

Introducción a la bioinformática

Laura Gómez-Romero

Instituto Nacional de Medicina Genómica

lgomez@inmegen.gob.mx

November 8, 2021

Tabla de contenidos

- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell
- 4 Automatizando acciones repetitivas
- 5 Homología: Bidirectional Best Hit

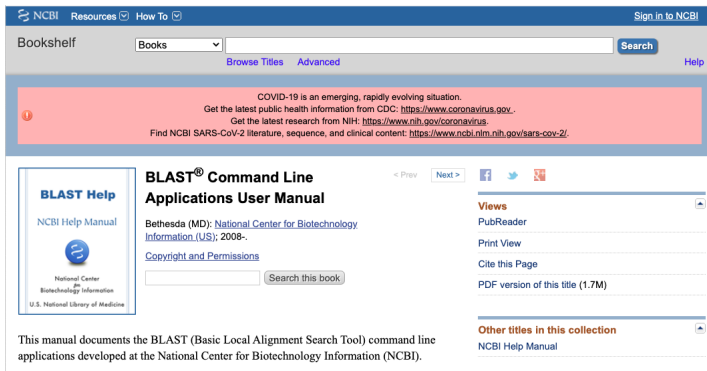
Tabla de contenidos

- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell
- 4 Automatizando acciones repetitivas
- 5 Homología: Bidirectional Best Hit

BLAST Command Line User Manual

BLAST puede ser ejecutado desde la línea de comandos. El manual de usuario se encuentra en:

<https://www.ncbi.nlm.nih.gov/books/NBK279690/>



The screenshot shows the NCBI Bookshelf interface. At the top, there's a navigation bar with 'NCBI', 'Resources', and 'How To' links, along with a 'Sign in to NCBI' button. Below this is a search bar with 'Books' selected in the dropdown and a 'Search' button. A red banner in the center contains COVID-19 related information and links to CDC, NIH, and NCBI SARS-CoV-2 resources. The main content area features the book cover for 'BLAST® Command Line Applications User Manual' on the left, which includes the NCBI logo and text: 'NCBI Help Manual', 'National Center for Biotechnology Information', and 'U.S. National Library of Medicine'. To the right of the cover, the title 'BLAST® Command Line Applications User Manual' is displayed, followed by the publisher information: 'Bethesda (MD): National Center for Biotechnology Information (US); 2008-'. Below this is a 'Copyright and Permissions' link and a search box with the text 'Search this book'. Further right, there are navigation links '< Prev' and 'Next >', social media icons for Facebook, Twitter, and YouTube, and a list of options: 'Views', 'PubReader', 'Print View', 'Cite this Page', and 'PDF version of this title (1.7M)'. At the bottom right, there's a section for 'Other titles in this collection' with a link to 'NCBI Help Manual'. A footer at the very bottom of the page contains navigation icons.

NCBI Resources How To Sign in to NCBI

Bookshelf Books Search Browse Titles Advanced Help

COVID-19 is an emerging, rapidly evolving situation.
Get the latest public health information from CDC: <https://www.coronavirus.gov>.
Get the latest research from NIH: <https://www.nih.gov/coronavirus>.
Find NCBI SARS-CoV-2 literature, sequence, and clinical content: <https://www.ncbi.nlm.nih.gov/sars-cov-2/>.

BLAST® Command Line Applications User Manual

Bethesda (MD): National Center for Biotechnology Information (US); 2008-.

[Copyright and Permissions](#)

Search this book

BLAST Help
NCBI Help Manual
National Center for Biotechnology Information
U.S. National Library of Medicine

This manual documents the BLAST (Basic Local Alignment Search Tool) command line applications developed at the National Center for Biotechnology Information (NCBI).

< Prev Next > Facebook Twitter YouTube

Views

PubReader

Print View

Cite this Page

PDF version of this title (1.7M)

Other titles in this collection

NCBI Help Manual

Pasos básicos de BLAST

- Instalar BLAST
- Descargar/construir una base de datos
- Correr BLAST

Instalar BLAST

En tepeu ya se encuentra instalado este comando

¿Cómo lo sabemos?

- Escribe la palabra blast en la terminal
- Presiona tabulador dos veces

```
[lgomez@tepeu ~]$ blast
blast2sam.pl      blastdbcheck      blastn_vdb        blastpgp
blastall          blastdbcmd        blastn_vdb.2      blastToPsl
blastclust        blast_formatter   blastn_vdb.2.2.30-2.4.5 blastx
blastdb_aliastool blastn            blastp            blastXmlToPsl
```

Tenemos varias versiones: blastn, blastp, blastx

Para ver las instrucciones de los desarrolladores:

<https://www.ncbi.nlm.nih.gov/books/NBK537770/>

¿Cuáles bases de datos necesitamos?

```
## Ayuda sobre el comando
```

```
update_blastdb.pl --help
```

```
## Para ver todas las bases de datos disponibles para descargar
```

```
update_blastdb.pl --showall [*]
```

¿Cuáles bases de datos necesitamos?

- ref_euk_rep_genomes
- refseq_rna
- refseq_protein

¿Cómo las descargamos directamente de NCBI?

```
## Reference RNA sequences
```

```
update_blastdb.pl --decompress refseq_rna
```

```
## RefSeq Genome Database
```

```
update_blastdb.pl --decompress ref_prok_rep_genomes
```

```
## Reference proteins
```

```
update_blastdb.pl --decompress refseq_protein
```

Así se ve el proceso

```
[lgomez@tepeu blast-databases]$ update_blastdb.pl --decompress refseq_rna
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
Connected to NCBI
Downloading refseq_rna (8 volumes) ...
Downloading refseq_rna.00.tar.gz...
[OK]
Downloading refseq_rna.01.tar.gz... [OK]
Downloading refseq_rna.02.tar.gz... [ ]
```

También se puede crear una base de datos personalizada a partir de un archivo multi-fasta

```
## Crear una base de datos a partir de un archivo multi-fasta  
makeblastdb -in mydb.fsa -dbtype nucl -parse_seqids  
  
## mydb.fsa - archivo multi.fasta  
## nucl - tipo de secuencias, opciones [nucl, prot]  
## -parse_seqids - para parsear los IDs de las secuencias  
## a partir del archivo FASTA
```

Base de datos de BALST en tepeu

En tepeu las bases de datos de BLAST las descarga el administrador y las encontramos en:

`/export/storage/data/blast/db/`

```
(base) -bash-4.4$ date
Mon Nov  8 19:05:30 CST 2021
(base) -bash-4.4$
(base) -bash-4.4$
(base) -bash-4.4$ du -h /export/storage/data/blast/db/
131G    /export/storage/data/blast/db/FASTA
770G    /export/storage/data/blast/db/
```

No queremos tener 700Gb de información en el \$HOME de cada usuario.

Correr BLAST

Sintaxis básica

```
blastn -db nt -query nt.fsa -out results.out -remote
```

Todas las opciones

```
blastn -help
```

- db - Nombre de la base de datos
- query - Secuencia problema
- out - Archivo de resultados
- remote - Para mandar a ejecutar la búsqueda en los servidores y bases de datos de NCBI

Configurando nuestro directorio de trabajo

Antes de empezar, asegurémonos de tener todo listo

```
## Crea el directorio practica 8 y posícionalate ahí  
mkdir practica9  
cd practica9
```

```
## Crea una liga simbolica al directorio con  
## las bases de datos de NCBI  
ln -s /export/storage/data/blast/db/ db
```

```
## Crea una liga simbolica a los datos de la practica8  
ln -s ../practica8/data/ .
```

Enviando un BLAST a NCBI

```
## Nuestro primer blast  
blastn -db refseq_rna \  
        -query data/TP53a.fasta \  
        -out TP53a_refseq_rna.blastn \  
        -remote
```

- db - Base de datos refseq_rna
- query - La secuencia TP53a.fasta
- out - Los resultados se escribirán en TP53a_refseq_rna.blastn
- remote - Lo mandaremos ejecutar en los servidores y bases de datos de NCBI

Corriendo un BLAST en nuestros servidores

```
## Enviando el mismo blast utilizando una base de datos local  
blastn -db db/refseq_rna \  
        -query data/TP53a.fasta \  
        -out TP53a_refseq_rna_local.blastn
```

- db - **Base de datos db/refseq_rna (LOCAL)**
- query - La secuencia TP53a.fasta
- out - Los resultados se escribirán en TP53a_refseq_rna_local.blastn

¡Esto es mucho más rápido!

Formatos de salida: varios

Todos los formatos de BLAST contienen casi la misma información. Sin embargo, las herramientas para parsearlos son diferentes.

```
*** Formatting options
-outfmt <String>
  alignment view options:
    0 = Pairwise,
    1 = Query-anchored showing identities,
    2 = Query-anchored no identities,
    3 = Flat query-anchored showing identities,
    4 = Flat query-anchored no identities,
    5 = BLAST XML,
    6 = Tabular,
    7 = Tabular with comment lines,
    8 = Seqalign (Text ASN.1),
    9 = Seqalign (Binary ASN.1),
    10 = Comma-separated values,
    11 = BLAST archive (ASN.1),
    12 = Seqalign (JSON),
    13 = Multiple-file BLAST JSON,
    14 = Multiple-file BLAST XML2,
    15 = Single-file BLAST JSON,
    16 = Single-file BLAST XML2,
    17 = Sequence Alignment/Map (SAM),
    18 = Organism Report
```

Formatos de salida: tabular y XML

Formato de salida XML

```
blastn -db db/refseq_rna \  
      -query data/TP53a.fasta -outfmt 5 \  
      -out TP53a_refseq_rna_local.xml
```

Formato de salida tabular

```
blastn -db db/refseq_rna \  
      -query data/TP53a.fasta -outfmt 6 \  
      -out TP53a_refseq_rna_local.txt
```

- db - Base de datos db/refseq_rna (LOCAL)
- query - La secuencia TP53a.fasta
- out - Archivo para escribir los resultados (atención en la extensión)
- outfmt - **5 para XML, 6 para tabular**

¿Y si tenemos mil archivos iniciales?

Escribamos los comandos que corrimos en un archivo de texto (sin extensión)

```
## Crear y abrir un archivo de texto
vi comandos

## blastn -db db/refseq_rna -query data/TP53a.fasta -outfmt 6 -out TP53a_refseq_rna_local_script.txt
## blastn -db db/refseq_rna -query data/TP53a.fasta -outfmt 5 -out TP53a_refseq_rna_local_script.xml
```

Convirtiendo un archivo de texto en un ejecutable

Cualquier archivo de texto puede convertirse en un archivo ejecutable y ejecutarse

```
## Cambiando los permisos (+ejecucion)
```

```
chmod +x comandos
```

```
## Ejecutando los comandos
```

```
./comandos
```

Hemos convertido el archivo de texto comandos en un script de shell.

Tabla de contenidos

- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell
- 4 Automatizando acciones repetitivas
- 5 Homología: Bidirectional Best Hit

Una variable en matemáticas

“A variable is a quantity that may change within the context of a mathematical problem or experiment. Typically, we use a single letter to represent a variable.

— Nykamp DQ, “Variable definition.” From Math Insight. <http://mathinsight.org/definition/variable>

Una variable en programación

“A variable is a symbolic name for (or reference to) information. The variable's name represents what information the variable contains. They are called variables because the represented information can change but the operations on the variable remain the same. In general, a program should be written with "Symbolic" notation, such that a statement is always true symbolically. For example if I want to know the average of two grades, We can write "average = (grade_1 + grade_2) / 2.0;" and the variable average will then contain the average grade regardless of the scores stored in the variables, grade_1 and grade_2.

— <https://www.cs.utah.edu/~germain/PPS/Topics/variables.html>

Nuestra primer variable en bash

Declarando y utilizando nuestra primer variable en bash

```
## Declarando una variable
```

```
DIR='prueba-dir'
```

```
## Viendo el contenido de nuestra variable
```

```
$DIR
```

```
## Utilizando nuestra variable
```

```
mkdir $DIR
```

La interpretación de una variable depende de su contexto

La variable \$DIR es una cadena de texto, la interpretación de esta cadena de texto dependerá del contexto de los comandos utilizados

```
## Declarando una variable  
DIR='prueba-dir-2'  
  
## Utilizando nuestra variable  
vi $DIR
```

Podemos utilizar variables dentro de scripts

```
## Nuestro script  
vi variables  
## VAR="hola mundo"  
## echo $VAR  
  
## Cambiando permisos de ejecución  
chmod +x variables  
  
## Ejecutandolo  
./variables
```

Tabla de contenidos

- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell**
- 4 Automatizando acciones repetitivas
- 5 Homología: Bidirectional Best Hit

Preparando el directorio de trabajo

```
## Crea una copia identica de la carpeta practica9  
cp -r practica9 practica9-bis  
  
## Verificando  
ls practica9  
ls practica9-bis
```

Tu primer script de shell

En la carpeta practica9, diseña un script de shell que ejecute las siguientes acciones (una acción por comando, un comando por línea):

- Crear la carpeta formatos
- Crear la carpeta fuentes
- Mover los archivos con la extensión blastn a la carpeta fuentes
- Mover el resto de los archivos a la carpeta formatos

Tu primer script de shell

En la carpeta practica9-bis, diseña un script de shell que ejecute las siguientes acciones (una acción por comando, un comando por línea)

IMPORTANTE: Define los nombres de las carpetas como variables al inicio del script.

- Crear la carpeta formatos
- Crear la carpeta fuentes
- Mover los archivos con la extensión blastn a la carpeta fuentes
- Mover el resto de los archivos a la carpeta formatos

Si no te funciona, crea otra carpeta y vuelve a intentarlo.

Tabla de contenidos

- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell
- 4 Automatizando acciones repetitivas**
- 5 Homología: Bidirectional Best Hit

Ejercicio original

Para cada una de las siguientes secuencias:

- TP53a.fasta
- TP53b.fasta
- TP53c.fasta

Realiza un BLAST contra las siguientes bases de datos:

- Reference RNA sequences
- RefSeq Genome Database
- Reference proteins

¿Qué puedes concluir de las distintas secuencias FASTA? ¿Cómo interpretas los valores de identidad, cobertura y e.valor para cada BLAST?

Vectores en bash

```
## Creemos el siguiente script (vectores)

#!/bin/bash

declare -a VECTOR=("10" "20" "30")

for NUMBER in ${VECTOR[@]}; do
    echo $NUMBER
done
```

Vectores en bash por partes

```
## Indica cual es el comando que ejecutara el script
#!/bin/bash

## Se declara un vector de cadenas
## Los elementos estan delimitados por comas, separados por espacios
## La variable se llama VECTOR
declare -a VECTOR=("10" "20" "30")

## Se accede a cada elemento en VECTOR
## Cada elemento se almacena en la variable NUMBER
## Se imprime la variable NUMBER
for NUMBER in ${VECTOR[@]}; do
    echo $NUMBER
done
```

Acciones dentro del ciclo

```
## Creemos el siguiente script (acciones)

#!/bin/bash

declare -a VECTOR=("10" "20" "30")

for NUMBER in ${VECTOR[@]}; do
    FILE=$NUMBER'.txt'
    touch $FILE
done
```

Acciones dentro del ciclo, por partes

```
#!/bin/bash

declare -a VECTOR=("10" "20" "30")

## Se accede a cada elemento en VECTOR
## Se crea una variable nueva formada por:
##      la variable NUMBER y la extension txt
## Se ejecuta el comando touch con la variable FILE como argumento
for NUMBER in ${VECTOR[@]}; do
    FILE=$NUMBER'.txt'
    touch $FILE
done
```

Acciones dentro del ciclo, por partes

```
#!/bin/bash

declare -a VECTOR=("10" "20" "30")

## Antes de ejecutar directamente los comandos finales
## es buena idea imprimir el comando a ejecutar para verificarlo
for NUMBER in ${VECTOR[@]}; do
    FILE=$NUMBER'.txt'
    echo "touch $FILE"
done
```

Ejercicio original: solución automatizada

Creen un bash script que ejecute todos los BLAST necesarios para resolver este ejercicio:
Para cada una de las siguientes secuencias:

- TP53a.fasta
- TP53b.fasta
- TP53c.fasta

Realiza un BLAST contra las siguientes bases de datos:

- Reference RNA sequences
- RefSeq Genome Database
- Reference proteins

Por ahora no ejecutes los comandos de blast solo imprime el texto que se ejecutaría

Tabla de contenidos

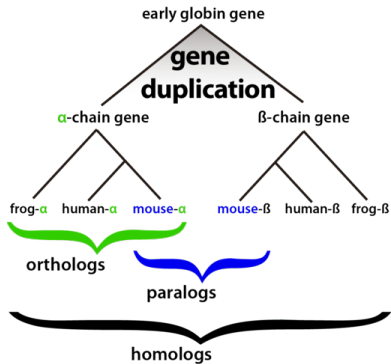
- 1 BLAST en línea de comandos
- 2 Variables
- 3 Práctica: Scripts de shell
- 4 Automatizando acciones repetitivas
- 5 Homología: Bidirectional Best Hit**

Cuando dos instancias biológicas provienen del mismo ancestro común [?].

Existen:

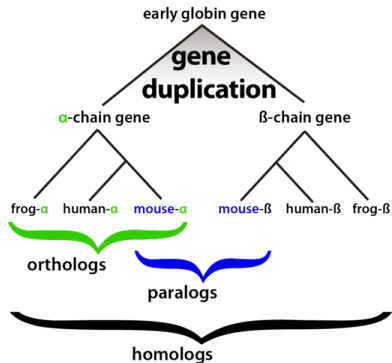
- Genes homólogos
- Nucleótidos homólogos
- Partes corporales

Homología



Todos los genes en esta imagen son homólogos

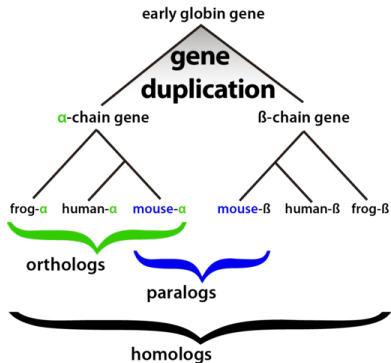
Imagen tomada de <https://bitesizebio.com/26762/homology-terminology-never-say-wrong-word/>



Los genes ortólogos:

- Proviene de un evento de especiación
- Generalmente mantienen la misma función

Imagen tomada de <https://bitesizebio.com/26762/homology-terminology-never-say-wrong-word/>



Los genes parálogos:

- Proviene de un evento de duplicación
- Generalmente sus funciones son diferentes pero relacionadas

Imagen tomada de <https://bitesizebio.com/26762/homology-terminology-never-say-wrong-word/>

Inferencia funcional basada en ortología

(a) ancestral duplication

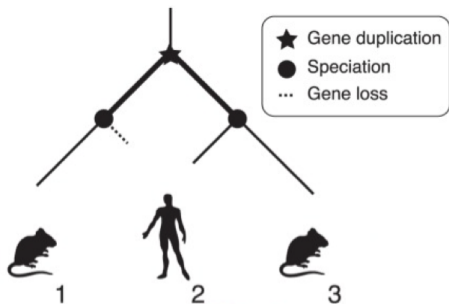


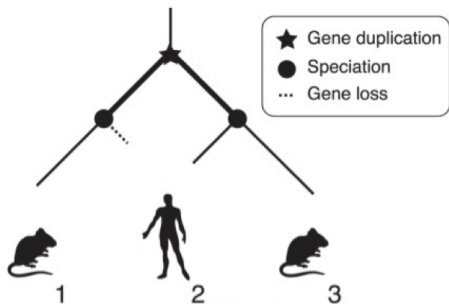
Imagen tomada de [?]

Imagina el siguiente escenario

- Sucedió un evento de duplicación y un evento de especiación en la historia evolutiva de un gene para los linajes de ratón y humano
- En la modernidad, sólo se conservan tres de las copias que algunas vez existieron de ese gene: 2 en ratón y 1 en humano
- Se sabe la función de ambas copias del gene de ratón y se quiere hacer una inferencia funcional de dicho gene en humano

Inferencia funcional basada en ortología

(a) ancestral duplication



- Se sabe la función de ambas copias del gene de ratón y se quiere hacer una inferencia funcional de dicho gene en humano

Recordando:

- Ortólogos: tienden a mantener la misma función
- Parálogos: tienden a tener funciones ligeramente diferentes pero relacionadas

Imagen tomada de [?]

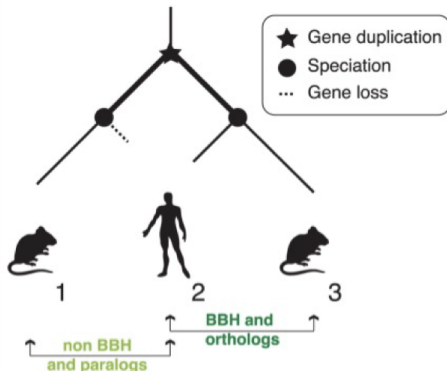
Bidirectional best hit

“One simple and widespread approach to identifying orthology is the bidirectional best hit (BBH) method (also known as reciprocal best hit or reciprocal Blast hit): call as orthologs all pairs of genes between two species that are more similar (i.e., with highest alignment score) to one another than to any other gene in the other species”

—[?]

Inferencia funcional basada en ortología

(a) ancestral duplication



Bidirectional best hit:

- 3 es el mejor alineamiento de 2 en ratón
- 2 es el mejor alineamiento de 3 en humano

Por lo tanto:

- Ortólogos: Gene 2 y 3 tendrán la misma función
- Parálogos: Gene 1 tendrá una función ligeramente diferente pero relacionada a la función de Gene 2 y 3

Imagen tomada de [?]

Práctica: obtener grupos de proteínas ortólogas

Preparando el directorio de trabajo:

- Crear el directorio practica9-blast
- Crear el directorio practica9-blast/fasta
- Copiar las secuencias problema de:
/home/lgoomez/WelcomeBioinfo/datos/practica9 a practica9-blast/fasta
- Entra al directorio practica9-blast

Práctica: obtener grupos de proteínas ortólogas

Procedimiento:

- Obtener los organismos con los que estamos trabajando
- Realizar un blast de cada proteína query contra la base de datos proteínas para cada organismo en nuestro conjunto de datos, seleccionando un formato de output que permita realizar el siguiente paso de manera ágil
- Analizar cuáles parejas de secuencias corresponden a **Bidirectional Best Hits**

Recuerda automatizar procesos. Haz uso de variables de bash y shell scripts.
Recuerda que correr un blast de manera local es mucho más rápido

BLAST contra un organismo específico

```
## Limitar la búsqueda de BLAST por taxid  
blastn -db $DB -query $SEQ -taxids $TAXID -outfmt $OUTFMT -out $OUT
```

Para obtener el taxid consulta la base de datos de Taxonomy de NCBI

El Tax ID de Homo sapiens es 9606.

Presentación de tabla de análisis

El análisis debe incluir una tabla como la siguiente:

| BEST HITS | Homo sapiens | Xenopus tropicalis | Bos taurus | Y EL RESTO DE LOS ORGANISMOS |
|-----------------------------|--------------|--------------------|------------|------------------------------|
| NP_000508 | NA | | | |
| NP_001005092 | | NA | | |
| NP_001070890 | | | NA | |
| Y EL RESTO DE LAS PROTEÍNAS | | | | |

NP_000508 es una proteína de humano por lo tanto se buscan ortólogos en el resto de los organismos, por eso se reporta como NA.

Ejemplo de un par de proteínas homólogas

Query: NP_00508.1 (human, 9606). Database: 9544

```
[lgomez@tepeu practica9-blast]$ head NP_000508.fasta_9544.txt
```

| | | | | | | | | | | | |
|-------------|----------------|--------|-----|----|---|----|-----|----|-----|----------|-----|
| NP_000508.1 | NP_001253705.1 | 96.479 | 142 | 5 | 0 | 1 | 142 | 1 | 142 | 3.82e-97 | 276 |
| NP_000508.1 | NP_001038189.1 | 96.479 | 142 | 5 | 0 | 1 | 142 | 1 | 142 | 7.46e-97 | 276 |
| NP_000508.1 | XP_014980671.2 | 95.775 | 142 | 6 | 0 | 1 | 142 | 36 | 177 | 3.91e-95 | 273 |
| NP_000508.1 | XP_002802384.1 | 63.380 | 142 | 52 | 0 | 1 | 142 | 1 | 142 | 1.38e-60 | 184 |
| NP_000508.1 | XP_014980670.2 | 58.451 | 142 | 59 | 0 | 1 | 142 | 1 | 142 | 6.09e-56 | 172 |
| NP_000508.1 | XP_028696705.1 | 57.746 | 142 | 60 | 0 | 1 | 142 | 1 | 142 | 2.47e-55 | 171 |
| NP_000508.1 | NP_001180633.2 | 45.390 | 141 | 77 | 0 | 2 | 142 | 1 | 141 | 9.76e-41 | 134 |
| NP_000508.1 | NP_001157900.1 | 43.448 | 145 | 74 | 3 | 3 | 141 | 4 | 146 | 1.05e-32 | 114 |
| NP_000508.1 | XP_028696189.1 | 50.450 | 111 | 55 | 0 | 32 | 142 | 46 | 156 | 3.39e-32 | 112 |
| NP_000508.1 | XP_028689611.1 | 40.690 | 145 | 78 | 3 | 3 | 141 | 4 | 146 | 5.63e-32 | 112 |

Query: NP_001253705.1 (macaca, 9544). Database: 9696

```
[lgomez@tepeu practica9-blast]$ head NP_001253705.fasta_9606.txt
```

| | | | | | | | | | | | |
|----------------|----------------|--------|-----|-----|---|----|-----|----|-----|----------|------|
| NP_001253705.1 | NP_000508.1 | 96.479 | 142 | 5 | 0 | 1 | 142 | 1 | 142 | 5.29e-97 | 276 |
| NP_001253705.1 | NP_005322.1 | 62.676 | 142 | 53 | 0 | 1 | 142 | 1 | 142 | 4.15e-60 | 183 |
| NP_001253705.1 | NP_005323.1 | 59.155 | 142 | 58 | 0 | 1 | 142 | 1 | 142 | 3.21e-55 | 171 |
| NP_001253705.1 | NP_001003938.1 | 46.099 | 141 | 76 | 0 | 2 | 142 | 1 | 141 | 5.12e-42 | 137 |
| NP_001253705.1 | NP_000509.1 | 43.448 | 145 | 74 | 3 | 3 | 141 | 4 | 146 | 1.27e-31 | 111 |
| NP_001253705.1 | NP_000175.1 | 41.379 | 145 | 77 | 3 | 3 | 141 | 4 | 146 | 4.64e-31 | 110 |
| NP_001253705.1 | NP_005321.1 | 39.007 | 141 | 78 | 3 | 7 | 141 | 8 | 146 | 6.13e-27 | 99.4 |
| NP_001253705.1 | XP_005257062.1 | 27.397 | 146 | 97 | 2 | 6 | 142 | 22 | 167 | 9.75e-14 | 66.6 |
| NP_001253705.1 | NP_001349775.1 | 25.676 | 148 | 104 | 1 | 1 | 142 | 1 | 148 | 5.96e-07 | 47.8 |
| NP_001253705.1 | XP_016879605.1 | 27.957 | 93 | 64 | 1 | 53 | 142 | 10 | 102 | 4.48e-05 | 42.4 |

Ejemplo de un par de proteínas homólogas

| BEST HITS | 9606 Homo | 8364 | 9544 Macaca | 9913 | 10090 | 9615 Canis | 7955 Danio | 9598 Pan |
|--------------|-------------|------|----------------|------|-------|------------|------------|----------|
| NP_000508 | NA | | NP_001253705.1 | | | | | |
| NP_001005092 | | NA | | | | | | |
| NP_001070890 | | | | NA | | | | |
| NP_001188320 | | | | | NA | | | |
| NP_001257812 | | | | | | NA | | |
| NP_005323 | NA | | | | | | | |
| NP_571332 | | | | | | | NA | |
| XP_508242 | | | | | | | | NA |
| NP_000549 | NA | | | | | | | |
| NP_001038189 | | | NA | | | | | |
| NP_001077424 | | | | | NA | | | |
| NP_001253705 | NP_000508.1 | | NA | | | | | |
| NP_001257814 | | | | | | NA | | |
| NP_032244 | | | | | NA | | | |
| NP_988860 | | NA | | | | | | |

NP_000508.1 y NP_001253705.1 son proteínas homólogas

References I