

Ramón Tamarit Agusti

FUNDAMENTOS DE INFORMÁTICA EN ENTORNOS BIOINFORMÁTICOS

PEC1 — Primera Prueba de evaluación continua

Ejercicio 1 – Descripción de los catálogos de genes. Análisis comparativo de los genomas de varias especies (catálogos de genes disponibles a día de hoy).

Ejercicio 1 – Descripción de los catálogos de genes [30%]

Análisis comparativo de los genomas de varias especies (catalogos de genes disponibles a día de hoy). De la misma forma que hemos realizado durante los ejercicios prácticos, conectaos al servidor UCSC para acceder a los ficheros "refGene.txt" de varias especies. El objetivo es que relleneis la siguiente Tabla con los datos que obtendreis usando los comandos apropiados en vuestro terminal de LINUX. Añadid una pequeña interpretación biológica de los resultados anotados:

Species	# Transcripts	# Genes	Average Length Transcripts	Average # exons per transcript
<i>H. sapiens</i>				
<i>M. musculus</i>				
<i>G. gallus</i>				
<i>D. melanogaster</i>				
<i>C. elegans</i>				

NOTA: El símbolo "#" denota "número de" (cardinal)

Recordad que la estructura de estos ficheros es una línea por transcrito (RNA mensajero, RNAm) y que un gen puede dar lugar a más de un transcrito alternativo (isoforma).

Por ejemplo, este es un extracto del correspondiente fichero en *D.melanogaster*:

```

741 NM_170159 chr3R + 20475664 20479099 20475692 20478924 8
20475664,20475956,20476126,20476371,20477582,20478031,20478503,20478819,
20475899,20476066,20476311,20476716,20477963,20478312,20478761,20479099, 15170
ash2 cml cml 0,0,2,1,1,1,0,0,

741 NM_176558 chr3R + 20475809 20479099 20475950 20478924 7
20475809,20476126,20476371,20477582,20478031,20478503,20478819,
20476066,20476311,20476716,20477963,20478312,20478761,20479099, 15171
ash2 cml cml 0,2,1,1,1,0,0,

741 NM_170160 chr3R + 20477248 20479098 20477324 20478924 5
20477248,20477582,20478031,20478503,20478819,
20477352,20477963,20478312,20478761,20479098, 15172
ash2 cml cml 0,1,1,0,0,
    
```

Donde hay 3 RNAm (cada uno con su identificador NM) para un mismo gen (*ash2*). Esto significa que este gen *ash2* tiene 3 formas alternativas de transcripción. Gráficamente, este es el gen y sus tres isoformas (con sus exones):



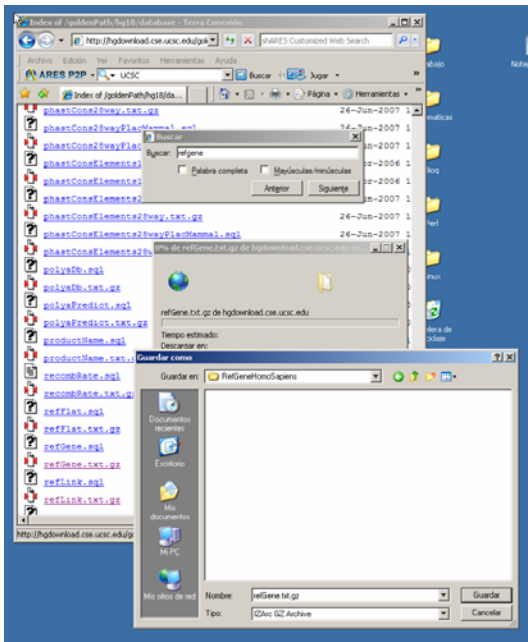
Ficheros refGene.txt:

- UCSC web: <http://hgdownload.cse.ucsc.edu/downloads.html>
[seleccionad la especie y buscar el directorio "Annotation database"]

1. Paso 1. Preaparación y Obtención de los ficheros.

Para trabajar los datos usare el terminal de bash de cygwin. Previamente he instalado las ultimas versiones de los módulos de gawk. Como editor de texto usaré indistintamente el notepad2 (<http://www.flos-freeware.ch/notepad2.html>) y xemacs. La ejecución de todos estos comandos en cygwin es equivalente a hacerlo en un Terminal "Linux" o "Unix", con la salvedad que el entorno grafico nos permite observar con facilidad la estructura de los ficheros para analizarlos.

Una vez obtenidos los ficheros del servidor de UCSC (en fecha 18.11.2008), los descomprimo (*gunzip nombrefichero*) los ficheros en cada una de sus carpetas y los renombro de la siguiente forma. (Uso *cp nombrei nombref* para copiar y renombrar a la vez)

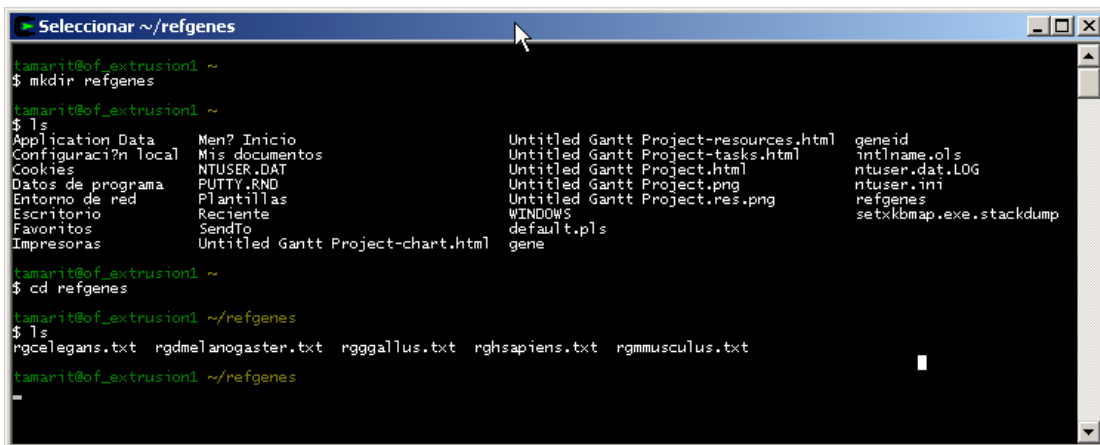


- H. sapiens rghsapiens.txt
- M. musculus rgmmusculus.txt
- G. gallus rgggallus.txt
- D. melanogaster rgdmelanogaster.txt
- C. elegans rgcelegans.txt

Los ficheros txt son los que usaré en el terminal de Unix.

El directorio de trabajo sera `~/refgenes` , dentro de el creo otro directorio `/cpy` y pongo otra copia sobre la que trabajaré. (ver figura de abajo)

Vamos al servidor de UCSC y descargamos los ficheros de anotaciones de cada especie. Cada uno de los ficheros lo guardo en un directorio para que no se mezclen.



Estructura de directorios después de descomprimir, renombrar los ficheros y traspasarlos al directorio de trabajo de Unix

2. Estructura de los ficheros. RefGene.txt.

Del servidor de UCSC se puede consultar la estructura de la tabla RefGene. La estructura la detallo a continuación:

field	example	Description
bin	1643	Indexing field to speed chromosome range queries.
name	NM_016459	Name of gene (usually transcript_id from GTF)
chrom	chr5	Reference sequence chromosome or scaffold
strand	-	+ or - for strand
txStart	138751155	Transcription start position
txEnd	138753504	Transcription end position
cdsStart	138751352	Coding region start
cdsEnd	138753444	Coding region end
exonCount	4	Number of exons
exonStarts	138751155,138751608,1387520...	Exon start positions
exonEnds	138751509,138751719,1387521...	Exon end positions
id	0	Unique identifier
name2	MGC29506	Alternate name (e.g. gene_id from GTF)
cdsStartStat	cmpl	enum('none','unk','incompl','cmpl')
cdsEndStat	cmpl	enum('none','unk','incompl','cmpl')
exonFrames	2,2,0,0,	Exon frame {0,1,2}, or -1 if no frame for exon

Los campos que nos interesan son:

- Campo 5 "txStart" . Inicio del transcrito
- Campo 6 "txEnd", final del transcrito
- Campo 9 "exonCount" : Número de exones
- Campo 12 "name2" : Nombre del gen

3. Cálculo del número de transcritos.

El número de transcritos se puede obtener del conteo de las filas de cada fichero. Para ello usará directamente el comando `wc rg*`. La primera columna es el número de líneas (transcritos). Igualmente se puede ejecutar alternativamente el comando `wc -l rg*` que nos dará únicamente el número de líneas:

```
tamarit@of_extrusion1 ~/refgenes
$ wc rg*
 24978   398938   4983895  rgcelegans.txt
 25591   409456   4324523  rgdmelanogaster.txt
  4355    69685   1205268  rgggallus.txt
 29363   469808   8615054  rghsapiens.txt
 22246   355936   6472117  rgmmusculus.txt
106533  1703823  25600857  total
```

```
tamarit@of_extrusion1 ~/refgenes
$ wc -l rg*
 24978  rgcelegans.txt
 25591  rgdmelanogaster.txt
  4355  rgggallus.txt
 29363  rghsapiens.txt
 22246  rgmmusculus.txt
106533  total
```

Vamos a ordenarlos para estudiar el fichero. Para ello voy a ir preparando un script que puede serme útil en el futuro. En principio a partir de ahora trabajare sobre los ficheros de ~/refgenes/cpy.

```
ordenar.scr - Notepad2
1 #!/bin/bash
2
3 echo *****ORDENANDO LOS FICHEROS*****
4 wc rg*
5 echo *****
6 sort -n rgcelegans.txt>rgcelegansS1.txt
7 sort -n rghsapiens.txt>rghsapiensS1.txt
8 sort -n rgdmelanogaster.txt>rgdmelanogasterS1.txt
9 sort -n rgmmusculus.txt>rgmmusculusS1.txt
10 sort -n rggallus.txt>rggallusS1.txt
11 echo *****
12 wc rg*
13 echo *****FIN ORDENANDO LOS FICHEROS*****

~/ordenar.scr
*****ORDENANDO LOS FICHEROS*****
24978 398938 4983895 rgcelegans.txt
25591 409456 4324523 rgdmelanogaster.txt
4355 69685 1205268 rggallus.txt
29363 469808 8615054 rghsapiens.txt
22246 355936 6472117 rgmmusculus.txt
106533 1703823 25600857 total
ordenar.scr rgcelegans.txt rgdmelanogaster.txt rggallus.txt rghsapiens.txt
ordenar.scr rgcelegansS1.txt rgcelegansS1.txt rgdmelanogaster.txt rgdmelanogasterS1.txt
ordenar.scr rggallus.txt rggallusS1.txt rgmmusculus.txt rgmmusculusS1.txt
24978 398938 4983895 rgcelegans.txt
24978 398938 4983895 rgcelegansS1.txt
25591 409456 4324523 rgdmelanogaster.txt
25591 409456 4324523 rgdmelanogasterS1.txt
4355 69685 1205268 rggallus.txt
4355 69685 1205268 rggallusS1.txt
29363 469808 8615054 rghsapiens.txt
29363 469808 8615054 rghsapiensS1.txt
22246 355936 6472117 rgmmusculus.txt
22246 355936 6472117 rgmmusculusS1.txt
213066 3407646 51201714 total
*****FIN ORDENANDO LOS FICHEROS*****
tamarit@of_extrusion1: ~/refgenes/cpy
$
```

Y el resultado:

```
rgcelegansS1.txt - Notepad2
1 1 NM_001028783 chrV + 8387334 8391926 8387334 8391926 19 {
2 1 NM_001028784 chrV + 8387334 8392303 8387334 8392303 21 {
3 1 NM_001029476 chrX - 8387903 8392521 8387903 8392521 9 83
4 1 NM_001029477 chrX - 8387409 8392521 8387903 8392521 10 {
5 1 NM_001029478 chrX - 8387406 8392521 8387903 8392521 8 83
6 1 NM_001029480 chrX - 8387903 8392522 8387903 8392521 10 {
7 1 NM_001129046 chrI - 8378297 8392731 8378297 8392590 9 83
8 1 NM_001129140 chrII + 8388055 8389465 8388266 8389397 8 83
9 1 NM_001129550 chrV + 16764228 16787944 16764228 1678794
10 1 NM_059873 chrI - 8378297 8390020 8378297 8390011 8 837829
11 1 NM_063362 chrII + 8388266 8389816 8388266 8389693 8 838826
12 1 NM_066449 chrIII - 8388425 8389558 8388425 8389558 3 83884
13 1 NM_070587 chrIV + 16773788 16785016 16773788 16785016 :
14 1 NM_074838 chrV + 16747529 16788386 16747592 16787944 :
15 1 NM_182066 chrI - 8377505 8392731 8378297 8392590 10 8377
16 1 NM_023951 chrII + 8388266 8389693 8389693 8389693 8 838826
17 9 NM_001026075 chrIII + 7338927 7343553 7338927 7343500 14
18 9 NM_001026076 chrIII + 7338927 7343500 7338927 7343500 14
19 9 NM_001026929 chrII + 7339202 7349047 7339202 7348907 10 :
20 9 NM_001026931 chrII + 7339168 7348907 7346455 7348907 10 :
21 9 NM_001026932 chrII + 7339175 7349061 7339202 7348907 10 :
22 9 NM_001027445 chrIII + 7336513 7343559 7336549 7338116 20
23 9 NM_001027721 chrIII + 2092748 2099871 2092750 2099622 7 :
24 9 NM_001027722 chrIII + 2095749 2099622 2095749 2099622 6 :
```

El tamaño del transcrito será el campo 6 menos el 5

```
$ gawk '{print $13,$6-$5}' rgcelegansS1.txt > test.txt
```

El numero de exones esta en la posición 9

```
$ gawk '{print $13,$6-$5,$9}' rgcelegansS1.txt > test.txt
```

El nombre del gen único esta en el campo 13

4. Contando los genes de cada especie

Vamos a contar los genes. Se identifican por el campo 13. Partimos de las copias y de los ficheros ordenados por el script, es decir los *S1.txt

Ejemplo:

```
$ gawk '{print $13}' rgcelegansS1.txt > rgcelegansGenes.txt
```

Luego ordenamos con sort -u (para eliminar los duplicados). Con wc contamos las ocurrencias.

Al final el script queda:

```

ordenar1.scr - Notepad2
File Edit View Settings ?
1 #!/bin/bash
2
3 #*****
4 #borrando los ficheros de pruebas anteriores
5 rm rg*S1*
6 rm rg*Genes*
7 rm rg*GenesCount*
8
9
10
11 for fn in `ls rg*`
12 do
13
14 #Se repite para cada fichero rg* en el directorio del
15 #script. en la variable fn se guarda el nombre del
16 # fichero en curso.....
17
18 echo "*****fichero $fn *****"
19 echo "*****CONTEO DE EXONES Y TAMAÑO*****"
20 #contar exones y su promedio
21
22 #l es el contador de exones >>>en el campo $9
23 #m es el contador de longitud >>> campo $6-$5
24 #NR es una variable interna que nos da el numero de filas
25 gawk 'BEGIN{l=0;m=0} {l=l+$9;m=m+($6-$5)}
26         END {print FILENAME, "número Exones =", l,
27         "promedio Exones =",l/NR,
28         "promedio Longitud =", m/NR}' $fn
29
30
31 #echo "*****ORDENANDO LOS FICHEROS*****"
32 #wc -l $fn ##me sirve para comprobar descomentando
33 #echo "*****"
34
35 FichOUT=$fn.S1
36 sort -n $fn > $FichOUT
37 fichIN=$fn.S1
38 FichOUT=$fn.Genes
39
40
41 #LOS FICHEROS *.Genes me sirven para comprobar que estoy
42 #haciendo las cuentas bien.....
43
44 #
45 gawk '{print $6-$5,$9,$13}' $fichIN | sort > $FichOUT
46
47 #echo "*****"
48 fichIN=$fn.Genes
49 FichOUT=$fn.GenesCount
50
51 #los fichero *.GenesCount contienen una linea por gen con
52 # Su nombre correspondiente.
53 # uniq fichero : quita los repetidos
54
55 gawk '{print $3}' $fichIN | uniq > $FichOUT
56 echo "*****FIN ORDENANDO LOS FICHEROS*****"
57 echo
58 done
59
60 # Presenta en pantalla el conteo de genes
61
62 echo "***** NUMERO DE GENES *****"
63 wc rg*Genes*

```

Ln 9:63 Col 1 Sel 0 1,72 KB ANSI LF INS Default Text

Y en la pantalla:

```

Seleccionar ~/refgenes/cpy
tamarit@of_extrusion1 ~/refgenes/cpy
$ ./ordenar1.scrpt
rm: cannot remove `rg*GenesCount*': No such file or directory
*****fichero rgcelegans.txt *****
*****CONTEO DE EXONES Y TAMANO*****
rgcelegans.txt número Exones = 159741 promedio Exones = 6.39527 promedio Longitud = 3300.38
*****FIN ORDENANDO LOS FICHEROS*****

*****fichero rgdmelanogaster.txt *****
*****CONTEO DE EXONES Y TAMANO*****
rgdmelanogaster.txt número Exones = 111007 promedio Exones = 4.33774 promedio Longitud = 11139.6
*****FIN ORDENANDO LOS FICHEROS*****

*****fichero rgggallus.txt *****
*****CONTEO DE EXONES Y TAMANO*****
rgggallus.txt número Exones = 42987 promedio Exones = 9.87072 promedio Longitud = 27876.6
*****FIN ORDENANDO LOS FICHEROS*****

*****fichero rghsapiens.txt *****
*****CONTEO DE EXONES Y TAMANO*****
rghsapiens.txt número Exones = 302388 promedio Exones = 10.2983 promedio Longitud = 58158.8
*****FIN ORDENANDO LOS FICHEROS*****

*****fichero rgmmusculus.txt *****
*****CONTEO DE EXONES Y TAMANO*****
rgmmusculus.txt número Exones = 219060 promedio Exones = 9.84716 promedio Longitud = 47155.9
*****FIN ORDENANDO LOS FICHEROS*****

***** NUMERO DE GENES *****
24978 74934 370435 rgcelegans.txt.Genes
23787 23787 188057 rgcelegans.txt.GenesCount
25591 76773 393757 rgdmelanogaster.txt.Genes
19401 19401 141327 rgdmelanogaster.txt.GenesCount
4355 13065 61974 rgggallus.txt.Genes
4329 4329 27058 rgggallus.txt.GenesCount
29363 88089 434382 rghsapiens.txt.Genes
23886 23886 158975 rghsapiens.txt.GenesCount
22246 66738 340210 rgmmusculus.txt.Genes
20968 20968 151399 rgmmusculus.txt.GenesCount
198904 411970 2267574 total

tamarit@of_extrusion1 ~/refgenes/cpy
    
```

Y una muestra de los ficheros intermedios.

```

rgmmusculus.txt.Genes - Notepad2
1 10000 2 B1cap
2 100000 20 Rgs3
3 100006 31 sce1
4 10001 5 Spink8
5 10002 2 OTTMUSG00000002038
6 10002 7 Cwc15
7 100020 12 012Ertd553e
8 100036 29 Ccdc18
9 10004 14 Irak1
10 100054 11 Map2k4
11 100064 9 Scm14
12 10007 5 AV320801
13 10007 5 OTTMUSG00000018782
14 10008 3 Nr2f1
15 10009 5 AV320801
16 10009 5 OTTMUSG00000018782
17 10010 2 K1hdc6
18 10012 6 Zfand5
19 100124 14 Epc2
20 10015 12 Lck
21 10015 7 Tbx22
22 100162 8 9030409G11R1k
23 100176 9 Nrg1
24 100184 15 Mtm1
25 100197 27 Inpp5d
26 100197 27 Inpp5d
27 100197 28 Inpp5d
28 1002 1 Krtap5-4
29 1002 1 olfr12
30 1002 1 olfr304
31 1002 1 olfr324
32 1002 1 olfr518
33 1002 1 Tas2r110
34 1002 1 Tas2r122

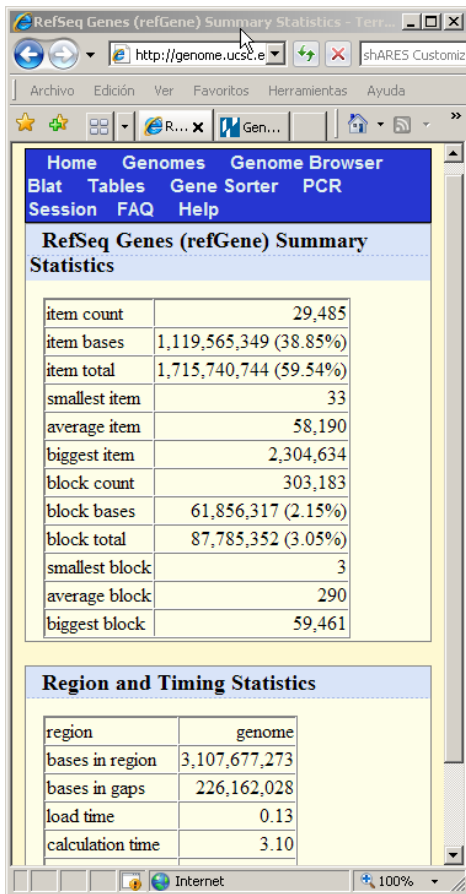
rgmmusculus.txt.GenesCount - Notepad2
232 Hemgn
233 Rbp3
234 B430406I07R1k
235 G1rp1
236 Clmn
237 Ppil5
238 K1h114
239 Fpr-rs6
240 V1rc17
241 Slc7a15
242 Slc7a60s
243 Fat1
244 Prr8
245 Cope
246 AI646023
247 Mapksp1
248 Atp11b
249 E2f1
250 Ccdc5
251 Kctd4
252 Cript
253 Cntn4
254 Tgif21x
255 Ttc9c
256 Dnajc5b
257 OTTMUSG00000010432
258 Sacs
259 Crtc3
260 Ankrd34c
261 Ppargc1b
262 Tdpoz5
263 Defb11
264 4930451I11R1k
265 An2k1
    
```

>> Estos ficheros los uso para comprobar que la salida que obtengo es realmente la que busco con los comandos.

5. Resultados finales

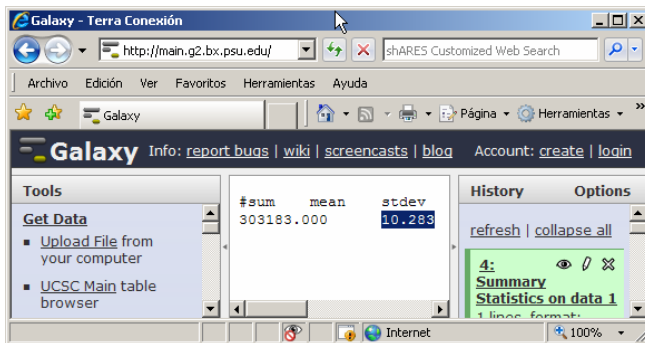
	Nº de transcritos	Promedio longitud transcritos	Promedio Exones por transcrito	Nº de genes
H. sapiens	29363	58159	10,298	23886
M. musculus	22246	47156	9,847	20968
G. gallus	4355	27897	9,871	4329
D. melanogaster	25591	11140	4,338	19401
C. elegans	24987	3300	6,395	23787

(Algunos valores los he redondeado)



6. Interpretación biológica de los resultados.

- Como demuestran los datos, el número de genes no es proporcional a lo desarrollado de una especie. Como curiosidad el C. elegans (un pequeño gusanito) tiene casi tantos genes como los humanos y mas de 4 veces que los gallos.
- El organismo comparado con las estadísticas (se entiende que son las de la tabla de la sección 5) más similares a los humanos es el ratón (m. musculus). De ahí que sea uno de los organismos mayormente utilizados en laboratorio para experimentación.
- Parece que el mejor parámetro para medir la complejidad bioquímica de un organismo es el promedio de exones por transcrito. Hay claramente dos ordenes de magnitud: El de los vertebrados (sobre 10) y el de los insectos (entre 6 y 4).
- Igualmente, un organismo complejo muestra una mayor longitud de los transcritos. Este hecho se constata no solo por los resultados obtenidos sino también por la cantidad de citas bibliográficas que emplean este parámetro.
- El número de transcritos en principio no puede ser comparable, ya que puede depender de la cantidad de análisis que se haya hecho del organismo, y del estado de resolución experimental de su genoma.



Comprobación a 28.11.2008 de RefGene para H. sapiens. Vemos que ya ha crecido el numero de registros en de 29363 a 29485, el promedio de longitud es 58190 (58159 en nuestro cálculo). Y la media de exones por transcrito es de 10.28 (desde galaxy) y coincide con nuestros cálculos.

Probando todo en el OpenSuse Linux.

Para comprobar que todo el trabajo es reproducible lo ejecuto desde el OpenSuse que tengo instalado. Muevo con el winspc al home y allí ejecuto el ordenar1.scr

Y vemos que da lo mismo.

```

Linux - X-Win32
xterm
linux:/home/tamarit/refgenes/cpy # ls
.,directory  ordenar1.scr  rgdmelanogaster.txt  rghsapiens.txt
ordenar1.scr  rgcelegans.txt  rgggallus.txt  rgmmusculus.txt
linux:/home/tamarit/refgenes/cpy # ./ordenar1.scr
bash: ./ordenar1.scr: Permission denied
linux:/home/tamarit/refgenes/cpy # chmod 777 ./ordenar1.scr
linux:/home/tamarit/refgenes/cpy # ./ordenar1.scr
rm: cannot remove `rg*S1*`: No such file or directory
rm: cannot remove `rg*Genes*`: No such file or directory
rm: cannot remove `rg*GenesCount*`: No such file or directory
*****fichero  rgcelegans.txt  *****
*****CONTEO DE EXONES Y TAMA *****
rgcelegans.txt n ero Exones = 159741 promedio Exones = 6,39527 promedio Longitud
= 3300,38
*****FIN ORDENANDO LOS FICHEROS*****
*****fichero  rgdmelanogaster.txt  *****
*****CONTEO DE EXONES Y TAMA *****
rgdmelanogaster.txt n ero Exones = 111007 promedio Exones = 4,33774 promedio Lon
gitud = 11139,6
*****FIN ORDENANDO LOS FICHEROS*****
*****fichero  rgggallus.txt  *****
*****CONTEO DE EXONES Y TAMA *****
rgggallus.txt n ero Exones = 42987 promedio Exones = 9,87072 promedio Longitud =
27876,6
*****FIN ORDENANDO LOS FICHEROS*****
*****fichero  rghsapiens.txt  *****
*****CONTEO DE EXONES Y TAMA *****
rghsapiens.txt n ero Exones = 302388 promedio Exones = 10,2983 promedio Longitud
= 58158,8
*****FIN ORDENANDO LOS FICHEROS*****
*****fichero  rgmmusculus.txt  *****
*****CONTEO DE EXONES Y TAMA *****
rgmmusculus.txt n ero Exones = 219060 promedio Exones = 9,84716 promedio Longitu
d = 47155,9
*****FIN ORDENANDO LOS FICHEROS*****
***** NUMERO DE GENES *****
24978  74934  370435  rgcelegans.txt,Genes
23787  23787  188057  rgcelegans.txt,GenesCount
25591  76773  393757  rgdmelanogaster.txt,Genes
19401  19401  141327  rgdmelanogaster.txt,GenesCount
4355  13065  61974  rgggallus.txt,Genes
4329  4329  27058  rgggallus.txt,GenesCount
29363  88089  434382  rghsapiens.txt,Genes
23886  23886  158975  rghsapiens.txt,GenesCount
22246  66738  340210  rgmmusculus.txt,Genes
20968  20968  151399  rgmmusculus.txt,GenesCount
198904  411970  2267574  total
linux:/home/tamarit/refgenes/cpy #

```

Ejercicio 2 – Anotación funcional de genes

Ejercicio 2 – Anotación funcional de genes [30%]

GeneOntology es un diccionario de conceptos biológicos que son utilizados para definir las funciones de los genes. Echale un vistazo a estas páginas y describe en pocas líneas en qué consiste:

-
- <http://www.geneontology.org/>
 - http://en.wikipedia.org/wiki/Protein_ontology
-

Ahora vamos a trabajar con la anotación funcional GO del genoma de la mosca de la fruta. En este caso, en lugar de usar los identificadores NM, utilizaremos el fichero de genes de la base de datos de esta especie (FlyBase). Adjuntos a este enunciado tenéis los ficheros (comprimidos) `gene_ontology.dat` y `gene_association.fb` (ambos de texto sin formato).

El fichero `gene_ontology.dat` contiene todas las definiciones (indexadas por su GOid) que pueden asignarse a los genes:

```
GO:0000001    mitochondrion inheritance
GO:0000002    mitochondrial genome maintenance
GO:0000003    reproduction
```

El fichero `gene_association` contiene las definiciones GO que están asociadas a los genes de la mosca de la fruta (*D.melanogaster*):

```
FB      FBgn0000139      ash2      GO:0048096  FB:FBF0085406|PMID:8555105 IMP P
absent,  small,  or  homeotic  discs  2  1124/11|291.8|703|ASH2|CG6677|ash-
2|1(3)112411|1(3)8112411|1(3)8G65|mad|many  abnormal  discs  gene  taxon:7227
20060803  FlyBase
```

Debeis contestar las siguientes preguntas usando los comandos apropiados de LINUX en vuestro terminal:

- ¿ Cuántos y cuáles genes de la mosca están relacionados con la regulación de la transcripción?
- ¿ Cuántos y cuáles genes están asociados a los discos imaginales de ala de la mosca?

NOTA: Podeis ver qué es un disco imaginal en la siguiente dirección:

http://es.wikipedia.org/wiki/Disco_imaginal

Definición de GeneOntology

El proyecto GO es básicamente un vocabulario de términos biológicos que describen los genes y los productos génicos, y su regulación, estableciendo anotaciones de los términos a la información disponible.

El vocabulario consta de tres diccionarios de términos u ontologías:

- Función molecular de los genes y sus productos (proteínas, mARN, etc): Ej: "Actividad catalítica",
- Procesos biológicos Ej: "Transporte alfa-glucosidico),
- Componentes celulares. Ej "Ribosoma",

Cada entrada del diccionario tiene asociado un índice (GOid) único sus sinónimos y su definición. Cada uno de los términos está asociado con el resto como un grafo acíclico, de forma que se pueden establecer relaciones diversas entre los términos (un grafo acíclico no tiene principio ni fin y a un vértice se puede llegar desde distintos caminos).

Por otra parte, se desarrollan bases de datos de estudios genéticos de los organismos en donde se incluyen las asociaciones de los productos génicos con los términos go. Una entrada en la tabla de asociaciones: En donde se indica principalmente (en el siguiente apartado incluyo los campos de la tabla):

- El gen o producto génico relacionado,
- El GOid,
- Quien y cuando ha hecho la anotación,
- La base de datos donde localizar la información,
- La evidencia de la anotación.

Mediante el uso combinado de las tablas de términos y de anotaciones podemos realizar consultas que nos ayudarán a localizar toda la información relacionada con un termino Go.

El formato de anotación GO

De la web de UCSC busco la definición de la tabla que usaremos más adelante. Los campos campos

The gene_association.cgd.gz file uses the standard file format for gene_association files of the Gene Ontology (GO) Consortium. A more

complete description of the file format is found here:

<http://www.geneontology.org/doc/GO.annotation.html#file>

Columns are:	Contents:
1) DB file (always "CGD" for this file)	- database contributing the
2) DB_Object_ID	- CGDID
3) DB_Object_Symbol	- see below
4) Qualifier (optional) 'colocalizes_with' qualifier for a GO annotation, when needed	- 'NOT', 'contributes_to', or
5) GO ID for the GO term	- unique numeric identifier
6) DB:Reference(DB:Reference) with the GO annotation	- the reference associated
7) Evidence GO annotation	- the evidence code for the
8) With (or) From (optional) for the GO annotation	- any With or From qualifier
9) Aspect belongs in (see note below)	- which ontology the GO term
10) DB_Object_Name(Name) (optional) words, e.g. 'acid phosphatase'	- a name for the gene product in
11) DB_Object_Synonym(Synonym) (optional)	- see below
12) DB_Object_Type e.g. gene, protein, etc.	- type of object annotated,
13) taxon(taxon) encoding gene product	- taxonomic identifier of species
14) Date	- date GO annotation was made
15) Assigned_by (always "CGD" for this file)	- source of the annotation

Note on CGD nomenclature (pertaining to columns 3 and 11):

Column 3 - When a Standard Gene Name (e.g. CDC28, COX2) has been conferred, it will be present in Column 3. When no Gene Name has been conferred, the ORF Name (e.g., orf19.6632) will be present in column 3.

Column 11 - The ORF Name (e.g., orf19.6632) will be the first name present in Column 11. Any other names (except the Standard Name, which will be in Column 3 if one exists), including Aliases used for the gene will also be present in this column.

Note on Aspect (column 9):
 C = Cellular Component
 F = Molecular Function
 P = Biological Process

7. Genes de la mosca implicados en la regulación de la transcripción.

Primero buscamos la palabra "transcription" y "regulation" en el diccionario

```
tamarit@of_extrusion1 ~/go
$ grep transcription gene_ontology.dat>go_trasncrption.txt
```

(grep nos separa las líneas que contienen el argumento "gene" y lo guardamos en un fichero de texto desviando la salida con ">")

```
tamarit@of_extrusion1 ~/go
$ grep transcription gene_ontology.dat | grep regulation>go_busqueda.txt
```

```
tamarit@of_extrusion1 ~/go
$
```

```
58 GO:0045945 positive regulation of transcription from RNA polymerase III promoter
59 GO:0046024 positive regulation of transcription from RNA polymerase III promoter, mitotic
60 GO:0007221 positive regulation of transcription of Notch receptor target
61 GO:0045889 positive regulation of transcription of homeotic gene (Polycomb group)
62 GO:0006339 positive regulation of transcription of homeotic gene (trithorax group)
63 GO:0007072 positive regulation of transcription on exit from mitosis
64 GO:0007073 positive regulation of transcription on exit from mitosis, from RNA polymerase I promoter
65 GO:0007074 positive regulation of transcription on exit from mitosis, from RNA polymerase II promoter
66 GO:0007075 positive regulation of transcription on exit from mitosis, from RNA polymerase III promoter
67 GO:0045893 positive regulation of transcription, DNA-dependent
68 GO:0045895 positive regulation of transcription, mating-type specific
69 GO:0051039 positive regulation of transcription, meiotic
70 GO:0045897 positive regulation of transcription, mitotic
71 GO:0045899 positive regulation of transcriptional preinitiation complex assembly
72 GO:0050434 positive regulation of viral transcription
73 GO:0060194 regulation of antisense RNA transcription
74 GO:0032583 regulation of gene-specific transcription
75 GO:0060147 regulation of posttranscriptional gene silencing
76 GO:0010551 regulation of specific transcription from RNA polymerase II promoter
77 GO:0045449 regulation of transcription
78 GO:0045990 regulation of transcription by carbon catabolites
79 GO:0000409 regulation of transcription by galactose
80 GO:0046015 regulation of transcription by glucose
81 GO:0009373 regulation of transcription by pheromones
82 GO:0051090 regulation of transcription factor activity
83 GO:0042990 regulation of transcription factor import into nucleus
84 GO:0006356 regulation of transcription from RNA polymerase I promoter
85 GO:0046017 regulation of transcription from RNA polymerase I promoter, mitotic
```

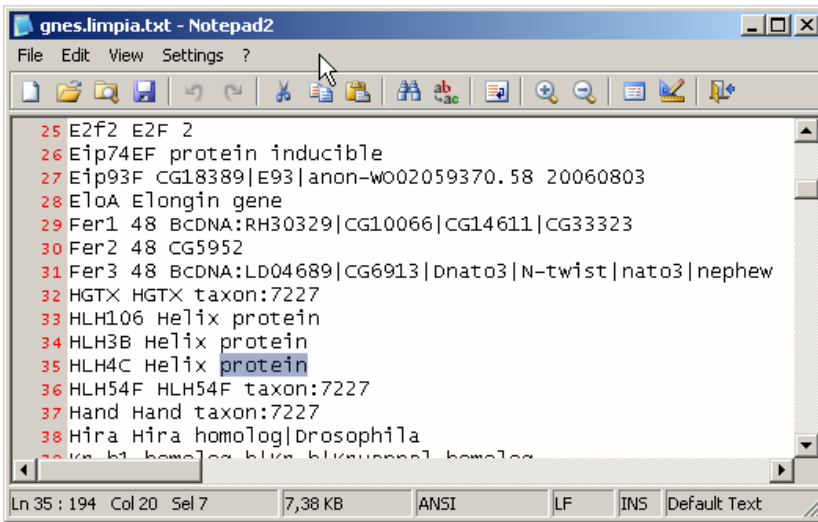
La salida anterior nos sirve para identificar el GOid que usaremos en la búsqueda de anotaciones.

```
Separamos las entradas con GO:0045449
tamarit@of_extrusion1 ~/go
$ grep GO:0045449 gene_association.fb>gnes.txt
```

```
tamarit@of_extrusion1 ~/go
```

```
136 FB FBgn0008651 lbl GO:0045449 FB:FBF0105495 ISS UniProt:P52955 P ladybird late CG6570|Ladybird|Lb|Lb|NK-cluster homeobox 4|NK?|NKch4|lac
137 FB FBgn0039039 lmd GO:0045449 FB:FBF0136559 ISS P lame duck CG4677|sleefu|KlLame duck|Lame duck|Lameduck|Lmd|Hobblasts 4|incompetent|anor
138 FB FBgn0040765 luna GO:0045449 FB:FBF0157541 NAS P luna CG17326|CG33473|CG9087|DKLF|KLF|K1F gene taxon:7227 20060803 FlyBase
139 FB FBgn0005511 mid GO:0045449 FB:FBF0137013 PMID:11404084 ISS FB:FBgn0022740 P net CG11450|Group IId|Shout|shout gene taxon:7227 200608
140 FB FBgn0027548 nito GO:0045449 FB:FBF0174215 IEA InterPro:IPR010912 P spn1to bcDNA:G011110|CG2910|bm44/spn1to|dms5|one twenty two|cc
141 FB FBgn0003013 osa GO:0045449 FB:FBF0107412 PMID:9895321 IMP P osa CG7467|E(E2F)3C|OSA|osa|anon-W00118547.314|anon-W00172774.126|eld|en|
142 FB FBgn0003013 osa GO:0045449 FB:FBF0124978 PMID:10601025 IDA P osa CG7467|E(E2F)3C|OSA|osa|anon-W00118547.314|anon-W00172774.126|eld|en|
143 FB FBgn0015524 otp GO:0045449 FB:FBF0105495 ISS EMBL:AY651764 P orthopedia bcDNA:RE58095|CG10036|CG2965|orthopedia|otpl|w26|bk24|w26 gene
144 FB FBgn0003028 ovo GO:0045449 FB:FBF0159232 PMID:12612640 TAS P ovo CG15467|CG6824|Fs(1)K1103|Fs(1)K1237|Fs(1)K155|ovo|ovo-d|Shavent
145 FB FBgn0069093 p170 GO:0045449 FB:FBF0173638 IMP P p170 gene taxon:7227 20060803 FlyBase
146 FB FBgn0069093 p170 GO:0005634 FB:FBF0173638 IC GO:0045449|GO:0003677 C p170 gene taxon:7227 20060803 FlyBase
147 FB FBgn0003053 peb GO:0045449 FB:FBF0105495 ISS EMBL:AF013754 P pebbled CG12212|EG:66A1.1|EP55|HNT|Hind|Hindsight|Hnt|PEB|anon-4Cg|hindsi
148 FB FBgn0003053 peb GO:0045449 FB:FBF0151872 PMID:12231351 NAS P pebbled CG12212|EG:66A1.1|EP55|HNT|Hind|Hindsight|Hnt|PEB|anon-4Cg|hindsi
149 FB FBgn0003053 peb GO:0045449 FB:FBF0113498 NAS P pleiohomeotic CG17743|PHO|Pho|Pleiohomeotic|yv1|1(4)102Efc|1(4)29|1(4)BU-2|1(4)OC-1|1(
150 FB FBgn0002521 pho GO:0045449 FB:FBF0200379 IMP P pleiohomeotic CG17743|PHO|Pho|Pleiohomeotic|yv1|1(4)102Efc|1(4)29|1(4)BU-2|1(4)OC-1|1(
151 FB FBgn0002521 pho GO:0045449 FB:FBF0200379 IMP P pleiohomeotic CG17743|PHO|Pho|Pleiohomeotic|yv1|1(4)102Efc|1(4)29|1(4)BU-2|1(4)OC-1|1(
152 FB FBgn0028579 phrF GO:0045449 FB:FBF0174215 IEA InterPro:IPR001092 P phrF bcDNA:GH08636|bcDNA:GH08636|CG32681d-phrF gene taxon:7227
153 FB FBgn0003117 pnt GO:0045449 FB:FBF0141164 ISS P pannier CG3978|GATA|GATAA|GATAA|Pannier|Pannier|Pnr|dsGATAA|dsGATAA|pannier|1(3)8963|prr
154 FB FBgn0003118 pnt GO:0045449 FB:FBF0105495 ISS MGI:MGI:95455 P pointed 0123/09|0608/07|0998/12|3520|CG17077|Ets-2|D-ets-2|DMP|PNT|IA|E|E
155 FB FBgn0082831 pps GO:0045449 FB:FBF0105495 ISS protein_id:BAAL3438 P protein partner of snf CG6525|SNF5 Protein Partner|SPP|Spp|spp gene
156 FB FBgn0003145 prd GO:0045449 FB:FBF0105495 ISS MGI:MGI:97487 P paired CG6716|Paired|Prd|pr gene taxon:7227 20060803 FlyBase
```

Limpamos un poco la tabla para verla mejor

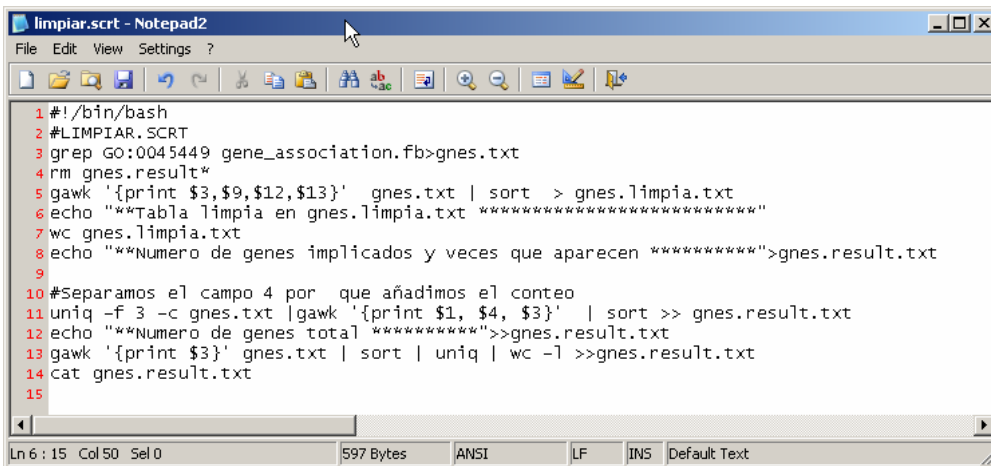


¡¡Ojo!! por que no todos son genes hay de todo (veasé la figura de arriba "protein"). Si nos fijamos en el formato de gene_association.gb, vemos que los genes están identificados con la palabra "gene" en la columna 12:

- 10) DB_Object_Name(|Name) (optional) - a name for the gene product in words, e.g. 'acid phosphatase'
- 11) DB_Object_Synonym(|Synonym) (optional) - see below
- 12) DB_Object_Type - type of object annotated, e.g. gene, protein, etc.
- 13) taxon(|taxon) - taxonomic identifier of species encoding gene product

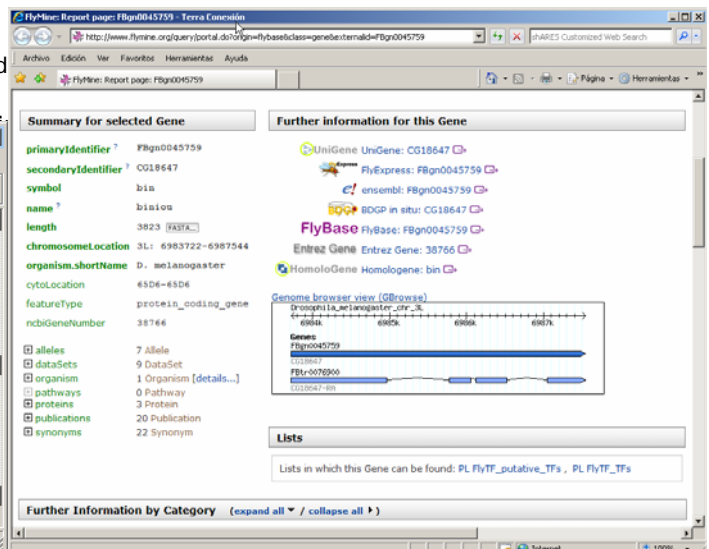
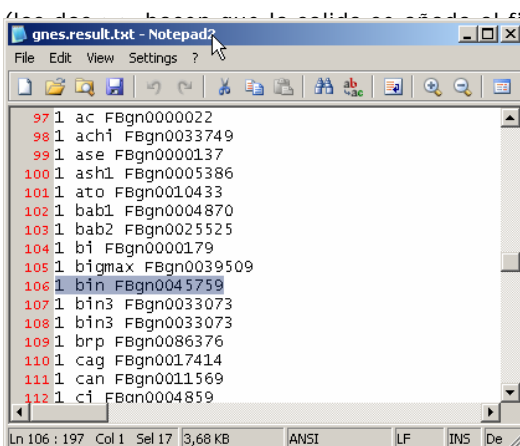
Procedemos a separar los "gene":

Preparo un script limpiar.scrpt . Al final el script queda



(gawk '{print \$i...}' imprime los campos seleccionados del fichero) (sort los ordena) (wc cuenta las líneas, palabras y caracteres)

(uniq -f 3 Hace que se eliminen los repetidos)

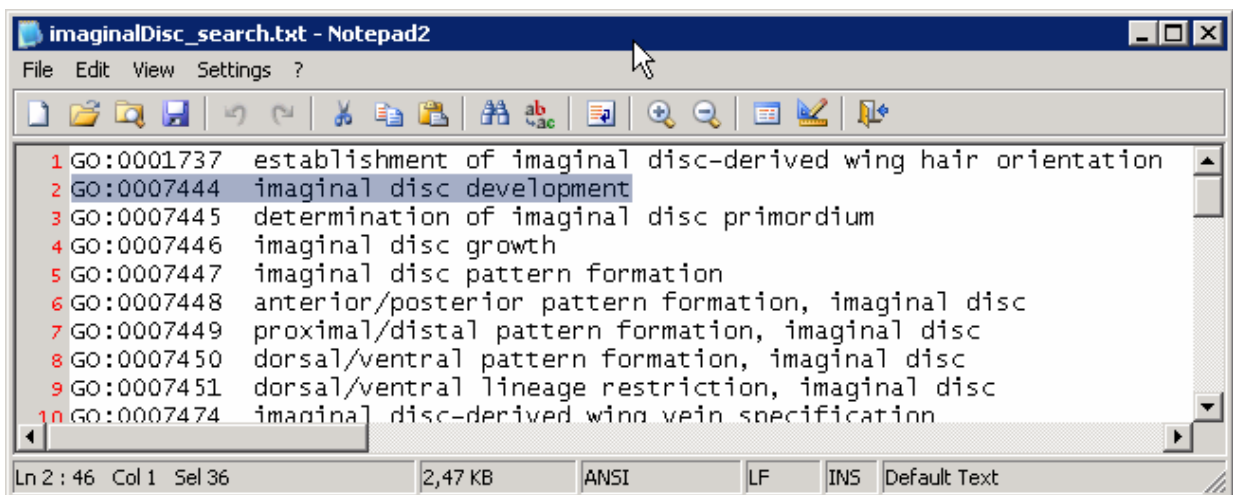


Genes asociados a los discos imaginales

Repetimos el proceso anterior pero con las palabras "imaginal discs"

```
tamarit@of_extrusion1 ~/go
$ grep imaginal gene_ontology.dat | grep disc > imaginalDisc_search.txt
tamarit@of_extrusion1 ~/go
```

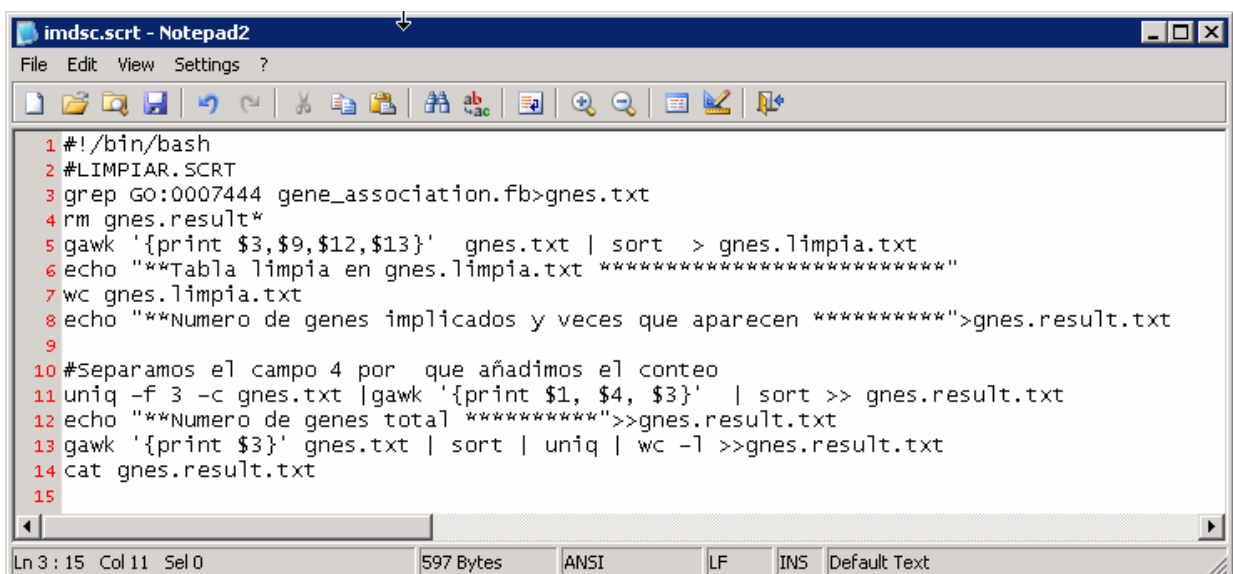
Visualizamos el resultado y comprobamos que existe una entrada igual a la que buscamos:



```
imaginalDisc_search.txt - Notepad2
File Edit View Settings ?
1 GO:0001737 establishment of imaginal disc-derived wing hair orientation
2 GO:0007444 imaginal disc development
3 GO:0007445 determination of imaginal disc primordium
4 GO:0007446 imaginal disc growth
5 GO:0007447 imaginal disc pattern formation
6 GO:0007448 anterior/posterior pattern formation, imaginal disc
7 GO:0007449 proximal/distal pattern formation, imaginal disc
8 GO:0007450 dorsal/ventral pattern formation, imaginal disc
9 GO:0007451 dorsal/ventral lineage restriction, imaginal disc
10 GO:0007474 imaginal disc-derived wing vein specification
Ln 2 : 46 Col 1 Sel 36 2,47 KB ANSI LF INS Default Text
```

La entrada que nos interesa lleva el Goid GO:0007444.

Retocamos el script del punto anterior. Lo renombramos a: imsrch.scr

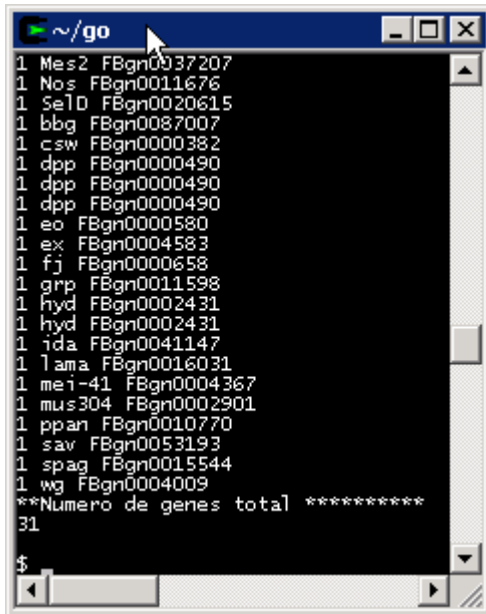


```
imdsch.scr - Notepad2
File Edit View Settings ?
1 #!/bin/bash
2 #LIMPIAR.SCRT
3 grep GO:0007444 gene_association.fb>gnes.txt
4 rm gnes.result*
5 gawk '{print $3,$9,$12,$13}' gnes.txt | sort > gnes.limpia.txt
6 echo "***Tabla limpia en gnes.limpia.txt *****"
7 wc gnes.limpia.txt
8 echo "***Numero de genes implicados y veces que aparecen *****">gnes.result.txt
9
10 #Separamos el campo 4 por que añadimos el conteo
11 uniq -f 3 -c gnes.txt |gawk '{print $1, $4, $3}' | sort >> gnes.result.txt
12 echo "***Numero de genes total *****">>gnes.result.txt
13 gawk '{print $3}' gnes.txt | sort | uniq | wc -l >>gnes.result.txt
14 cat gnes.result.txt
15
Ln 3 : 15 Col 11 Sel 0 597 Bytes ANSI LF INS Default Text
```

(Con el comando cat muestro el resultado contenido en el fichero gnes.result.txt)

(Cuando utilizo ">>" es para que no me genere un uno archivo, sino que ponga la salida al final)

El resultado:



```

~/go
1 Mes2 FBgn0037207
1 Nos FBgn0011676
1 Se1D FBgn0020615
1 bbg FBgn0087007
1 csw FBgn0000382
1 dpp FBgn0000490
1 dpp FBgn0000490
1 dpp FBgn0000490
1 eo FBgn0000580
1 ex FBgn0004583
1 fj FBgn0000658
1 grp FBgn0011598
1 hyd FBgn0002431
1 hyd FBgn0002431
1 ida FBgn0041147
1 lama FBgn0016031
1 mei-41 FBgn0004367
1 mus304 FBgn0002901
1 ppan FBgn0010770
1 sav FBgn0053193
1 spag FBgn0015544
1 wg FBgn0004009
**Numero de genes tota| *****
31
$

```

En total hay 31

<<<< el conteo de genes se hace con `gawk '{print $3}' gnes.txt | sort | uniq | wc -l` (en realizad el sort sobra, pero como es rápido tampoco me pare a bórralo. El script se podría optimizar para que fuera más rápido si lo quisiéramos volver a usar). La utilidad de hacerlo así es que no hay que estar tecleando series de comandos continuamente para hacer comprobaciones, lo podemos reutilizar, o incluso añadirle modificaciones más adelante.

Comprobaciones de los cálculos

Vamos a web de amigo y filtramos por imaginal disc development en drosophyla y que solo nos muestre los genes:

AmiGO: Term Association Details - Terra Conexión

http://amigo.geneontology.org/cgi-bin/amigo/term-assoc.cgi?gptype=genes&speciesdb=FB&taxid=7227&evcode=all&term_assoc=direct&t...

the Gene Ontology AmiGO

Search GO terms genes or proteins exact match Enviar consulta

imaginal disc development

Term associations Term information Term lineage External references

Gene Product Associations to imaginal disc development ; GO:0007444

gene association format RDF/XML

Current filters
Species: *Drosophila melanogaster*
Gene Product Type: gene
Data source: FlyBase

Filter associations displayed

Filter by Gene Product: Gene Product Type: All, complex, gene, protein
Data source: All, CGD, dicyBase, EcoCyc
Species: All, Anaplasma phagocy..., Arabidopsis thaliana, Bacillus anthraci...

Filter by Association: Evidence Code: All, IC, IDA, EXP

View associations: All, Direct associations
Remove all filters

imaginal disc development ; GO:0007444 [show def] [view in tree]

Symbol, full name	Information	Qualifier	Evidence	Reference	Assigned by
<input type="checkbox"/> 14-3-3epsilon	view associations gene from <i>Drosophila melanogaster</i>		TAS	FB:FBf0134532	FlyBase
<input type="checkbox"/> Awh	view associations gene from <i>Drosophila melanogaster</i>		IMP	FB:FBf0098764	UniProtKB

Guardamos el fichero y comprobamos:

```

1 FB->FBgn0020238>14-3-3epsilon->GO:0007444 -
2 FB->FBgn0013751>Awh->GO:0007444->FB:FBf00
3 FB->FBgn0023407>B4->GO:0007444->FB:FBf01
4 FB->FBgn0011766>E2f->GO:0007444->FB:FBf01
5 FB->FBgn0024371>E2f2->GO:0007444->FB:FBf
6 FB->FBgn0003731>Egfr->GO:0007444->FB:FBf
7 FB->FBgn0003731>Egfr->GO:0007444->FB:FBf
8 FB->FBgn0003731>Egfr->GO:0007444->FB:FBf
9 FB->FBgn0020416>Idgf1->GO:0007444->FB:FBf
10 FB->FBgn0020415>Idgf2->GO:0007444->FB:FBf
11 FB->FBgn0020414>Idgf3->GO:0007444->FB:FBf
12 FB->FBgn0026415>Idgf4->GO:0007444->FB:FBf
13 FB->FBgn0064237>Idgf5->GO:0007444->FB:FBf
14 FB->FBgn0086359>Invadolysin->GO:0007444->F
15 FB->FBgn0037207>Mes2->GO:0007444->FB:FBf
16 FB->FBgn0011676>Nos->GO:0007444->FB:FBf01
17 FB->FBgn0020615>Se1D->GO:0007444->FB:FBf
18 FB->FBgn0087007>bbg->GO:0007444->FB:FBf01
19 FB->FBgn0000382>csw->GO:0007444->FB:FBf00
20 FB->FBgn0000490>dpp->GO:0007444->FB:FBf01
21 FB->FBgn0000490>dpp->GO:0007444->FB:FBf01
22 FB->FBgn0000490>dpp->GO:0007444->FB:FBf01
23 FB->FBgn0000580>eo->GO:0007444->FB:FBf00
24 FB->FBgn0004583>ex->GO:0007444->FB:FBf01
25 FB->FBgn0000658>fj->GO:0007444->FB:FBf01
26 FB->FBgn0011598>grp->GO:0007444->FB:FBf01
27 FB->FBgn0002431>hyd->GO:0007444->FB:FBf00
28 FB->FBgn0002431>hyd->GO:0007444->FB:FBf01
29 FB->FBgn0041147>ida->GO:0007444->FB:FBf01
30 FB->FBgn0016031>lama->GO:0007444->FB:FBf
31 FB->FBgn0004367>mei-41->GO:0007444->FB:FB
32 FB->FBgn0002901>mus304->GO:0007444->FB:FB
33 FB->FBgn0010770>ppan->GO:0007444->FB:FBf
34 FB->FBgn0053193>sav->GO:0007444->FB:FBf01
35 FB->FBgn0015544>spag->GO:0007444->FB:FBf
36 FB->FBgn0004009>wg->GO:0007444->FB:FBf01
37

```

Obtenemos 36 entradas 5 mas. Pero en realidad si comprobamos el campo:

6) DB:Reference(|DB:Reference) - the reference associated with the GO annotation

Vemos que en realidad son 31 entradas. Están repetidas por que existen varias referencias (experimentaciones) asociadas con el mismo elemento y además con "evidencias diferentes".

En el caso de "Transcription Regulation", seguimos el mismo proceso, aquí la base nos devuelve 149 registros, menos de los que nos da a nosotros. Si cojemos uno de ellos:

FB FBgn0032248 CG5343 GO:0045449
FB:FBf0105495|EMBL:AY675080 ISS EMBL:AY675080.

Que si nos sale a nosotros pero no en el listado que nos hemos bajado del amiGO. Vamos a la pagina de amiGO y lo comprobamos. Nos llevamos la sorpresa , encontramos su entrada, pero no asociada a al GO:0045449, sino a otros más específicos.

Pero si buscamos específicamente su entrada: Si que lo encontramos y además relacionado a "regulation of transcription". Nos quedamos tranquilos de que lo que hemos hecho esta bien.

CG5343

CG5343
gene from *Drosophila melanogaster* (fruit fly)

Term associations **↓** Gene product information **→** Peptide Sequence **→** Sequence information **→**

Term Associations

gene association format RDF/XML

Current filters
Data source: FlyBase

Filter associations displayed

Filter Associations

Ontology	Evidence Code
All	All
biological process	IC
cellular component	IDA
molecular function	EXP

Perform an action with the selected terms...

	Accession, Term	Ontology	Qualifier	Evidence	Reference	Assigned by
<input type="checkbox"/>	GO:0048813 : dendrite morphogenesis 139 gene products view in tree	biological process		IMP	FB:FBrf0190556	FlyBase
<input type="checkbox"/>	GO:0007517 : muscle development 182 gene products view in tree	biological process		IMP	FB:FBrf0190556	FlyBase
<input type="checkbox"/>	GO:0048666 : neuron development 378 gene products view in tree	biological process		IMP	FB:FBrf0190556	FlyBase
<input type="checkbox"/>	GO:0045449 : regulation of transcription 531 gene products view in tree	biological process		ISS With EMBL:AY675080	FB:FBrf0105495	FlyBase
<input type="checkbox"/>	GO:0005634 : nucleus 1435 gene products view in tree	cellular component		ISS With EMBL:AY675080	FB:FBrf0105495	FlyBase
<input type="checkbox"/>	GO:0003700 : transcription factor activity 303 gene products view in tree	molecular function		ISS With EMBL:AY675080	FB:FBrf0105495	FlyBase

Perform an action with the selected terms...

[Back to top](#)

GO database release 2008-11-30
[Cite this data](#) • [Terms of use](#) • [GO helpdesk](#)
Copyright © 1999-2008 the Gene Ontology

Ejercicio 3 – Predicción computacional de genes

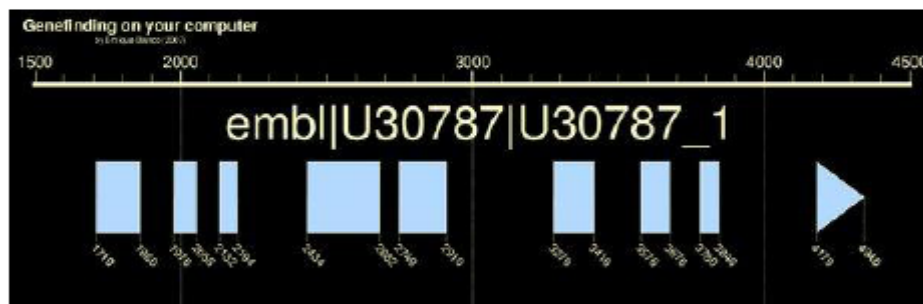
Ejercicio 3 – Predicción computacional de genes [30%]

(A) Un investigador nos comenta que la secuencia U30787.fasta (adjunta a este enunciado) contiene un gen humano. Debeis usar el programa **geneid** en vuestro terminal para averiguar cuál es la estructura éxonica más probable de ese gen según este programa.

(B) Ahora debeis representar gráficamente este gen usando el programa **gff2ps**, explicadme como obteneis una imagen como ésta y qué significan los colores asignados a los exones:



(C) Finalmente, antes de mostrarle nuestra predicción a nuestro colaborador, nos gustaría convertir esta imagen a un formato más atractivo. Cread un fichero de parámetros para **gff2ps** que reproduzca un formato como éste:



ATENCIÓN: Como Título de la imagen (en lugar de Genefinding on your computer) escribid vuestro nombre y el de la asignatura.

MATERIALES:

Recordad que las predicciones (el gen detectado por **geneid**) debe estar en formato GFF. Aquí teneis la descripción de éste, columna por columna:

http://genome.imim.es/software/geneid/docs/chapter4/formats_html/gff.html

Para llamar a GFF2PS con un fichero de parámetros "gff2ps.param", debeis hacer:
% gff2ps -C gff2ps.param predicciones.gff

Os adjunto al enunciado un fichero de parametros semivacio para que veais el formato. Asimismo teneis tambien el manual de usuario del programa GFF2PS (paginas 16-26 o apéndice B para las definiciones de campos).

Nota.- He usado el **geneid** sobre **cygwin**- Para hacerlo funcionar hay que compilarlo de una forma un poco especial ya que el **makefile** por defecto da errores de linkado. Adjunto como anexo las indicaciones para compilarlo (como **geneid.exe**) para **cygwin**.

a) Estructura exónica más probable.

```
tamarit@of_extrusion1 ~/geneid
```

```
$ ./geneid -P param/human3iso.param U30787.fa > U30787.out
```

```

emacs: U30787.out
File Edit View Cmds Tools Options Buffers
U30787.out
## date Fri Nov 21 16:49:43 2008
## source-version: geneid v 1.2 -- geneid@imim.es
# Sequence emb1|U30787|U30787 - Length = 4514 bps
# Optimal Gene Structure. 1 genes. Score = 16.20
# Gene 1 (Forward). 9 exons. 391 aa. Score = 16.20
Internal 1710 1860 -0.11 + 0 1 -0.36 5.38 4.68 0.00 AA 1: 51 emb1|U30787|U30787_1
Internal 1976 2055 0.24 + 2 0 4.17 4.71 -0.22 0.00 AA 51: 77 emb1|U30787|U30787_1
Internal 2132 2194 0.44 + 0 0 4.23 0.58 6.38 0.00 AA 78: 98 emb1|U30787|U30787_1
Internal 2434 2682 4.66 + 0 0 5.09 0.16 16.28 0.00 AA 99:181 emb1|U30787|U30787_1
Internal 2749 2910 3.19 + 0 0 4.65 5.34 5.50 0.00 AA 182:235 emb1|U30787|U30787_1
Internal 3279 3416 0.97 + 0 0 1.52 3.24 7.80 0.00 AA 236:281 emb1|U30787|U30787_1
Internal 3576 3676 3.23 + 0 2 2.70 3.95 10.61 0.00 AA 282:315 emb1|U30787|U30787_1
Internal 3780 3846 -0.96 + 1 0 2.83 1.86 3.06 0.00 AA 315:337 emb1|U30787|U30787_1
Terminal 4179 4340 4.55 + 0 0 2.65 0.00 19.89 0.00 AA 338:391 emb1|U30787|U30787_1

>emb1|U30787|U30787_1|geneid_v1.2_predicted_protein_1|391_AA
HTDTYYPHPLIARPGFPELKNMDFLRAAWGEETDYPVWCHRQAGRYLPEFRETRAADQ
FFSTCRSPEACCELTLQPLRRFLDAAIIFSDILVVPQALGHEVTHVPGKGPSFPEPLRE
EQDLERLRDPEVVASELGYVFAITLTRQLAGRVPLIGFAGAPVHWDRASTRGAGRSLW
KWTLMTYMVEGGSSMAQAKRWLQRPQASHQLRLITDALVPLVVGQVVAQAQALQLF
ESHAGHLGQLFNKFLPFIIRDVAKQVKARLEAGLAPVPHIFAKDGHFALEELAQAQY
EYVGLDWTVA PKKARECVGKVTTLQGNLDPICALYASEEEIGQLVKQMLDDFCPHRYIANL
GHGLYPMDDPEHVGAFVDAVHKHSRLLRQW*
ISOS--*-XEmacs: U30787.out (Fundamental)-----all-----
ESC -

```

La interpretación del resultado es la siguiente:

- Se predice 1 gen con 9 exones en dirección "forward" (5'→3'), con un score de 16.20 (que es la suma de los scores de de los 9 exones)
- Todos los exones son de tipo "internal" o internos

Se pueden encontrar cuatro tipos (los nuestros son internal):

- Iniciales: Delimitados por un codón inicial (ATG) y un donnor site.
 - Internos: Delimitados por un "acceptor site" y un "donnor".
 - Terminales: Delimitados por un donnor site y un codón STOP (TGA, TAG y TAA).
 - Singles: Delimitados por un codón inicial y un STOP.
- Las coordenadas internas de cada exón están en las columnas 2 y 3
 - La probabilidad (store) de cada exón se obtiene de la columna 4.
 - El + indica que todos van en dirección "forward"
 - El frame del exón esta el campo 6, el remanente "remaninder" en el 7,
 - Los campos siguientes 8,9,10,11 son scores variados (ver en el manual).
 - La localización de cada exón en la secuencia de aminoácidos de la predicción esta en los campos 12 y 13 (inicio y fin)
 - El ultimo campo es el identificador del gen.
 - Debajo tenemos la proteína predecida en formato fasta.

Con el parámetro -X se puede obtener una salida más completa.

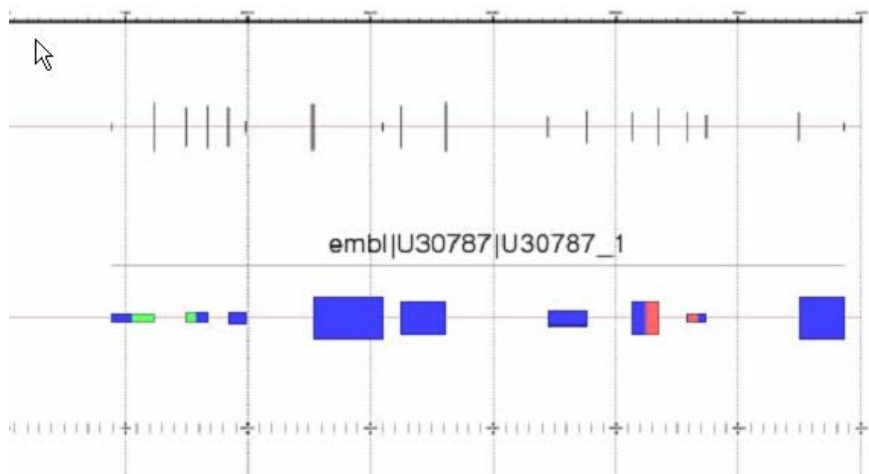
En el siguiente ejercicio comprobamos que podemos obtener la misma imagen modificando la salida con un fichero de parámetros modificando los valores por defecto de:

Por ejemplo

```
*::track_scale=2
*::range=0..20
Comprobado
```

De esta forma podemos usar la versión 0981.>>>

Imagen de la derecha obtenida con el fichero de parámetros modificado.



```
U30787.gff2psrc - Notepad2
File Edit View Settings ?
415 geneid_v1.2::track_scale=1
416 geneid_v1.2::source_line_color=verydarkred
417 geneid_v1.2::source_line=default
418 geneid_v1.2::source_style=default
419 geneid_v1.2::vert_align=default
420 geneid_v1.2::show_source_positions=false
421 #
422 *::track_scale=2
423 *::range=0..10
Ln 423 : 423 Col 15 Sel 0 18,84 KB ANSI LF INS Default Text
```

Significado de los colores

El significado de los colores por defecto esta especificado en el manual, e indica el frame del exon y del "remainder" (Exon a la izquierda – remainder a la derecha). Sirve para comprobar visualmente la consistencia de los frames de exones adyacentes. Dos exones son compatibles si los colores de sus caras enfrentadas (las cajas) son del mismo color (tiene el mismo frame).

Los colores que se definen por defecto son:

frame0_color=blue (Azul)

frame1_color=red (Rojo)

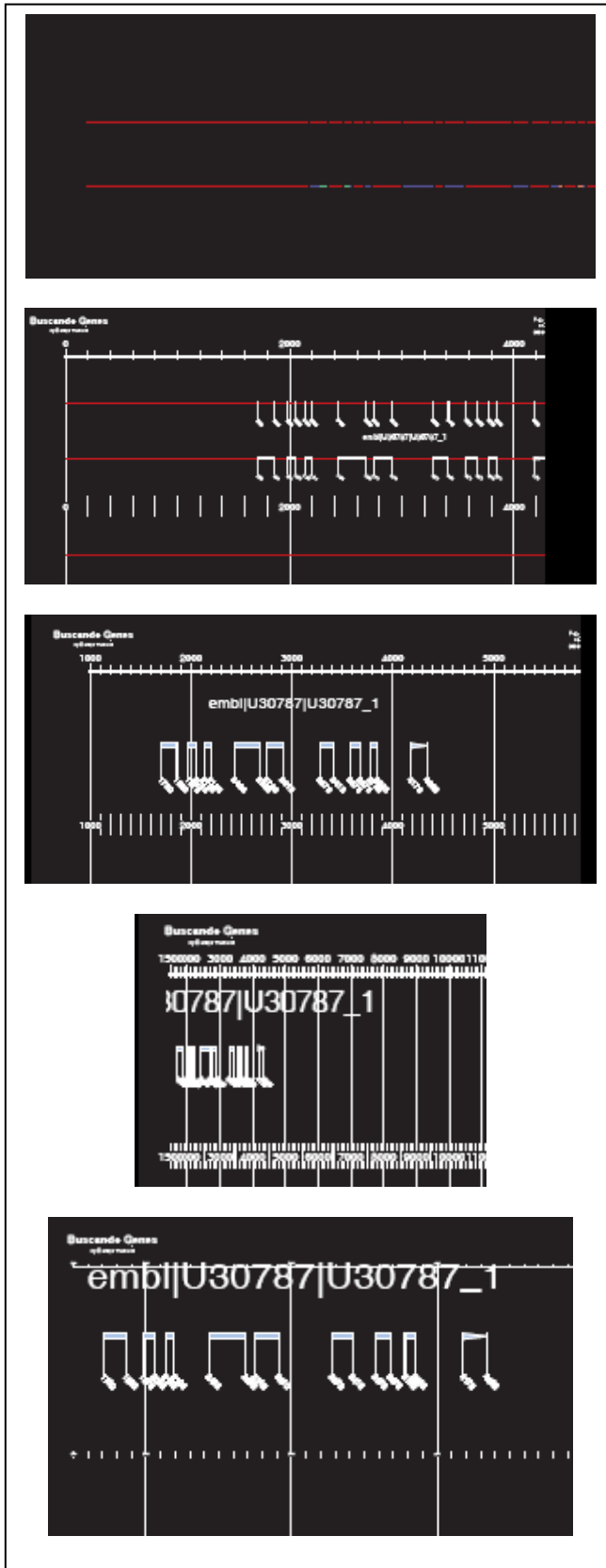
frame2_color=green (verde)

Para calcular el frame de los posibles exones a partir de las señales de splicing hay que tener en cuenta que los codones de los exones tienen que estar en la misma pauta que el codón stop que delimita el ORF. Entonces:

- Si el número de bases entre el inicio del ORF y el final del intrón es divisible entre 3 será frame 0.
- Si al dividir entre tres el residuo es dos: frame 1
- Si al dividir entre tres el residuo es uno: frame 2

El frame de los exones iniciales y de los single (un solo exón) siempre es 0.

c) Imagen especial



Para resolver este problema he comenzado desde el fichero de muestra entregado con la PEC. Además me he generado un .gff2ps (fichero de parámetros por defecto) con el comando `-d` para ver más o menos como hay que hacer el fichero de parámetros. Igualmente me he servido del manual para ir probando las distintas opciones

En particular me he fijado en:

- Como cambiar el título
- El fondo tiene que ser negro
- Tiene que salir la escala en blanco
- Una sola secuencia en la parte superior
- Que salga las marcas de posición de los nucleótidos iniciales y finales de cada exón.
- Ajustar el tamaño de las etiquetas

Dejo una secuencia de cómo me ha ido saliendo el dibujo a medida que cambiaba y añadía parámetros

El fichero de parámetros que al final he dado por bueno es el siguiente:

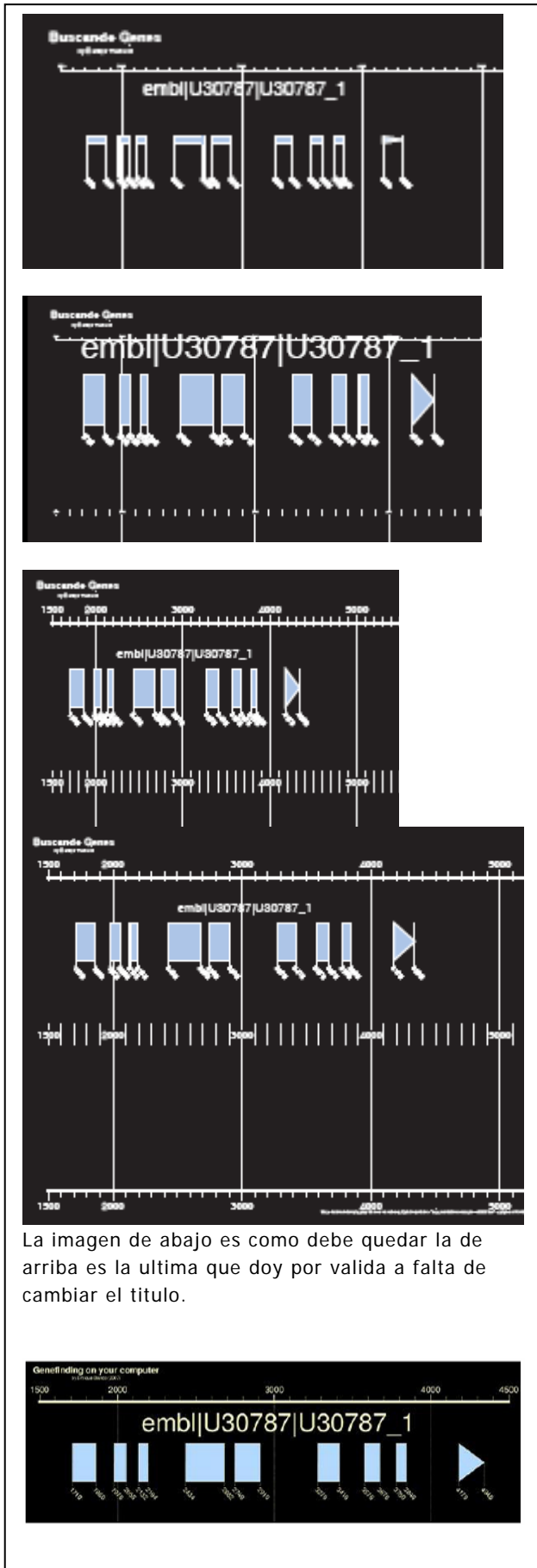
```
# L #
##pagina a4
##Para forzar a que entre todo en la
pagina
page_size=a4

## se me va hacia la derecha
##Retocamos el margen
margin_right= 2cm

##Fondo de color negro
background_color=black

##bajo fondo de líneas en blanco
##¡¡¡sino no se ven mas que puntitos
foreground_color=white

title= Ramón Tamarit - Fundamento de
informatica
subtitle=by Ramón Tamarit
##ajuste de escala
##probado con 2000 y no sale bien
major_tickmarks_nucleotides=1000
```



La imagen de abajo es como debe quedar la de arriba es la ultima que doy por valida a falta de cambiar el titulo.

```
##mostrar las posiciones de los exones
show_positions=on
```

```
##ponemos la escala
show_grid=on
show_outer_scale=on
```

```
## Pruebo con varios
#default_scale_width=1
#default_scale_width=2
##tamaño etiqueta escala
default_scale_width=0.8
```

```
##un solo bloque en la pagina
blocks_x_page=1
```

```
#default_track_width=0.5cm
```

```
##para acercar la secuencia a la izda
zoom=1500..5500
```

```
##ajuste de las etiquetas de escala
left_source_label_width=0.9cm
group_label_scale=1
position_label_scale=1
```

```
# G #
```

```
*::group_line=none
## todas las cajas del mismo color azulado
*::feature_color=verylightskyblue
```

```
# F #
```

```
##pone un triangulo en el exon terminal
terminal::shape=right_triangle
```

```
##los internos como caja
internal::shape=box
## que no se dividan
single::shape=single
```

```
##todos de color azul y que no muestre los traks
*::feature_color=verylightskyblue
*::fill_shape_mode=1_color
```

```
##Desctivamos los bloques reverse e
independiente
## para que no molesten
strand_show_reverse=off
strand_show_independent=off
```

```
# S #
```

```
*::left_label::++none++
```

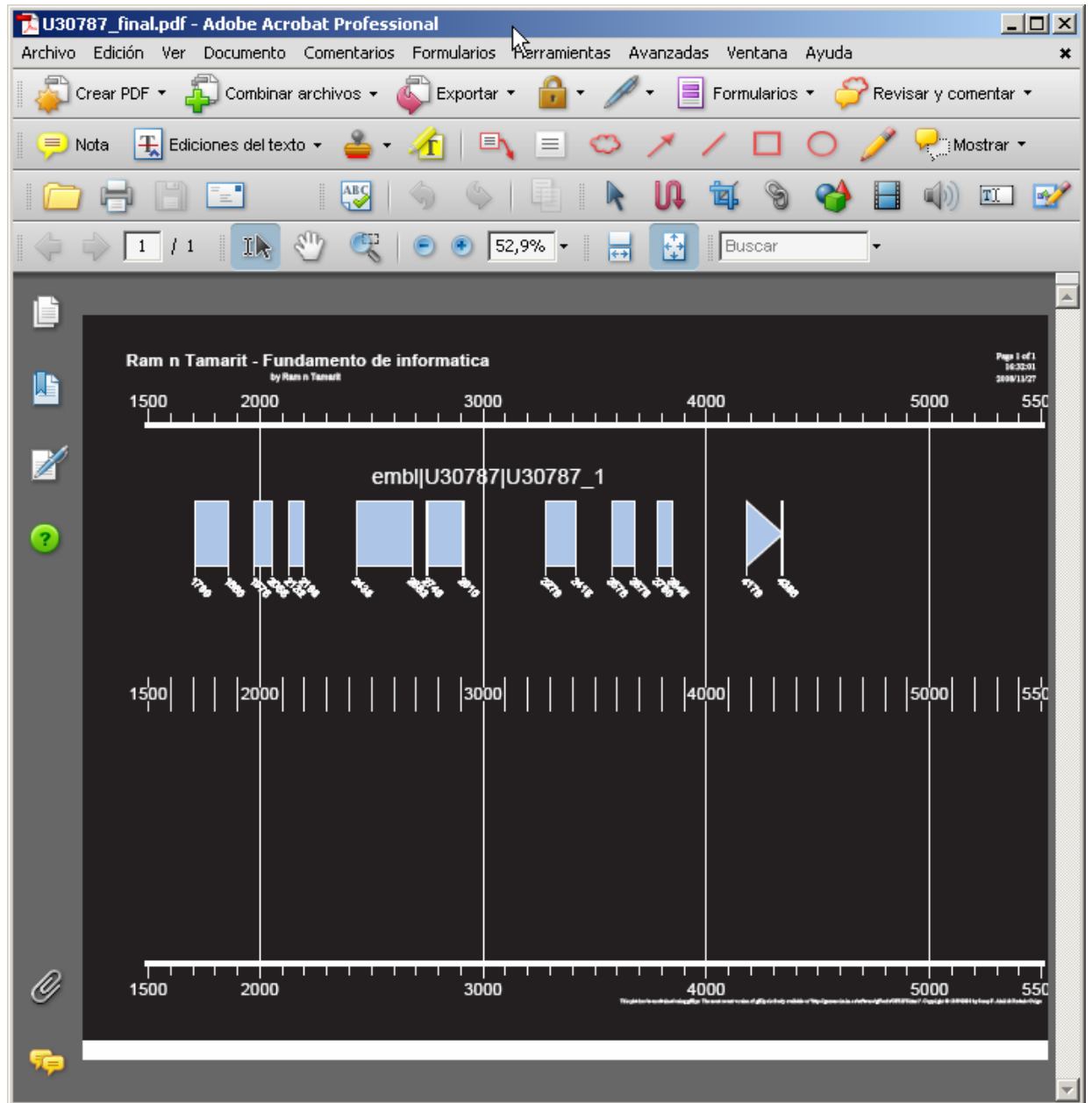
```
##hacemos desaparecer las lineas
*::source_line=none
*::unfold_ungrouped_line=off
```

```
*::track_spacing_scale=1.5
```

```
##alargar las cajas.. y evitar que salgan los scores
*::track_scale=1
```

```
*::range=none ##antes he probado con
## 2..2 pero no salen a mi gusto
*::source_label_scale=0.3
```

Después de muchos cambio y ajustes de tamaño (en total 37 pruebas) lo dejo como en la imagen siguiente.



Ejercicio 4 – Diseño de un pipeline automático de anotación.

Ejercicio 4 – Diseño de un pipeline automático de anotación [10%]

Como ejercicio conceptual que nos ayudará a asociar este tema de LINUX con el tema de PERL, imaginaos que en lugar de hacer la anotación de un gen (como en el apartado anterior), debéis realizar este proceso para 100 genes cuyas secuencias se encuentran en un directorio.

Debeis proponerme un script en PERL que automatice este proceso de forma que realice la predicción de los genes en esas secuencias con `geneid` y despues construya la representacion gráfica con `gff2ps`.

IMPORTANTE:

Este ejercicio numero 4 puede ser resuelto en dos modos según vuestros conocimientos prácticos adquiridos hasta ahora. Podeis animaros a implementar una versión funcional del script, o bien presentarme por escrito un esbozo de lo que creéis que debería hacer este programa (su flujo de datos). Os lo dejo a vuestra elección, el sistema de calificación en ambos casos será igual. En cualquier caso es importante que expliquéis que hace en cada instrucción este programa.

Adjunto el programa en perl. Creo que lo he comentado todo lo posible. Lo he probado y funciona.

```
#!/usr/bin/perl -w
#*****pipegen*****
use strict;

# Los argumentos se ponen en la línea de comandos
# Hay que poner :
# 1.- la ruta de los archivos fasta
# que queremos calcular,
# 2.- El archivo de parámetros de geneid
# 3.- El archivo de parámetros de gff2ps -opcional-
##por ejemplo
##pipegen ~/geneid ~/geneid/param/human3iso.param ~/geneid/.gff2ps
# Si no le llegan argumentos hay que salir del programa
# imprimiremos la variable $COMOUSAR que mostrará el un
# mensaje en pantalla
# $0 es una variable que contiene el nombre del programa

my($COMOUSAR) = "$0 - \n\t
*****\n\t
Para usar pipegen hay que pasar como argumentos \n\t
los ficheros a calcular \n\t
*****\n\t
Ejemplo:
~/geneid ~/geneid/param/human3iso.param ~/geneid/.gff2ps\n\t
*****\n\n";

my($LEYENDAPROCESO) = "
*****\n\t
PROCESANDO.....\n\n";

# @ARGV Matriz que contiene los argumentos de la linea de comandos
unless(@ARGV) {
    print $COMOUSAR;
    ## Si esta vacía salimos del programa
    exit;
}
```

```

#inicializamos las variables
my($RutaFicheros) = $ARGV[0];
my($RutaGeneidPar) = $ARGV[1];
my $Rutagff2psPar="";
print $RutaFicheros;
chdir $RutaFicheros #nos cambiamos al directorio de trabajo
or
die " Error - no encuentro $RutaFicheros: $!"; ##Si no podemos>> salimos

#sino damos un fichero de parametros ponemos la ruta vacia
unless(my($Rutagff2psPar) = $ARGV[2]) {
    #Ruta vacia
    $Rutagff2psPar="";
    exit;
}

## Si le hemos puesto la ruta del archivo de parámetros de gff2ps
# le añadimos -C
if ($Rutagff2psPar cmp "")
{
    #le añadimos el modificador
    # el . lo uso para unir cadenas
    $Rutagff2psPar = "-C " . $Rutagff2psPar;
    print "El fichero de parametros de gff2ps es : ", $Rutagff2psPar, "\n";
}

##obtenemos un listado de los ficheros con extensión fasta y los guardamos
# en la matriz @Ficheros. Esta es una de las formas de llamar a la línea
# de comandos del sistema
my @Ficheros = `ls -a *.fa`;

##Inicializamos las variables escalares para hacer los inputs
#nombre del fichero fasta para el calculo
my $FicheroFasta = "";

#nombre del fichero de salida
my $FicheroFastaOUT="";
my $FicheroGff2psOUT="";

#Contador de ficheros
my $numero = 0;

#Respuesta del geneid --- log
my $FicheroLog="";

##para cada fichero de la matriz ficheros
foreach (@Ficheros) {
    #recuperamos el nombre del fichero con la variables $_
    $FicheroFasta = $_ ;

    #Aumentamos en uno el contador
    $numero ++ ;

    #eliminamos el terminador de línea \n sino luego el input de
    #geneid tendrá problemas
    chomp($FicheroFasta );

    #Preparamos el nombre del fichero de salida de geneid y gff2ps
    $FicheroFastaOUT= $FicheroFasta . ".pipegen.gff";

    $FicheroGff2psOUT=$FicheroFasta . "pipegen.ps";
}

```

```

###nota####
## Podriamos haber hecho las llamadas en una función para mayor
# Claridad .....

#preparamos el input
my $Calculo= "./bin/geneid.exe -vGP " . $RutaGeneidPar . " "
                . $FicheroFasta . ">"
                . $FicheroFastaOUT;

print $LEYENDAPROCESO;
print "fichero fasta :", $numero ,"\t", $FicheroFasta,"\n";
print "fichero param :", $numero ,"\t", $RutaGeneidPar,"\n";
print "comando      :", $numero ,"\t", $Calculo,"\n";
print $Calculo, "\n";

#abrimos el geneid con las opciones que queremos y desviamos su
#salida a pantalla.
open(GENEID,"$Calculo|")|| die " Error... \n"; ## Si falla nos envía un error
while(<GENEID>) ##mientras tengamos salida
{
    #imprime la salida en pantalla
    #print "$_"; ##probando
    $FicheroLog = $FicheroLog . $_ ;
}
close(GENEID);

#abrimos el gff2ps con las opciones que queremos

## Aquí en vez de obtener un ps para cada fichero podriamos
## haber enviado todos los *.gff a gff2ps y haber obtenido un solo ps con cada
## resultado en un bloque. Para ello habria que sacar esta parte del bucle foreach, y
## después reconstruir la entrada de parámetros al gff2ps.

$Calculo= "./gff2ps " . $Rutagff2psPar . " "
                . $FicheroFastaOUT . ">"
                . $FicheroGff2psOUT;
open(GFF2PS,"$Calculo|") || die " Error... \n";
while(<GFF2PS>) ##mientras tengamos salida
{
    #imprime la salida en pantalla
    print "$_";
    $FicheroLog = $FicheroLog . $_ ;
}
close(GFF2PS);

print "*****" ;
}

##lo dejo preparado por si queremos evitar la salida
##por pantalla, guardarla en un fichero para verla después.
# en principio lo dejo comentado
#print $FicheroLog ;

exit;

#####

```

Compilar geneid para cygwin en windows

Descargar los programas necesarios.

El IDE de gcc de blodshead

<http://www.bloodshed.net/download.html>

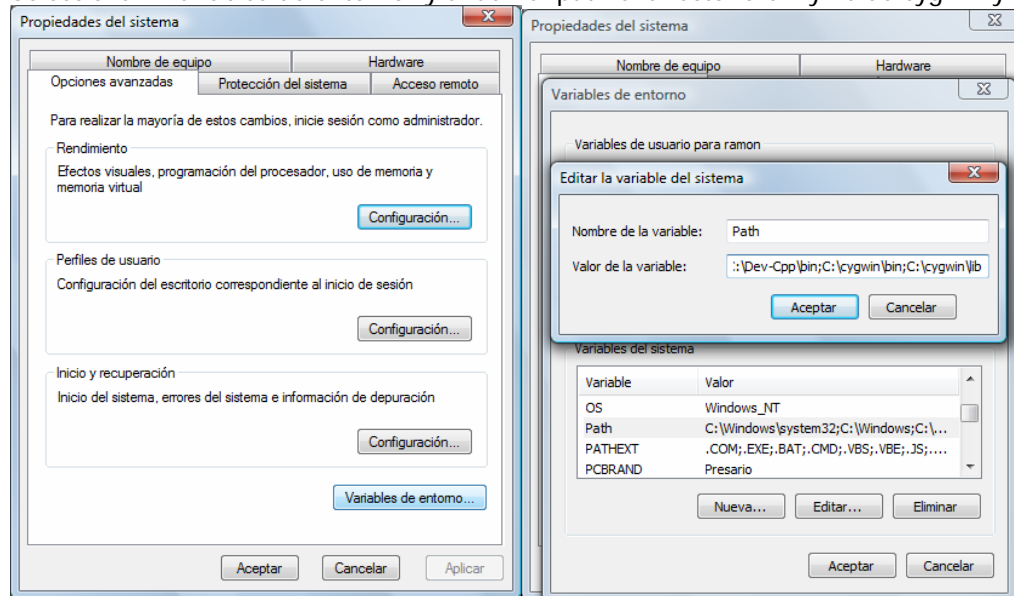
<http://sourceforge.net/projects/dev-cpp> yo he usado la versión 4.9.9.2

El cygwin. Asegurarse de descargar también los paquetes "devel" con el compilador.

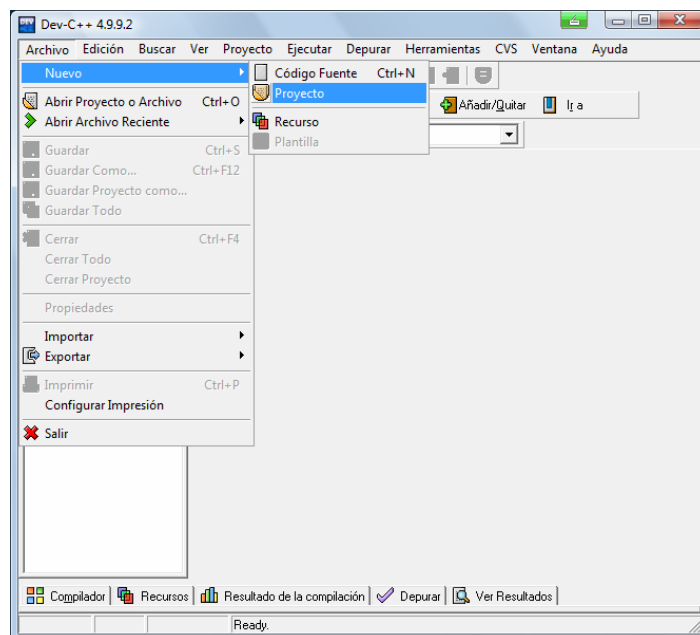
<http://www.cygwin.com/>

Configurar las rutas de acceso a las librería de sistema

Seleccionar "Variables de entorno" y añadir al path el directorio bin y lib de cygwin y el bin del ide.



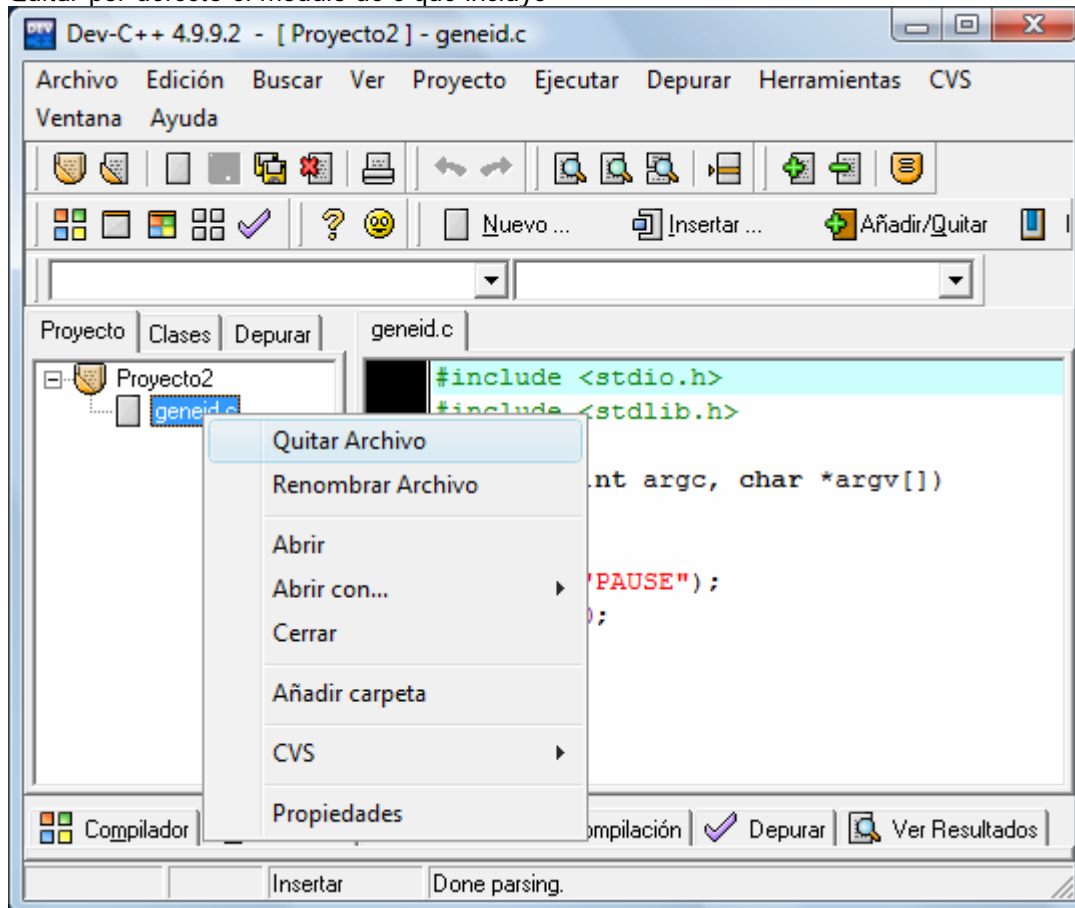
Crear un Nuevo proyecto con el Dev-C++



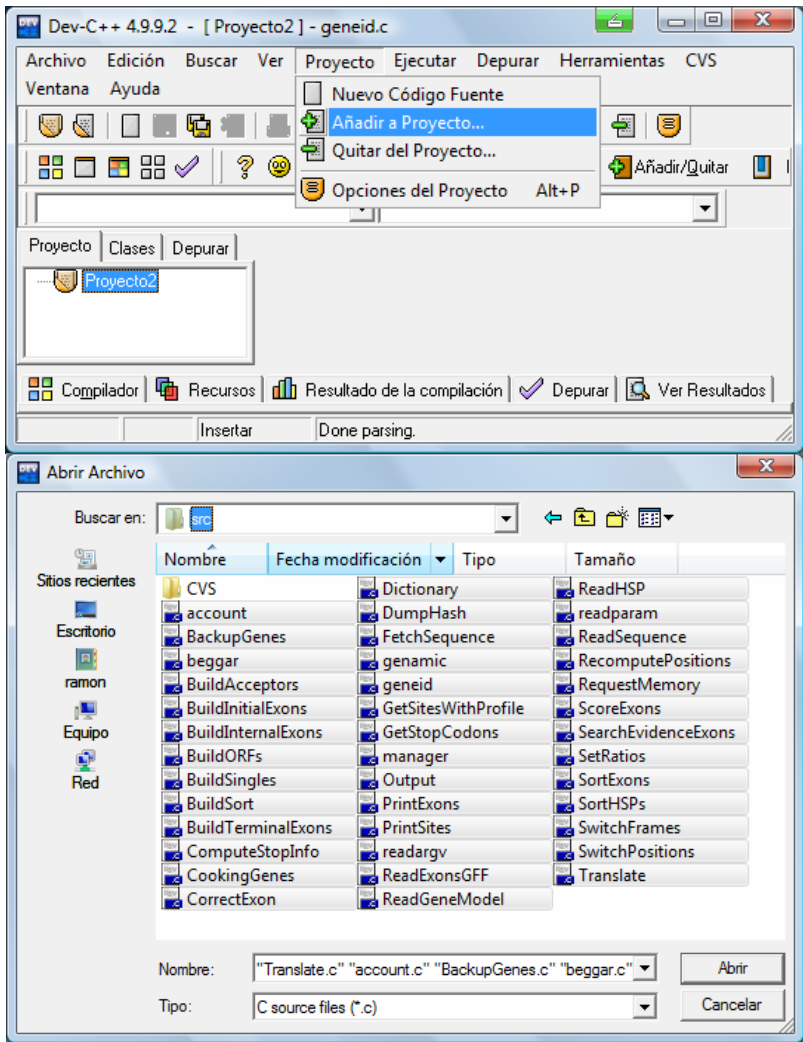
El proyecto será "Console Application" y en C.



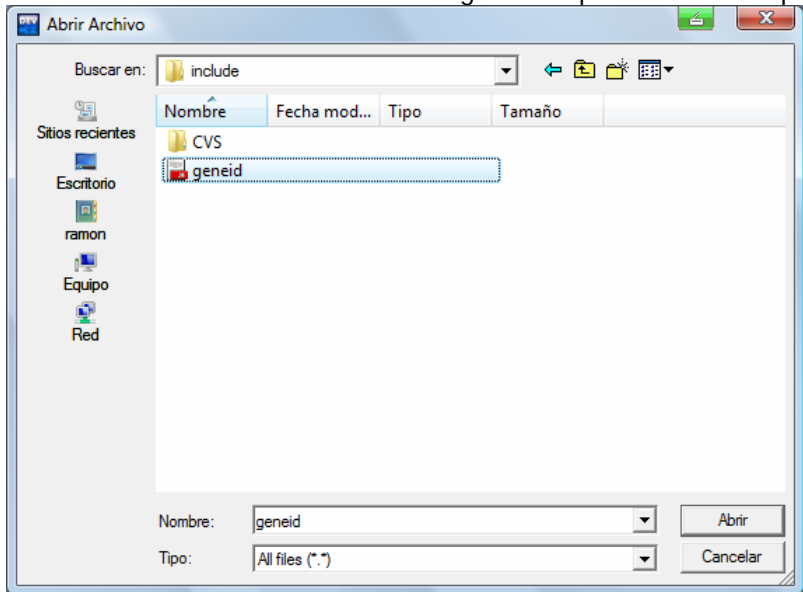
Quitar por defecto el modulo de c que incluye

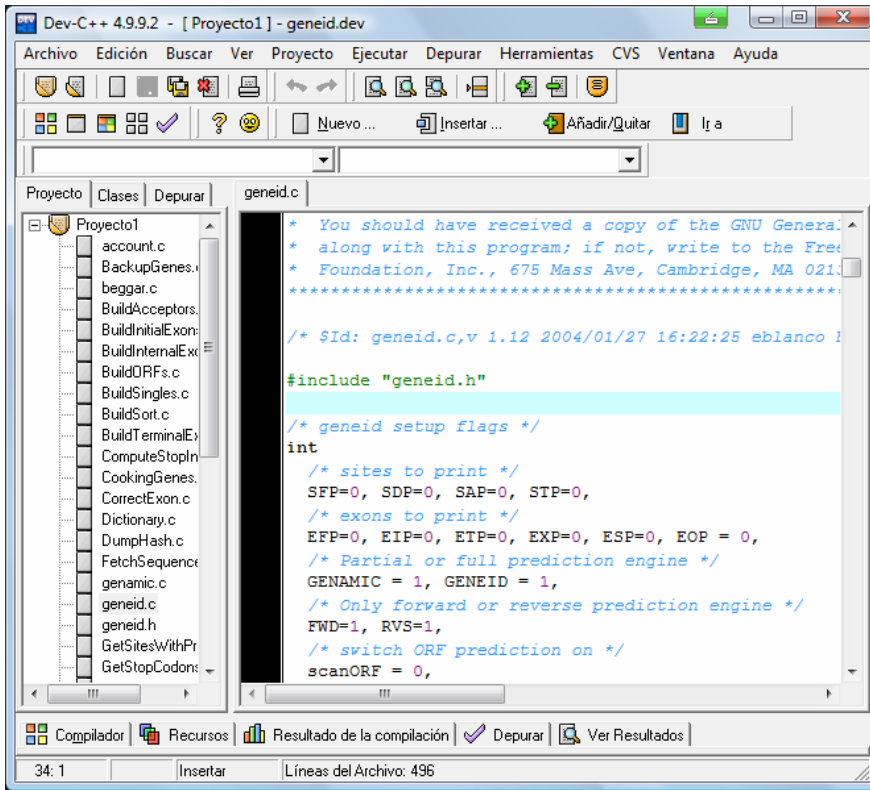


Añadir todos los archivos de geneid.



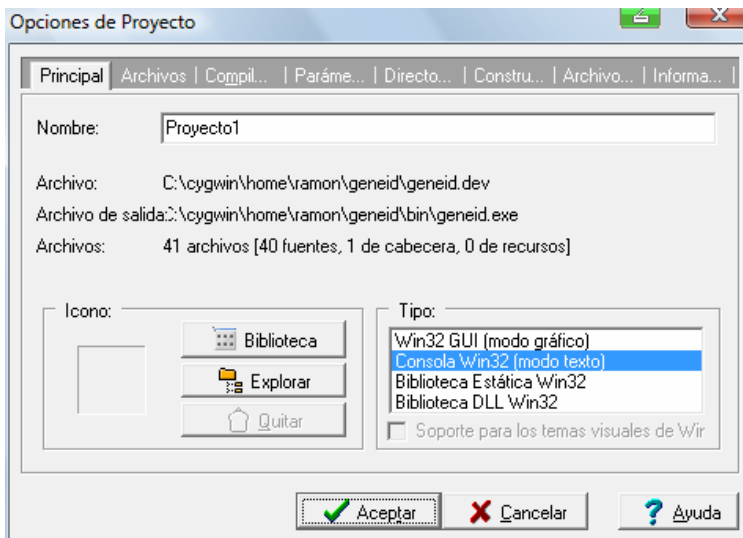
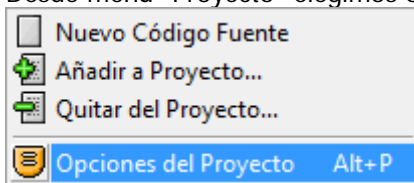
Añadir también el fichero de cabecera geneid.h que esta en la carpeta includes

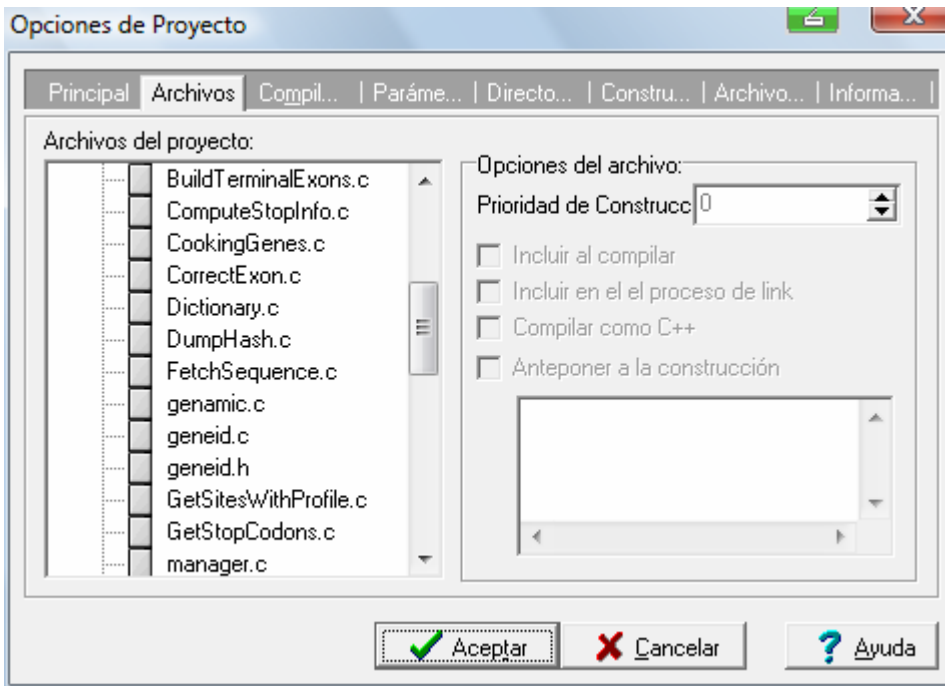




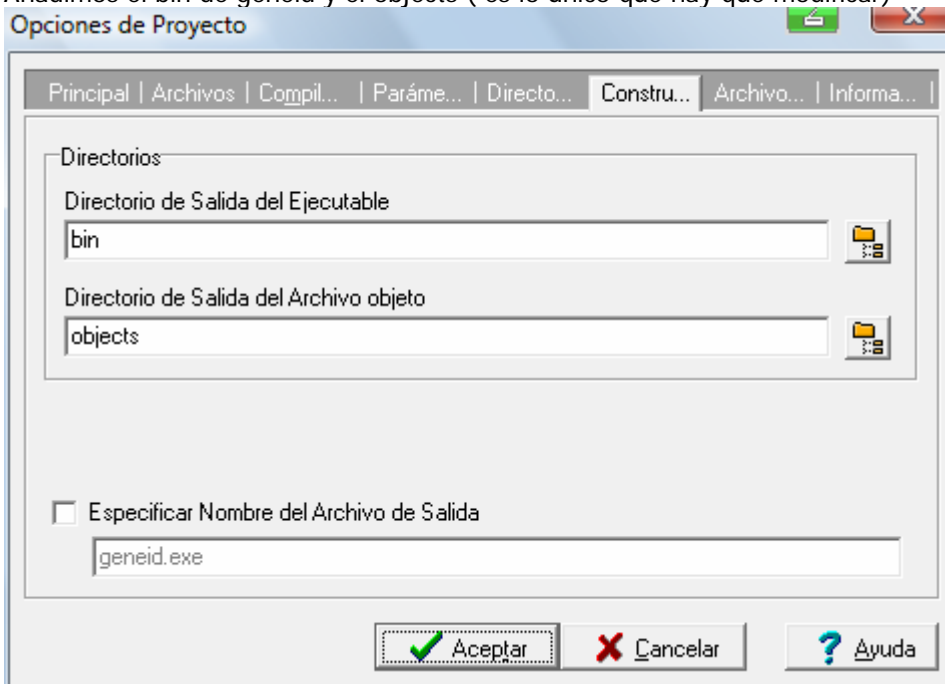
Configurar las opciones del proyecto

Desde menú "Proyecto" elegimos opciones del proyecto y modificamos lo que pone por defecto.



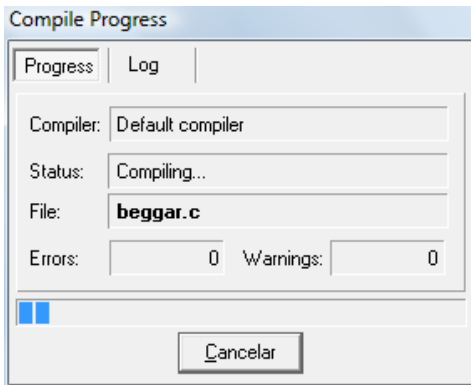
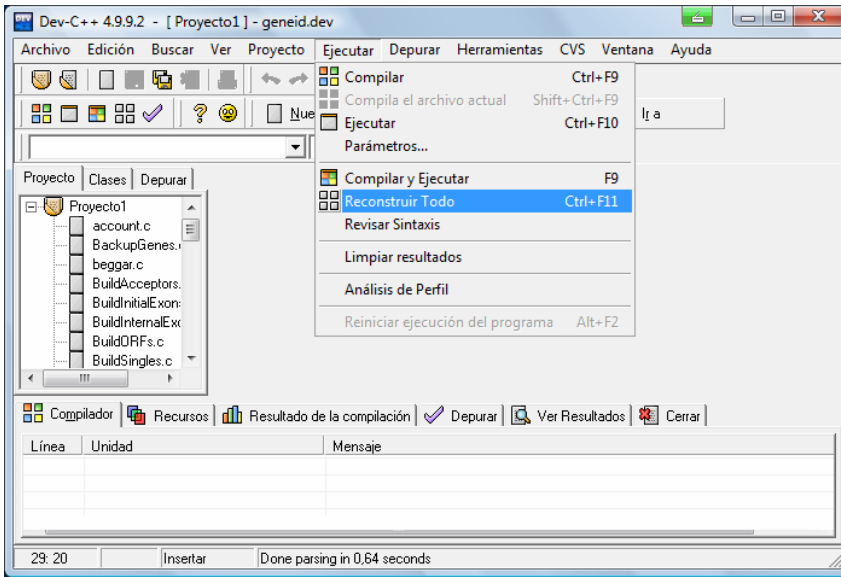


Añadimos el bin de geneid y el objects (es lo único que hay que modificar)

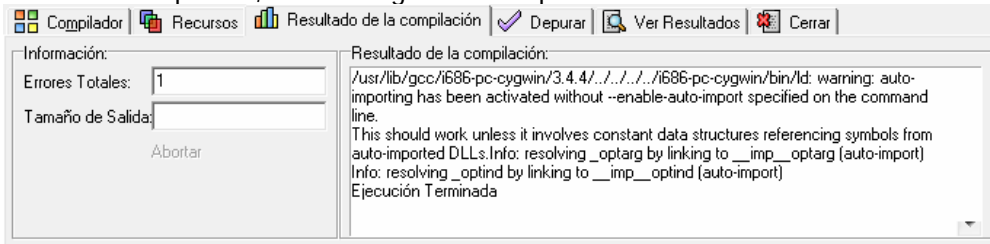


Lo demás lo dejamos igual. (principal, archivos, compilador, parámetros...

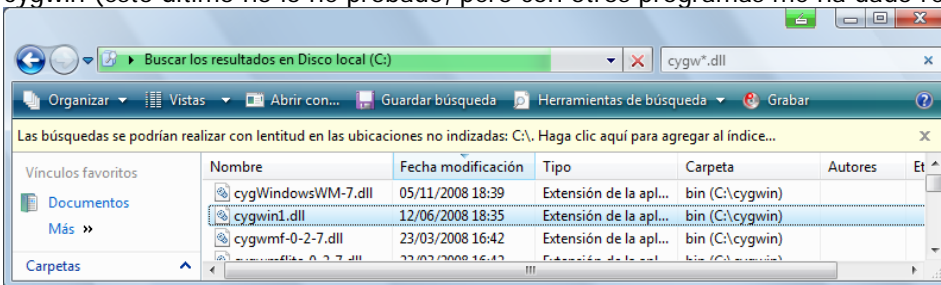
Reconstruir y compilar



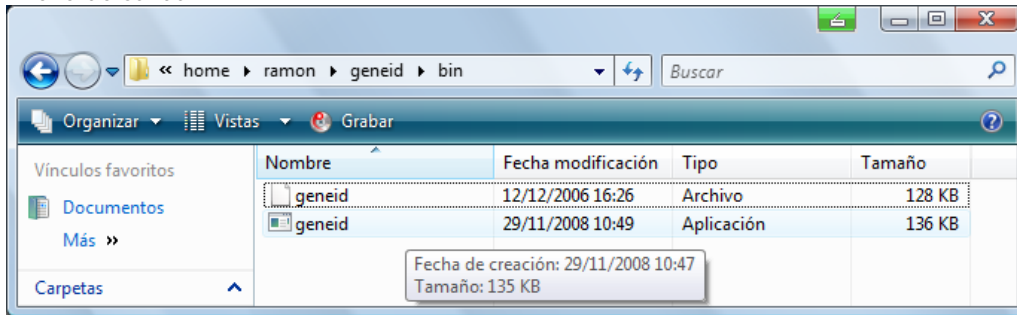
Salida del compilador, los warnings no son importantes.



Si queremos que funcione fuera de cygwin u en otro PC hay que distribuir la librería cygwin1.dll junto con geneid.exe. O bien usar algún programa tipo "alloy" para insertarla en el código ensamblado de cygwin (esto ultimo no lo he probado, pero con otros programas me ha dado resultado)



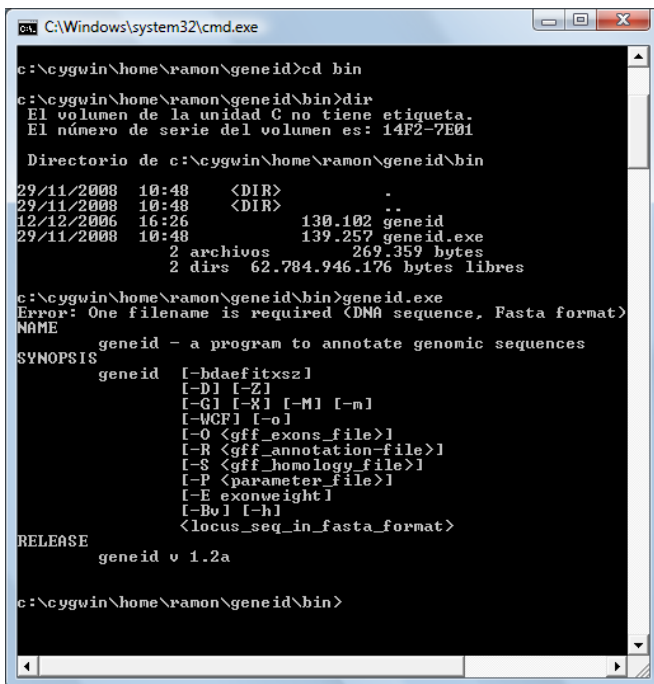
El exe de salida



(es un poco más grande porque incluye las cabeceras necesarias para rodar en windows)

Probando el geneid.exe

En el cmd de windows



En el cygwin como si fuera un programa unix (evidentemente, en un Linux no funcionará el exe)

