EMBOSS Toolbox

Introducción:

EMBOSS es una *suite* bioinformática con una multitud de herramientas elementales en biología molecular y genética. Creada y mantenida por EMBnet, EMBOSS es la clase de herramientas que siempre es mejor tener que no tener, a pesar de que todo lo que podemos hacer con ésta, también lo podemos hacer *manualmente* (esto es, en papel o con algún software específico). La conveniencia radica en que el software no solo maneja información biológica en varios formatos para realizar distintos tipos de tareas, sino que además lo hace muy rápidamente (lo cual significa que es computacionalmente escalable) y con esfuerzo mínimo, dado que la *suite* provee al usuario con una interfaz unificada para todas las aplicaciones. La lista de herramientas disponibles es ENORME:

Lista de utilidades en EMBOSS Suite

▶ Ver lista completa

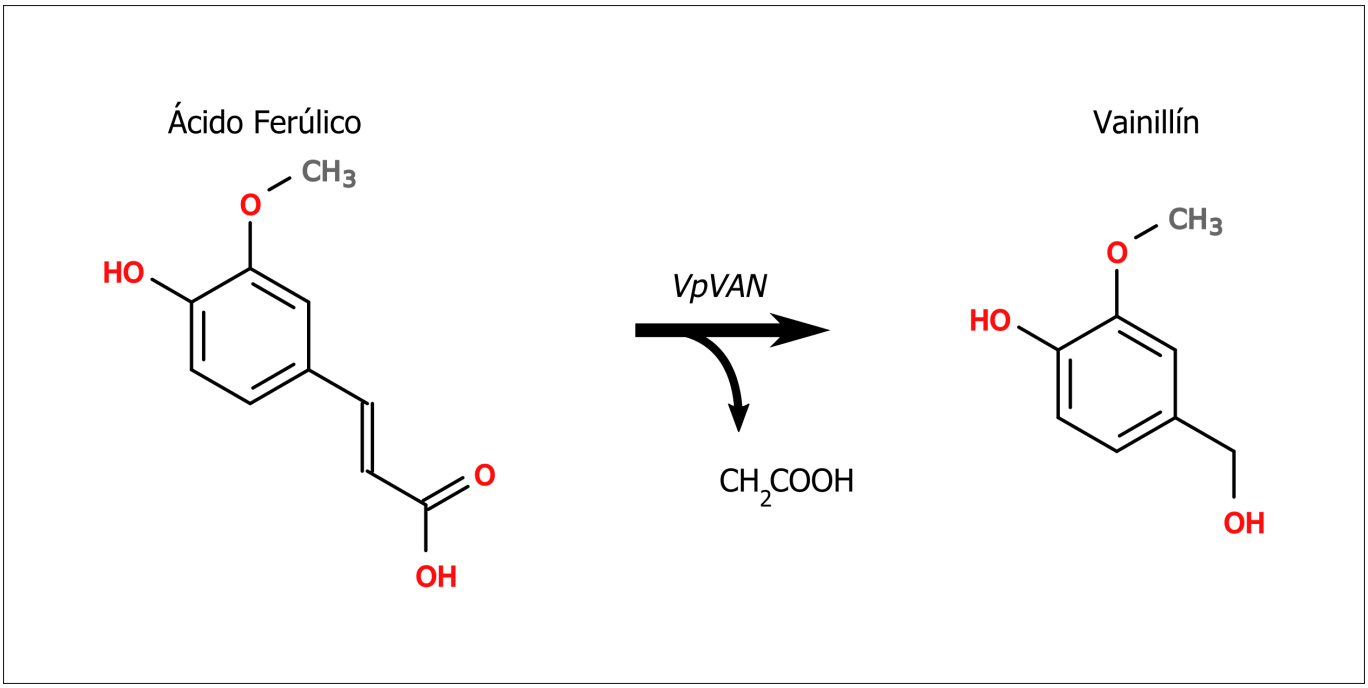
La verdad sea dicha, todos los biotecnológos hemos jugado (o jugamos) con secuencias de ácidos nucleicos o aminoácidos, y lo hemos hecho incluso sin saber que esta clase de herramientas existe: Así como podemos irnos de camping sin una de esas herramientas suizas multipropósito, también es cierto que podemos armar estrategias de clonado o hacer alineamientos múltiples sin EMBOSS.

En el TP de hoy vamos a familiarizarnos con EMBOSS y algunas herramientas del paquete, aplicándolas al diseño de una estrategia de clonado, puntualmente para diseñar/optimizar proteinas para expresion recombinante heteróloga.

En los últimos años, se ha simplificado cuantiosamente la ejecución de un proceso de clonado/expresión; por un lado gracias a la aparición de múltiples herramientas de Ing. Genética y por la posibilidad de sintetizar largas secuencias de ácidos nucleicos *in vitro*, lo que quita el peso de *levantar un gen* de interés o el riesgo de *meter errores* durante la PCR que ejecutamos para hacerlo. Para este TP, consideraremos que hace rato compramos groupón *90% off* en ADN sintético, que está por vencer y que, por ende,tenemos/podemos usar.

Como buenos biotecnólogxs (o *biotec-wannabes*), ya sabemos que una de las industrias biotecnológicas más antigua es la industria alimenticia. Centenas de microorganismos distintos y decenas de enzimas son utilizados en esta industria para distintos procesos. Algunos muy complejos, como la fermentación de un buen vino (y de uno malo también); y otros muy simples y puntuales, como la degradación de lactosa en productos lacteos para intolerantes a este azúcar. Lo procesos enzimáticos simples pueden resolverse *fácilmente* mediante la producción de la enzima de interés en forma heteróloga. Con el fin de dar rienda suelta a nuestro *científico entrepeneur* montaremos las bases de una empresa biotecnológica: vamos a producir enzimas.

La enzima que queremos producir es la *VpVan*, la enzima encargada de convertir el ácido ferúlico en **¡nada menos que vainillín!**



Esta enzima ha sido aislada (y secuenciada) de *Vanilla planifolia* . Encontrarán la secuencia correspondiente entre sus materiales de trabajo (`VpVAN.fasta`) y más información acerca de esta *million-dollar-idea* en este *paper*.

Vamos a clonar en forma direccionada, usando las enzimas

- BamHI
- HindIII

Intentaremos expresarla en las siguientes condiciones:

Sistema de expresión heterólogo:

- En *E. coli BL21*

Sistema de purificación:

- Con His-tag
- Con MBP-tag
- Con FLAG-tag

Por cuestiones de practicidad, todos los tags van a estar el C-terminal.

> **NOTA**: Estamos asumiendo que TODOS saben de qué estamos hablando con las condiciones expuestas. Si no es el caso, **pregunten**.

En lineas generales vamos a:

1. Generar secuencia de aminoácidos VpVAN-Tag para cada tag de interés.
2. Obtener las secuencias codificantes del organismo de interés y generar tabla de uso de codones para el organismo de interés.
3. Generar la secuencia nucleotídica VpVAN-tag con los codones optimizados para el organismo de interés.

4. Verificar que no hayan quedado sitios de restricción propios de la estrategia de clonado DENTRO de la secuecnia VpVAN-tag optimizada.

¡Manos a la obra!

1. Secuencias VpVan-Tag

Vamos a generar las secuencias quiméricas VpVan-Tag (donde Tag = His/MBP/FLAG). En su directorio de trabajo tienen los siguientes archivos

- VpVan.fasta
- His-tag.fasta
- MBP-tag.fasta
- FLAG-tag.fasta

El primer objetivo será generar un nuevo fasta por cada construcción. Pueden hacerlo por *copy-paste*. Nadie los va a juzgar. PEEEEEEERO, ya que estamos con la linea de comando, pueden aprovechar para practicar un poco de *scripting*.

```
# Acá va una propuesta. Pueden pensar en alguna alternativa, si quieren.
for TAG in `ls *-tag.fasta`; # Por cada Fasta de tag disponible...
do
   vpvanseq=`cat VpVAN.fasta | grep -v ">"`; # leer el archivo VpVAN y quitarle el
header (>), guardarlo en una variable
   tagseq=`cat $TAG | grep -v ">"`; # leer un archivo tag y quitarle el header
(>), guardarlo en otra variable
   printf ">VpVAN-$TAG\n$vpvanseq$tagseq" > VpVAN-$TAG; # imprimir y guardar en un
archivo separado para cada tag
done;
```

¡Ya tenemos nuestras secuencias quiméricas!

Comencemos por instalar EMBOSS en nuestro sistema. Si no tuviéramos la máquina virtual (que ya tiene todo instlado), podríamos instalar EMBOSS como sigue:

En una linea de comando, ingresaríamos:

```
sudo apt-get install emboss emboss-data emboss-doc
```

> ¿Se acuerdan qué son estos comandos? Si no se acuerdan, siempre pueden acudir al manual:

```
man sudo
man apt-get
```

2. Construir tabla de frecuencias de uso de codones

Para poder calcular la tabla de uso de codones, podemos usar `cusp`, de EMBOSS. Por si no recurdan, del manual de cusp (`tfm cusp` en la linea de comando), "*cusp calculates a codon usage table for one or more nucleotide coding sequences and writes the table to file*".

Esto significa que necesitaremos una lista de secuencias codificantes.

El formato más común que usamos los bioinformáticos para guardar (y usar) listas de secuencias es el formato FASTA. Vamos a construir un archivo tipo fasta con nuestras secuencias codificantes.

2.1 Obtener secuencias codificantes

Para bacterias descargar secuencias codificantes en bacterias, debemos seguir las instrucciones descritas en https://www.ncbi.nlm.nih.gov/genome/doc/ftpfaq/ *How can I download RefSeq data for all complete bacterial genomes?*, que básicamente se resumen en los siguientes pasos:

1. Bajar un resumen de todos los proyectos genoma disponibles para descargar en la base de datos RefSeq:

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

2. Una vez descargado el resumen, podemos buscar entre todas las secuencias cuáles podrían interesarnos. Prueben usando distintos comandos de unix, como `grep`, `awk` y `cat` para obtener el link ftp de todas las secuencias de *E. coli* de la cepa BL21 (comunmente utilizada para expresión heteróloga).

▶ Ver links a los genomas

---

3. Una vez identificada la manera de filtrar el `summary` hasta obtener el/los genomas de interés, bajar el/los archivos `<genoma>_cds_from_genomic.fna.gz`, donde `<genoma>` sería el link al genoma de interés.

Podemos agregar manualmente (buh! 💩) o podemos hacer todo en un solo paso (eeh! 🎉):

Una forma simple pero efectiva:

```
# Obtengo los links
grep "BL21" assembly_summary.txt | grep "coli" | awk -F'\t' '{print $8,$20}'
# y los pego en el navegador para descargarlos manualmente
```

Otra forma más prolija y programática

```
# Obtengo los links <- Esto puede ser muy distinto a lo que hicieron ustedes para
obtener sus links, revisen las diferencias :)
cat assembly_summary.txt | awk -F "\t" '{ if ($12 == "Complete Genome" && $11 ==
"latest" && $8 ~ "BL21") {print $20}}'  > ftpdirpaths
# Agrego el sufijo _cds_from_genomic_fna.gz
```

```
awk 'BEGIN{FS=OFS="/";filesuffix="cds_from_genomic.fna.gz"}
{ftpdir=$0;asm=$10;file=asm"_"filesuffix;print ftpdir,file}' ftpdirpaths >
ftpfilepaths
# Descargo de todas las URLs adentro de ftpfilepaths
wget -i ftpfilepaths
```

▶ Ver links a los genomas

2.2 Obtener la frecuencia de uso de codones

Veamos cómo puede ayudarnos EMBOSS a hacer esto.

Entrada en calor con EMBOSS

Para familiarizarnos con EMBOSS, comencemos por buscar qué herramientas vamos a usar durante el TP. Si emboss-doc está instalado, se puede ver la documentación de los paquetes en la linea de comando. Ya sabemos que vamos a estar optimizando codones para algún organismo así que arranquemos por ahí: Para buscar comandos que hacen cosas, usar `wossname` con palabras clave (en inglés, ej: 'codon'). El comando `wossname` nos da una lista de comandos asociados con esas palabras clave y lo que hace cada programa.

> Esto solo funcionará si instalamos la documentación de EBMOS.

```
    sudo apt-get install emboss-doc
```

Luego, podremos usar wossname para buscar comandos que trabajen con codones.

```
wossname codon

    Find programs by keywords in their short description
    SEARCH FOR 'CODON'
    cai         Calculate codon adaptation index
    checktrans  Report STOP codons and ORF statistics of a protein
    chips       Calculate Nc codon usage statistic
    codcmp      Codon usage table comparison
    codcopy     Copy and reformat a codon usage table
    cusp        Create a codon usage table from nucleotide sequence(s) # <----
TABLA DE USO DE CODONES ALERT!
    cutgextract Extract codon usage tables from CUTG database
    syco        Draw synonymous codon usage statistic plot for a nucleotide
sequence
```

Si queremos consultar el manual de alguna aplicación en particular, como por ejemplo `cusp`, lo que haremos será usar el comando `tfm`

```
tfm cusp

    Display full documentation for an application cusp

    Wiki

    The master copies of EMBOSS documentation are available at
    http://emboss.open-bio.org/wiki/Appdocs on the EMBOSS Wiki.

    Please help by correcting and extending the Wiki pages.

    Function

    Create a codon usage table from nucleotide sequence(s)

    Description

    cusp calculates a codon usage table for one or more nucleotide coding
    sequences and writes the table to file.

    The codon usage table gives for each codon: i. Sequence of the codon.
    ii. The encoded amino acid. iii. The proportion of usage of the codon
    among its redundant set, i.e. the set of codons which code for this
    codons amino acid. iv. The expected number of codons, given the input
    sequence(s), per 1000 bases. v. The observed number of codons in the
    input sequences.

    Usage

    Here is a sample session with cusp

    ....
```

Ya tenemos nuestra lista de secuencias codificantes así que ya estamos en condiciones de calcularel uso de codones usando `cusp`.

El *Team procariota*:

```
# Podemos hacerlo así (3 veces, una por cada archivo):
zcat GCF_000009565.1_ASM956v1_cds_from_genomic.fna.gz | cusp -sequence stdin -
outfile ecoli-clase.cusp

# O así (3 veces, una por cada archivo):
gzip -d GCF_000009565.1_ASM956v1_cds_from_genomic.fna.gz
cusp -sequence GCF_000009565.1_ASM956v1_cds_from_genomic.fna -outfile
GCF_000009565.1_ASM956v1_cds_from_genomic-clase.cusp

# O así (una sola vez!):
files=`ls GCF*.gz`
```

```
for file in ${files}; do zcat ${file} | cusp -auto -sequence stdin -outfile
${file}.cusp; done

# Todas las alternatias son correctas (incluso podría haber más... )
```

Dado que este comando puede demorar mucho en calcular todo, **ya tienen el archivo .cusp listo para usar en su carpeta de trabajo.**

Revisen la tabla de codones. ¿Qué representa? Pregunta para el *team procarita*, ¿Notan diferencias entre las frecuencias de uso de codones de los distintos proyectos genoma que analizaron?

```
head -20 <organismo-de-interes.cusp>
#CdsCount: 5421

#Coding GC 51.71%
#1st letter GC 58.60%
#2nd letter GC 40.86%
#3rd letter GC 55.68%

#Codon AA Fraction Frequency Number
GCA    A     0.215    20.289  27644
GCC    A     0.269    25.392  34597
GCG    A     0.354    33.395  45501
GCT    A     0.163    15.373  20946
TGC    C     0.550     6.681   9103
TGT    C     0.450     5.461   7440
GAC    D     0.372    18.796  25609
GAT    D     0.628    31.698  43189
GAA    E     0.691    39.176  53377
GAG    E     0.309    17.482  23819
TTC    F     0.425    16.517  22504
TTT    F     0.575    22.385  30499
```

▶ Ayuda-memoria con aminoácidos y sus abreviaturas

Si se pusieron a buscar diferencias a ojo entre un archivo y otro, todavía no aprendieron nada: EMBOSS tiene una herrramienta específicamente diseñada para hacer esta comparación (y, además, validarla estadísticamente - porque a ojo igual no íbamos a poder sacar ninguna conclusión).

```
codcmp <primer.cusp> <segund.cusp> cusp-comparison.out
```

Comparen dos tablas de frecuencia de uso de la misma especie (distinto proyecto) y entre especies. ¿Qué pueden decir al respecto? *Nota: El resultado aparecerá en el archivo cusp-comparison.out.*

3. Optimizar la secuencia en función de la tabla de uso de codones

Recuerden que todo este embrollo devino de la necesidad de expresar en forma heteróloga una proteína de interés (a no perder el foco, que ya falta poco! 😃), de modo que el siguiente paso es **adaptar nuestra secuencia quimérica al uso de codones del organismo en el que expresaremos nuestra proteína recombinante**.

Para ello, podemos usar `backtranseq`: Dada una secuencia de aminoácidos correspondiente a una proteína de interés, este programa nos permite (des?)-traducirla a la secuencia de DNA que, con mayor probabilidad, le dio origen. Es decir, backtranseq utilizará una tabla de uso de codones provista para escribir una secuencia de DNA a partir de una secuencia de aminoácidos. Revisen el manual de EMBOSS en búsqueda de información sobre cómo usar la herramienta.

```
# A Leer?
tfm backgranseq
# Igual acá está el que anda, vagxs.
backtranseq -auto -sequence <myprotein.fasta> -cfile <ecoli.cusp> -outfile
<myprotein.ecoli.codons.dna.fasta>
```

¡Tenemos que hacerlo para todas nuestras secuencias quiméricas!

El archivo `myprotein.ecoli.codons.dna.fasta` (o como hayan gustado llamarlo) tiene la secuencia en la que nos vamos a gastar nuestro Groupon.

4. Analizar los patrones de restriccion de mi nueva secuencia (optimizada)

FINALMENTE, lo ultimísimo que vamos a hacer es verificar que las enzimas que vamos a usar para clonar no están en nuestras secuencias quiméricas optimizadas. Podemos hacer utilizando `remap` (también de EMBOSS). Pero antes de eso, tendremos que indicarle a EMBOSS una base de datos de enzimas de restricción.

4.1. Instalar la base de datos de enzimas de restricción (REBASE) y configurarla para que la pueda usar EMBOSS

Bajar los archivos withrefm.907 y proto.907 desde aquí o desde aquí... aunque ya deberían estar descargados en la carpeta del TP.

(hoy la ultima version es 907, mañana esto puede cambiar!)

```
rebaseextract -infile withrefm.907 -protofile proto.907
```

Es posible que Ubuntu no nos deje hacer esto, dado que vamos a estar modificando cosas del sistema operativo, y para eso requerimos permisos. Podemos enseñarle quién manda con un `sudo`

```
#la contraseña es "unsam"
sudo rebaseextract -infile withrefm.907 -protofile proto.907
```

4.1. Verificar sitios de restricción

```
remap -auto -sequence myprotein.ecoli.codons.dna.fasta -single -width 80 -
commercial -sitelen 6 -frame 1 -enzymes all -outfile remap
```

Donde:

- -single = enzimas que cortan solo una vez (mincuts = maxcuts = 1)
- -commercial = only enzymes with commercial supplier
- -sitelen = min length of enzyme recognition site
- -width = ancho de secuencia en el outfile

Abran el archivo `remap` que acaban de generar. Pueden revisar la secuencia en busca de las enzimas que íbamos a usar para cortar (HindIII/BamHI). Si no están, estamos listos para agregarlas a la secuencia que vamos a pedir. ¿Qué pasa si están? ¿Qué alternativas tenemos? Si tuvieramos otras enzimas en el labo ¿Cuáles podríamos usar? ¿Qué enzimas podríamos usar para seleccionar clones una vez que tengamos nuestras construcciones transformadas?