Introducción

A partir de una secuencia de ADN se pueden calcular una serie de propiedades composicionales que a veces nos pueden facilitar información acerca de su contenido, posibles funciones y su organismo portador. El ejemplo clásico es la búsqueda del origen de replicación en bacterias. Se recomienda la lectura de este capitulo: https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1

El contenido en G+C de una secuencia es simplemente la fracción de guaninas y citosinas sobre el total de las 4 bases (en ausencia de bases ambiguas, este número es idéntico a la longitud de la secuencia). Su análisis a lo largo de la secuencia nos puede dar una idea donde se localizan promotores y exones,

regiones que se suelen caracterizar por destacar en su contenido en G+C sobre su entorno.

Sin embargo, aún más informativo es la frecuencia de los dinucleótidos. En mamíferos existen metiltransferasas que metilan las citosinas en un contexto CpG. Una citosina metilada desamina espontáneamente hacia timina, y no hacia uracilo como una citosina nometilada. Por ello, la metilación es el principal causante de mutaciones puntuales y cerca de dos tercios de todos los SNPs en la población humana se encuentran en el contexto CpG. La metilación de citosinas por lo tanto causa la pérdida evolutiva de dinucleótidos CpG y en una secuencia típica de un mamífero se detectan solamente la quinta parte de los CpGs esperados. Por lo tanto, solamente mediante un análisis de las frecuencias de dinucleótidos podemos determinar si la secuencia anónima proviene de un mamífero o no.

Ejercicios paso por paso

Para el análisis composicional, disponemos de una serie de programas de EMBOSS instalados en el servidor de docencia. Para hacer los ejercicios que vienen a continuación, nos conectamos primero con el servidor (véase tema 2).

Determinar la longitud y el contenido global en G+C (%GC)

El programa que vamos a usar se llama *infoseq*.

Podemos lanzar el programa escribiendo en línea de comando:

```
infoseq --help
```

Buscamos el inicio de la salida y observamos lo siguiente:

```
michael@sRNAtoolbox:~$ infoseq --help
Display basic information about sequences
Version: EMBOSS:6.6.0.0

   Standard (Mandatory) qualifiers:
  [-sequence]          seqall     (Gapped) sequence(s) filename and optional
                                  format, or reference (input USA)

   Additional (Optional) qualifiers:
   -outfile            outfile    [stdout] If you enter the name of a file
                                  here then this program will write the
                                  sequence details into that file.
   -html               boolean    [N] Format output as an HTML table

   Advanced (Unprompted) qualifiers:
   -[no]columns        boolean    [Y] Set this option on (Y) to print the
                                  sequence information into neat, aligned
                                  columns in the output file. Alternatively,
                                  leave it unset (N), in which case the
                                  information records will be delimited by a
                                  character, which you may specify by using
                                  the -delimiter option. In other words, if
                                  -columns is set on, the -delimiter option is
                                  overriden.
   -delimiter          string     [|] This string, which is usually a single
                                  character only, is used to delimit
                                  individual records in the text output file.
```

El único parámetro obligatorio es la secuencia de entrada. Vamos a analizar una secuencia con los parámetros por defecto:

Descargamos la secuencia de ejemplo: sec29 (es conveniente hacer una carpeta para este tema y guardar la secuencia dentro de esta carpeta). Después de descargar la secuencia veremos algo parecido si lanzamos:

```
ls -la
```

```
michael@sRNAtoolbox:~/tema3$ ls -la
total 2316
drwxrwxr-x 1 michael michael      18 Oct  1 10:45 .
drwxr-xr-x 1 michael michael     168 Oct  1 10:45 ..
-rw-rw-r-- 1 michael michael 2369307 Dec 18  2018 sec29.txt
```

Lanzamos el infoseq con la secuencia de entrada (OJO, la siguiente línea de comando se basa en rutas relativas, es decir solo funcionará si estamos dentro de la carpeta que contiene la secuencia cuando lanzamos la orden)

```
infoseq -sequence seq29.txt
```

Deberíamos observar la siguiente salida. La primera línea es el encabezado y la segunda nos muestra los valores.

```
michael@sRNAtoolbox:~/tema3$ infoseq -sequence sec29.txt
Display basic information about sequences
USA                    Database  Name          Accession      Type Length %GC    Organism          Description
fasta::sec29.txt:sec29 -                sec29          -              N    2278171 36.45
```

La salida usa tabuladores, y por eso las columnas se pueden desplazar en comparación con el encabezado. En este caso podemos ver los valores que toman la longitud y el %G+C

- Length : 2278171
- %GC : 36.45

Practicar mas

- Redireccionar la salida a un fichero (o con la opción -outfile o mediante el operador >). En este apartado se especifica más acerca la redirecciones
- Copiar el fichero del resultado al ordenador del estudiante
- Abrir el fichero con excel

Analizar la variación espacial del %GC a lo largo de una secuencia

Para calcular el %GC a lo largo de una secuencia vamos a emplear un método de ventana deslizante (o ventana móvil).

Los métodos de ventana deslizante (sliding window) consiste en los siguientes pasos

1. Posicionar un puntero en la secuencia
2. Extraer una subsecuencia de tamaño N desde la posición actual. N es el tamaño de la ventana.
3. Ejecutar una operación sobre esta subsecuencia (como por ejemplo calcular el %G+C) y guardar tanto la posición como el valor
4. Desplazar el puntero M posiciones. M es el tamaño del paso. Valores frecuentes son 1 o poner M igual que N (tamaño de la ventana)
5. Reiniciar en el punto 2 hasta llegar al final de la secuencia

Para ilustrar este método vamos a considerar la siguiente secuencia. Calculamos el %G+G en ventanas de longitud 10 y paso 10

```
ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG

ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG

ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG

ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG

ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG

ACGCAGACTAGGCATCCCCGATTACAAAATCGGTTAGCCTCTGGGACCAGACTTACGACGCTG
```

Posición	%G+C
1	50
11	80
21	10
31	60
41	70
51	50

El programa freak implementa un algoritmo como se explicó en el gráfico anterior. Nos permite analizar el contenido en G+C (%GC) a lo largo de una secuencia o más en general, cualquier combinación de letras en una secuencia simbólica.

Lanzamos el programa:

```
freak -help
```

```
michael@sRNAtoolbox:~/tema3$ freak -help
Generate residue/base frequency table or plot
Version: EMBOSS:6.6.0.0

   Standard (Mandatory) qualifiers (* if not always prompted):
  [-seqall]           seqall     Sequence(s) filename and optional format, or
                                 reference (input USA)
   -letters           string     [gc] Residue letters (Any string)
*  -graph             xygraph    [$EMBOSS_GRAPHICS value, or x11] Graph type
                                 (ps, hpgl, hp7470, hp7580, meta, cps, x11,
                                 tek, tekt, none, data, xterm, png, gif, pdf,
                                 svg)
*  -outfile           outfile    [*.freak] Output file name

   Additional (Optional) qualifiers:
   -step              integer    [1] Stepping value (Any integer value)
   -window            integer    [30] Averaging window (Any integer value)

   Advanced (Unprompted) qualifiers:
   -plot              toggle     [N] Produce graphic

   General qualifiers:
   -help              boolean    Report command line options and exit. More
                                 information on associated and general
                                 qualifiers can be found with -help -verbose
```

En la ayuda podemos ver los parámetros implementados:
-seqall : el fichero de entrada
-letters : una cadena de caracteres que define los nucleótidos que se van a usar en el análisis. En nuestro caso estamos interesados un G+C así que usaremos el valor por defecto 'gc'
-step : el tamaño del salto (véase ilustración)
-window : el tamaño de la ventana (véase ilustración)

Ejercicio:

- Descargar la secuencia del MHC humano (cromosoma 6) y la del genoma del Thermus thermophilus HB8 (NC_006461.1)
- Lanzar el freak con valores por defecto para la secuencia del MHC

```
freak -seqall MHC_LH_clean.txt -step 100 -window 100 -letters gc
-outfile MHC.freak
```

```
michael@sRNAtoolbox:~/tema3$ freak -seqall MHC_LH_clean.txt -step 100 -window 100 -outfile MHC.freak
Generate residue/base frequency table or plot
Residue letters [gc]:
michael@sRNAtoolbox:~/tema3$ freak -seqall MHC_LH_clean.txt -step 100 -window 100 -letters gc -outfile MHC.freak
Generate residue/base frequency table or plot
```

Si visualizamos el contenido con more

```
more MHC.freak
```

Observamos que la salida es como explicamos en la ilustración arriba. Para cada posición de la ventana (primera columna) obtenemos el valor del G+C (expresado como fracción y no como porcentaje en este caso)

```
FREAK of MHC from 1 to 1543020 Window 100 Step 100

1           0.550000
101         0.490000
201         0.590000
301         0.550000
401         0.510000
501         0.560000
601         0.340000
701         0.340000
801         0.450000
901         0.360000
1001        0.470000
1101        0.430000
1201        0.330000
1301        0.520000
1401        0.470000
1501        0.590000
1601        0.490000
1701        0.630000
1801        0.450000
1901        0.470000
2001        0.470000
```

Para obtener una visualización de estos resultados podemos o descargar los datos y representarlos gráficamente o usamos una opción del freak para crear directamente un gráfico.

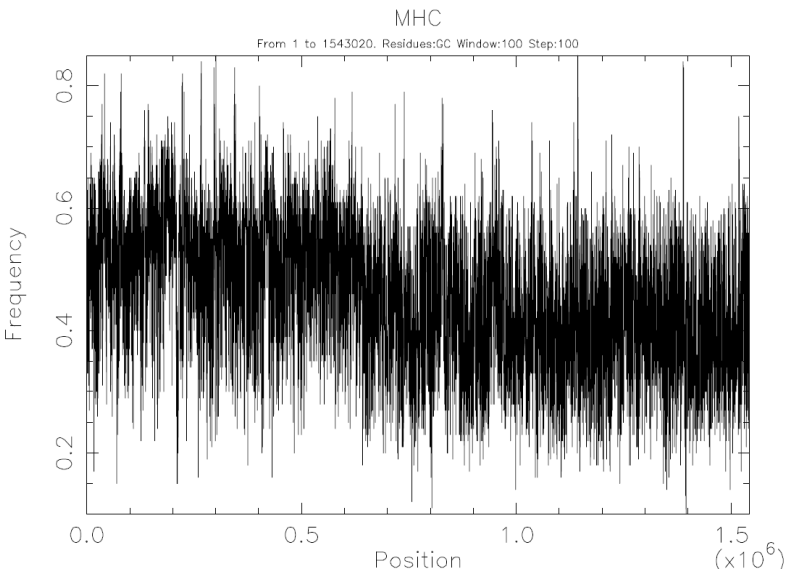
```
freak -seqall MHC_LH_clean.txt -step 100 -window 100 -letters gc
-outfile MHC.pdf -graph pdf -plot
```

El programa crea el fichero freak.pdf (no MHC.pdf como especificamos en la línea de comando (se trata de un pequeño *'bug'* del programa). Para recordar que freak.pdf corresponde a la salida del fichero MHC lo vamos a renombrar.

```
mv freak.pdf MHC_w100_s100.pdf
```

De esta forma, el nombre del fichero lleva tanto la información acerca del fasta de entrada (MHC) como de los parámetros más importantes (tamaño de ventana y salto).

Si copiamos el fichero al ordenador local y lo abrimos con un programa para visualizar un PDF podemos observar el resultado que vemos a la derecha. Existen ventanas con %GC por encima del 80%, pero también ventanas con un %GC cerca de 0. En general se observa mucha fluctuación

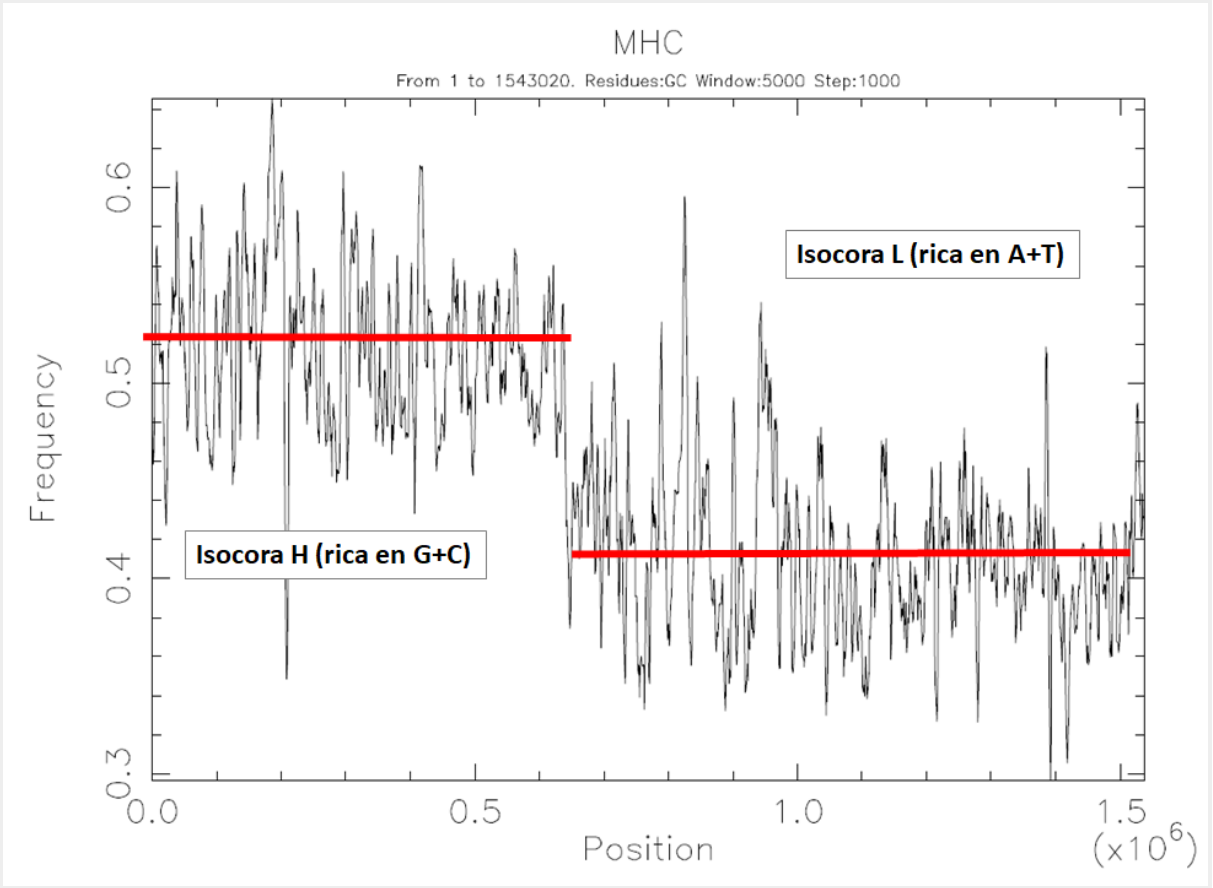


lo que nos puede indicar que la ventana ha sido muy pequeña. Aun así se puede intuir que a la izquierda existe un %G+C más alto que a la derecha. Vamos a comprobarlo aumentando el tamaño de la ventana.

```
freak -seqall MHC_LH_clean.txt -step 1000 -window 5000 -letters gc
-outfile MHC.pdf -graph pdf -plot
```

Ahora podemos observar que el gráfico es mucho menos ruidoso. Detectamos claramente dos dominios, uno rico en G+C (a la izquierda, isocora H) y otro rico en A+T (a la derecha, isocora L)
Opcional (más información acerca de isocoras):

- Figura 2 en esta publicación: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC441537/
- https://www.sciencedirect.com/science/article/pii/S0022283699930048



Relación entre ventana y ruido:

Fijar el tamaño de la ventana (y del salto) no es trivial. Si elegimos el tamaño de ventana muy pequeña, obtendremos mucho ruido. En el primer gráfico apenas se puede intuir la estructura de isocoras presente. Si al contrario seleccionamos una ventana demasiado grande, desaparecen las estructuras pequeñas. Por ejemplo, el promotor de un gen abarca

pocos cientos de pares de bases. Si aplicamos una ventana muy grande probablemente el pico en G+C desaparezca.

Analizar la frecuencia de N-meros

N-mero es simplemente la forma genérica de referirse a una secuencia simbólica corta de longitud N. Por ejemplo, los 1-meros, 2-meros y 3-meros se llaman mononucleótidos, dinucelótidos y trinucleótidos cuando se trata de una secuencia de nucleótidos. Para detectar su frecuencia se desliza una ventana móvil a lo largo de la secuencia (***compseq***). El tamaño de la ventana corresponde a la longitud del N-mero que queremos detectar. Generalmente se detectan de forma solapante, es decir con salto 1. Por ejemplo, para determinar la frecuencia de dinucleótidos se usa una ventana de tamaño 2 y salto de 1.

```
Secuencia:     ATTCCGTGAACTG…
Dinucleótidos: AT, TT, TC, CG, GT…
```

Ejercicio

Descargamos la secuencia secuencia_N
`Ejecutamos el programa compseq`

```
compseq -sequence secuencia_con_N.txt -outfile
secuencia_con_N_w2.txt -word 2
```

-word 2 indica que la longitud de ventana será 2 (dinucleótidos)

```
michael@sRNAtoolbox:~/tema3$ more secuencia_con_N_w2.txt
#
# Output from 'compseq'
#
# The Expected frequencies are calculated on the (false) assumption that every
# word has equal frequency.
#
# The input sequences are:
#       MHC


Word size       2
Total count     43152

#
# Word  Obs Count       Obs Frequency   Exp Frequency   Obs/Exp Frequency
#
AA      2519            0.0583750       0.0625000       0.9340007
AC      2034            0.0471357       0.0625000       0.7541713
AG      3333            0.0772386       0.0625000       1.2358176
AT      1834            0.0425009       0.0625000       0.6800148
CA      3071            0.0711670       0.0625000       1.1386726
CC      3404            0.0788839       0.0625000       1.2621431
CG      748             0.0173341       0.0625000       0.2773452
CT      3123            0.0723721       0.0625000       1.1579533
GA      2713            0.0628708       0.0625000       1.0059325
GC      2410            0.0558491       0.0625000       0.8935855
GG      3421            0.0792779       0.0625000       1.2684464
GT      2054            0.0475992       0.0625000       0.7615869
TA      1418            0.0328606       0.0625000       0.5257694
TC      2498            0.0578884       0.0625000       0.9262143
TG      3095            0.0717232       0.0625000       1.1475714
TT      2476            0.0573786       0.0625000       0.9180571

Other   3001            0.0695449       0.0000000       10000000000.0000000
michael@sRNAtoolbox:~/tema3$
```

En la salida vemos las siguientes columnas:
Word : el N-mero (el dinucleótido en este caso)
Obs Count : el número de veces que la secuencia contiene este N-mero
Obs. Frequency : La frecuencia observada (count/'número total de palabras'). El número de palabras existentes en este caso es 'longitud de la secuencia' -1.
Exp. Frequency : La frecuencia esperada según las frecuencias de los mononucleótidos
Obs/Exp Frequency : La ratio entre la frecuencia observada y la esperada. Un valor de 1 indica que la secuencia contiene exactamente el número que hubiéramos esperado. Una desviación de 1 puede indicar la existencia de procesos biológicos que causan que observemos más o menos dinucleótidos de cierto tipo.

Antes de analizar el resultado, nos fijamos en las frecuencias esperadas que toman todas el mismo valor. 0.0625 es un cuarto por un cuarto que sería la probabilidad de encontrar cualquier dinucleótido en una secuencia que contiene el mismo número de las 4 bases; 0.25 cada una. Pocas secuencias reales cumplen este supuesto, y por lo tanto es mejor usar las frecuencias reales de los mononucleótidos. Compseq tiene una opción para esto:

```
compseq -sequence secuencia_con_N.txt -outfile
secuencia_con_N_w2_correcto.txt -word 2 -calcfreq
```

```
Calculate the composition of unique words in sequences
michael@sRNAtoolbox:~/tema3$ more secuencia_con_N_w2_correcto.txt
#
# Output from 'compseq'
#
# The Expected frequencies are calculated from the observed single
# base or residue frequencies in these sequences
#
# The input sequences are:
#       MHC


Word size       2
Total count     43152

#
# Word  Obs Count       Obs Frequency   Exp Frequency   Obs/Exp Frequency
#
AA      2519            0.0583750       0.0507458       1.1503430
AC      2034            0.0471357       0.0540084       0.8727475
AG      3333            0.0772386       0.0553239       1.3961160
AT      1834            0.0425009       0.0495295       0.8580938
CA      3071            0.0711670       0.0540084       1.3177028
CC      3404            0.0788839       0.0574808       1.3723524
CG      748             0.0173341       0.0588809       0.2943922
CT      3123            0.0723721       0.0527139       1.3729221
GA      2713            0.0628708       0.0553239       1.1364125
GC      2410            0.0558491       0.0588809       0.9485096
GG      3421            0.0792779       0.0603151       1.3143962
GT      2054            0.0475992       0.0539979       0.8815012
TA      1418            0.0328606       0.0495295       0.6634553
TC      2498            0.0578884       0.0527139       1.0981618
TG      3095            0.0717232       0.0539979       1.3282601
TT      2476            0.0573786       0.0483423       1.1869225

Other   3001            0.0695449       0.0695201       1.0003565
```

Observamos que ahora cada dinucleótido tiene su 'propia' frecuencia esperada. El dinucleótido CG es el que más desviación de 1 representa (0.29), es decir que en esta secuencia se observa tan solo el 30% de los CG que hubiéramos esperado. Esto se debe a la metilación del ADN. En virtualmente todos los vertebrados hay metiltransferasas activas que metilan Citosinas preferentemente en el contexto CG (Citosinas seguidas por una Guanina 5'-CG-3'), normalmente llamado CpG). Una Citosina metilada desamina espontáneamente hacia Timina, lo que explica la pérdida evolutiva de dinucleótidos CG. Una Citosina no-metilada desamina hacia Uracilo, base que se reconoce más eficazmente en la reparación del ADN.

IMPORTANTE: USAR siempre la opción -calcfreq

Uso de codones

Es importante subrayar que los codones no se cuentan cómo los trinucleótidos. Estos últimos se cuentan mediante compseq con una ventana solapante. Los codones son también trinucleótidos, pero NO solapantes. Otra cuestión a tener en cuenta es que los codones hay que contarlos en el marco de lectura correcto. Para analizar la frecuencia de codones podemos usar por ejemplo el programa cusp.

```
cusp -help
```

```
michael@sRNAtoolbox:~$ cusp -help
Create a codon usage table from nucleotide sequence(s)
Version: EMBOSS:6.6.0.0

   Standard (Mandatory) qualifiers:
  [-sequence]          seqall     Nucleotide sequence(s) filename and optional
                                  format, or reference (input USA)
  [-outfile]           outfile    [*.cusp] Output file name

   Additional (Optional) qualifiers: (none)
   Advanced (Unprompted) qualifiers: (none)
   General qualifiers:
   -help                boolean    Report command line options and exit. More
                                   information on associated and general
                                   qualifiers can be found with -help -verbose
```

Cusp solo tiene dos parámetros, el fichero de entrada y el fichero de salida. Recordamos que la entrada tiene que corresponder a una secuencia CDS (región codificante). Cómo obtener dicha secuencia lo vimos en este vídeo

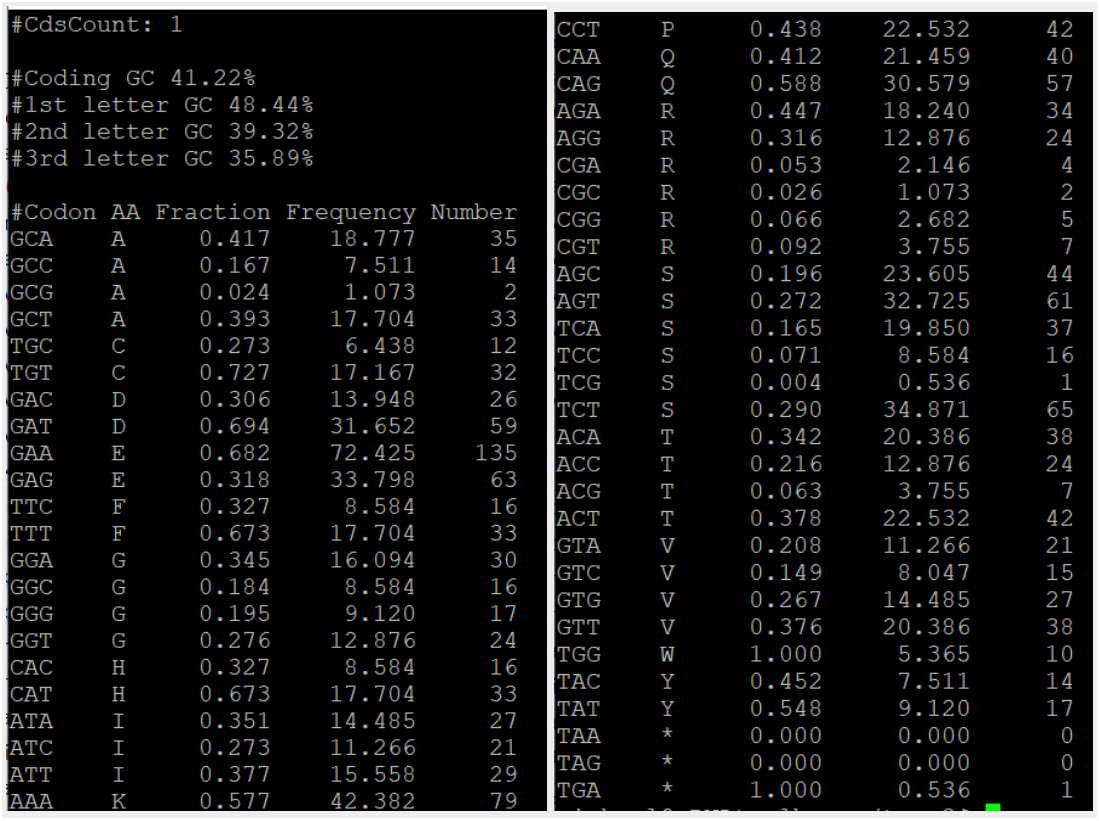
Ejercicio:

Descargamos primero la CDS de la isoforma 1 del gene BRCA1 y la guardamos con el nombre de NM_007295_CDS.fa

```
cusp -sequence NM_007295_CDS.fa -outfile NM_007295_CDS.cusp
```

La salida de cusp nos reporta:

#codon	: la secuencia del codón
AA	: el aminoácido correspondiente
Fraction	: la fracción del codón entre todos que codifican el mismo AA
Frequency	: la frecuencia esperada de este codón cada 1000 bases
Number	: La frecuencia observada de este codón



```
#CdsCount: 1

#Coding GC 41.22%
#1st letter GC 48.44%
#2nd letter GC 39.32%
#3rd letter GC 35.89%

#Codon AA Fraction Frequency Number
GCA     A      0.417    18.777     35
GCC     A      0.167     7.511     14
GCG     A      0.024     1.073      2
GCT     A      0.393    17.704     33
TGC     C      0.273     6.438     12
TGT     C      0.727    17.167     32
GAC     D      0.306    13.948     26
GAT     D      0.694    31.652     59
GAA     E      0.682    72.425    135
GAG     E      0.318    33.798     63
TTC     F      0.327     8.584     16
TTT     F      0.673    17.704     33
GGA     G      0.345    16.094     30
GGC     G      0.184     8.584     16
GGG     G      0.195     9.120     17
GGT     G      0.276    12.876     24
CAC     H      0.327     8.584     16
CAT     H      0.673    17.704     33
ATA     I      0.351    14.485     27
ATC     I      0.273    11.266     21
ATT     I      0.377    15.558     29
AAA     K      0.577    42.382     79
CCT     P      0.438    22.532     42
CAA     Q      0.412    21.459     40
CAG     Q      0.588    30.579     57
AGA     R      0.447    18.240     34
AGG     R      0.316    12.876     24
CGA     R      0.053     2.146      4
CGC     R      0.026     1.073      2
CGG     R      0.066     2.682      5
CGT     R      0.092     3.755      7
AGC     S      0.196    23.605     44
AGT     S      0.272    32.725     61
TCA     S      0.165    19.850     37
TCC     S      0.071     8.584     16
TCG     S      0.004     0.536      1
TCT     S      0.290    34.871     65
ACA     T      0.342    20.386     38
ACC     T      0.216    12.876     24
ACG     T      0.063     3.755      7
ACT     T      0.378    22.532     42
GTA     V      0.208    11.266     21
GTC     V      0.149     8.047     15
GTG     V      0.267    14.485     27
GTT     V      0.376    20.386     38
TGG     W      1.000     5.365     10
TAC     Y      0.452     7.511     14
TAT     Y      0.548     9.120     17
TAA     *      0.000     0.000      0
TAG     *      0.000     0.000      0
TGA     *      1.000     0.536      1
```

Salida del programa CUSP

- El programa nos da primero el número de secuencias (ya que acepta multi-fasta): CdsCount: 1
- Antes de la tabla del uso de codones, podemos ver el %GC en la secuencia entera, y en primera, segunda y tercera posición de los codones
- Observamos que no todos los codones de un AA se usan con la misma frecuencia (hay mas sesgo para algunos AA). Por ejemplo GCA y GCG codifican Alanina, el primero se usa 35 veces y el segundo solamente 2 veces.
- Es importante fijarse en el número de codones de parada (al final con '*'). ¡Si hay más de un codón de parada, la secuencia de entrada no ha sido una CDS! En este caso solamente hay uno, TGA.

Trabajar con la secuencia anónima

- Analisis composicional
- Uso de codones del gen (genes) que contenga

Ejercicios y problemas

1. **Comparar la secuencia del MHC con una bacteria**

Comparar la fluctuación del %G+C de la secuencia MHC con la de NC_006461.1

¿Qué diferencias observamos si comparamos la distribución espacial del G+C entre las dos secuencias?

2. **Analizar el G+C a lo largo de esta secuencia.**
    - ¿Que observamos?
    - ¿Qué explicación tiene?

3. **Comparar la composición de dinucleótidos**

Comparar la composición de la sec29 con una secuencia obtenida de C. elegans (¿Como?)
    - ¿Qué diferencias observamos?
    - ¿Qué explicación puede tener?

4. **Comparar la composición en dinucleótidos con una secuencia aleatorizada**

Aleatorizar (suffleseq) la sec29 y determinar el uso de dinucleótidos.

- ¿Qué diferencias hay en el uso de dinucleótidos entre la sec29 y su secuencia randomizada? ¿Por qué?
- Existen otros dinucleótidos en la sec29 cuya ratio obs/esp se desvía de 1. ¿A que se podría deber?

5. **Determinar el uso de codones con compseq**

El programa ***compseq***, por defecto determina la frecuencia de N-meros mediante ventanas solapantes. Sin embargo existe un parámetro que nos permite definir un 'salto' (como en el programa freak).

- ¿como se llama el parámetro y que valor tenemos que especificar?
- ¿Coinciden las frecuencias de codones?

6. **Uso de codones en una bacteria**

Analizar el uso de codones de la bacteria Thermus thermophilus

- ¿Qué observamos si comparamos el %G+C en las diferentes posiciones del codón?
- ¿Qué explicación puede tener?

Que hemos aprendido

- Ciertos elementos genómicos como promotores o elementos transponibles como los retrotransposones Alu son ricos en G+C. Otros como los LINE1 son ricos en A+T.

Analizando la distribución del %GC a lo largo de una secuencia, estos elementos se pueden manifestar como picos en %GC (o valles si son ricos en %AT)
- Mamíferos de sangre caliente tienen un genoma estructurado según el contenido en G+C. Regiones homogéneas en G+C se llaman isocoras. Isocoras ricas en G+C son mucho más densas en genes y tienen intrones más cortos entre otras propiedades.
- Para calcular la ratio entre la frecuencia observada y esperada de N-meros, tenemos basarnos en las frecuencias observadas de mononucleótidos.
- En el genoma de un mamífero detectamos típicamente sólo el 20-25% del número esperado del dinucleótido CG
- La ausencia se debe a la pérdida evolutiva del dinucleótido CG debido a la desaminación espontánea

Uso de línea de comando avanzado

Redirigir 'salidas'

Lanzamos el programa infoseq para obtener la ayuda

```
infoseq --help
```

Vemos que la salida no cabe en una pantalla y por lo tanto no podemos ver en el terminal el inicio de la ayuda. Para poder verla entera, recordamos algunas utilidades de Linux en línea de comando:

- Los pipe's → enlazar diferentes programas mediante el operador | en línea de comando (usar la salida estándar de un programa como entrada de otro)
- Redirigir la salida estandar a un fichero con > (o 2> para redirigir la salida de error 'stderr')
- Los programas more (or less) para visualizar el contenido de un fichero

Por lo tanto, las siguiente línea nos debería visualizar la ayuda en el editor 'more'

```
infoseq --help | more
```

No funciona, ya que este programa escribe la ayuda no por stdout sino por stderr. Como redirigir en este caso se explica en [este artículo](). Sería la siguiente orden:

```
infoseq --help  2>&1 > /dev/null | more
```