

Deber TC3_ForLoop

Ejercicios 1.10 del libro Allesina&Wilmes_ComputingSkills4Biologists-Toolbox

Estudiante: Laura Lázara Rodríguez Castillo

Se ingresó al directorio CSB-master/unix/sandbox:

```
Usuario@DESKTOP-TPNNH6H MINGW64 ~
$ cd OneDrive

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive
$ cd CSB-master

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master
$ cd unix

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix
$ cd sandbox
```

1.10.1 Next Generation Sequencing Data:

```
GNU nano 6.4 Ejercicio1.10.1.sh
## Para saber el tamaño del archivo

echo "Para saber el tamaño del archivo se pone el siguiente código"
ls -lh ../data/Marra2014_data.fasta

## Creación de copia de Marra2014_data.fasta en el directorio sandbox

echo "Para crear la copia de Marra2014_data.fasta en el directorio de sandbox y nombrarle como"
cp ../data/Marra2014_data.fasta my_file.fasta

# Para saber cuantos contigs se clasifican como isogruppo00036

echo "Para saber cuantos contigs se clasifican como isogruppo00036 se ejecuta el siguiente código"
grep -c isogruppo00036 my_file.fasta

# Separación del delimitador original de dos espacios con una coma

echo " para separar el delimitador de dos espacios con una coma se utiliza el siguiente código "
cat my_file.fasta | tr -s ' ' ',' | head -n 3

[ Read 50 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

```
GNU nano 6.4 Ejercicio1.10.1.sh
# Guardar y sobrescrito de archivo temporal

echo "Para guardar y sobrescribir lo del archivo se ejecuta lo siguiente"
cat my_file.fasta | tr -s ' ' ',' > my_file.tmp
mv my_file.tmp my_file.fasta

# Para saber cuantos isogrupos únicos hay en el archivo

echo "Para saber cuantos isogrupos únicos hay en el archivo se utiliza el siguiente comando"
grep '>' my_file.fasta | cut -d ',' -f 4 | sort | uniq | wc -l

# Para saber que contig tiene el mayor número de lectura

echo "Para saber que contig tiene el mayor número de lectura se ejecuta el siguiente código"
grep '>' my_file.fasta | cut -d ',' -f 1,3 | head -n 3

#Para ordenar según el número de lee

echo "Para ordenar según el número de lee se ejecuta el siguiente código"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
```

```
#Para ordenar según el número de lee
echo "Para ordenar según el número de lee se ejecuta el siguiente código"
grep '>' my_file.fasta | cut -d ',' -f 1,3 | sort -t '=' -k 2 -n | head -n 5
```

MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/sandbox

```
Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/sandbox
$ nano Ejercicio1.10.1.sh

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/sandbox
$ bash Ejercicio1.10.1.sh
Para saber el tamaño del archivo se pone el siguiente código
-rw-r--r-- 1 Usuario 197121 553K Jan 28 2019 ../data/Marra2014_data.fasta
Para crear la copia de Marra2014_data.fasta en el directorio de sandbox y nombrarle como my_file
.fasta se utiliza el siguiente comando
Para saber cuantos contigs se clasifican como isogrupos00036 se ejecuta el siguiente código
16
Para separar el delimitador de dos espacios con una coma se utiliza el siguiente código
>contig00001,length=527,numreads=2,gene=isogroup00001,status=it_thresh
ATCCTAGCTACTCTGGAGACTGAGGATTGAAGTTCAAAGTCAGCTCAAGCAAGAGATTG
TTTACAATTAACCCACAAAGGCTGTTACTGAAGGTGTGGCTTAAGTGTCAAGCAAGCAAG
Para guardar y sobrescribir lo del archivo se ejecuta lo siguiente
Para saber cuantos isogrupos únicos hay en el archivo se utiliza el siguiente comando
43
Para saber que contig tiene el mayor número de lectura se ejecuta el siguiente código
>contig00001,numreads=2
>contig00002,numreads=8
>contig00003,numreads=2
Para ordenar según el número de lee se ejecuta el siguiente código
>contig00089,numreads=1
>contig00176,numreads=1
>contig00210,numreads=1
>contig00001,numreads=2
>contig00003,numreads=2
```

1.10.2 Hormone Levels in Baboons:

MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/sandbox

```
GNU nano 6.4 Ejercicios1.10.2.sh
# Ejercicio 1.10.2 Gesquiere
# 1. Cuántas veces fueron los niveles de los individuos 3 y 27 grabados ?
echo "Para saber cuantas veces fueron los niveles de los individuos 3 y 27 se utiliza el siguiente código"
echo "Si deseas primero puedes ver la estructura del archivo de la siguiente manera"
head -n 3 ../data/Gesquiere2011_data.csv
echo "Eneste caso solo se va a utilizar la columna 1, para esto"
cut -f 1 ../data/Gesquiere2011_data.csv | head -n 3
echo "Para contar el número de ocurrencias"
#Identificación masculina 3
echo "Para saber la identificación masculina 3 se utiliza el siguiente código"
cut -f 1 ../data/Gesquiere2011_data.csv | grep -c -w 3
#Identificación masculina 27
echo "Para la identificación masculina 27 se utiliza el siguiente código"
cut -f 1 ../data/Gesquiere2011_data.csv | grep -c -w 27
# 2. Escriba un script tomando como entrada el nombre del archivo y el DNI del individuo, y devolviendo el numero de registros para esa ID.
echo "Script tomando como entrada el nombre del archivo y el ID del individuo"
bash Ejercicio1.10.2.2.sh ../data/Gesquiere2011_data.csv 27
```

```
# 3. Escriba un script que devuelva el número de veces que se tomaron muestras de cada individuo
echo "Número de veces que se tomaron las muestras de cada individuo"
echo "Primero se obtienen las identificaciones únicas"
bash Ejercicio1.10.2.3.sh
```

MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/sandbox

```
GNU nano 6.4 Ejercicio1.10.2.2.sh
#Para leer la entrada del comando línea
cut -f 1 $1 | grep -c -w $2
```

```
myIDS=`tail -n +2 ../data/Gesquiere2011_data.csv | cut -f 1 | sort -n | uniq`
for id in $myIDS
do
    mycounts=`bash Ejercicio1.10.2.2.sh ../data/Gesquiere2011_data.csv $id`
    echo "ID:" $id "counts:" $mycounts
done
```

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/sandbox

\$ bash Ejercicios1.10.2.sh

Para saber cuantas veces fueron los niveles de los individuos 3 y 27 se utiliza el siguiente código

Si deseas primero puedes ver la estructura del archivo de la siguiente manera

```
maleID  GC      T
1       66.9    64.57
1       51.09   35.57
```

Eneste caso solo se va a utilizar la columna 1, para esto

```
maleID
```

```
1
```

```
1
```

Para contar el número de ocurrencias

Para saber la identificación masculina 3 se utiliza el siguiente código

```
61
```

Para la identificación masculina 27 se utiliza el siguiente código

```
5
```

Script tomando como entrada el nombre del archivo y el ID del individuo

```
5
```

Número de veces que se tomaron las muestras de cada individuo

Primero se obtienen las identificaciones únicas

```
ID: 1 counts: 10
ID: 2 counts: 2
ID: 3 counts: 61
ID: 4 counts: 46
ID: 5 counts: 28
ID: 6 counts: 7
ID: 7 counts: 5
ID: 8 counts: 17
ID: 9 counts: 4
ID: 10 counts: 21
ID: 11 counts: 26
```

```
ID: 101 counts: 26
ID: 102 counts: 15
ID: 103 counts: 26
ID: 104 counts: 29
ID: 105 counts: 6
ID: 106 counts: 46
ID: 107 counts: 7
ID: 108 counts: 41
ID: 109 counts: 28
ID: 110 counts: 3
ID: 111 counts: 24
ID: 112 counts: 3
ID: 113 counts: 1
ID: 114 counts: 1
ID: 115 counts: 1
ID: 116 counts: 14
ID: 118 counts: 23
ID: 119 counts: 1
ID: 120 counts: 42
ID: 121 counts: 12
ID: 122 counts: 9
ID: 123 counts: 39
ID: 124 counts: 1
ID: 125 counts: 39
ID: 126 counts: 15
ID: 127 counts: 13
```

Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/sand

1.10.3 Plant-Pollinator Networks:

```
MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/data/Saavedra2013
GNU nano 6.4 Ejercicios1.10.3.sh
# 1. 1) Escriba un script que tome uno de estos archivos y determina el número de filas (polinizadores) y columnas
echo "Conteo de las filas"
wc -l n10.txt
echo "Regresión"
cat n10.txt | wc -l
echo "Conteo del número de columnas"
echo "Primero se ocupa la primera línea"
head -n 1 n10.txt
echo "Eliminación de los espacios y del terminador de línea"
head -n 1 n10.txt | tr -d ' ' | tr -d '\n'
echo "Conteo de caracteres"
head -n 1 n10.txt | tr -d ' ' | tr -d '\n' | wc -c
```

```
# 2. Escriba un script que imprima el número de filas y columnas para cada red
echo "Impresión del número de filas y columnas para cada red"
bash Ejercicio1.10.3.2.sh
# 3.Cuál es la red con el mayor número de filas? ¿Cuál es el que tiene el mayor número de columnas?
echo "Red con más filas"
bash Ejercicio1.10.3.2.sh | sort -n -r -k 2 | head -n 1
echo "Red con más columnas"
bash Ejercicio1.10.3.2.sh | sort -n -r -k 3 | head -n 1
```

```
MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/data/Saavedra2013
GNU nano 6.4 Ejercicio1.10.3.2.sh
FILES=*.txt
for f in $FILES
do
    myrow=`cat $f | wc -l`
    mycol=`head -n 1 $f | tr -d ' ' | tr -d '\n' | wc -c`
    echo $f $myrow $mycol
done
```

```
MINGW64/c/Users/Usuario/OneDrive/CSB-master/unix/data/Saavedra2013
Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/data/Saavedra2013
$ bash Ejercicios1.10.3.sh
Conteo de las filas
14 n10.txt
Regresión
14
Conteo del número de columnas
Primero se ocupa la primera línea
0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
Eliminación de los espacios y del terminador de línea
01000001000000000100Con el conteo de caracteres
20
Impresión del número de filas y columnas para cada red
n1.txt 97 80
n10.txt 14 20
n11.txt 270 91
n12.txt 7 72
n13.txt 61 17
n14.txt 35 15
n15.txt 38 11
n16.txt 118 24
n17.txt 76 31
n18.txt 13 14
n19.txt 10 16
n2.txt 62 41
n20.txt 18 7
n21.txt 19 45
n22.txt 19 36
n23.txt 179 26
```

```
n24.txt 80 28
n25.txt 17 16
n26.txt 82 40
n27.txt 27 5
n28.txt 90 19
n29.txt 61 25
n3.txt 25 36
n30.txt 8 19
n31.txt 28 25
n32.txt 45 21
n33.txt 70 20
n34.txt 79 25
n35.txt 14 8
n36.txt 40 169
n37.txt 44 13
n38.txt 51 99
n39.txt 33 25
n4.txt 101 11
n40.txt 28 18
n41.txt 12 10
n42.txt 42 8
n43.txt 55 29
n44.txt 56 9
n45.txt 36 61
n46.txt 58 17
n47.txt 139 41
n48.txt 118 49
n49.txt 47 23
n5.txt 21 7
n50.txt 45 46
```

```
n5.txt 21 7
n50.txt 45 46
n51.txt 8 15
n52.txt 33 7
n53.txt 34 13
n54.txt 126 25
n55.txt 14 50
n56.txt 110 207
n57.txt 14 11
n58.txt 678 90
n59.txt 663 130
n6.txt 9 31
n7.txt 16 25
n8.txt 19 33
n9.txt 12 22
Red con más filas
n58.txt 678 90
Red con más columnas
n56.txt 110 207
```

1.10.4 Data Explorer:

```
GNU nano 6.4 Ejercicios1.10.4.sh
# Escriba un script que, para un archivo csv dado y número de columna, imprime:
# - Nombre de la columna
# - Número de valores distintos
# - Valor mínimo
# - Valor máximo

echo "Extracción del nombre de la columna"

cut -d ',' -f 7 ../data/Buzzard2015_data.csv | head -n 1

echo "Número de valores distintos"

cut -d ',' -f 7 ../data/Buzzard2015_data.csv | tail -n +2 | sort | uniq | wc -l

echo "Valor máximo"

cut -d ',' -f 7 ../data/Buzzard2015_data.csv | tail -n +2 | sort -n | tail -n 1

echo "Valor mínimo"

cut -d ',' -f 7 ../data/Buzzard2015_data.csv | tail -n +2 | sort -n | head -n 1
```

```
Usuario@DESKTOP-TPNNH6H MINGW64 ~/OneDrive/CSB-master/unix/data
$ bash Ejercicios1.10.4.sh
Extracción del nombre de la columna
biomass
Número de valores distintos
285
Valor máximo
14897.29471
Valor mínimo
1.048466198
```