all_samples.fastq` = 33
7. `head -n 15 blast_sample.txt | cut -f 2,12`
8. `cut -f 1 blast_sample.txt | sort -u | wc -l` = 32
9. `cut -f 2 blast_sample.txt | sort -u | wc -l` = 887
10. `sed 's/|/_/g' blast_sample.txt > ../tmp.txt`
11. `grep ">" at_prot.fasta | cut -c 1-4 | sort | uniq -c`
12. AT1: 7078, AT2: 4245, AT3: 5437, AT4: 4128, AT5: 6318, ATC: 88, ATM: 122