```
Noes-MacBook-Pro:~ Noe$
```

Introduction to UNIX command-line

Boyce Thompson Institute
2015

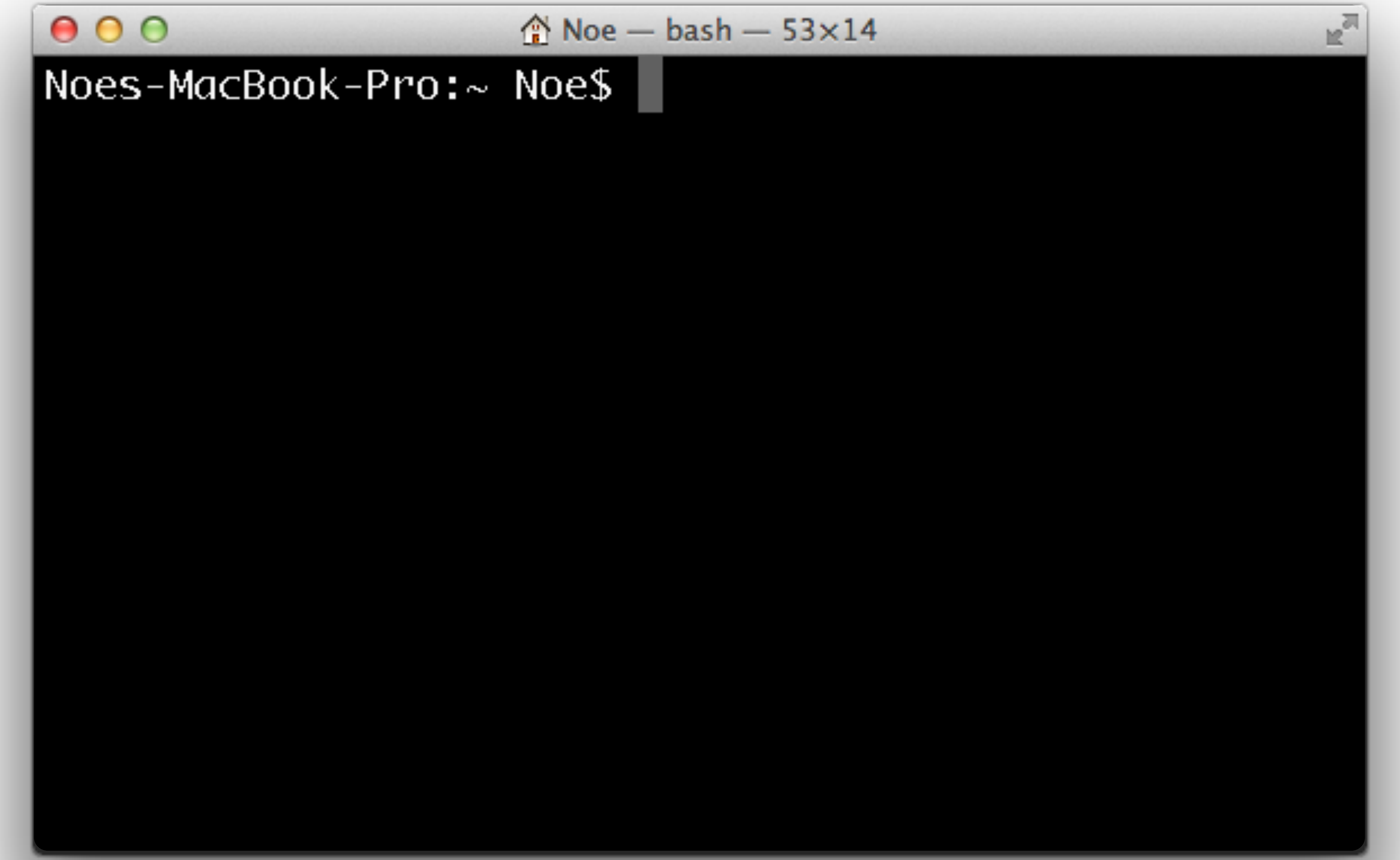
Noe Fernandez

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

What is a terminal?



```
Noes-MacBook-Pro:~ Noe$
```

Origins of Linux. The UNIX operating system

Ken Thompson and Dennis Ritchie at work at a PDP-11 at Bell Labs, ca. 1971

Ritchie also developed the C language

Why use command-line?

- Most software for biological data analysis is used through UNIX command-line terminal and 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 servers 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
- Bioinformatics allows the automatization of processes to study hundred or thousand data (genes, proteins, etc.)
- Data manual analysis is tedious and can introduce errors
- Compression commands are very useful for NGS, since large data files usually are stored and shared as compressed files

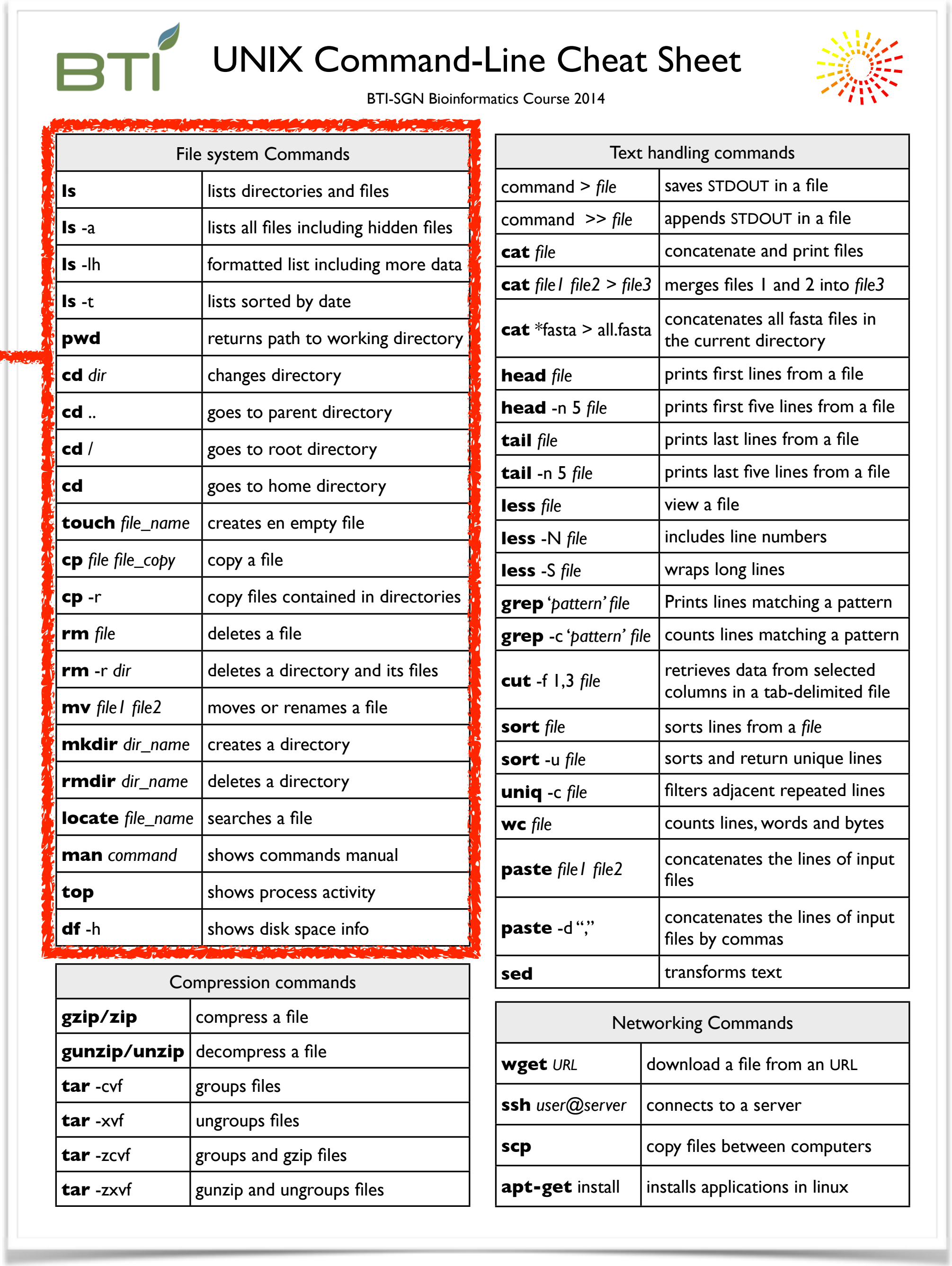
File system navigation

- File system commands

Download the cheat sheet from:

http://www.slideshare.net/NoFernndezPozo/unix-command-sheet2014

https://btiplantbioinfocourse.files.wordpress.com/2014/02/unix_command_sheet_2014.pdf



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

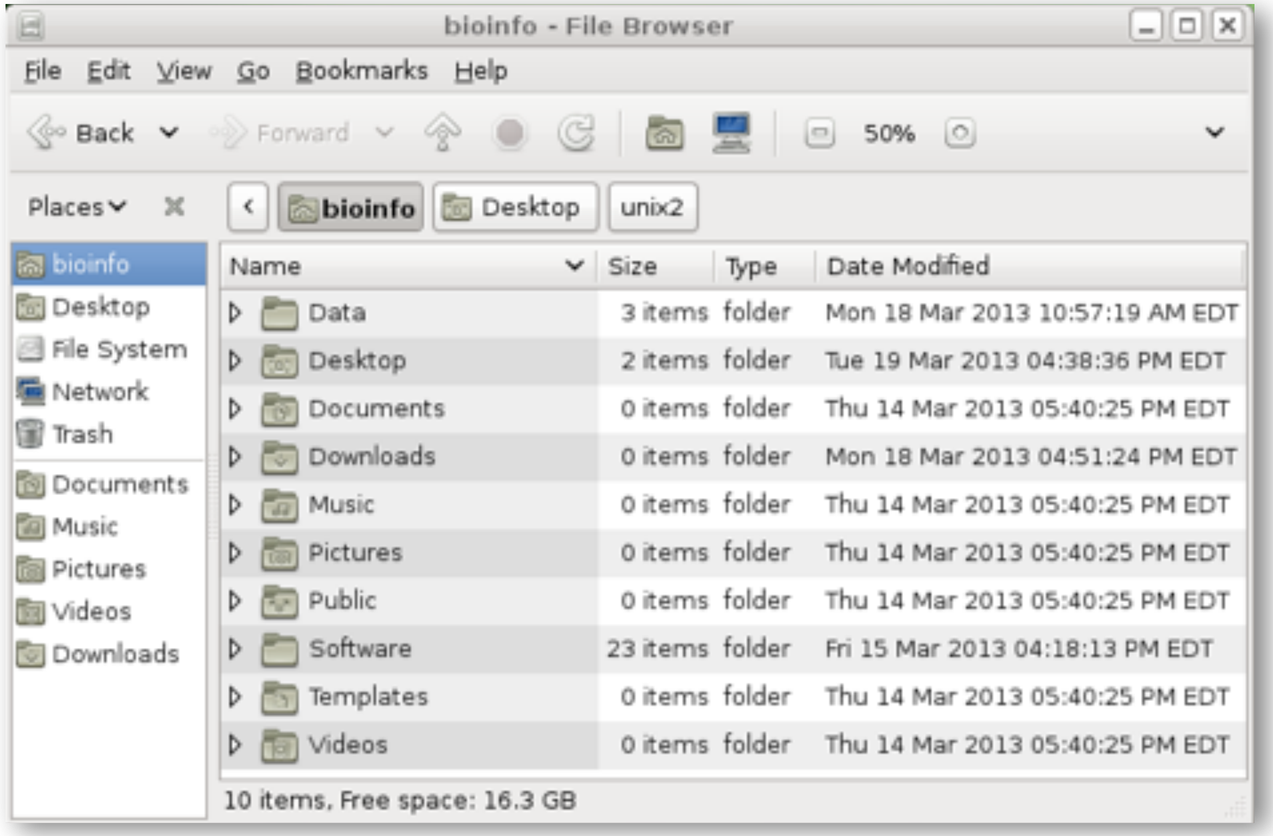
Compression commands	
gzip/zip	compress a file
gunzip/unzip	decompress a file
tar -cvf	groups files
tar -xvf	ungroups files
tar -zcvf	groups and gzip files
tar -zxvf	gunzip and ungroups files

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

File system navigation

File Browser

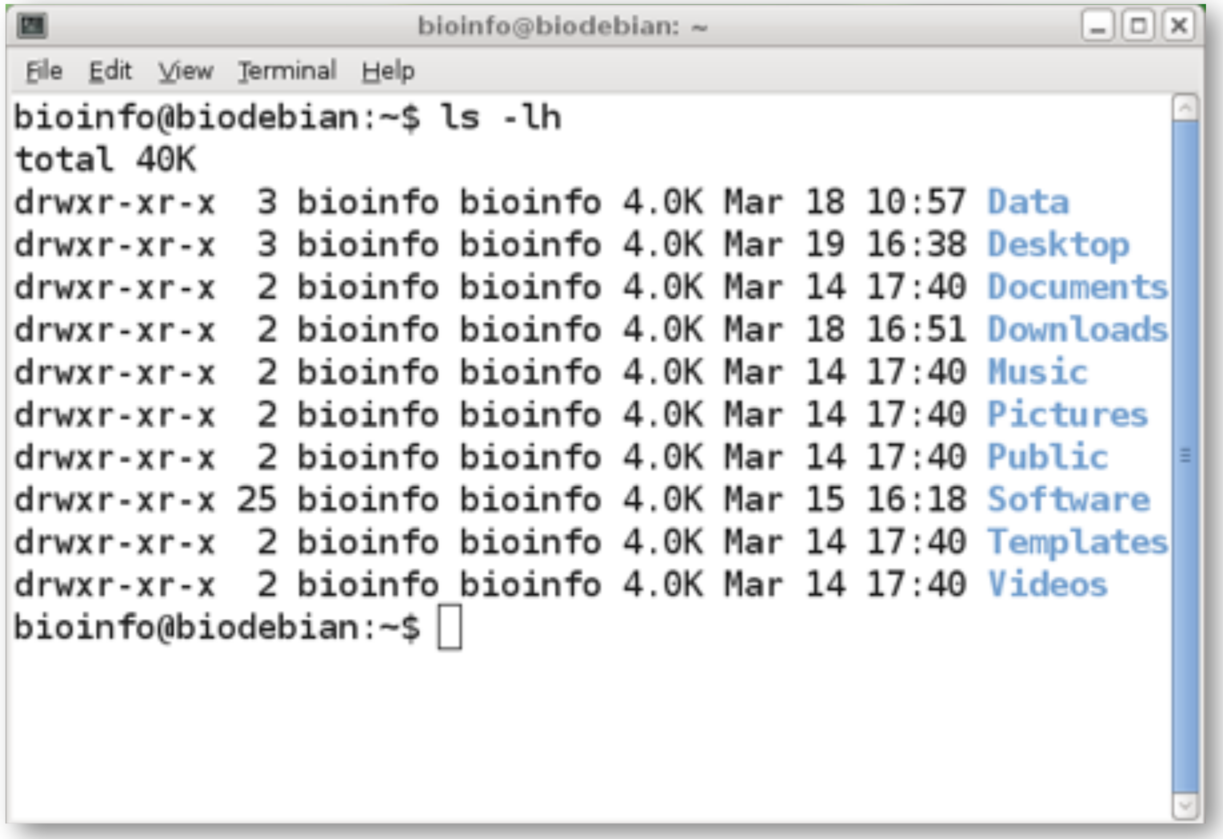


Name	Size	Type	Date Modified
Data	3 items	folder	Mon 18 Mar 2013 10:57:19 AM EDT
Desktop	2 items	folder	Tue 19 Mar 2013 04:38:36 PM EDT
Documents	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT
Downloads	0 items	folder	Mon 18 Mar 2013 04:51:24 PM EDT
Music	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT
Pictures	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT
Public	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT
Software	23 items	folder	Fri 15 Mar 2013 04:18:13 PM EDT
Templates	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT
Videos	0 items	folder	Thu 14 Mar 2013 05:40:25 PM EDT

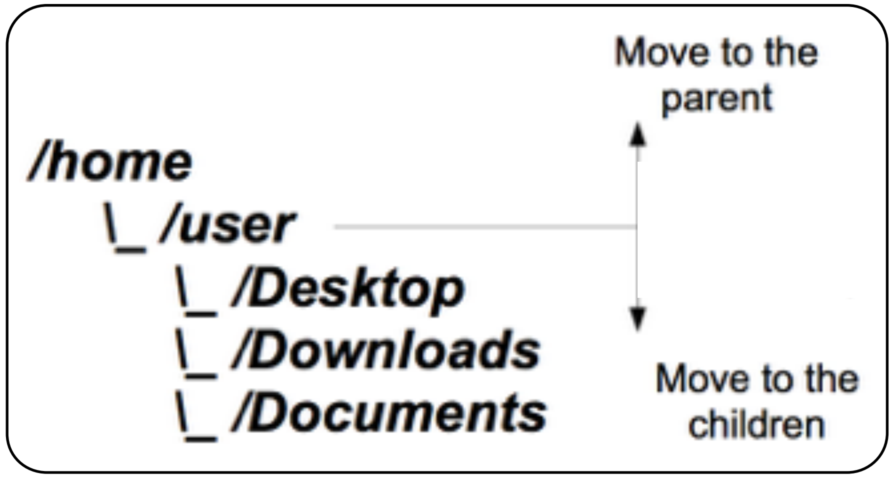
10 items, Free space: 16.3 GB

=

Terminal



```
bioinfo@biodebian:~$ ls -lh
total 40K
drwxr-xr-x  3 bioinfo bioinfo 4.0K Mar 18 10:57 Data
drwxr-xr-x  3 bioinfo bioinfo 4.0K Mar 19 16:38 Desktop
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Documents
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 18 16:51 Downloads
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Music
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Pictures
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Public
drwxr-xr-x 25 bioinfo bioinfo 4.0K Mar 15 16:18 Software
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Templates
drwxr-xr-x  2 bioinfo bioinfo 4.0K Mar 14 17:40 Videos
bioinfo@biodebian:~$
```

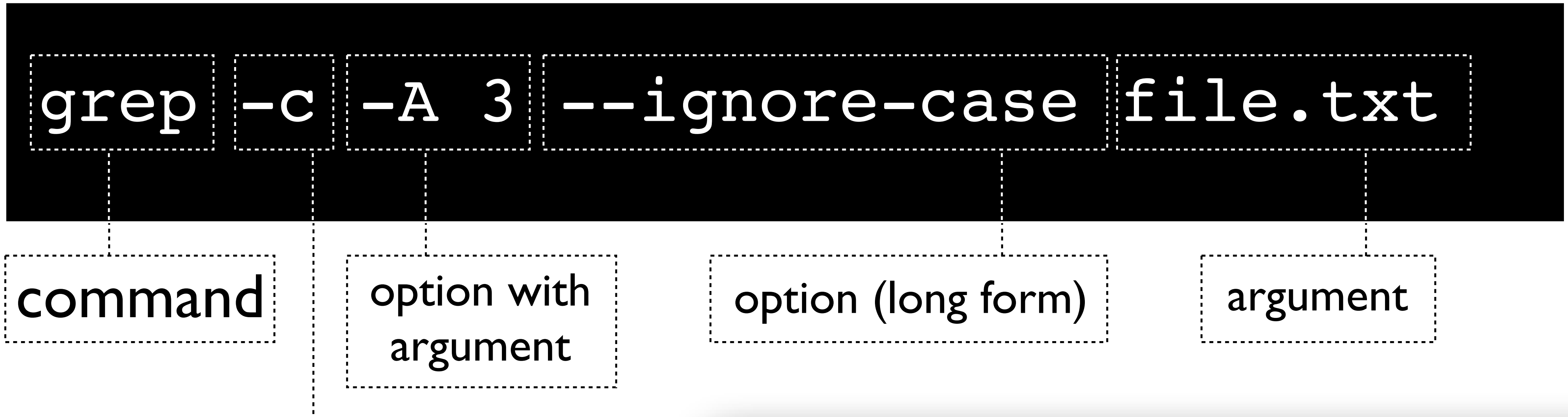


```
/home
  \_ /user
       \_ /Desktop
       \_ /Downloads
       \_ /Documents
```

Move to the parent

Move to the children

Anatomy of a UNIX command

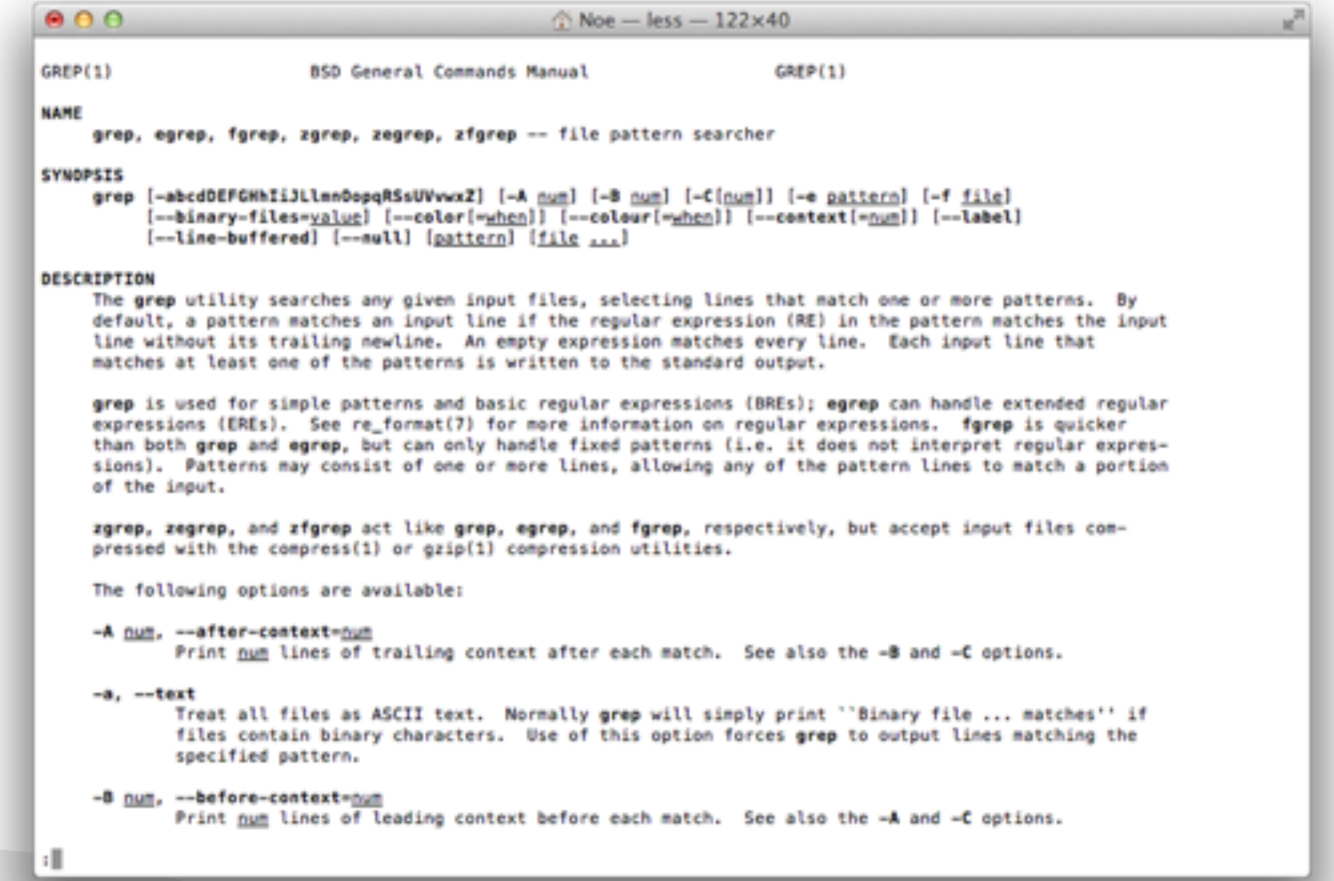


```
grep -c -A 3 --ignore-case file.txt
```

Simple option flag (short form)

print grep manual

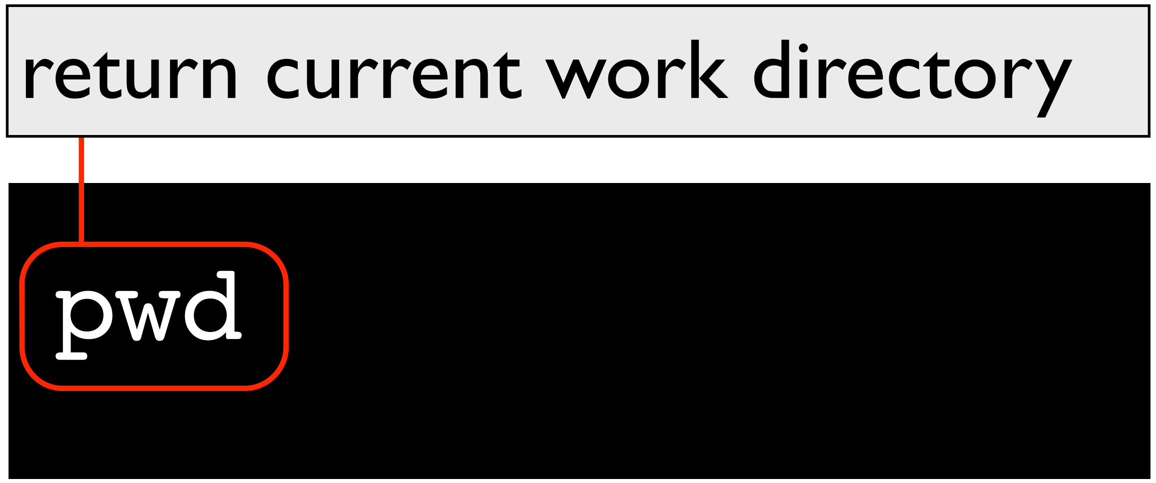
```
man grep
```



```
GREP(1)                   BSD General Commands Manual                  GREP(1)

NAME
     grep, egrep, fgrep, zgrep, zegrep, zfgrep -- file pattern searcher

SYNOPSIS
     grep [-abcdDEFGHhIiJLlmnOopqRSsUVvwxZ] [-A num] [-B num] [-C[num]] [-e pattern] [-f file]
          [--binary-files=value] [--color[=when]] [--colour[=when]] [--context[=num]] [--label]
          [--line-buffered] [--null] [pattern] [file ...]

DESCRIPTION
     The grep utility searches any given input files, selecting lines that match one or more patterns.  By
     default, a pattern matches an input line if the regular expression (RE) in the pattern matches the input
     line without its trailing newline.  An empty expression matches every line.  Each input line that
     matches at least one of the patterns is written to the standard output.

     grep is used for simple patterns and basic regular expressions (BREs); egrep can handle extended regular
     expressions (EREs).  See re_format(7) for more information on regular expressions.  fgrep is quicker
     than both grep and egrep, but can only handle fixed patterns (i.e. it does not interpret regular expres-
     sions).  Patterns may consist of one or more lines, allowing any of the pattern lines to match a portion
     of the input.

     zgrep, zegrep, and zfgrep act like grep, egrep, and fgrep, respectively, but accept input files com-
     pressed with the compress(1) or gzip(1) compression utilities.

     The following options are available:

     -A num, --after-context=num
             Print num lines of trailing context after each match.  See also the -B and -C options.

     -a, --text
             Treat all files as ASCII text.  Normally grep will simply print ``Binary file ... matches'' if
             files contain binary characters.  Use of this option forces grep to output lines matching the
             specified pattern.

     -B num, --before-context=num
             Print num lines of leading context before each match.  See also the -A and -C options.

:
```

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

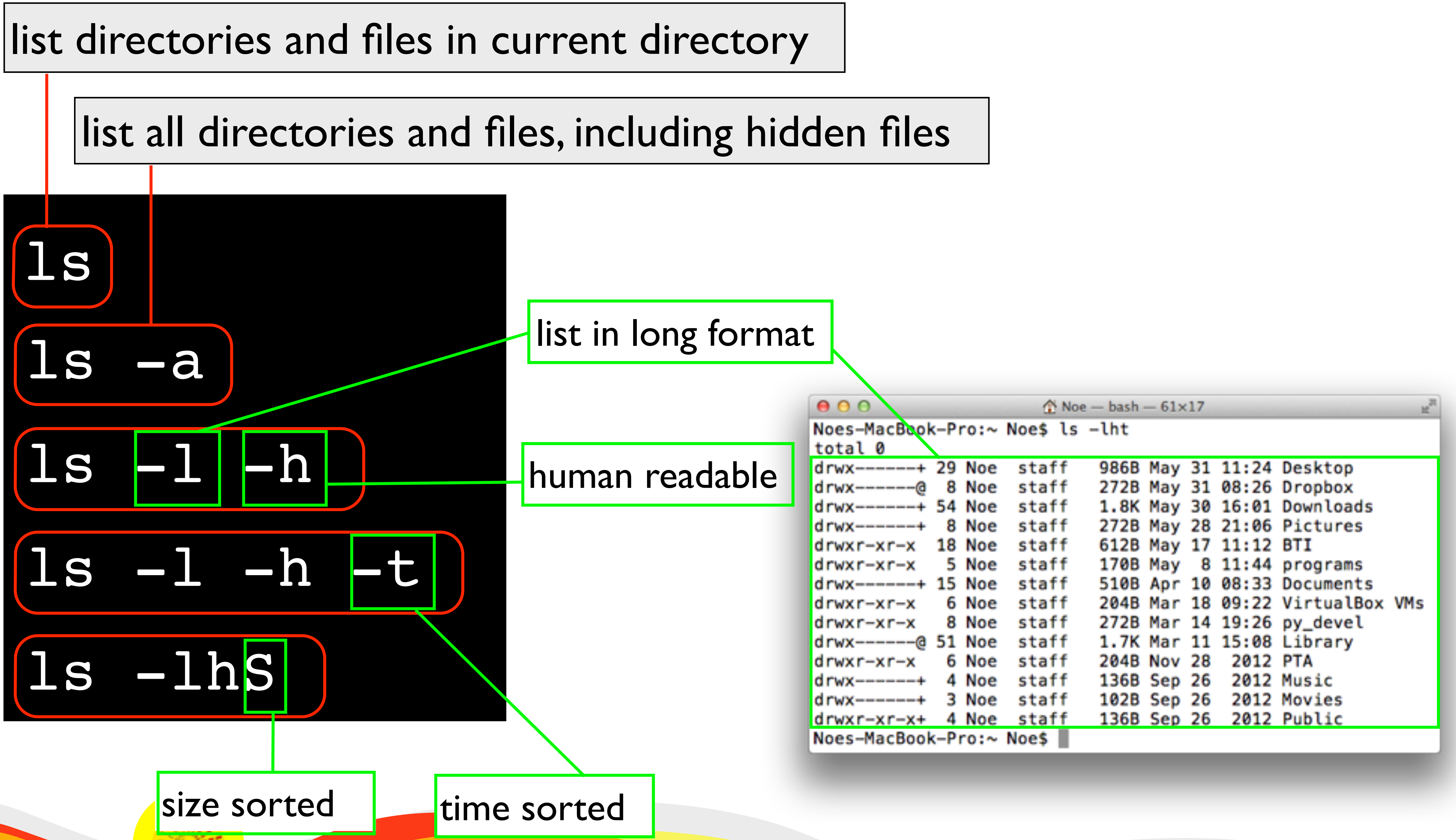
where am I?	pwd
how to change current directory	cd
what files and directories are in my current directory?	ls



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

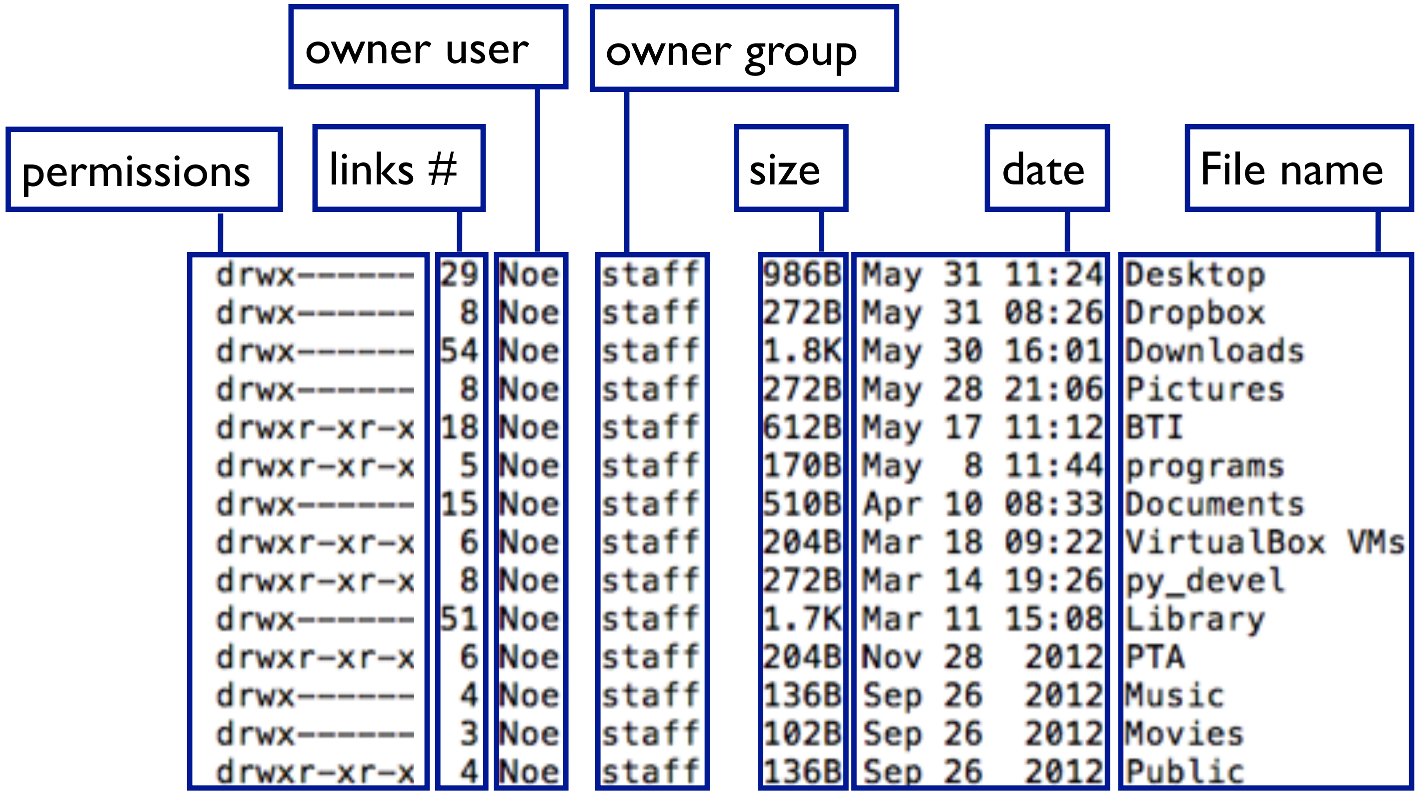
human readable

size sorted

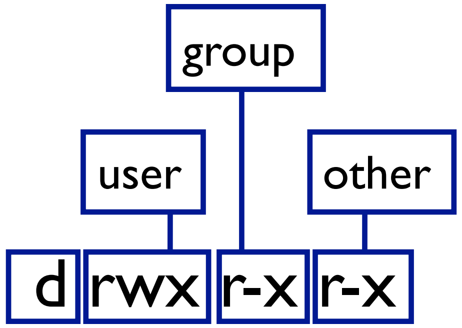
time sorted

```
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$ 
```

ls lists directories and files



```
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
```



d Directory
- Regular file

r readable
w writable
x executable or searchable
- not rwx

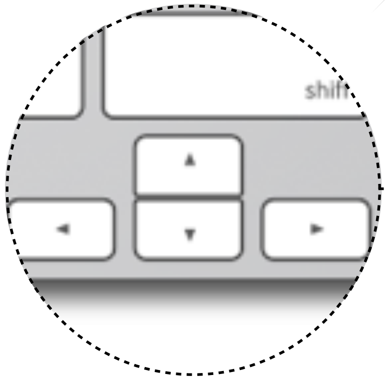
Wildcards, history and some shortcuts

```
ls *txt
```

list all txt files in current directory

```
ls P*s
```

list files starting with P and ending with s, e.g.: Pictures, Photos, Programs ...



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

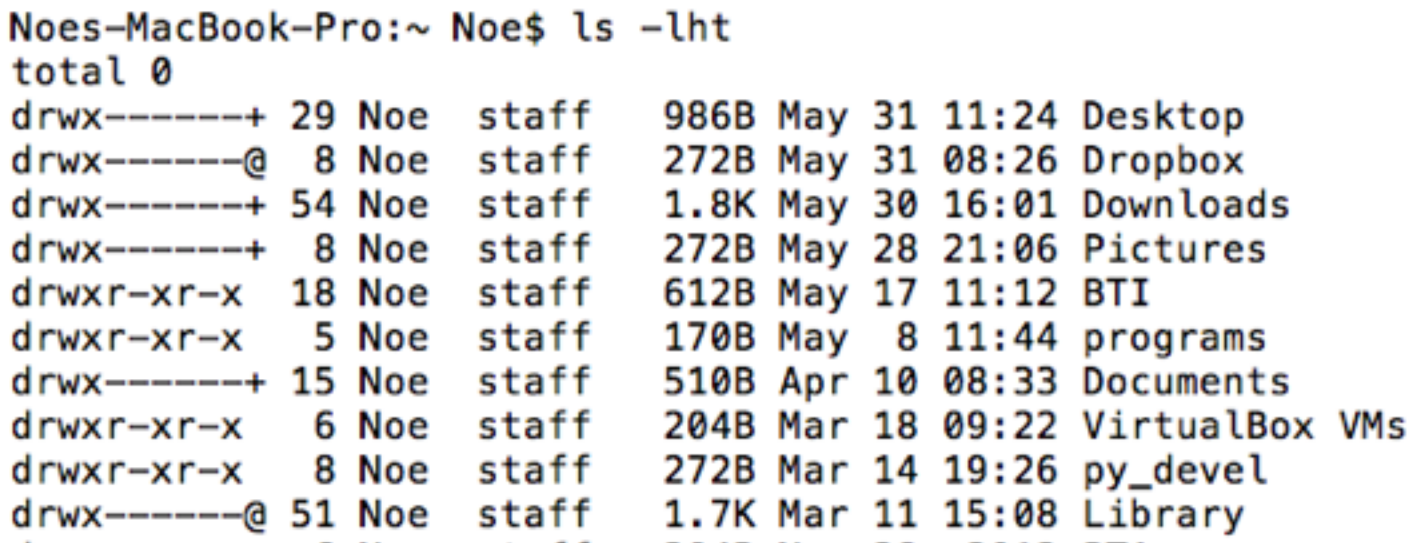
```
ls my\ folder
```

list a folder containing a space

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

Use tab key to autocomplete names

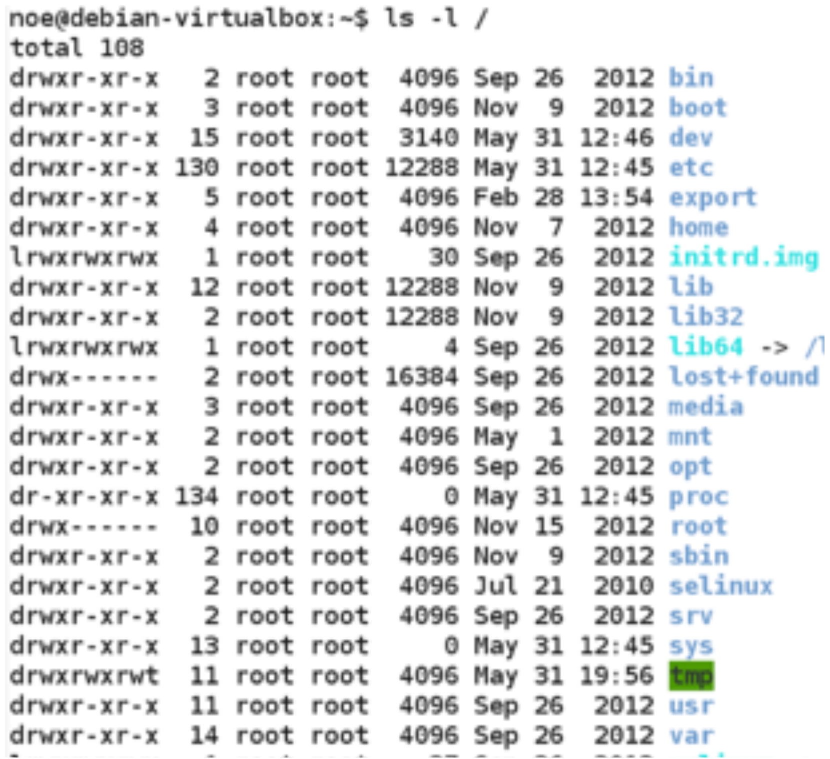
Home and Root directories



```
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
```

Home directory

/home/bioinfo
/home/noe
/home/noe/Desktop

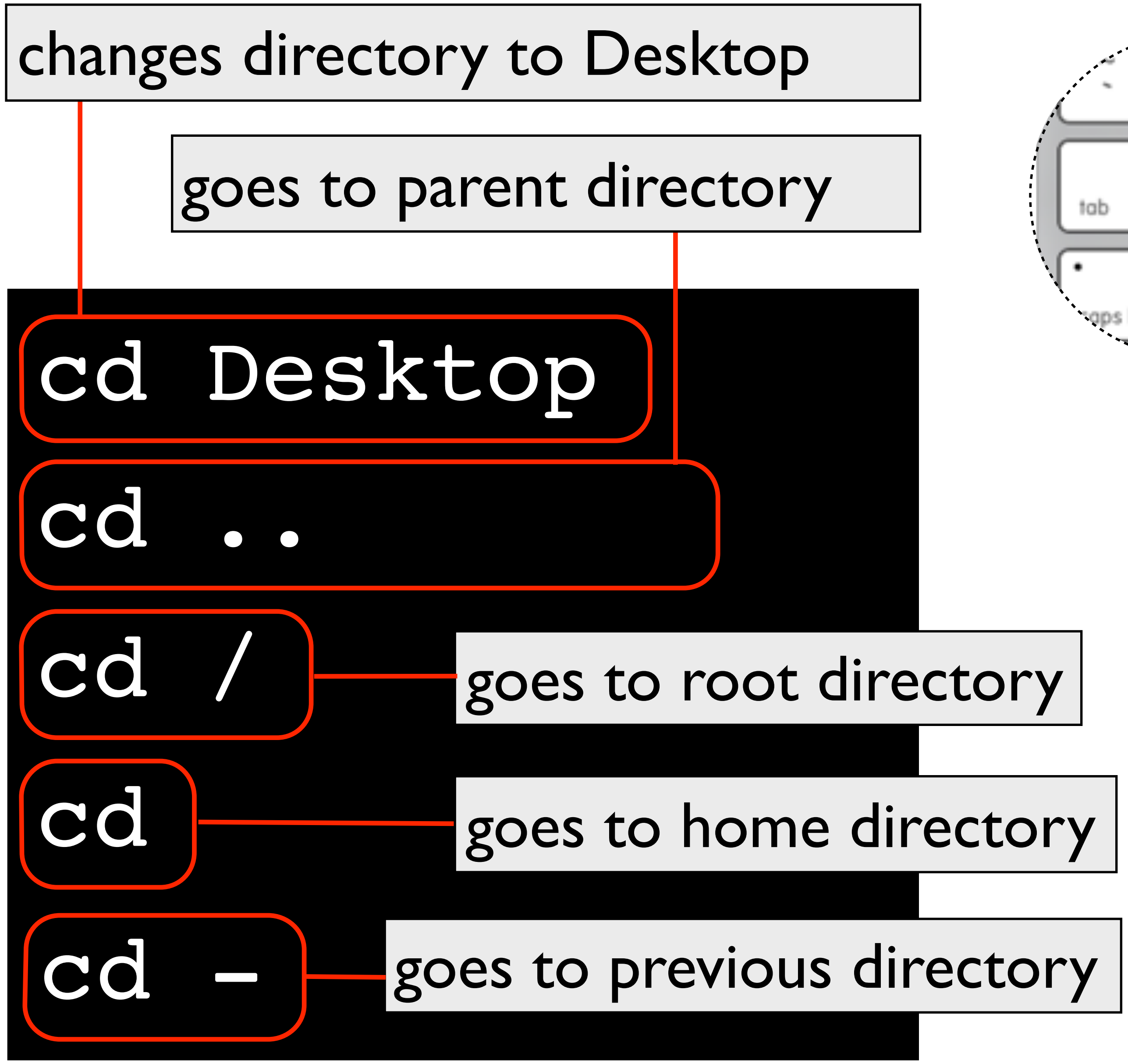


```
noe@debian-virtualbox:~$ ls -l /
total 108
drwxr-xr-x   2 root root  4096 Sep 26  2012 bin
drwxr-xr-x   3 root root  4096 Nov  9  2012 boot
drwxr-xr-x  15 root root  3140 May 31 12:46 dev
drwxr-xr-x 130 root root 12288 May 31 12:45 etc
drwxr-xr-x   5 root root  4096 Feb 28 13:54 export
drwxr-xr-x   4 root root  4096 Nov  7  2012 home
lrwxrwxrwx   1 root root    30 Sep 26  2012 initrd.img
drwxr-xr-x  12 root root 12288 Nov  9  2012 lib
drwxr-xr-x   2 root root 12288 Nov  9  2012 lib32
lrwxrwxrwx   1 root root     4 Sep 26  2012 lib64 -> /l
drwx------   2 root root 16384 Sep 26  2012 lost+found
drwxr-xr-x   3 root root  4096 Sep 26  2012 media
drwxr-xr-x   2 root root  4096 May  1  2012 mnt
drwxr-xr-x   2 root root  4096 Sep 26  2012 opt
dr-xr-xr-x 134 root root     0 May 31 12:45 proc
drwx------  10 root root  4096 Nov 15  2012 root
drwxr-xr-x   2 root root  4096 Nov  9  2012 sbin
drwxr-xr-x   2 root root  4096 Jul 21  2010 selinux
drwxr-xr-x   2 root root  4096 Sep 26  2012 srv
drwxr-xr-x  13 root root     0 May 31 12:45 sys
drwxrwxrwt  11 root root  4096 May 31 19:56 tmp
drwxr-xr-x  11 root root  4096 Sep 26  2012 usr
drwxr-xr-x  14 root root  4096 Sep 26  2012 var
```

Root directory

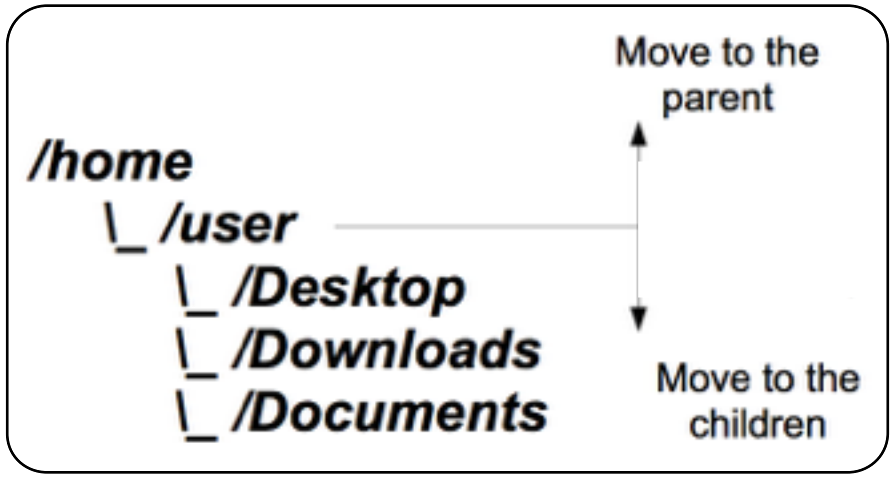
/bin, /lib, /usr	code and code libraries
/var	logs and other data
/home	user directories
/tmp	temporary files
/etc	configuration information
/proc	special file system in Linux

cd changes directory



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

Use tab key to autocomplete names



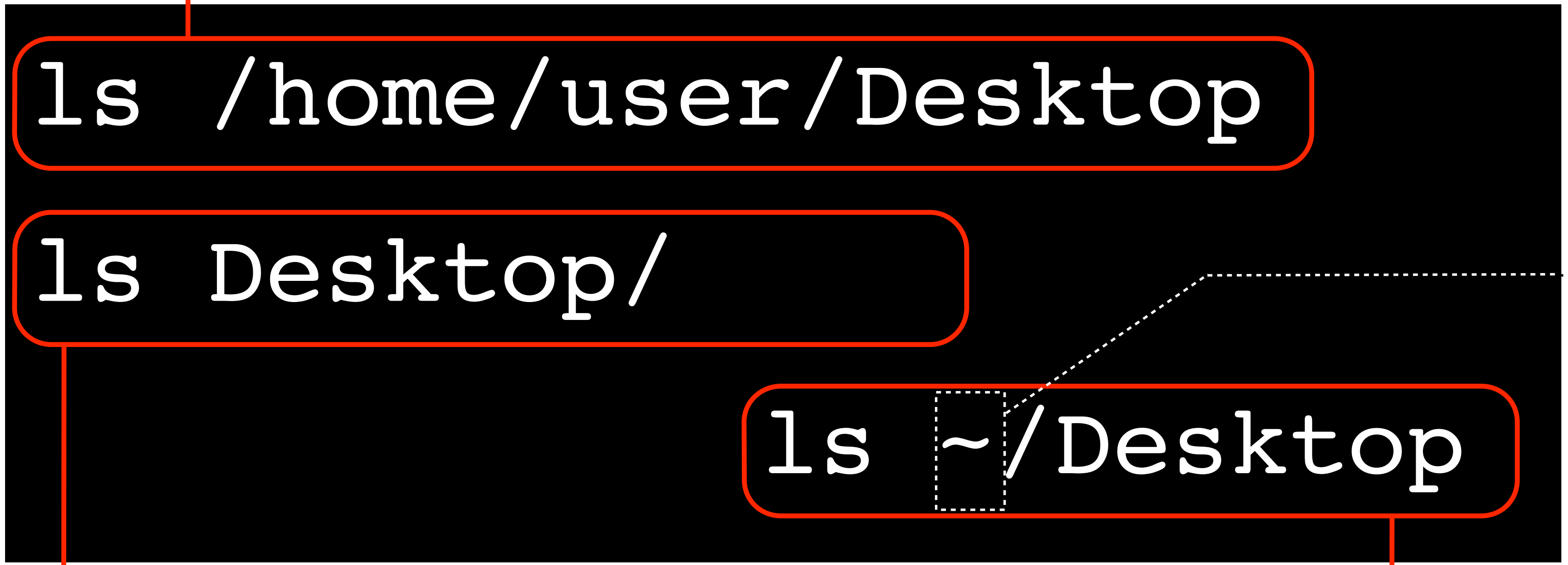
```
/home
   \_/user
       \_/Desktop
       \_/Downloads
       \_/Documents
```

Move to the parent

Move to the children

Absolute and relative paths

list files in Desktop using an absolute path

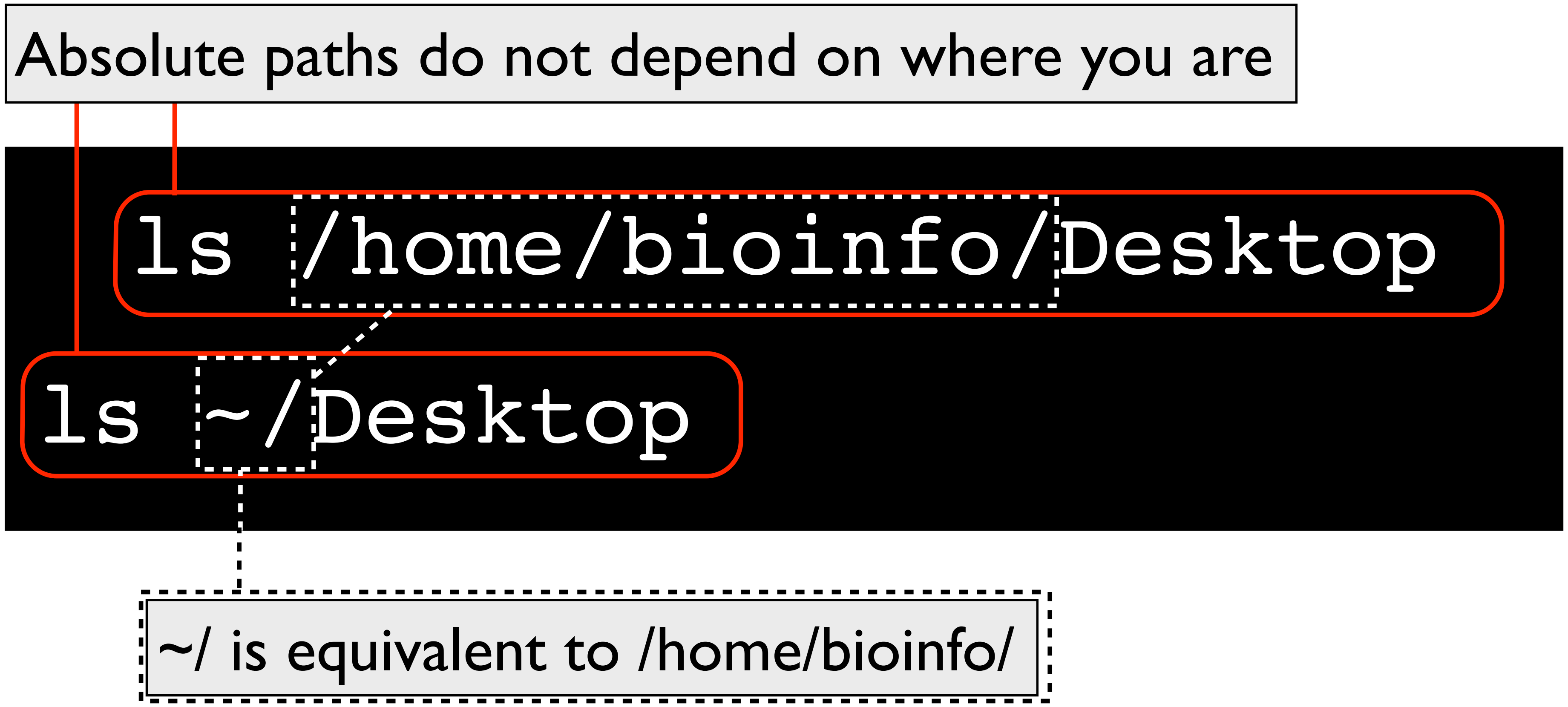


```
ls /home/user/Desktop
ls Desktop/
                 ls ~/Desktop
```

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



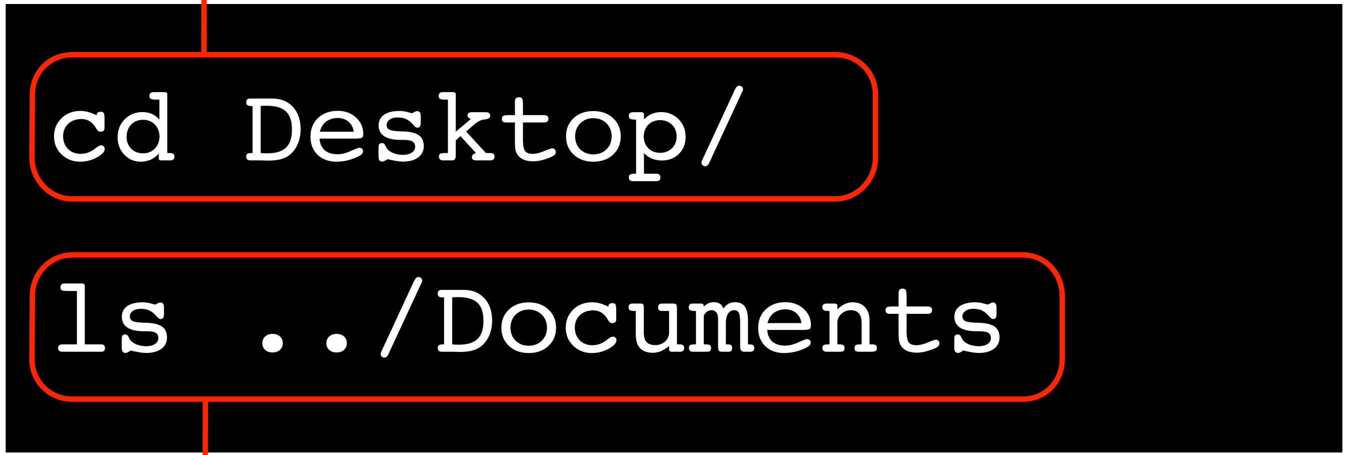
Absolute paths do not depend on where you are

```
  ls /home/bioinfo/Desktop
ls ~/Desktop
```

~/ is equivalent to /home/bioinfo/

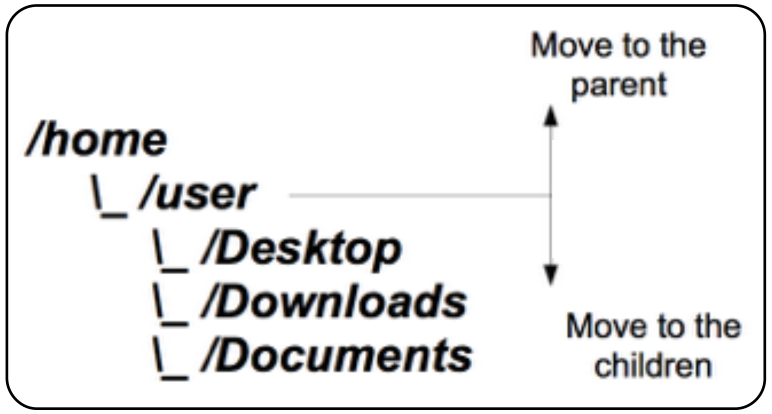
Absolute and relative paths

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



```
cd Desktop/

ls ../Documents
```

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



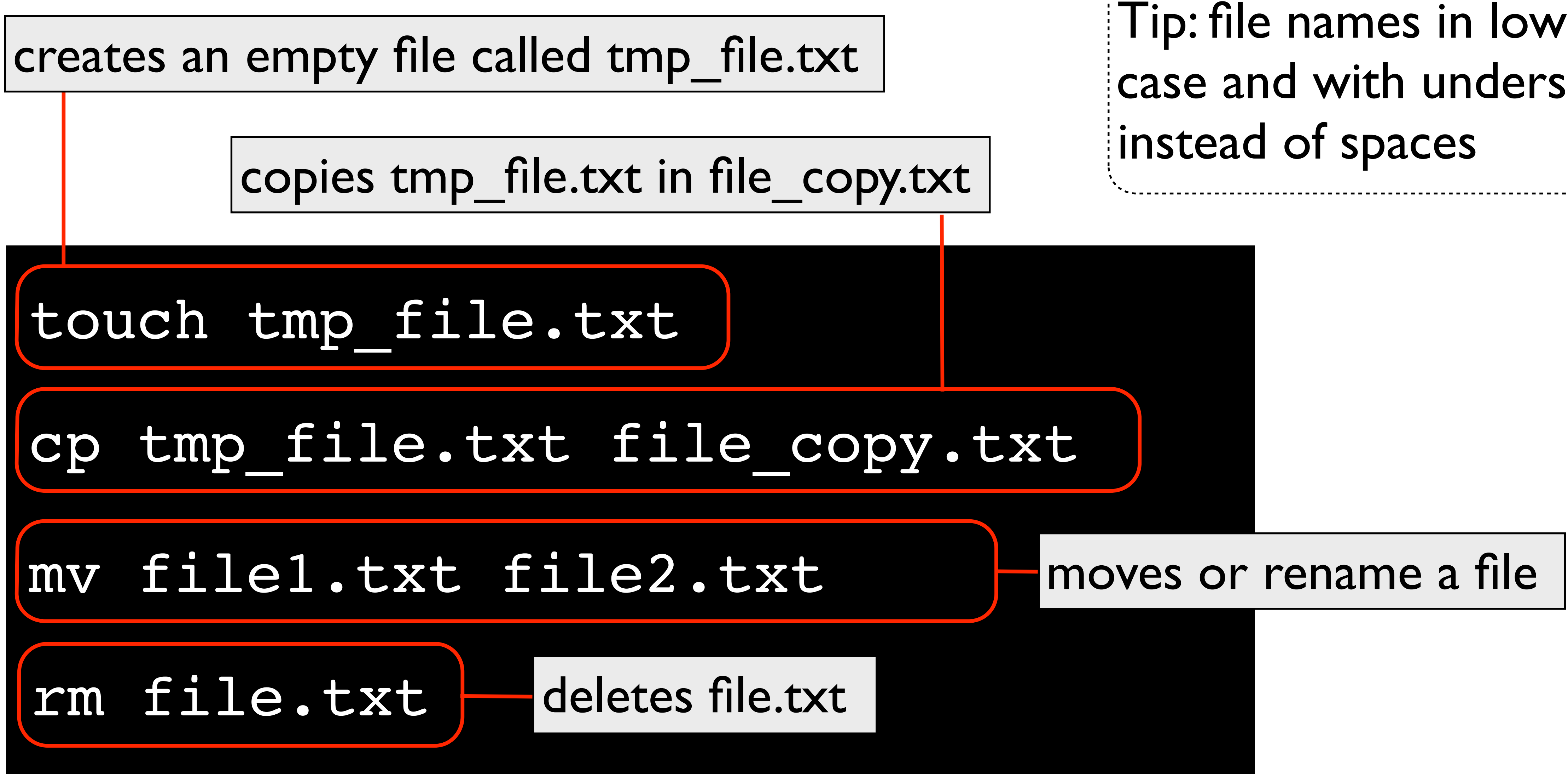
```
/home
   \_/user
      \_/Desktop
      \_/Downloads
      \_/Documents
```

Move to the parent

Move to the children

Create, copy, move and delete files

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



creates an empty file called tmp_file.txt

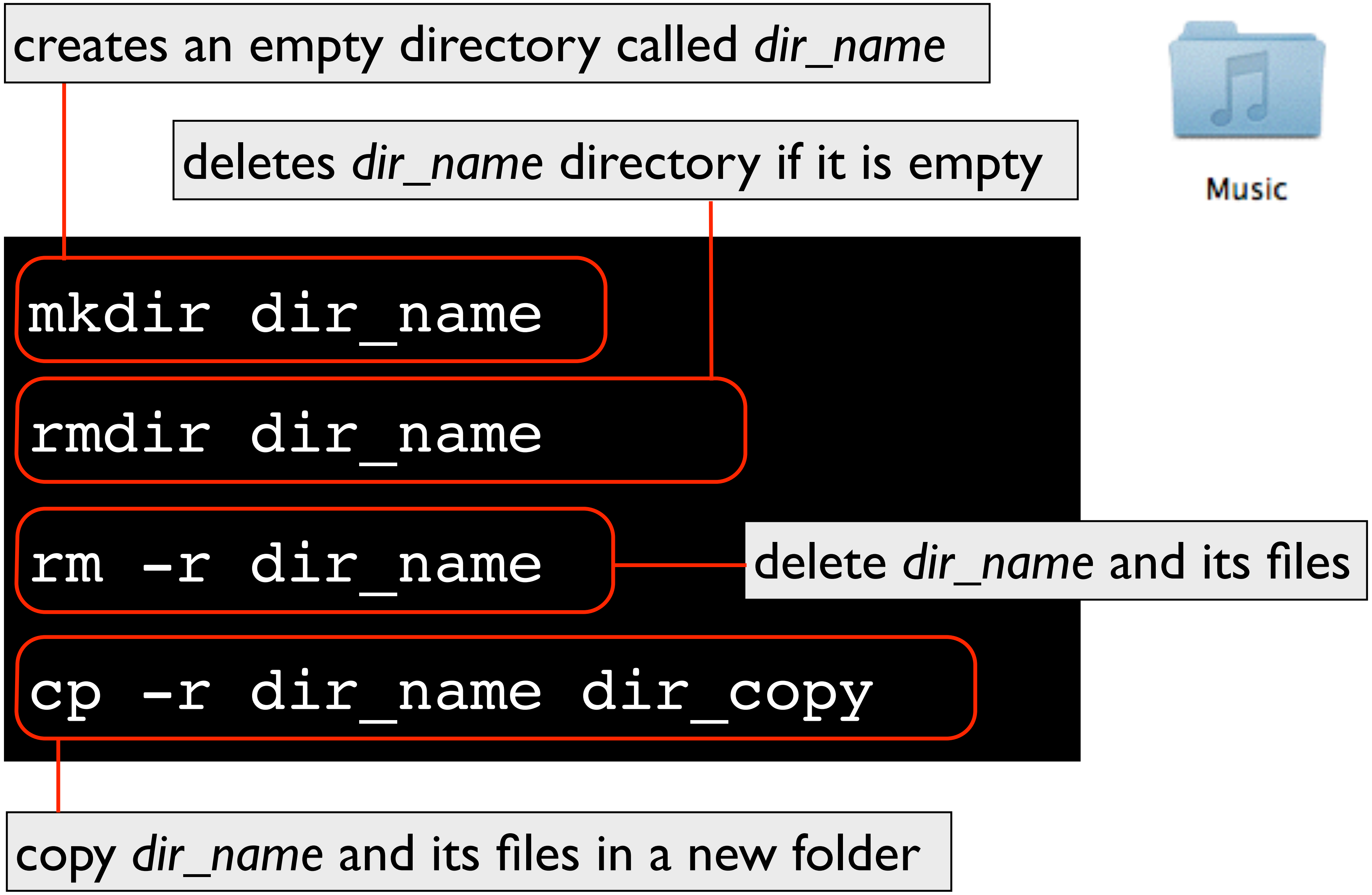
copies tmp_file.txt in file_copy.txt

```
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

Create, copy and delete directories



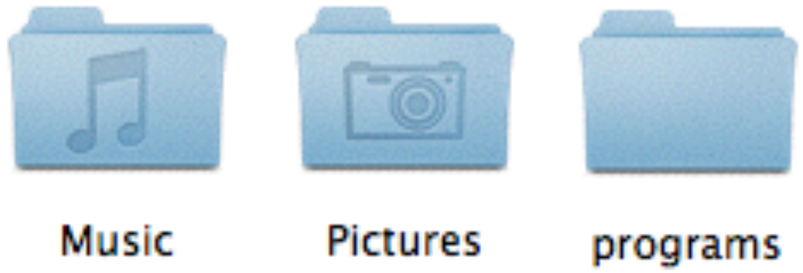
creates an empty directory called *dir_name*

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

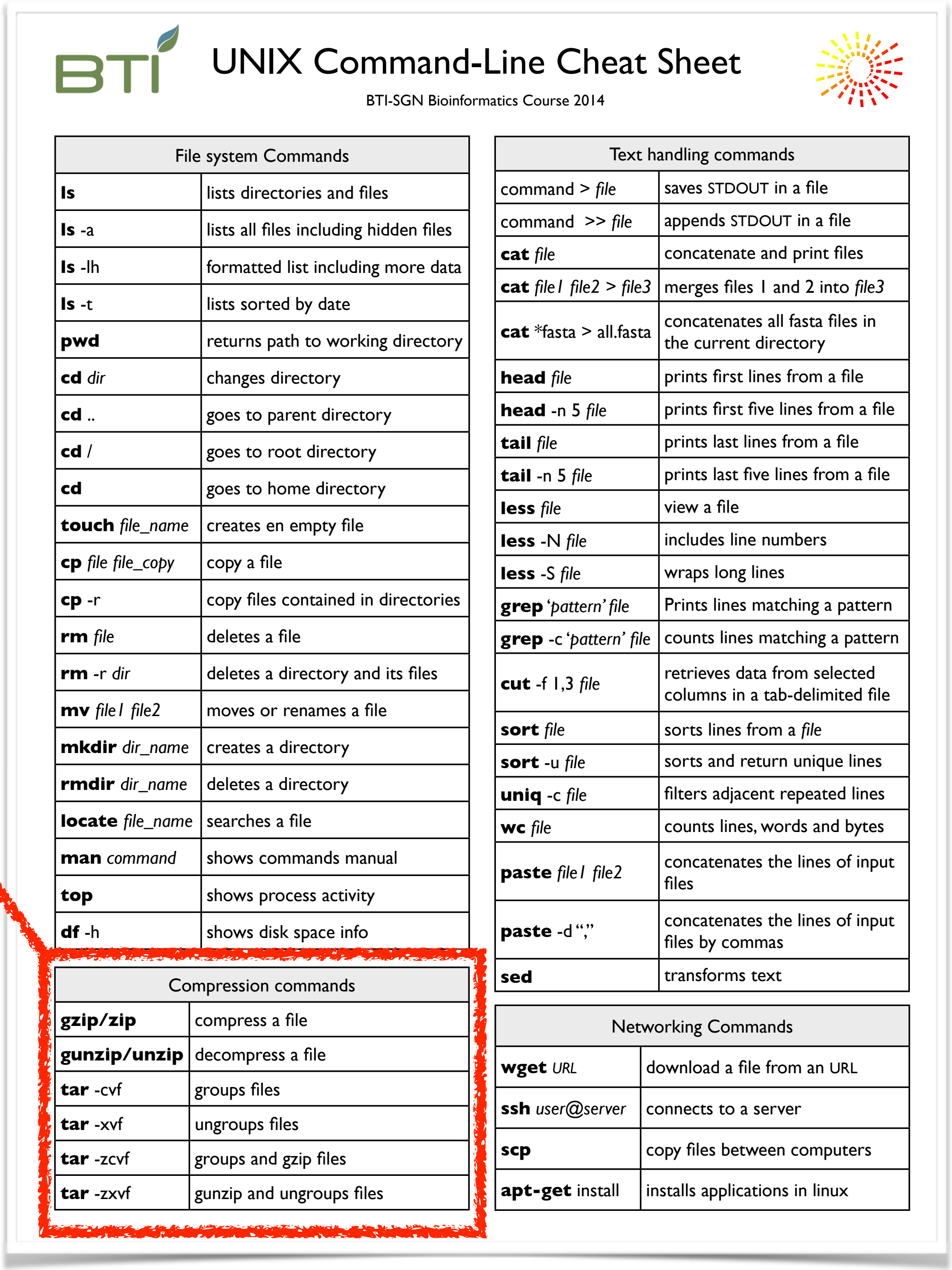
groups and compress files

```
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

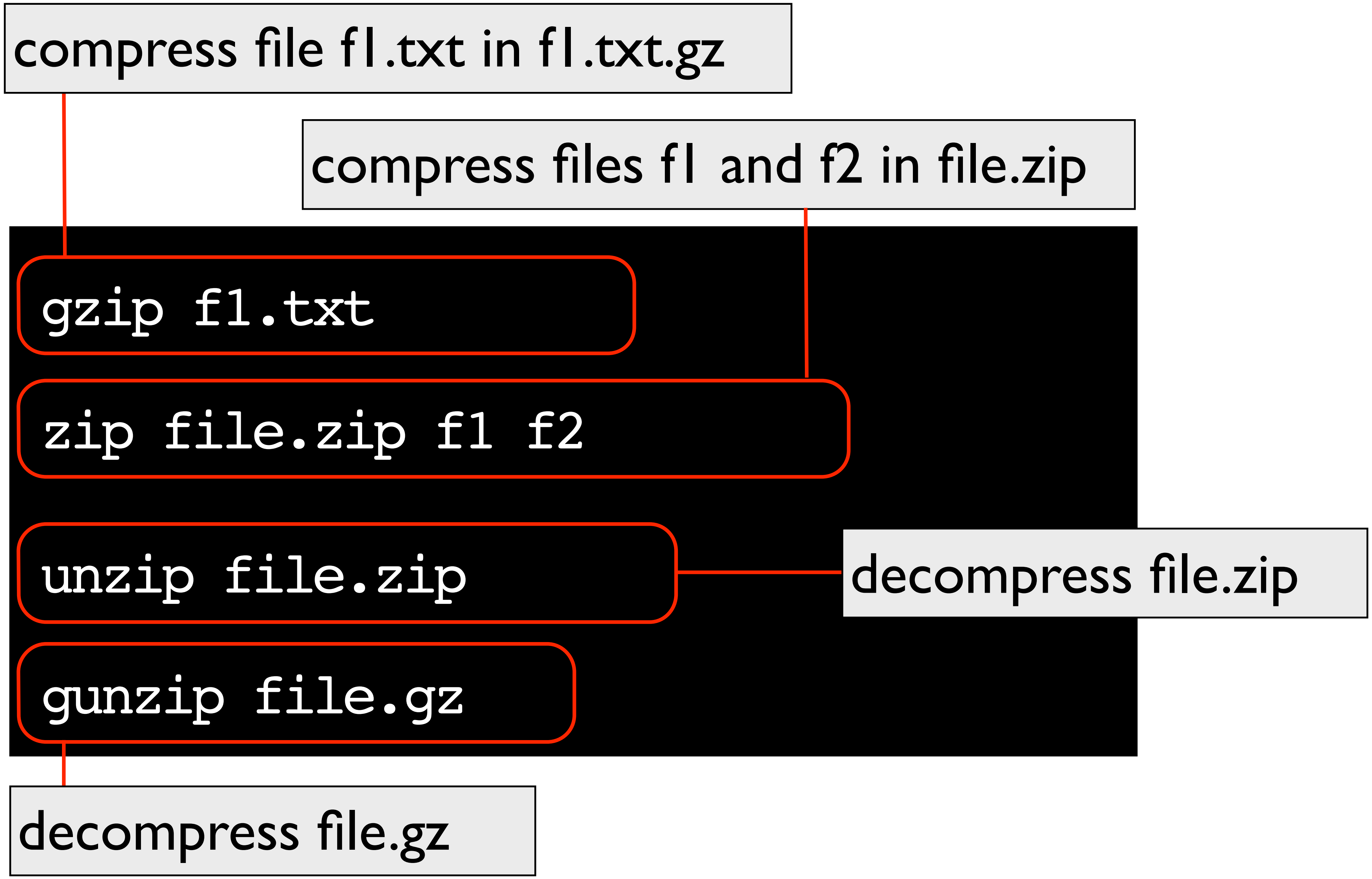
File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to 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

compress files f1 and f2 in file.zip

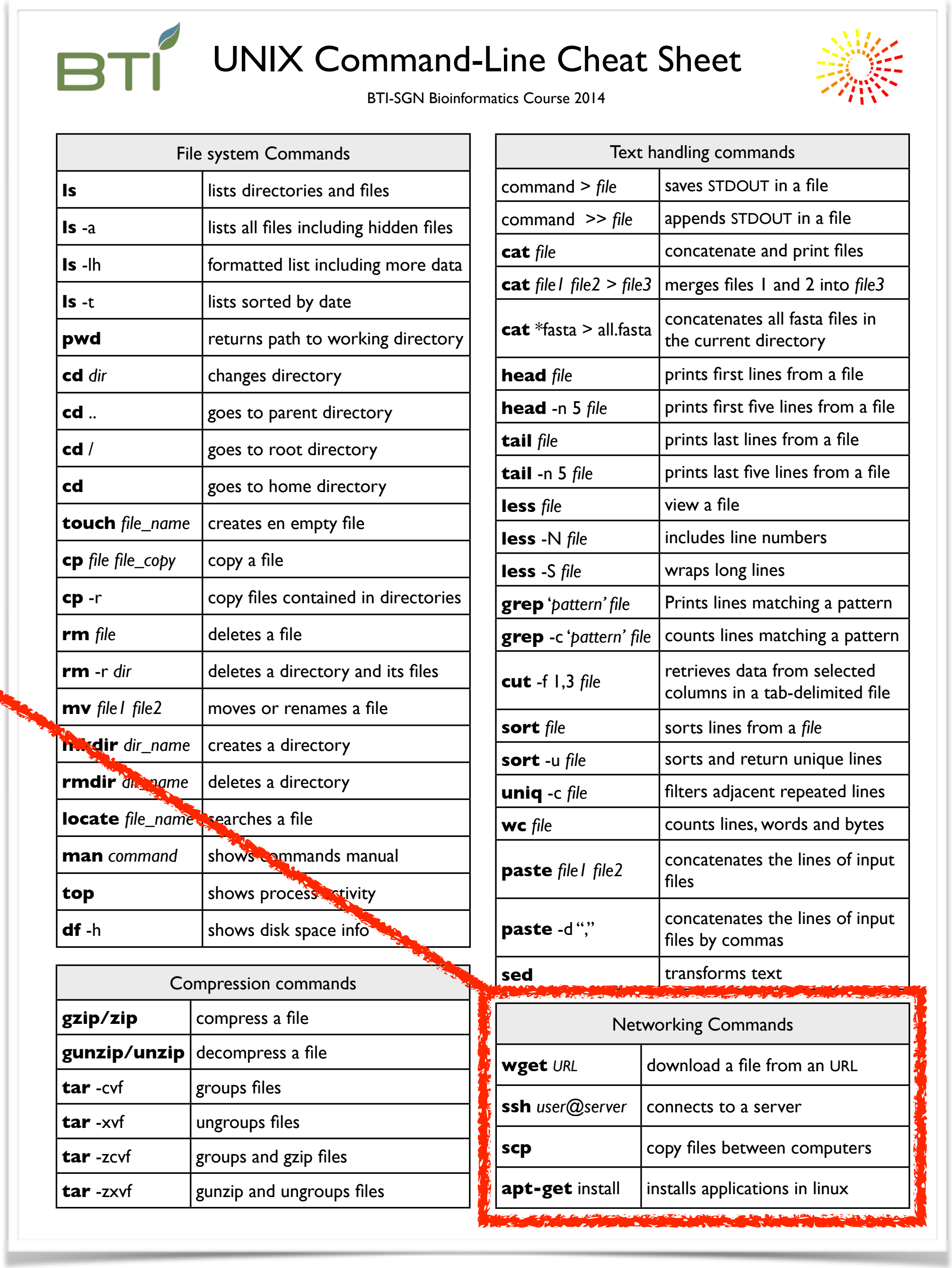
```
gzip f1.txt
zip file.zip f1 f2
unzip file.zip
gunzip file.gz
```

decompress file.zip

decompress file.gz

Networking Commands

- Networking commands



UNIX Command-Line Cheat Sheet

BTI-SGN Bioinformatics Course 2014

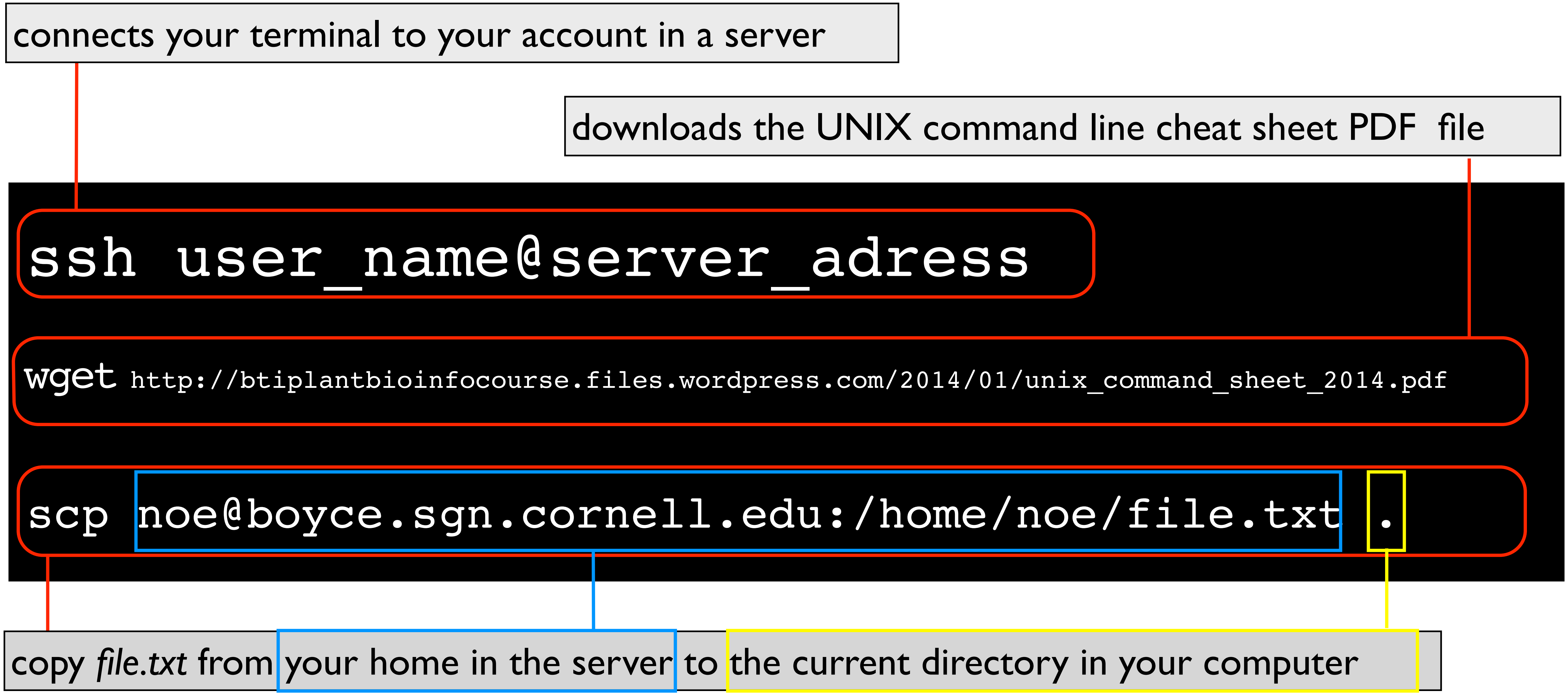
File system Commands	
ls	lists directories and files
ls -a	lists all files including hidden files
ls -lh	formatted list including more data
ls -t	lists sorted by date
pwd	returns path to working directory
cd *dir*	changes directory
cd ..	goes to parent directory
cd /	goes to root directory
cd	goes to 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



connects your terminal to your account in a server

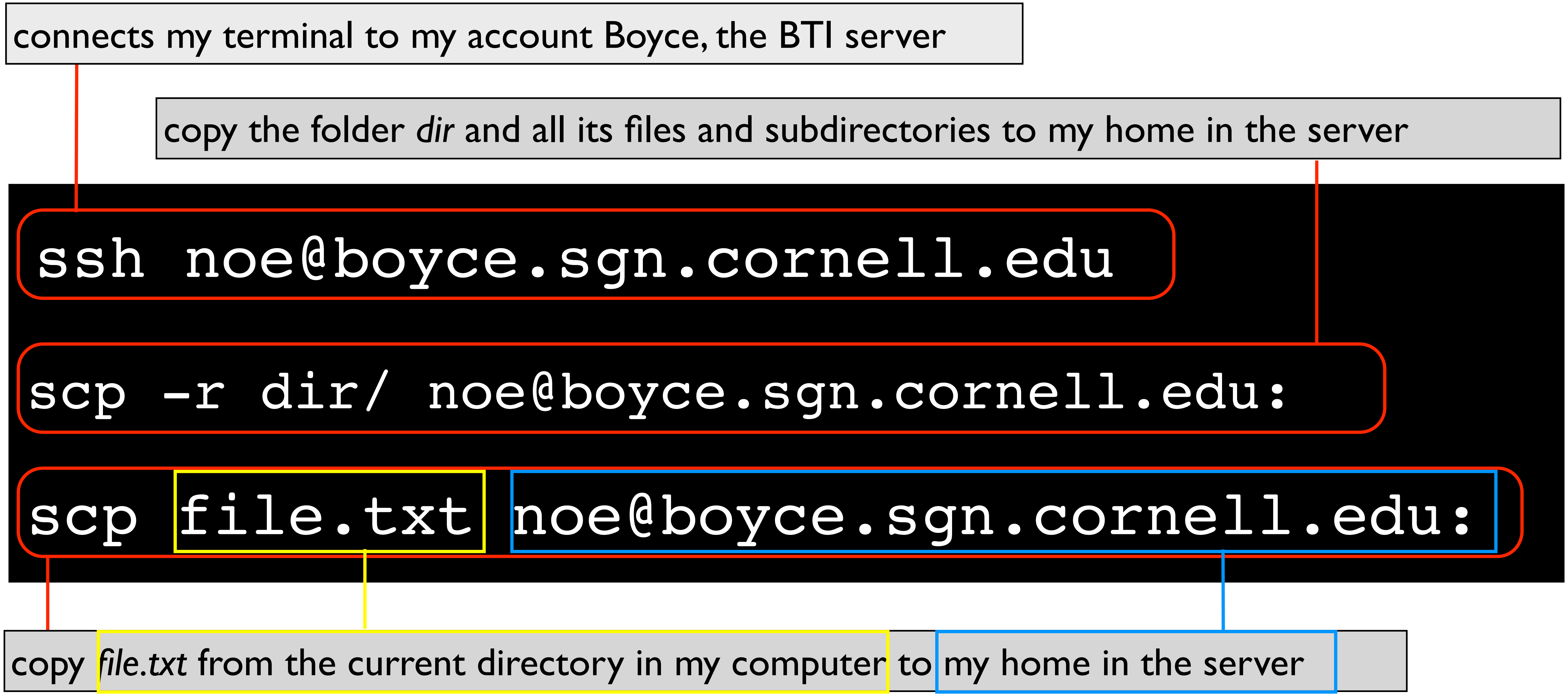
downloads the UNIX command line cheat sheet PDF file

```
ssh user_name@server_adress

wget http://btiplantbioinfocourse.files.wordpress.com/2014/01/unix_command_sheet_2014.pdf

scp noe@boyce.sgn.cornell.edu:/home/noe/file.txt .
```

copy *file.txt* from your home in the server to the current directory in your computer

Tip: use the command pwd to get the path for cp and scp

Networking Commands



connects my terminal to my account Boyce, the BTI server

copy the folder *dir* and all its files and subdirectories to my home in the server

```
ssh noe@boyce.sgn.cornell.edu

scp -r dir/ noe@boyce.sgn.cornell.edu:

scp file.txt noe@boyce.sgn.cornell.edu:
```

copy *file.txt* from the current directory in my computer to my home in the server

Exercises

1. Use the command mkdir to create a folder called unix_data in your desktop

2. Copy the file unix_class_file_samples.zip from your folder Data, in your home, to the folder unix_data, in your desktop

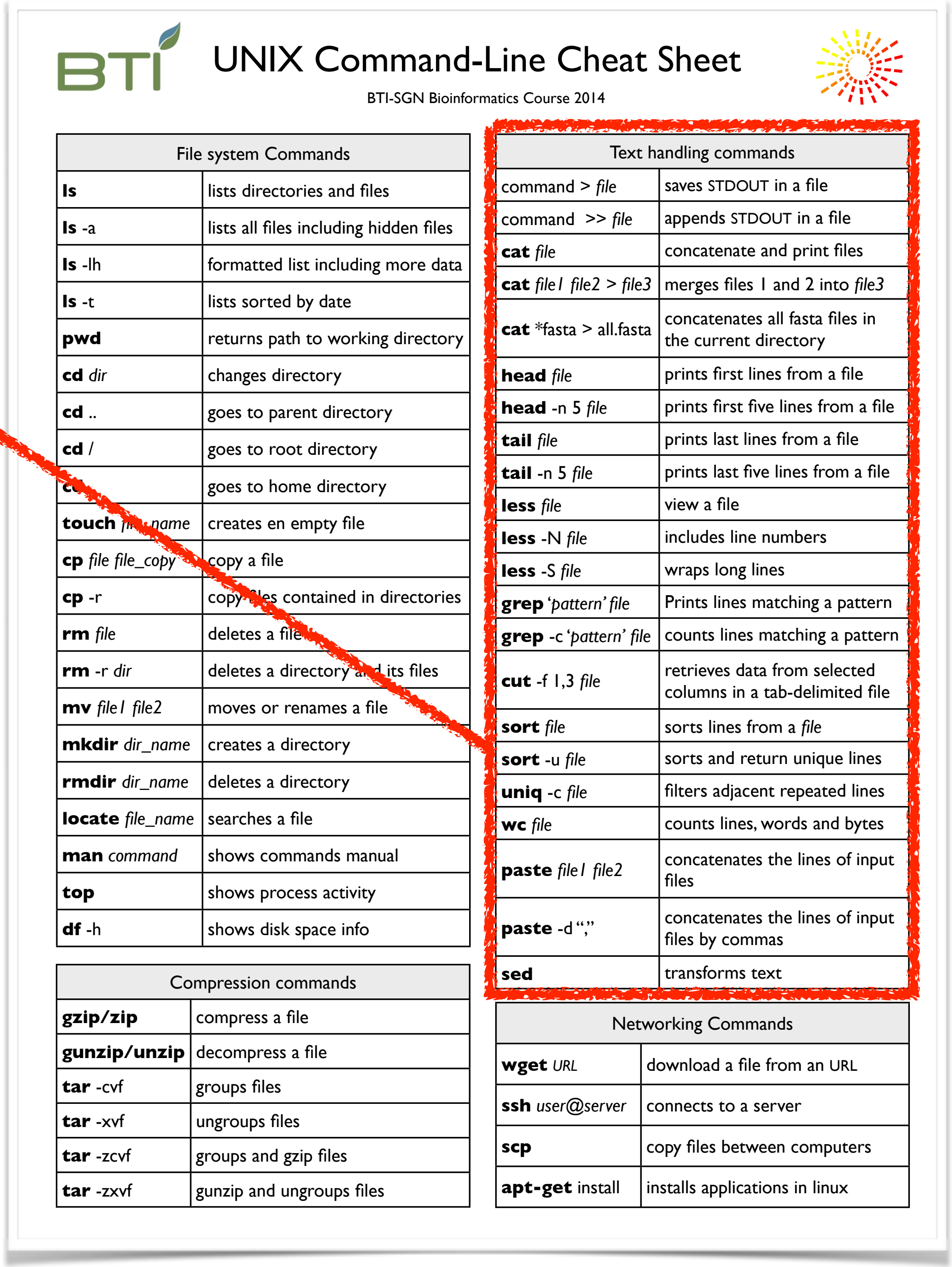
3. Uncompress the file unix_class_file_samples.zip in /home/bioinfo/Desktop/unix_data

4. Use the command wget to download the "UNIX command line cheat sheet" PDF from:

https://btiplantbioinfocourse.files.wordpress.com/2014/02/unix_command_sheet_2014.pdf

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

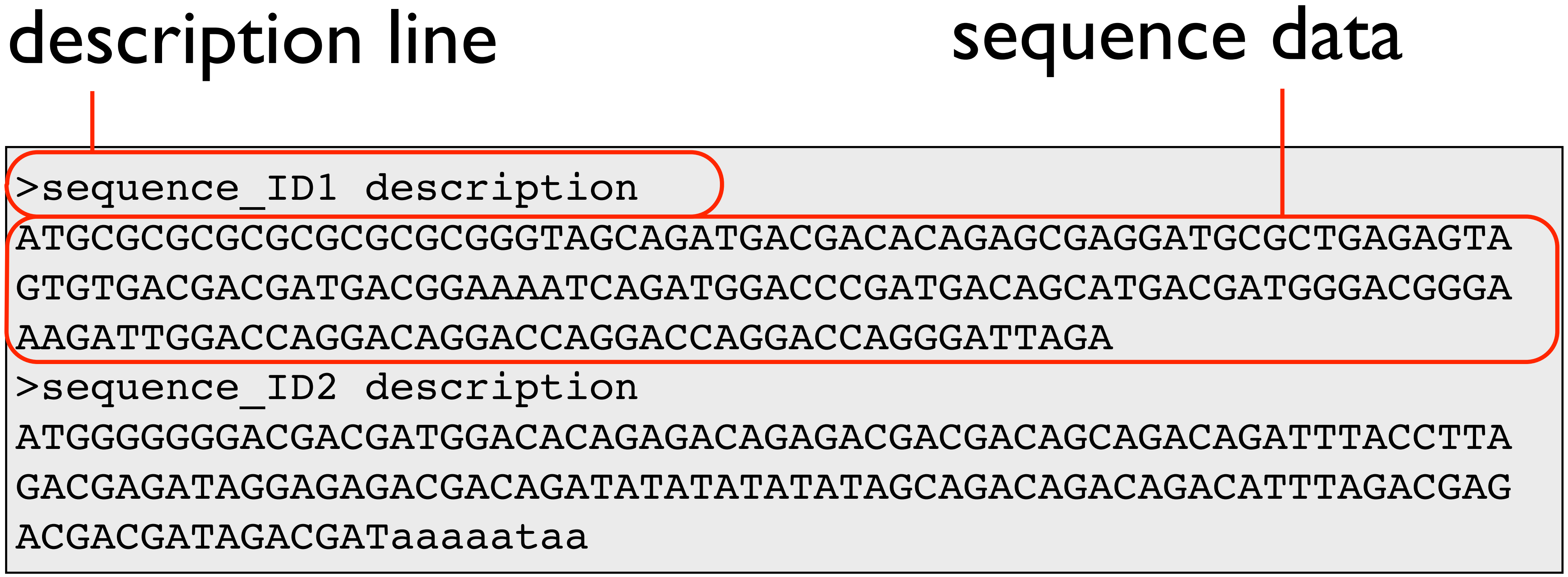
Text handling commands	
command > *file*	saves STDOUT in a file
command >> *file*	appends STDOUT in a file
cat *file*	concatenate and print files
cat *file1 file2 > file3*	merges files 1 and 2 into *file3*
cat *fasta > all.fasta	concatenates all fasta files in the current directory
head *file*	prints 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

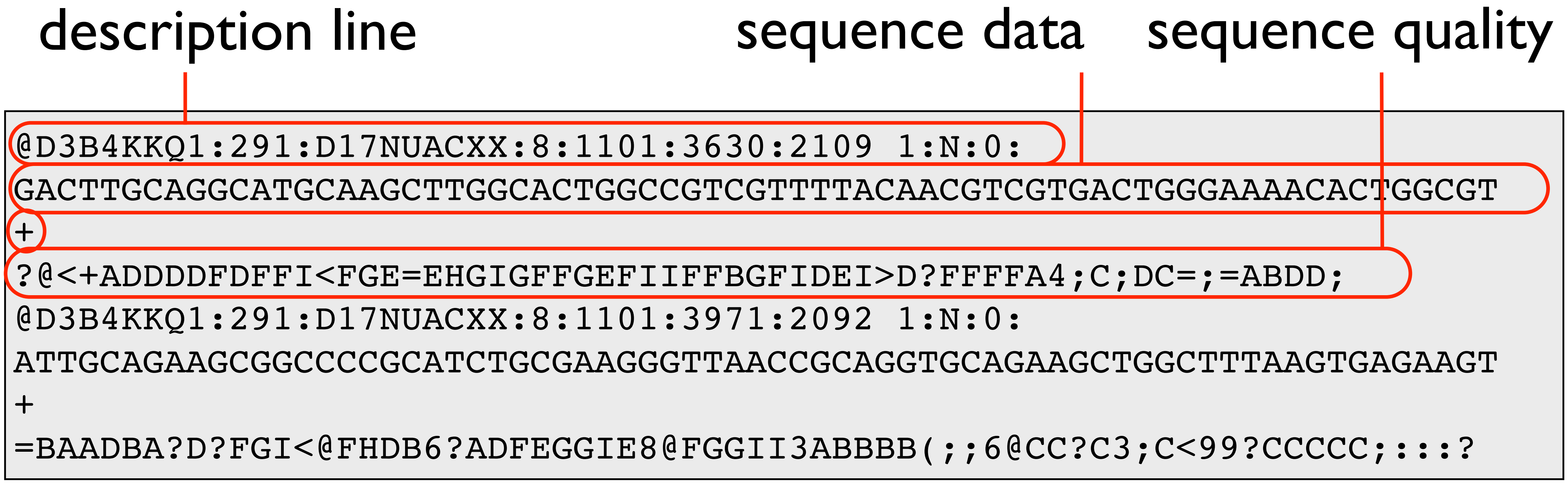
```
>sequence_ID1 description
ATGCGCGCGCGCGCGCGGGTAGCAGATGACGACACAGAGCGAGGATGCGCTGAGAGTA
GTGTGACGACGATGACGGAAAATCAGATGGACCCGATGACAGCATGACGATGGGACGGGA
AAGATTGGACCAGGACAGGACCAGGACCAGGACCAGGGATTAGA
>sequence_ID2 description
ATGGGGGGACGACGATGGACACAGAGACAGAGACGACGACAGCAGACAGATTTACCTTA
GACGAGATAGGAGAGACGACAGATATATATATATAGCAGACAGACAGACATTTAGACGAG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

FASTQ format

A FASTQ file normally uses four lines per sequence.

Line 1: begins with a '@' character, followed by a sequence identifier and an optional description.
Line 2: is the raw sequence letters.
Line 3: begins with a '+' character, is optionally followed by the same sequence identifier.
Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

wikipedia



description line

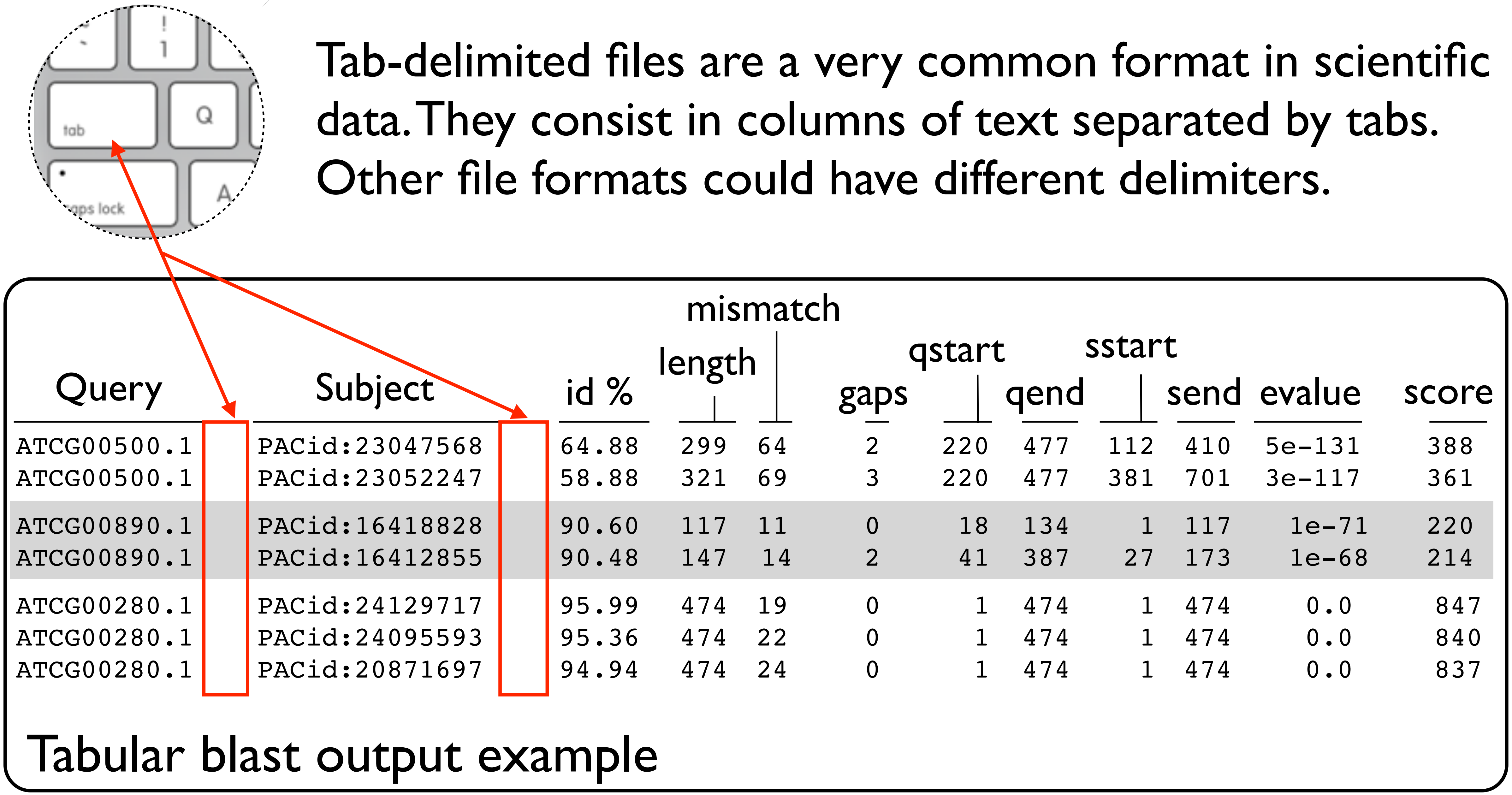
sequence data

sequence quality

```
@D3B4KKQ1:291:D17NUACXX:8:1101:3630:2109 1:N:0:
GACTTGCAGGCATGCAAGCTTGGCACTGGCCGTCGTTTTACAACGTCGTGACTGGGAAAACACTGGCGT
+
?@<+ADDDDFDFFI<FGE=EHGIGFFGEFIIFFBGFIDEI>D?FFFFA4;C;DC=;=ABDD;
@D3B4KKQ1:291:D17NUACXX:8:1101:3971:2092 1:N:0:
ATTGCAGAAGCGGCCCCGCATCTGCGAAGGGTTAACCGCAGGTGCAGAAGCTGGCTTTAAGTGAGAAGT
+
=BAADBA?D?FGI<@FHDB6?ADFEGGIE8@FGGII3ABBBB(;;6@CC?C3;C<99?CCCCC;:::?
```

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 ...

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*

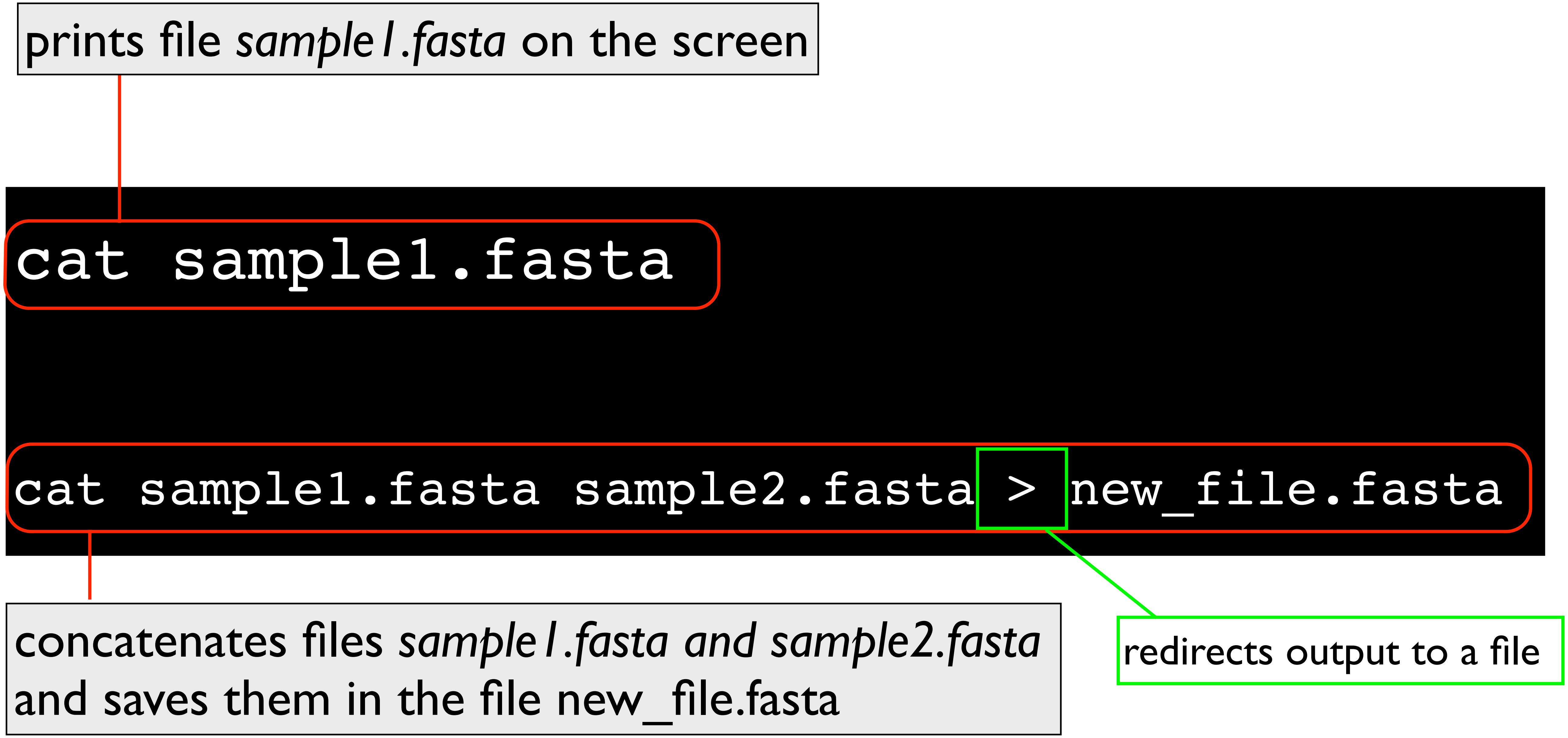
view file *blast_sample.txt*

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

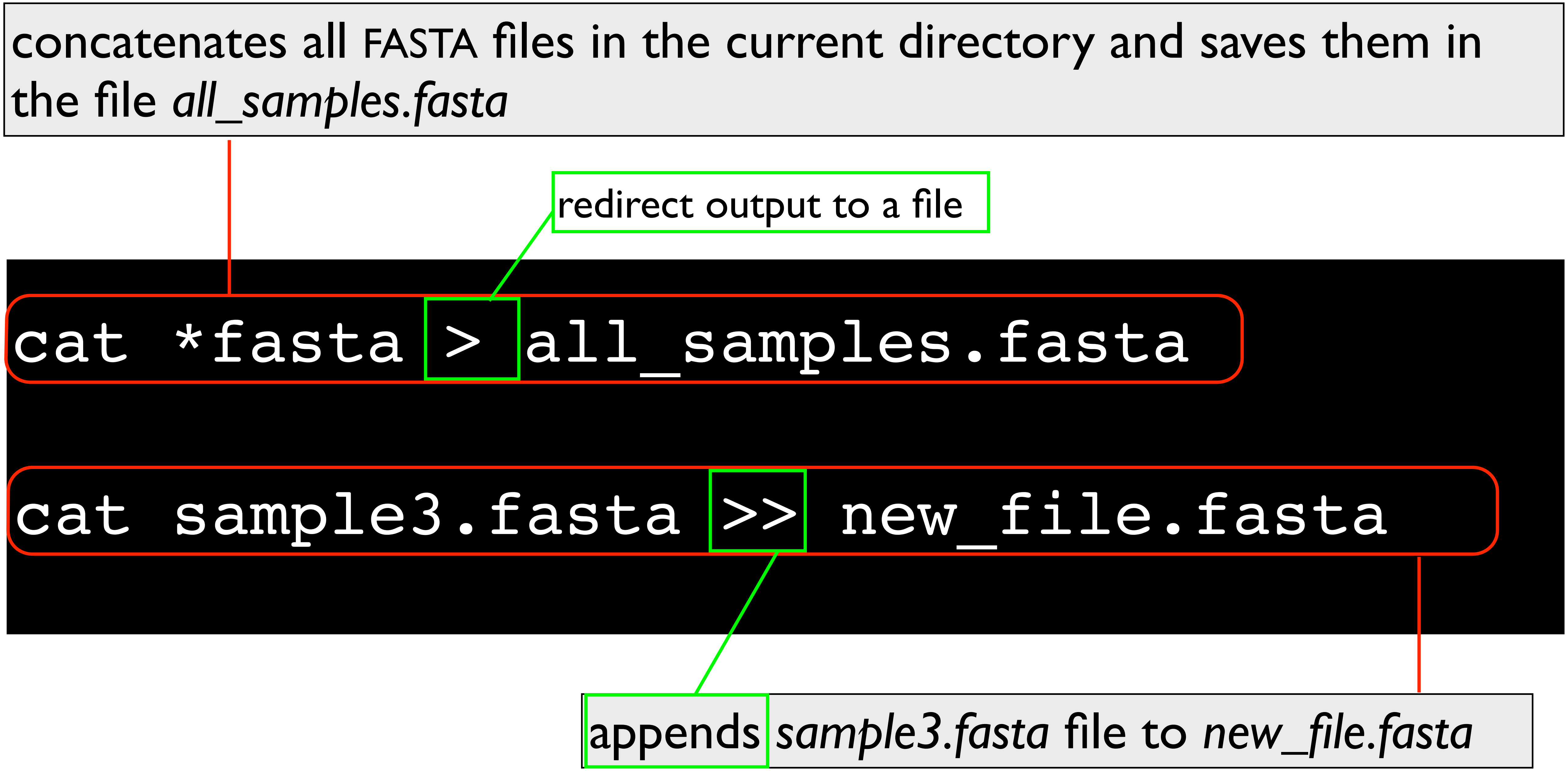
```
cat sample1.fasta
```

```
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



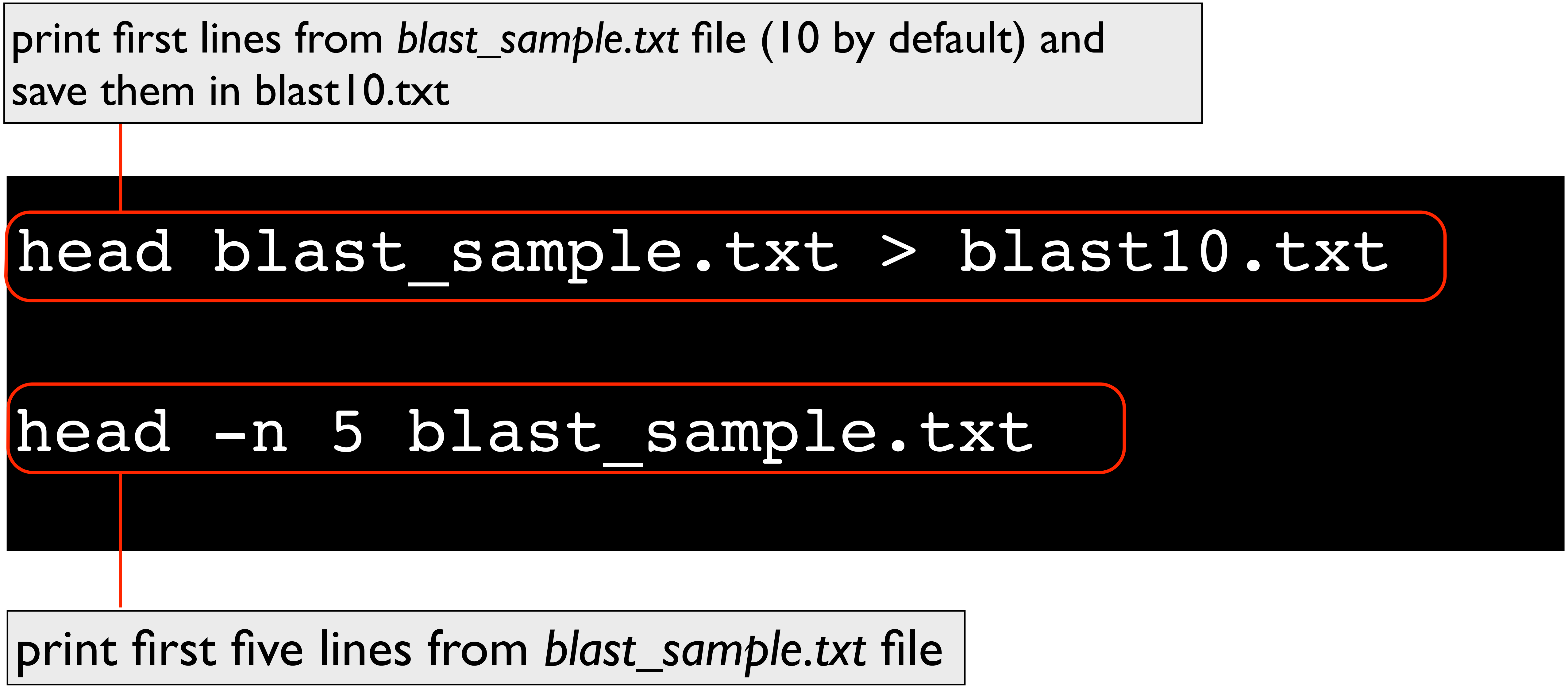
concatenates all FASTA files in the current directory and saves them in the file *all_samples.fasta*

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

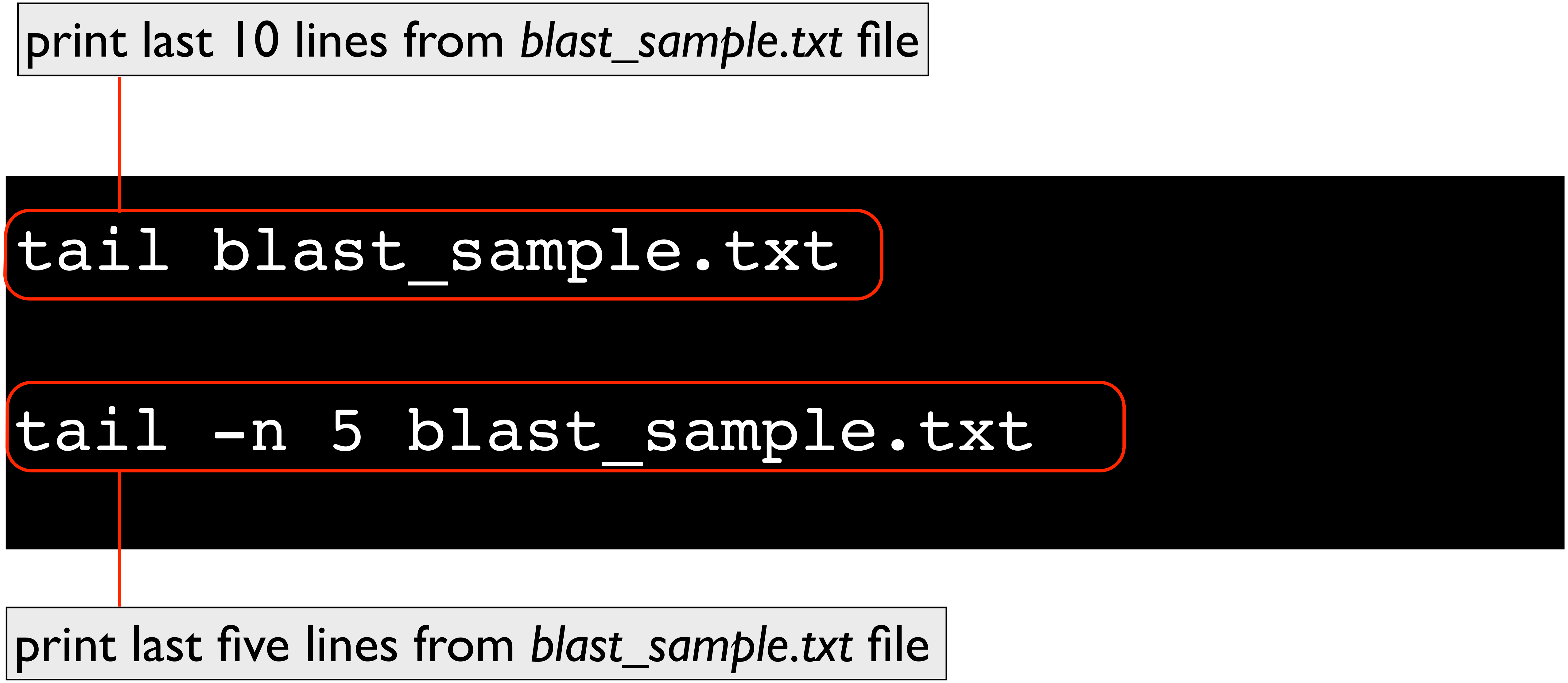


print first lines from *blast_sample.txt* file (10 by default) and save them in blast10.txt

```
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



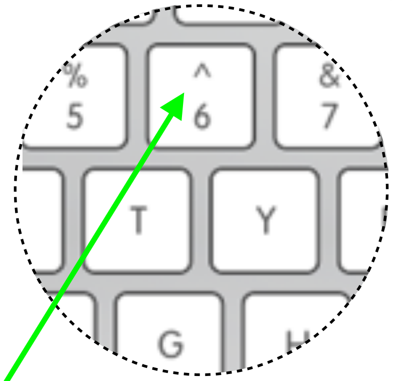
print last 10 lines from *blast_sample.txt* file

```
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

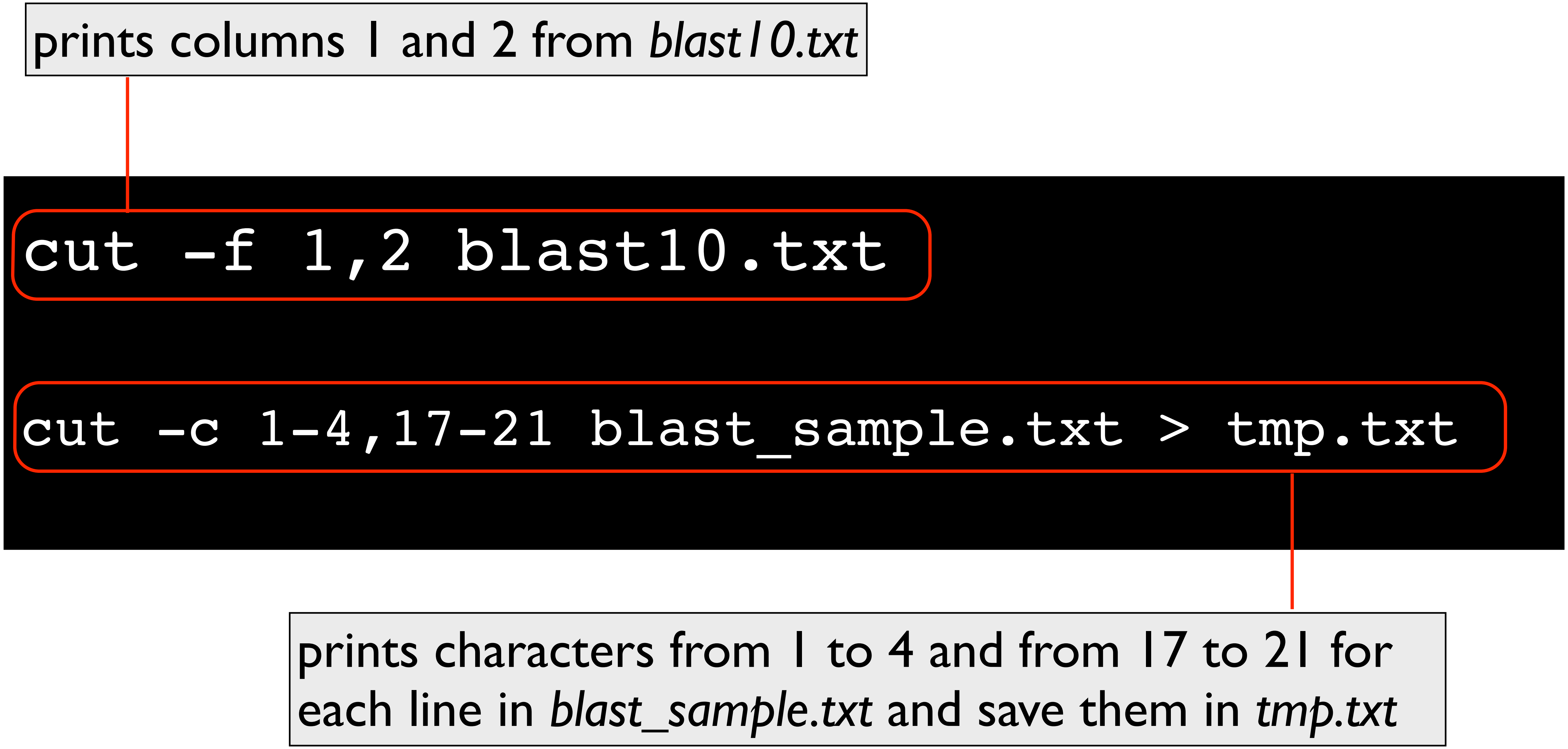
grep searches patterns in files

prints lines containing ‘Vvin’ and all their case combinations

```
grep -i ‘Vvin’ blast10.txt

grep -v ‘Vvin’ blast10.txt
```

prints all lines but the ones containing ‘Vvin’

cut gets columns from a tab-delimited file

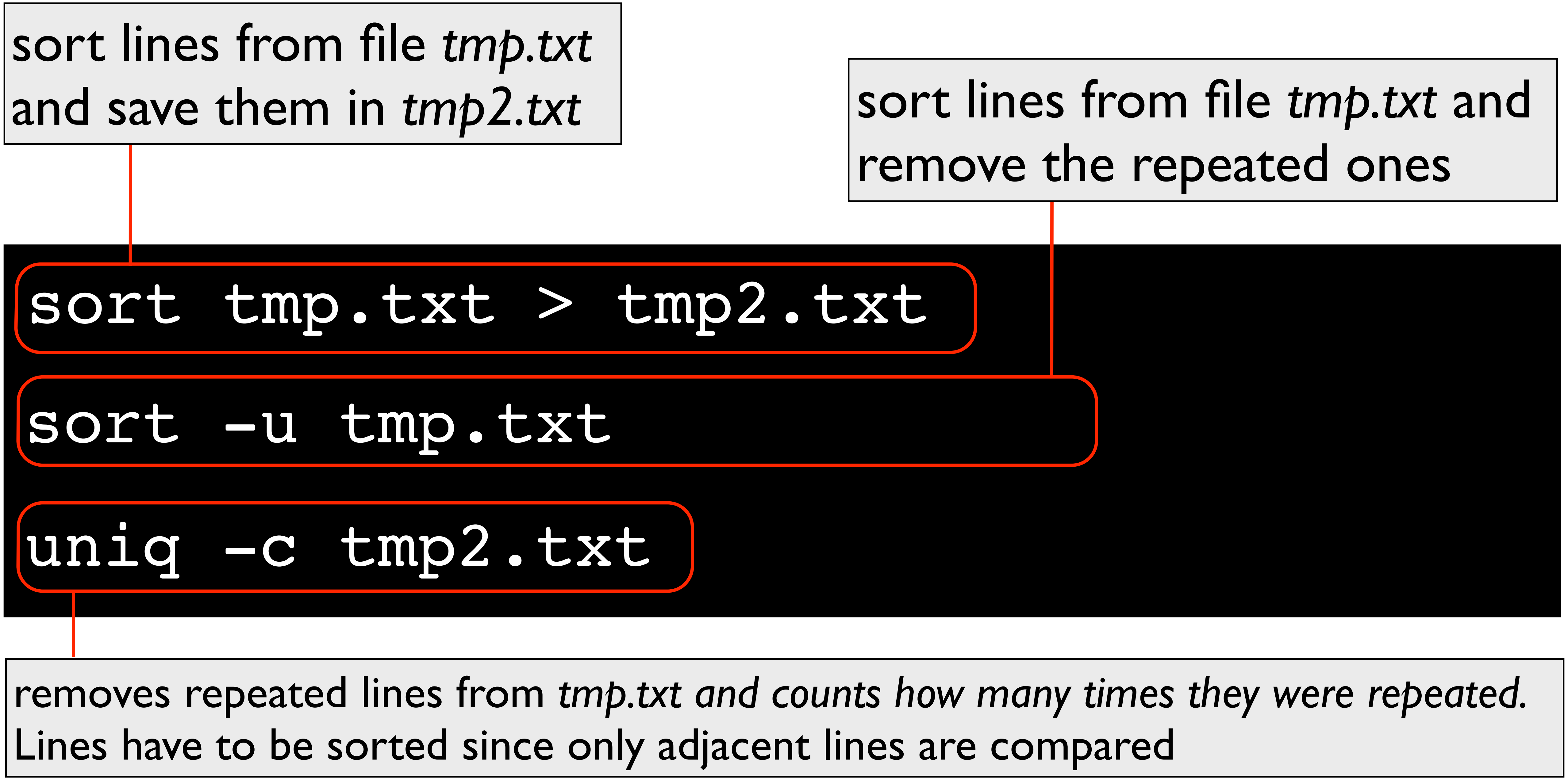


prints columns 1 and 2 from *blast10.txt*

```
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



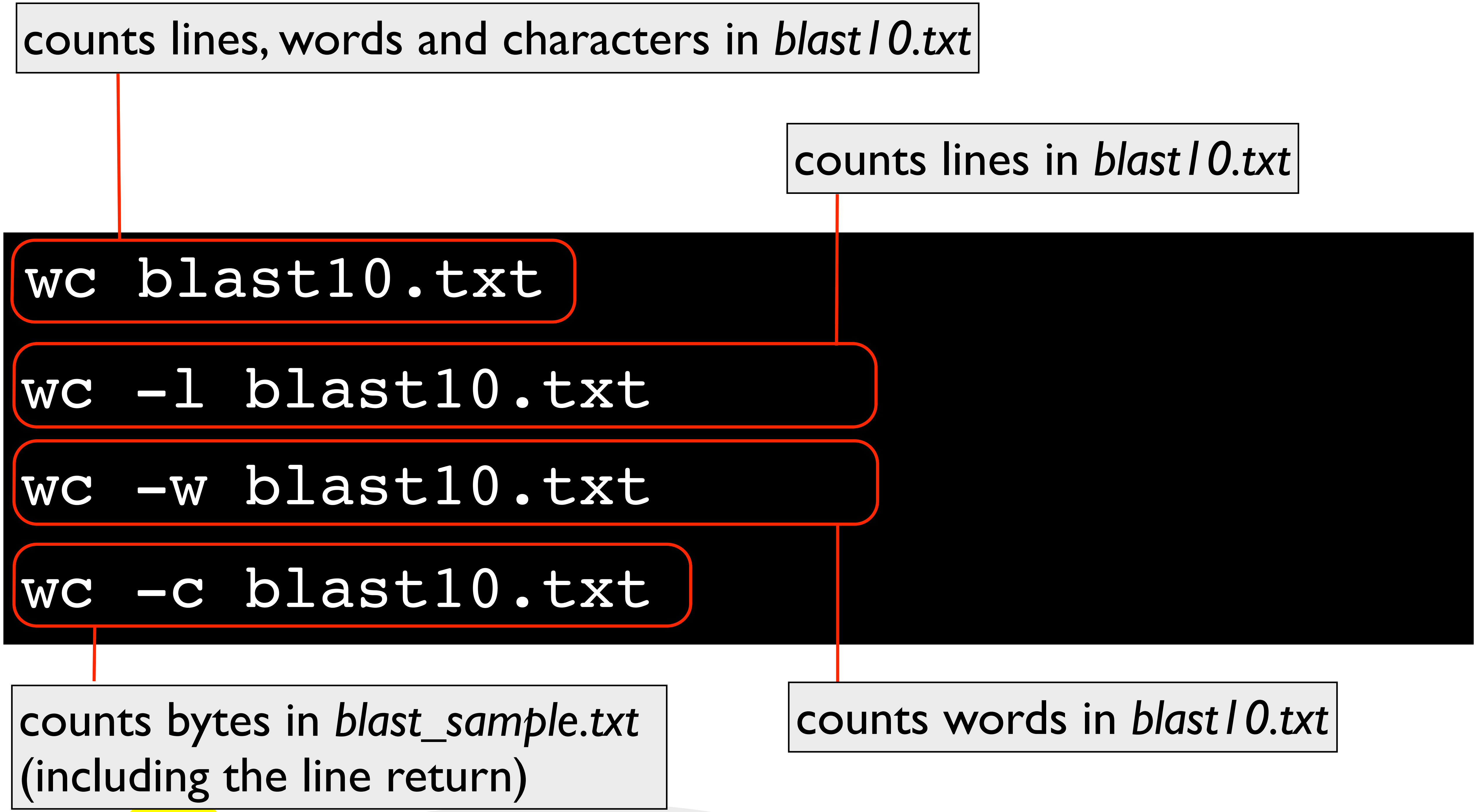
sort lines from file *tmp.txt* and save them in *tmp2.txt*

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*

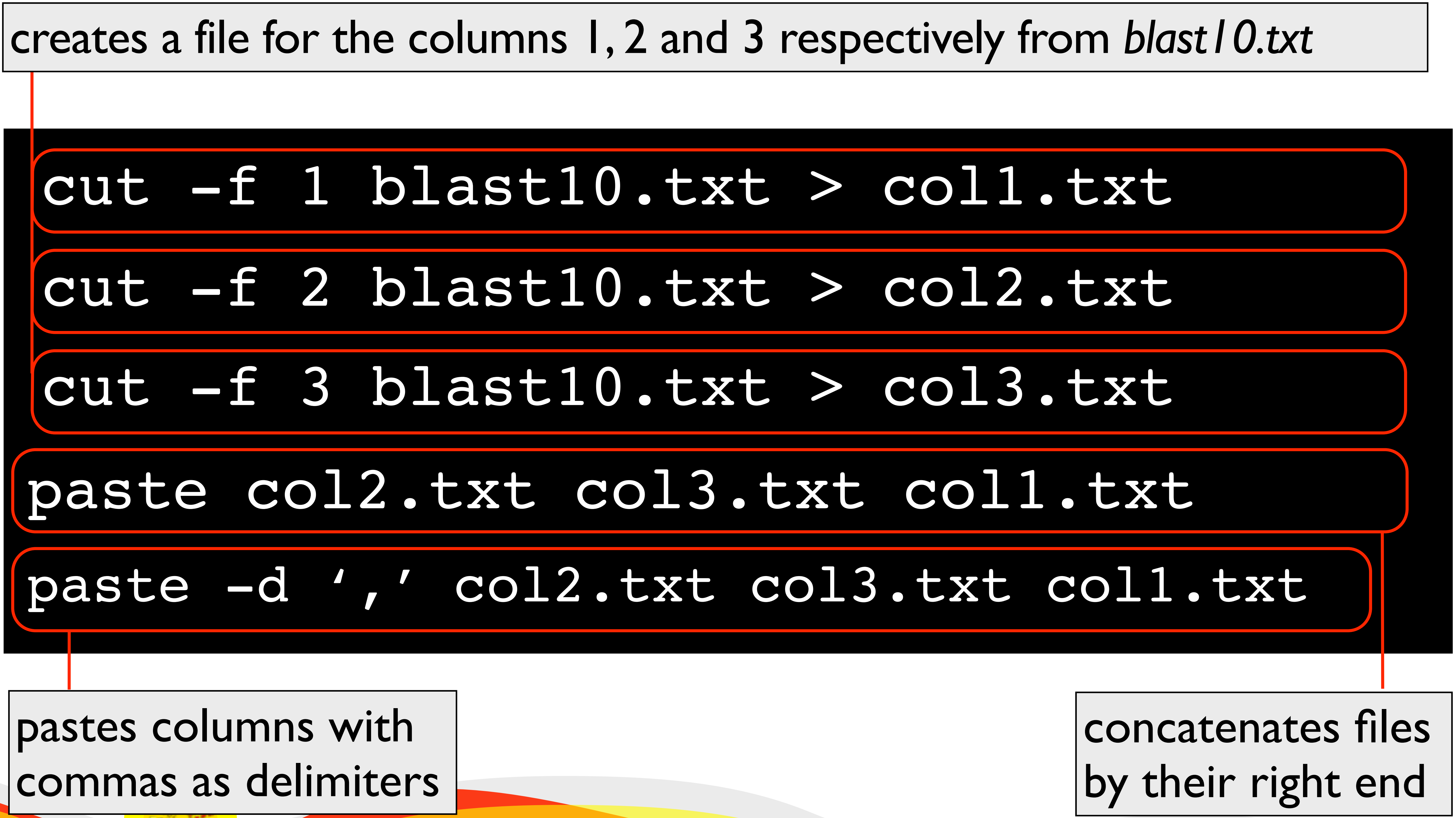
counts lines in *blast10.txt*

```
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*

paste concatenates files as columns



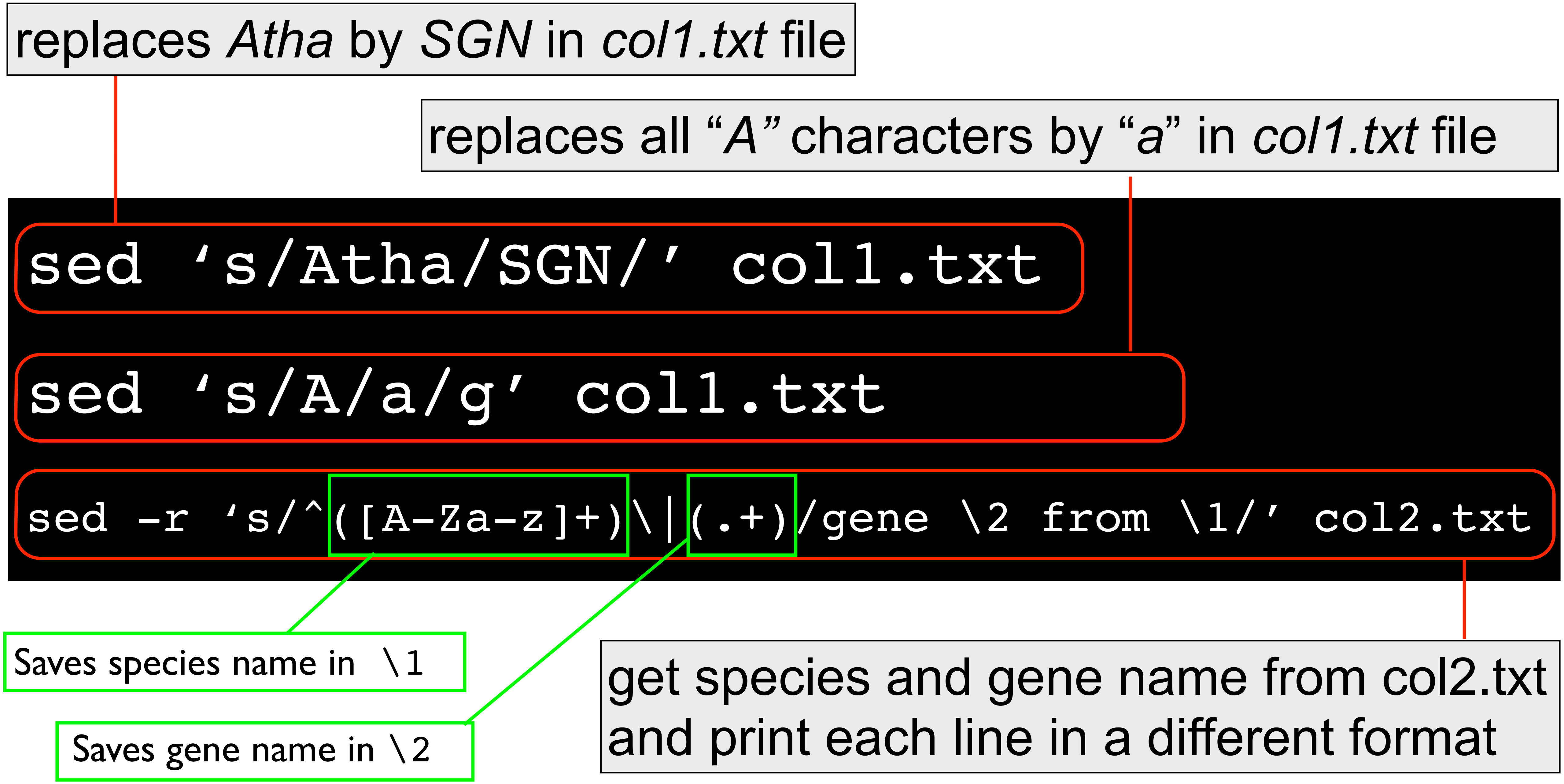
creates a file for the columns 1, 2 and 3 respectively from *blast10.txt*

```
cut -f 1 blast10.txt > col1.txt
cut -f 2 blast10.txt > col2.txt
cut -f 3 blast10.txt > col3.txt
paste col2.txt col3.txt col1.txt
paste -d ‘,’ col2.txt col3.txt col1.txt
```

pastes columns with commas as delimiters

concatenates files by their right end

sed replaces a pattern



replaces *Atha* by *SGN* in *col1.txt* file

replaces all “*A”* characters by “*a*” in *col1.txt* file

```
sed ‘s/Atha/SGN/’ col1.txt

sed ‘s/A/a/g’ col1.txt

sed -r ‘s/^([A-Za-z]+)\|(.+)/gene \2 from \1/’ col2.txt
```

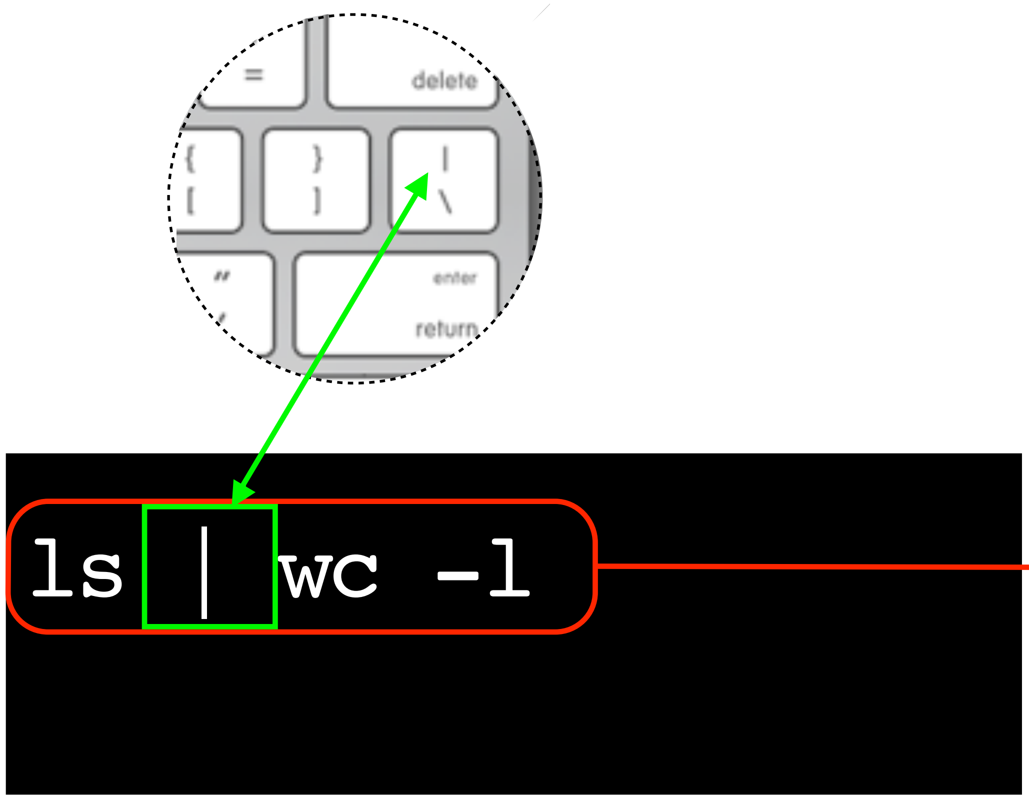
Saves species name in \1

Saves gene name in \2

get species and gene name from col2.txt and print each line in a different format

Pipelines

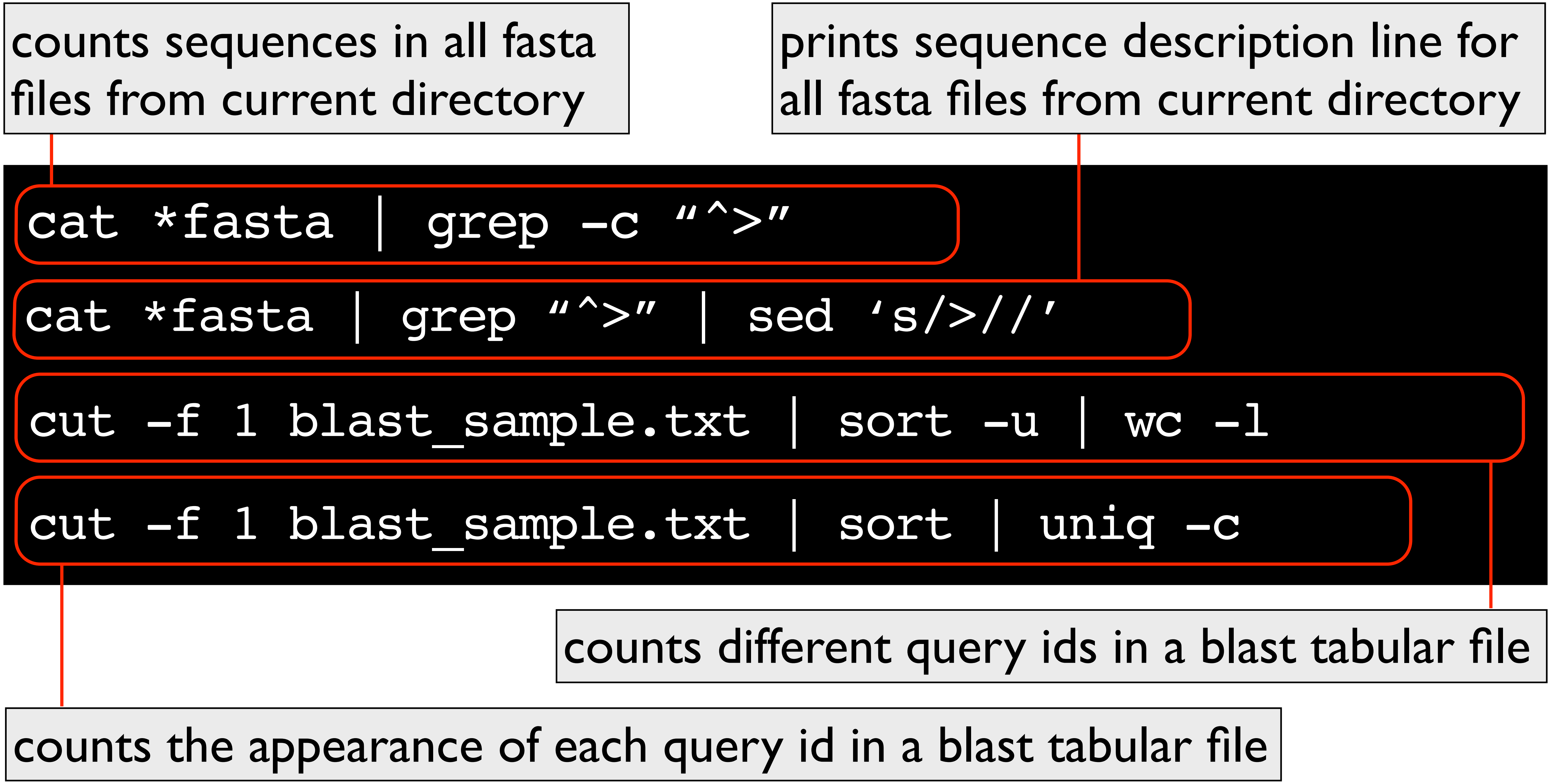
Pipelines consists in concatenate several commands by using the output of the first command as the input of the next one.
Two commands are connected placing the sign “|” between them.



```
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

Exercises

1. Merge the fasta files *sample1.fasta*, *sample2.fasta* and *sample3.fasta*, and save them in a new file called *all_samples.fasta*
2. How many sequences are in *all_samples.fasta*?
3. Save the first 100 lines from *blast_sample.txt* in a file called *blast100.txt*
4. 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

or

ftp://ftp.solgenomics.net/bioinfo_class/other/interns/2015/arabidopsis_proteins.fasta