Tema 1. Introducción al biocómputo en sistemas GNU/Linux

Pablo Vinuesa

2019-07-22

Contents

Tema 1. Introducción al biocómputo en sistemas GNU/Linux - Primer contacto

Este apunte fue creado para el Taller 3 - Análisis comparativo de genomas microbianos: Pangenómica y filoinformática de los Talleres Internacionales de Bioinformática - TIB2019, celebrados en el Centro de Ciencias Genómicas de la Universidad Nacional Autónoma de México, del 29 de julio al 2 de agosto de 2019 por Pablo Vinuesa, CCG-UNAM

version: 2019-07-22

Una vez que domines los comandos básicos que se presentarán seguidamente, recomiendo revisar tutoriales mucho más detallados y completos como los siguientes:

- Bash Reference Manual
- Advanced Bash Scripting Guide

Exploremos el sistema en el que estamos trabajando

ssh establecer sesion remota encriptada (segura) via ssh al servidor ivory

```
ssh -l $USER IP
```

hostname muestra el nombre del host (la máquina a la que estoy conectado) y la IP

```
hostname
hostname -i
```

```
## alisio
## 127.0.1.1
```

uname muestra el sistema operativo del host

```
uname
uname -a
```

```
## Linux
## Linux alisio 4.15.0-54-generic #58-Ubuntu SMP Mon Jun 24 10:55:24 UTC 2019 x86_64 x86_64 x86_64 GNU/
```

EXPLORACIÓN EL SISTEMA DE ARCHIVOS

pwd imprime la ruta absoluta del directorio actual

```
# dónde me encuentro en el sistema?
pwd
```

```
## /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux
```

ls lista contenidos del directorio

```
# Qué contiene el directorio actual?
ls

# mostrar todos (-a all) los archivos, incluidos los ocultos
ls -a
```

```
## assembly_summary.txt.gz
## empty_file
## github_TIB-filoinfo_screenshot.png
## intro2genomics
## intro_biocomputo_Linux_pt1.odp
## Intro_biocomputo_Linux_pt1.pdf
## linux_basic_commands.tab
## linux_commands.tab
## linux_very_basic_commands_table.csv
## sesion_local_capt_pantalla.png
## sesion_remota_bonampak_capt_pantalla1.png
## sesion_remota_bonampak_capt_pantalla2.png
## sesion_remota_bonampak_capt_pantalla.png
## Stenotrophomonas_complete_genomes_and_ftp_paths.txt
## working_with_linux_commands.code
## working_with_linux_commands.html
## working_with_linux_commands.Rmd
## .
## ..
```

```
## assembly_summary.txt.gz
## empty_file
## github_TIB-filoinfo_screenshot.png
## intro2genomics
## intro_biocomputo_Linux_pt1.odp
## Intro_biocomputo_Linux_pt1.pdf
## linux_basic_commands.tab
## linux_commands.tab
## .linux_commands.tab.swp
## linux_very_basic_commands_table.csv
## .Rhistory
## sesion_local_capt_pantalla.png
## sesion_remota_bonampak_capt_pantalla1.png
## sesion_remota_bonampak_capt_pantalla2.png
## sesion_remota_bonampak_capt_pantalla.png
## Stenotrophomonas_complete_genomes_and_ftp_paths.txt
## working_with_linux_commands.code
## working_with_linux_commands.html
## working_with_linux_commands.Rmd
```

Veamos el contenido del directorio raiz

```
ls /
```

```
## bin
## boot
## cdrom
## dev
## etc
## home
## initrd.img
## initrd.img.old
## lib
## lib32
## lib64
## lost+found
## media
## mnt
## opt
## proc
## root
## run
## sbin
## snap
## srv
## swapfile
## sys
## tmp
## usr
## var
## vmlinuz
## vmlinuz.old
```

Veamos el contenido del directorio /bin

```
ls /bin | head -20
```

```
## bash
## brltty
## bunzip2
## busybox
## bzcat
## bzcmp
## bzdiff
## bzegrep
## bzexe
## bzfgrep
## bzgrep
## bzip2
## bzip2recover
## bzless
## bzmore
## cat
## chacl
## chgrp
## chmod
## chown
```

```
# idem, pero con detalles de permisos etc

ls -l /bin | head -20
```

```
## total 12480
## -rwxr-xr-x 1 root root 1113504 jun  6 17:28 bash
## -rwxr-xr-x 1 root root  748968 ago 29  2018 brltty
## -rwxr-xr-x 3 root root   34888 jul  4 07:35 bunzip2
## -rwxr-xr-x 1 root root 2062296 mar  6 14:51 busybox
## -rwxr-xr-x 3 root root   34888 jul  4 07:35 bzcat
## lrwxrwxrwx 1 root root       6 jul  4 07:35 bzcmp -> bzdiff
## -rwxr-xr-x 1 root root    2140 jul  4 07:35 bzdiff
## lrwxrwxrwx 1 root root       6 jul  4 07:35 bzegrep -> bzgrep
## -rwxr-xr-x 1 root root    4877 jul  4 07:35 bzexe
## lrwxrwxrwx 1 root root       6 jul  4 07:35 bzfgrep -> bzgrep
## -rwxr-xr-x 1 root root    3642 jul  4 07:35 bzgrep
## -rwxr-xr-x 3 root root   34888 jul  4 07:35 bzip2
## -rwxr-xr-x 1 root root   14328 jul  4 07:35 bzip2recover
## lrwxrwxrwx 1 root root       6 jul  4 07:35 bzless -> bzmore
## -rwxr-xr-x 1 root root    1297 jul  4 07:35 bzmore
## -rwxr-xr-x 1 root root   35064 ene 18  2018 cat
## -rwxr-xr-x 1 root root   14328 abr 21  2017 chacl
## -rwxr-xr-x 1 root root   63672 ene 18  2018 chgrp
## -rwxr-xr-x 1 root root   59608 ene 18  2018 chmod
```

```
# idem, pero ordenando los archivos por fechas de modificacion (-t), listando los mas recientes al fina
ls -ltr /bin | head -20
```

```
## total 12480
## -rwxr-xr-x 1 root root      89 abr 26  2016 red
## -rwxr-xr-x 1 root root   51512 abr 26  2016 ed
## -rwxr-xr-x 1 root root   14328 ago 11  2016 ulockmgr_server
## -rwsr-xr-x 1 root root   30800 ago 11  2016 fusermount
```

```
## -rwsr-xr-x 1 root root   64424 mar  9  2017 ping
## -rwxr-xr-x 1 root root   40056 abr 21  2017 efibootmgr
## -rwxr-xr-x 1 root root   18424 abr 21  2017 efibootdump
## -rwxr-xr-x 1 root root   35512 abr 21  2017 setfacl
## -rwxr-xr-x 1 root root   23160 abr 21  2017 getfacl
## -rwxr-xr-x 1 root root   14328 abr 21  2017 chacl
## -rwxr-xr-x 1 root root    5047 abr 27  2017 znew
## -rwxr-xr-x 1 root root    1910 abr 27  2017 zmore
## -rwxr-xr-x 1 root root    2037 abr 27  2017 zless
## -rwxr-xr-x 1 root root    5938 abr 27  2017 zgrep
## -rwxr-xr-x 1 root root    2131 abr 27  2017 zforce
## -rwxr-xr-x 1 root root     140 abr 27  2017 zfgrep
## -rwxr-xr-x 1 root root     140 abr 27  2017 zegrep
## -rwxr-xr-x 1 root root    5764 abr 27  2017 zdiff
## -rwxr-xr-x 1 root root    1777 abr 27  2017 zcmp
```

Expansión de caracteres con * y ?

```
# lista los archivos en /bin que empiezan por las letras b y c
ls /bin/b*
ls /bin/c*
```

```
## /bin/bash
## /bin/brltty
## /bin/bunzip2
## /bin/busybox
## /bin/bzcat
## /bin/bzcmp
## /bin/bzdiff
## /bin/bzegrep
## /bin/bzexe
## /bin/bzfgrep
## /bin/bzgrep
## /bin/bzip2
## /bin/bzip2recover
## /bin/bzless
## /bin/bzmore
## /bin/cat
## /bin/chacl
## /bin/chgrp
## /bin/chmod
## /bin/chown
## /bin/chvt
## /bin/cp
## /bin/cpio
```

```
# lista los archivos en /bin que empiezan por la letra c seguida de uno o dos caracteres más
ls /bin/c?
ls /bin/c??
```

```
## /bin/cp
## /bin/cat
```

NAVEGAR EL SISTEMA DE ARCHIVOS: comando cd

de nuevo, ¿dónde estoy?

```
pwd
```

```
## /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux
```

sube un directorio usando RUTA RELATIVA

```
cd ..
```

donde estoy?

```
pwd
```

```
## /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux
```

regresa a tu home

```
cd $HOME

# que es equivalente a:
cd
```

cd cambiar directorios con rutas absolutas (/ruta/completa/al/dir) y relativas ../../

```
# a dónde nos lleva este comando?
cd /
```

- cambia de nuevo a tu home

```
cd

# o tambien usando la variable de ambiente $HOME
cd $HOME
```

- sube al directorio home/ usando la ruta relativa

```
cd ../
```

GENERACIÓN DE DIRECTORIOS: comando mkdir

```
# vamos a $HOME y generamos el directorio intro2genomics
cd
if [ -d intro2genomics ]; then
    echo "found dir intro2genomics"
else
    mkdir intro2genomics
fi
```

```
## found dir intro2genomics
```

- comprueba los **permisos** del nuevo directorio

```
ls -l
```

```
## total 11776
## -rw-r--r-- 1 vinuesa vinuesa 6780296 jul 21 19:26 assembly_summary.txt.gz
## -rw-r--r-- 1 vinuesa vinuesa       0 jul 22 21:43 empty_file
## -rw-r--r-- 1 vinuesa vinuesa   87065 jul 22 20:59 github_TIB-filoinfo_screenshot.png
## drwxr-xr-x 3 vinuesa vinuesa    4096 jul 22 21:23 intro2genomics
## -rw-r--r-- 1 vinuesa vinuesa 2186396 jul 22 21:11 intro_biocomputo_Linux_pt1.odp
## -rw-r--r-- 1 vinuesa vinuesa 1692567 jul 22 21:11 Intro_biocomputo_Linux_pt1.pdf
## -rwxr-xr-x 1 vinuesa vinuesa   10193 jul 21 11:21 linux_basic_commands.tab
## lrwxrwxrwx 1 vinuesa vinuesa      78 jul 22 21:43 linux_commands.tab -> /home/vinuesa/Cursos/TIB/TIB
## -rwxr-xr-x 1 vinuesa vinuesa    1705 jul 21 11:21 linux_very_basic_commands_table.csv
## -rw-rw-r-- 1 vinuesa vinuesa   51271 jul 22 18:55 sesion_local_capt_pantalla.png
## -rw-r--r-- 1 vinuesa vinuesa   40746 jul 22 18:56 sesion_remota_bonampak_capt_pantalla1.png
## -rw-r--r-- 1 vinuesa vinuesa    4468 jul 22 18:57 sesion_remota_bonampak_capt_pantalla2.png
## -rw-rw-r-- 1 vinuesa vinuesa  408580 jul 22 18:51 sesion_remota_bonampak_capt_pantalla.png
## -rw-r--r-- 1 vinuesa vinuesa   47651 jul 22 21:43 Stenotrophomonas_complete_genomes_and_ftp_paths.tx
## -rwxr-xr-x 1 vinuesa vinuesa    6047 jul 21 11:21 working_with_linux_commands.code
## -rw-r--r-- 1 vinuesa vinuesa  684706 jul 22 21:43 working_with_linux_commands.html
## -rw-r--r-- 1 vinuesa vinuesa   19004 jul 22 21:43 working_with_linux_commands.Rmd
```

- generemos un subdirectorio por debajo del que acabamos de crear:

```
mkdir -p intro2genomics/sesion1_linux && cd intro2genomics/sesion1_linux
```

permisos

- cambiamos a /home/vinuesa e intenta crear estos mismos directorios ahí

```
cd /home/vinuesa && mkdir -p intro2genomics/sesion1_linux
```

copiar, mover, renombrar y borrar archivos con: cp, mv y rm

```
# cambia a tu home, y luego a intro2genomics/sesion1_linux
cd && cd intro2genomics/sesion1_linux
```

- copia el archivo /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab al directorio actual

```
cp /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab . # <<< vean el punto
```

- otra manera, usando rutas absolutas y la variable de ambiente $HOME

```
cp /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab $HOME/intro2genomics/
```

- copiar el directorio /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/ a tu dir actual

```
# Noten el punto '.' y cp -r (recursively), necesario para copiar directorios completos
cp -r /home/vinuesa/cursos/intro2genomics/sesion1_linux/data .
```

Eliminar un directorio: comando rm -rf [recursively -r and force -f]

```
mkdir borrame

cp linux_basic_commands.tab borrame

ls borrame

rm -rf borrame
```

```
## linux_basic_commands.tab
```

GENERACIÓN DE LIGAS SIMBÓLICAS: comando ln -s /ruta/al/archivo/fuente nombre_la_liga

Esto es muy importante, ya que permite ahorrar mucho espacio en disco al evitar la multiplicación de copias fisicas en el disco duro del mismo archivo

```
hostn=$(hostname)

if [ "$hostn" == "Tenerife" ]; then
   ln -s /home/vinuesa/Cursos/OMICAS_UAEM_genomica/clase1_intro2linux/linux_basic_commands.tab comandos
elif [ "$hostn" == "buluc" ]; then
     ln -s /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab comandos_de_l
elif [ "$hostn" == "alisio" ]; then
   ln -s /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux/linux_basic_commands.tab comandos_de_linu
fi

# confirmamos que se genero la liga
ls -l
```

```
## total 11780
## -rw-r--r-- 1 vinuesa vinuesa 6780296 jul 21 19:26 assembly_summary.txt.gz
## lrwxrwxrwx 1 vinuesa vinuesa      78 jul 22 21:43 comandos_de_linux.tab -> /home/vinuesa/Cursos/TIB/T
## -rw-r--r-- 1 vinuesa vinuesa       0 jul 22 21:43 empty_file
## -rw-r--r-- 1 vinuesa vinuesa   87065 jul 22 20:59 github_TIB-filoinfo_screenshot.png
## drwxr-xr-x 3 vinuesa vinuesa    4096 jul 22 21:23 intro2genomics
## -rw-r--r-- 1 vinuesa vinuesa 2186396 jul 22 21:11 intro_biocomputo_Linux_pt1.odp
## -rw-r--r-- 1 vinuesa vinuesa 1692567 jul 22 21:11 Intro_biocomputo_Linux_pt1.pdf
## -rwxr-xr-x 1 vinuesa vinuesa   10193 jul 21 11:21 linux_basic_commands.tab
## lrwxrwxrwx 1 vinuesa vinuesa      78 jul 22 21:43 linux_commands.tab -> /home/vinuesa/Cursos/TIB/TIB
## -rwxr-xr-x 1 vinuesa vinuesa    1705 jul 21 11:21 linux_very_basic_commands_table.csv
## -rw-rw-r-- 1 vinuesa vinuesa   51271 jul 22 18:55 sesion_local_capt_pantalla.png
## -rw-r--r-- 1 vinuesa vinuesa   40746 jul 22 18:56 sesion_remota_bonampak_capt_pantalla1.png
## -rw-r--r-- 1 vinuesa vinuesa    4468 jul 22 18:57 sesion_remota_bonampak_capt_pantalla2.png
## -rw-rw-r-- 1 vinuesa vinuesa  408580 jul 22 18:51 sesion_remota_bonampak_capt_pantalla.png
## -rw-r--r-- 1 vinuesa vinuesa   47651 jul 22 21:43 Stenotrophomonas_complete_genomes_and_ftp_paths.tx
## -rwxr-xr-x 1 vinuesa vinuesa    6047 jul 21 11:21 working_with_linux_commands.code
## -rw-r--r-- 1 vinuesa vinuesa  684706 jul 22 21:43 working_with_linux_commands.html
## -rw-r--r-- 1 vinuesa vinuesa   19004 jul 22 21:43 working_with_linux_commands.Rmd
```

renombramos la liga (o cualquier archivo o directorio)

```
mv comandos_de_linux.tab linux_commands.tab
```

Visualización de contenidos de archivos: comando head, tail, cat, less, more

uso de head y tail para desplegar la cabecera y cola de archivos

```
head linux_commands.tab
```

```
## IEEE Std 1003.1-2008 utilities Name  Category    Description     First appeared
## admin    SCCS    Create and administer SCCS files    PWB UNIX
## alias    Misc    Define or display aliases
## ar   Misc    Create and maintain library archives    Version 1 AT&T UNIX
## asa  Text processing     Interpret carriage-control characters   System V
## at   Process management  Execute commands at a later time    Version 7 AT&T UNIX
## awk  Text processing     Pattern scanning and processing language    Version 7 AT&T UNIX
## basename     Filesystem  Return non-directory portion of a pathname; see also dirname    Version 7 A
## batch    Process management  Schedule commands to be executed in a batch queue
## bc   Misc    Arbitrary-precision arithmetic language     Version 6 AT&T UNIX
```

```
tail linux_commands.tab
```

```
## val  SCCS    Validate SCCS files     System III
## vi   Text processing     Screen-oriented (visual) display editor     1BSD
## wait     Process management  Await process completion    Version 4 AT&T UNIX
## wc   Text processing     Line, word and byte or character count  Version 1 AT&T UNIX
## what     SCCS    Identify SCCS files     PWB UNIX
## who  System administration   Display who is on the system    Version 1 AT&T UNIX
## write    Misc    Write to another user's terminal    Version 1 AT&T UNIX
## xargs    Shell programming   Construct argument lists and invoke utility     PWB UNIX
## yacc     C programming   Yet another compiler compiler   PWB UNIX
## zcat     Text processing     Expand and concatenate data     4.3BSD
```

```
# le podemos indicar el numero de lineas a desplegar
head -3 linux_commands.tab
```

```
## IEEE Std 1003.1-2008 utilities Name  Category    Description     First appeared
## admin    SCCS    Create and administer SCCS files    PWB UNIX
## alias    Misc    Define or display aliases
```

```
tail -1 linux_commands.tab
```

```
## zcat     Text processing     Expand and concatenate data     4.3BSD
```

cat despliega uno o más archivos, concatenándolos

```
cat linux_commands.tab | head
```

```
## IEEE Std 1003.1-2008 utilities Name  Category    Description     First appeared
## admin    SCCS    Create and administer SCCS files    PWB UNIX
## alias    Misc    Define or display aliases
## ar   Misc    Create and maintain library archives    Version 1 AT&T UNIX
## asa  Text processing     Interpret carriage-control characters   System V
```

```
## at   Process management  Execute commands at a later time    Version 7 AT&T UNIX
## awk  Text processing     Pattern scanning and processing language    Version 7 AT&T UNIX
## basename     Filesystem  Return non-directory portion of a pathname; see also dirname    Version 7 A
## batch    Process management  Schedule commands to be executed in a batch queue
## bc   Misc    Arbitrary-precision arithmetic language     Version 6 AT&T UNIX
```

cat -n nos permite añadir números de línea a los archivos desplegados

```
cat -n linux_commands.tab | head
```

```
##      1   IEEE Std 1003.1-2008 utilities Name     Category    Description     First appeared
##      2   admin   SCCS    Create and administer SCCS files    PWB UNIX
##      3   alias   Misc    Define or display aliases
##      4   ar  Misc    Create and maintain library archives    Version 1 AT&T UNIX
##      5   asa     Text processing     Interpret carriage-control characters   System V
##      6   at  Process management  Execute commands at a later time    Version 7 AT&T UNIX
##      7   awk     Text processing     Pattern scanning and processing language    Version 7 AT&T UNIX
##      8   basename    Filesystem  Return non-directory portion of a pathname; see also dirname    Vers
##      9   batch   Process management  Schedule commands to be executed in a batch queue
##     10   bc  Misc    Arbitrary-precision arithmetic language     Version 6 AT&T UNIX
```

el paginador less despliega archivos página a página

```
less linux_commands.tab | head
```

```
## IEEE Std 1003.1-2008 utilities Name  Category    Description     First appeared
## admin    SCCS    Create and administer SCCS files    PWB UNIX
## alias    Misc    Define or display aliases
## ar   Misc    Create and maintain library archives    Version 1 AT&T UNIX
## asa  Text processing     Interpret carriage-control characters   System V
## at   Process management  Execute commands at a later time    Version 7 AT&T UNIX
## awk  Text processing     Pattern scanning and processing language    Version 7 AT&T UNIX
## basename     Filesystem  Return non-directory portion of a pathname; see also dirname    Version 7 A
## batch    Process management  Schedule commands to be executed in a batch queue
## bc   Misc    Arbitrary-precision arithmetic language     Version 6 AT&T UNIX
```

Edición de archivos con los editores vim o [g|n]edit

vim (vi improved) es un poderoso editor programable presente en todos los sistemas UNIX. La principal característica tanto de Vim como de Vi consiste en que disponen de diferentes modos entre los que se alterna para realizar ciertas operaciones, lo que los diferencia de la mayoría de editores comunes, que tienen un solo modo en el que se introducen las órdenes mediante combinaciones de teclas (o interfaces gráficas). Se controla por completo mediante el teclado desde un Terminal, por lo que puede usarse sin problemas a través de conexiones remotas ya que no carga el sistema al no desplegar un entorno gráfico.

Es muy recomendable aprender a usar VIM, pero no tenemos tiempo de hacerlo en el TIB, por lo que les recomiendo este tutorial de uso de VIM en español, o directamente en su terminal tecleando el comando

```
vimtutor

# para salir de vim,

<ESC> # para estar seguros que estamos en modo ex
:q
```

En el taller usaremos generalmente el editor con ambiente gráfico gedit, de uso muy sencillo y simiar al block de notas de Windows o similar

```
# noten el uso de & al final de la sentencia para enviar el proceso al fondo
# para evitar que bloquee la terminal
gedit linux_commands.tab &
```

Edición de archivos con el editor de flujo sed (stream editor)

sed (stream editor) es un editor de flujo, una potente herramienta de tratamiento de texto para el sistema operativo Unix que acepta como entrada un archivo, lo lee y modifica línea a línea de acuerdo a un script, mostrando el resultado por salida estándar (normalmente en pantalla, a menos que se realice una redirección). Sed permite manipular flujos de datos, como por ejemplo cortar líneas, buscar y reemplazar texto (con soporte de xpresiones regulares), entre otras cosas. Posee muchas características de ed y ex.

La sintaxis general de la orden sed es:

```
$ sed [-n] [-e'script'] [-f archivo] archivo1 archivo2 ...
```

donde:

```
-n indica que se suprima la salida estándar.
-e indica que se ejecute el script que viene a continuación. Si no se emplea la opción -f se puede omiti
-f indica que las órdenes se tomarán de un archivo
```

Ejemplos de uso básico de sed:

- Cambia todas las minúsculas a mayúsculas de archivo:

```
$ sed 'y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/' archivo
```

- Borra la 1ª línea de archivo:

```
$ sed '1d' archivo
```

- Elimina las líneas en blanco. Nótese el uso de expresiones regulares, done:
  - // delimitan la expresión regular. Noten que hay que escaparla entre commillas sencillas.
  - ^ indica el inicio de la línea
  - $ indica el término de la línea

```
$ sed '/^$/d' archivo
```

- Genera una lista numerada de los nombres de campos o cabeceras del archivo linux_commands.tab
  - // delimitan la expresión regular. Noten que hay que escaparla entre commillas sencillas.
  - \t representa al tabulador
  - \m representa el salto de línea
  - //g la g indica que se reemplacen todas las instancias

```
head -1 linux_commands.tab | sed 's/\t/\n/g' | cat -n
```

```
##      1   IEEE Std 1003.1-2008 utilities Name
##      2   Category
##      3   Description
##      4   First appeared
```

Uso de tuberias de herramientas UNIX/Linux para filtrado de texto con cut, grep, sort, uniq, wc y |

UNIX y Linux ofrecen una gran cantidad de herramientas para todo tipo de trabajos, cada una generalmente con muchas opciones. En bioinformática y genómica, los archivos de texto plano (ASCII) son los más comunes. Por ello es muy útil dominar algunas de las herramientas de filtrado de texto más comunes. Como ejemplo, trabajaremos con el archivo assembly_summary.txt, que contiene los datos de ensambles genómicos de la división RefSeq de GenBank. Lo descargué y comprimí con los siguientes comandos:

```
wget -c ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/assembly_summary.txt
gzip assembly_summary.txt
```

- Exploremos el archivo comprimido (con compresión gnu zip) con usando los comandos zless o zcat

```
zless assembly_summary.txt.gz
```

```
# veamos las 5 primeras líneas del archivo
zcat assembly_summary.txt.gz | head -5
```

```
## #   See ftp://ftp.ncbi.nlm.nih.gov/genomes/README_assembly_summary.txt for a description of the colu
## # assembly_accession bioproject  biosample   wgs_master  refseq_category taxid   species_taxid   orga
## GCF_000010525.1  PRJNA224116 SAMD00060925        representative genome   438753  7   Azorhizobium cau
## GCF_000007365.1  PRJNA224116 SAMN02604269        representative genome   198804  9   Buchnera aphidi
## GCF_000007725.1  PRJNA224116 SAMN02604289        representative genome   224915  9   Buchnera aphidi
```

Ejemplos de herramientas de filtrado de texto en acción

- cut corta líneas de texto/tablas por delimitadores de campo (-d) específicos (TAB por defecto), extrayendo los campos indicados con -f (cut -d' ' -f1-3,5,9)
- sort ordena (sor -u; sort -nrk2; sort -dk1)
- wc cuenta líneas, palabras y caracteres (wc -l)
- uniq regresa listas de valores únicos (uniq -c)
- grep Filtra las lineas de un archivo que contienen (o no) caracteres o expresiones regulares (grep -E '^XXX|YYY|zzz$'; grep -v '^#')
- el pipe '|' conecta la salida de un comando con la entrada >STDIN> de otro
- ¿cuántas líneas tiene el archivo assembly_summary.txt.gz?

```
# ¿cuántas líneas tiene el archivo assembly_summary.txt.gz?
zcat assembly_summary.txt.gz | wc
zcat assembly_summary.txt.gz | wc -l
```

```
##  161297 3788695 48497020
## 161297
```

- la columna assembly_level (#12) indica el estado del ensamble. ¿Cuáles son los niveles de la variable categórica assembly_level (valores únicos de la misma?

```
# la columna assembly_level (#12) indica el estado del ensamble. ¿Cuáles son los niveles de la variable
zcat assembly_summary.txt.gz | grep -v "^#" | cut -f 12 | sort -u
```

```
## Chromosome
## Complete Genome
## Contig
## Scaffold
```

- ¿cuántos genomas hay por nivel de la variable categórica assembly_level?

```
# ¿cuántos genomas hay por nivel de la variable categórica assembly_level?
zcat assembly_summary.txt.gz | grep -v "^#" | cut -f 12 | sort | uniq -c
```

```
##    2018 Chromosome
##   13983 Complete Genome
##   82755 Contig
##   62539 Scaffold
```

- asocia cada nombre de columna de la cabecera con el número de la columna correspondiente

```
# asocia cada nombre de columna de la cabecera con el número de la columna correspondiente
zcat assembly_summary.txt.gz | head -2 | sed '1d; s/\t/\n/g' | cat -n
```

```
##      1   # assembly_accession
##      2   bioproject
##      3   biosample
##      4   wgs_master
##      5   refseq_category
##      6   taxid
##      7   species_taxid
##      8   organism_name
##      9   infraspecific_name
##     10   isolate
##     11   version_status
##     12   assembly_level
##     13   release_type
##     14   genome_rep
##     15   seq_rel_date
##     16   asm_name
##     17   submitter
##     18   gbrs_paired_asm
##     19   paired_asm_comp
##     20   ftp_path
##     21   excluded_from_refseq
##     22   relation_to_type_material
```

- genera una estadística del número de genomas por especie (columna # 8), y muestra sólo las 10 especies con más genomas secuenciados!

```
# genera una estadística del número de genomas por especie (columna # 8), y muestra sólo las 10 especie
zcat assembly_summary.txt.gz | grep -v "^#" | cut -f8 | sort | uniq -c | sort -nrk1 | head -10
```

```
##   14089 Escherichia coli
##    8039 Streptococcus pneumoniae
##    6398 Klebsiella pneumoniae
##    5924 Staphylococcus aureus
##    4556 Mycobacterium tuberculosis
##    4358 Pseudomonas aeruginosa
##    3164 Acinetobacter baumannii
##    2789 Listeria monocytogenes
##    2173 Salmonella enterica subsp. enterica serovar Typhi
##    1792 Clostridioides difficile
```

- ¿Cuántos genomas completos hay del género Acinetobacter?

```
# ¿Cuántos genomas completos hay del género Acinetobacter?
zcat assembly_summary.txt.gz | grep Acinetobacter | grep Complete | wc -l

# también puedes usar zgrep para evitar la llamada primero a zcat
zgrep Acinetobacter assembly_summary.txt.gz | grep Complete | wc -l
```

```
## 220
## 220
```

```
# ojo: Linux es sensible a mayúsculas y minúsculas: prueba este comando para comprobarlo
zgrep acinetobacter assembly_summary.txt.gz | grep Complete | wc -l # no encuentra nada
```

```
# grep -i lo hace insensible a la fuente
zgrep -i acinetobacter assembly_summary.txt.gz | grep Complete | wc -l
```

```
## 220
```

- filtra y cuenta las lineas que contienen Acinetobacter o Stenotrophomonas

```
# filtra y cuenta las lineas que contienen Acinetobacter o Stenotrophomonas
zgrep -E 'Acinetobacter|Stenotrophomonas' assembly_summary.txt.gz | wc -l
```

```
## 5170
```

- Cuenta los genomas de Acinetobacter, Pseudomonas y Klebsiella (por género) y presenta una lista ordenada por número decreciente de genomas

```
# Cuenta los genomas de Acinetobacter, Pseudomonas y Klebsiella (por género) y presenta una lista orden
zgrep -E 'Acinetobacter|Pseudomonas|Klebsiella' assembly_summary.txt.gz | cut -f 8 | cut -d' ' -f1 |sor
```

```
##    8951 Pseudomonas
##    8515 Klebsiella
##    4747 Acinetobacter
##       7 [Pseudomonas]
##       1 Candidatus
```

- Cuenta los genomas de Acinetobacter, Pseudomonas y Klebsiella (por género), con salida ordenada alfabéticamente por género

```
# filtra las lineas que contienen Filesystem o Text processing y ordénalas alfabéticamente según las en
# eliminando las entradas de Candidatus y [Pseudomonas]
zgrep -E 'Acinetobacter|Pseudomonas|Klebsiella' assembly_summary.txt.gz | cut -f 8 | cut -d' ' -f1 | gr
```

```
##    4747 Acinetobacter
##    8515 Klebsiella
##    8951 Pseudomonas
```

Veremos la gran utilidad y versatilidad de combinaciones de estos comandos para el procesamiento de archivos de secuencias en un ejercicio más adelante.

Manual de cada comando: man command

```
# mira las opciones de cut y sort en la manpage
man cut | head -20

man sort | head -20
```

```
## CUT(1)                                                User Commands
```

```
##
## NAME
##        cut - remove sections from each line of files
##
## SYNOPSIS
##        cut OPTION... [FILE]...
##
## DESCRIPTION
##        Print selected parts of lines from each FILE to standard output.
##
##        With no FILE, or when FILE is -, read standard input.
##
##        Mandatory arguments to long options are mandatory for short options too.
##
##        -b, --bytes=LIST
##               select only these bytes
##
##        -c, --characters=LIST
##               select only these characters
## SORT(1)                                              User Commands
##
## NAME
##        sort - sort lines of text files
##
## SYNOPSIS
##        sort [OPTION]... [FILE]...
##        sort [OPTION]... --files0-from=F
##
## DESCRIPTION
##        Write sorted concatenation of all FILE(s) to standard output.
##
##        With no FILE, or when FILE is -, read standard input.
##
##        Mandatory arguments to long options are mandatory for short options too.  Ordering options:
##
##        -b, --ignore-leading-blanks
##               ignore leading blanks
##
##        -d, --dictionary-order
```

redireccionado de la salida STOUT a un archivo con el comando >

```
zgrep Stenotrophomonas assembly_summary.txt.gz | cut -f8,20 > Stenotrophomonas_complete_genomes_and_ftp
```

Inicios de programación en Bash

Vermos aquí unas pocas construcciones muy básicas de probramación Shell

Asignación de variables

- La sintaxis básica de asignación es:

`varName=VALUE`

- para recuperar el valor de una varialbe, le añadimos el prefijo $. Para imprimir el valor asignado a la variable, usamos echo $varName

```
archivo_de_comandos_linux=linux_commands.tab
echo "$archivo_de_comandos_linux"
```

```
## linux_commands.tab
```

- para capturar la salida de un comando usammos $(comando)

```
wkdir=$(pwd)
date=$(date | awk '{print $3,$2,$6}' | sed 's/ //g')
h=$(hostname)
echo ">>> working in: $wkdir at <$h> on <$date>"
```

```
## >>> working in: /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux at <alisio> on <22jul2019>
```

- Modificación de variables y operaciones con ellas

```
wkdir=$(pwd)
echo "wkdir: $wkdir"

# 1. cortemos caracteres por la izquierda (todos los caracteres por la izquierda, hasta llegar a último
basedir=${wkdir##*/}
echo "basedir: $basedir  # \${wkdir##*/}"

# 2. cortemos caracteres por la derecha (cualqier caracter hasta llegar a /)
echo "path to basedir: ${wkdir%/*}  # \${wkdir%/*}"

# 3. contar el número de caracteres (longitud) de la variable
echo "basedir has ${#basedir} characters  # \${#basedir}"
```

```
## wkdir: /home/vinuesa/Cursos/TIB/TIB19-T3/sesion1_intro2linux
## basedir: sesion1_intro2linux  # ${wkdir##*/}
## path to basedir: /home/vinuesa/Cursos/TIB/TIB19-T3  # ${wkdir%/*}
## basedir has 19 characters  # ${#basedir}
```

Condicionales

Comparación de íntegros

```
i=5
j=3

if [ "$i" -lt "$j" ]; then
   echo "$i < $j"
elif [ "$i" -gt "$j" ]; then
   echo "$i > $j "
fi
```

```
## 5 > 3
```

Comparación de cadenas de caracteres

```
i=carla
j=juan
```

```
if [ "$i" == "$j" ]; then
   echo "$i = $j"
elif [ "$i" != "$j" ]; then
   echo "i:$i no es igual a i:$j "
fi
```

```
## i:carla no es igual a i:juan
```

Comprobación de la existencia de un archivo de tamaño > 0 bytes

```
touch empty_file
ls -l empty_file
ls -l *gz
f=$(ls *gz)

if [ -e empty_file  ]; then
   echo "empty_file file exists"
fi

if [ ! -s empty_file  ]; then
   echo "empty_file file exists but is empty"
fi

if [ -s "$f" ]; then
   echo "$f exists and is non-empty"
fi
```

```
## -rw-r--r-- 1 vinuesa vinuesa 0 jul 22 21:43 empty_file
## -rw-r--r-- 1 vinuesa vinuesa 6780296 jul 21 19:26 assembly_summary.txt.gz
## empty_file file exists
## empty_file file exists but is empty
## assembly_summary.txt.gz exists and is non-empty
```

```
# también podemos usar la versión corta del test:
f=$(ls *gz)
[ -s "$f" ] && echo "$f exists and is non-empty"
```

```
## assembly_summary.txt.gz exists and is non-empty
```

Bucles for

```
# veamos el contenido del directorio antes de correr el bucle
ls
```

```
## assembly_summary.txt.gz
## empty_file
## github_TIB-filoinfo_screenshot.png
## intro2genomics
## intro_biocomputo_Linux_pt1.odp
## Intro_biocomputo_Linux_pt1.pdf
## linux_basic_commands.tab
## linux_commands.tab
## linux_very_basic_commands_table.csv
## sesion_local_capt_pantalla.png
```

```
## sesion_remota_bonampak_capt_pantalla1.png
## sesion_remota_bonampak_capt_pantalla2.png
## sesion_remota_bonampak_capt_pantalla.png
## Stenotrophomonas_complete_genomes_and_ftp_paths.txt
## working_with_linux_commands.code
## working_with_linux_commands.html
## working_with_linux_commands.Rmd
```

```
#>>> AVANZADO: usa un bucle for, acoplado a las herramientas de filtrado arriba mostradas,
#              para generar archivos que contengan solo los comandos de las diferentes categorias
#              nombrando a los archivos por estas

# for type in $(cut -f2 linux_basic_commands.tab | sort -u); do grep "$type" linux_basic_commands.tab >
for type in $(cut -f2 linux_basic_commands.tab | sort -u); do
    grep "$type" linux_basic_commands.tab > ${type}.cmds
done
```

```
# veamos el contenido del directorio después de correr el bucle
ls
```

```
## administration.cmds
## assembly_summary.txt.gz
## Batch.cmds
## Category.cmds
## C.cmds
## empty_file
## Filesystem.cmds
## FORTRAN77.cmds
## github_TIB-filoinfo_screenshot.png
## intro2genomics
## intro_biocomputo_Linux_pt1.odp
## Intro_biocomputo_Linux_pt1.pdf
## linux_basic_commands.tab
## linux_commands.tab
## linux_very_basic_commands_table.csv
## management.cmds
## Misc.cmds
## Network.cmds
## Process.cmds
## processing.cmds
## programming.cmds
## Programming.cmds
## SCCS.cmds
## sesion_local_capt_pantalla.png
## sesion_remota_bonampak_capt_pantalla1.png
## sesion_remota_bonampak_capt_pantalla2.png
## sesion_remota_bonampak_capt_pantalla.png
## Shell.cmds
## Stenotrophomonas_complete_genomes_and_ftp_paths.txt
## System.cmds
## Text.cmds
## utilities.cmds
## working_with_linux_commands.code
## working_with_linux_commands.html
## working_with_linux_commands.Rmd
```

```
# veamos el contenido de uno de los nuevos archivos generados
cat programming.cmds
```

```
## cc/c99   C programming   Compile standard C programs     IEEE Std 1003.1-2001
## cflow    C programming   Generate a C-language call graph    System V
## command  Shell programming   Execute a simple command
## ctags    C programming   Create a tags file  3BSD
## cxref    C programming   Generate a C-language program cross-reference table     System V
## echo     Shell programming   Write arguments to standard output  Version 2 AT&T UNIX
## expr     Shell programming   Evaluate arguments as an expression     Version 7 AT&T UNIX
## false    Shell programming   Return false value  Version 7 AT&T UNIX
## fort77   FORTRAN77 programming   FORTRAN compiler    XPG4
## getopts  Shell programming   Parse utility options
## lex  C programming   Generate programs for lexical tasks     Version 7 AT&T UNIX
## logger   Shell programming   Log messages    4.3BSD
## nm   C programming   Write the name list of an object file   Version 1 AT&T UNIX
## printf   Shell programming   Write formatted output  4.3BSD-Reno
## read     Shell programming   Read a line from standard input
## sh   Shell programming   Shell, the standard command language interpreter    Version 7 AT&T UNIX (in
## sleep    Shell programming   Suspend execution for an interval   Version 4 AT&T UNIX
## strings  C programming   Find printable strings in files     2BSD
## strip    C programming   Remove unnecessary information from executable files    Version 1 AT&T UNIX
## tee  Shell programming   Duplicate the standard output   Version 5 AT&T UNIX
## test     Shell programming   Evaluate expression     Version 7 AT&T UNIX
## true     Shell programming   Return true value   Version 7 AT&T UNIX
## xargs    Shell programming   Construct argument lists and invoke utility     PWB UNIX
## yacc     C programming   Yet another compiler compiler   PWB UNIX
```

```
# finalmente borremos los nuevos archivos generados
rm *.cmds
```