Tutorial ensamblaje de metagenomica (https://notebook.community/germs-lab/frontiers-review-2015/frontiers-nb-2015)

Introduccion

El mayor desafío al que se enfrenta el uso de enfoques metagenómicos en microbiología es la necesidad de ampliar la formación en microbiología tradicional para incluir análisis de datos metagenómicos o de secuenciación. Sean Eddy (http://cryptogenomicon.org/2014/11/01/high-throughput-sequencing-para-neurociencia/#more-858) (biólogo computacional del Instituto Médico Howard Hughes) describe muy bien los impactos de la secuenciación de alto rendimiento en la biología y su capacitación en su discurso de apertura.

Para facilitar las barreras a los microbiólogos para el ensamblaje metagenómico, hemos complementado esta revisión con un tutorial sobre cómo estimar la abundancia de secuencias de referencia (por ejemplo, genes, contigs, etc.) en un metagenoma. Incluimos enfoques que incluyen el uso de referencias que son (i) referencias genómicas disponibles o (ii) ensambladas a partir del metagenoma. En general, para completar este tutorial y la mayoría del ensamblaje metagenómico, se necesitaría:

-Acceso a un servidor. La mayor parte del ensamblaje metagenómico requerirá más memoria de la que la mayoría de los investigadores tendrán en sus computadoras personales. En este tutorial, proporcionaremos capacitación sobre las instancias Amazon EC2 de acceso público que pueden ser alquiladas por cualquier persona con una cuenta registrada.

-Acceso a un conjunto de datos metagenómicos. Hemos seleccionado el uso del conjunto de datos HMP Mock Community WGS (http://www.hmpdacc.org/HMMC/) para este tutorial dada su disponibilidad, tamaño práctico y disponibilidad de genomas de referencia. Este conjunto de datos representa un metagenoma simulado de 22 organismos conocidos para los que se extrajo ADN de cultivos aislados, se combinó y se secuenció.

-Software para ensamblaje, lectura de mapas y anotación. Demostraremos la instalación de este software en un servidor basado en Ubuntu.

0.- Ponerse en la misma pagina

El primer paso de este tutorial es proporcionar a todos los usuarios acceso a un servidor para el que se pueden utilizar estas instrucciones, independientemente de la computadora en la que se encuentre. Para hacer esto, usaremos la computación en la nube. Más específicamente, Amazon Web Services Elastic Compute Cloud. Para alquilar tiempo de cómputo de Amazon Web Services (http://aws.amazon.com/ec2/pricing/), tendrá que registrarse y pagar con tarjeta de crédito. El costo es bastante manejable. Debería poder completar este tutorial en menos de cuatro horas, lo que equivale a <$ 1.

Una vez que se haya registrado en Amazon Web Services, debe seguir algunas instrucciones para iniciar una "instancia" o servidor en la nube. Para este tutorial, le sugerimos que utilice las instrucciones de Data Science Toolbox (http://datasciencetoolbox.org/#bundles). Un par de cosas a tener en cuenta antes de seguir esas instrucciones:

-Elija las instrucciones "en la nube"
-Puede elegir cualquier AMI, pero le sugerimos US EAST, ami-d1737bb8
-Cuando configure su instancia, elija la instancia m3.large
-No olvide "Agregar regla" como se describe en las instrucciones de la Caja de herramientas de ciencia de datos. Paso 2: -Agregue una "regla TCP personalizada" para el puerto "8888" y la fuente "En cualquier lugar".
-Complete las instrucciones de ciencia de datos hasta el paso 4. Una vez que llegue al paso 5, consulte a continuación.

Si tiene problemas para iniciar sesión en su instancia y está en un sistema operativo Mac o Linux:
-Verifique que haya cambiado los permisos en su archivo de clave (el que termina en * .pem)
-Asegúrese de ejecutar el comando ssh para iniciar sesión en la instancia en el mismo directorio que su archivo de seguridad o especificar la ubicación de ese archivo

Una vez que haya iniciado sesión en la instancia, por ejemplo, habrá ejecutado correctamente el siguiente comando (excepto que tendrá su propio archivo de seguridad con un nombre exclusivo y su propia dirección EC2 especial):

In [ ]:
```
$ssh -i MyKeyPair.pem ubuntu@ec2-XX-XX-XX-XXX.compute-1.amazonaws.com
```

Y ahora tiene una línea de comando que se parece a:

In [ ]:
```
ubuntu@ip-10-181-106-120:
```

Necesita hacer un par de cosas para que este tutorial se ejecute:

Copie y pegue los siguientes comandos uno por uno en su línea de comando y presione ENTER después de cada uno:

In [ ]:
```
cd /mnt sudo git clone https://github.com/germs-lab/frontiers-review-2015.git
```

A continuación, copie y pegue el siguiente comando e ingrese una contraseña de cuaderno de su elección cuando se le solicite:

In [ ]:
```
dst setup base
```

Luego, copie y pegue este comando:

In [ ]:
```
sudo ipython notebook --profile=dst --notebook-dir=/mnt/frontiers-review-2015
```

Esto iniciará un cuaderno IPython para este tutorial. Deje la pantalla del terminal abierta y busque su navegador de Internet, preferiblemente Google Chrome. También necesitará la dirección para el DNS público de su instancia EC2 que utilizó para iniciar sesión arriba "por ejemplo, ec2-XX-XX-XX-XXX". Si no lo sabe, siempre puede consultar su panel de AWS EC2 (consulte las instancias en ejecución) aquí (https://console.aws.amazon.com/ec2/v2/).

En su navegador web, vaya a (https://ec2-XX-XX-XX-XXX:8888 (https://ec2-XX-XX-XX-XXX:8888)) (excepto con su DNS público de EC2 específico). Casi todos los navegadores web tienen un mensaje que dice que se dirige a un lugar inseguro. No se alarme. En Chrome, puede presionar el enlace de opciones "Avanzadas" y presionar "Continuar de todos modos". Luego, escriba la contraseña (la contraseña es la que eligió anteriormente) y listo, verá un cuaderno que contiene este texto llamado "frontiers-nb-2015".

1. Cómo utilizar este cuaderno de IPython

Los cuadernos de IPython son muy útiles para entrenar bioinformática de forma colaborativa. Estos cuadernos han aparecido recientemente en Nature News (http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261 (http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261) and http://www.nature.com/news/programming-pick-up-python-1.16833 (http://www.nature.com/news/programming-pick-up-python-1.16833))

Al usar estos cuadernos, hay algunas cosas importantes a tener en cuenta. Hay dos tipos de contenido en este tutorial: texto y código. Este contenido se coloca en este cuaderno como "celdas". Si hace clic en esta página, verá diferentes celdas resaltadas. Para ejecutar cada celda (independientemente del contenido), presiona SHIFT + ENTER en el teclado. Si la celda contiene texto, el contenido se mostrará directamente. Si la celda contiene código, el código se ejecutará. Además, puede ejecutar todas las celdas en el cuaderno yendo a la pestaña Celda donde "Archivo, Editar, Ver, Insertar, Celda ..." están en la parte superior izquierda de esta página web y seleccionando *Ejecutar todo.*

2. Descargue el conjunto de datos del tutorial.

Comenzaremos este tutorial descargando el metagenoma simulado de HMP de los archivos de lectura corta de NCBI (SRA). Muchos metagenomas públicos se almacenan como archivos SRA en el NCBI. La forma más sencilla de obtener estos archivos SRA es utilizar un conjunto especial de herramientas llamado *sratoolkit*. Si tiene su ID de ejecución SRA del conjunto de datos (en este caso SRR172903), puede descargar el conjunto de datos y convertirlo al formato de secuencia estándar "fasta" o "fastq" para utilizar un programa especial para convertir el archivo.

In [ ]:
```
!wget http://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.4.5-2/sratoolkit.2.4.5-2-ubuntu64.tar.gz
```

In [ ]:
```
!tar -xvf sratoolkit.2.4.5-2-ubuntu64.tar.gz
```

Puede ver que ahora tenemos un archivo que contiene el software con el comando "ls". También verá este cuaderno en la lista de archivos en la ubicación actual en la que estamos trabajando.

In [ ]:
```
!ls
```

Ahora usaremos el programa sratoolkit instalado para descargar el conjunto de datos simulado de HMP en formato "fastq". (Esto lleva uno o dos minutos. Descubrirá que se requiere paciencia para trabajar con metagenomas. Lo bueno de trabajar en la nube es que está "alquilando" la potencia computacional para no utilizar la memoria de su computadora personal. liberándolo para las cosas que puede hacer mientras espera. Notará que puede ver que "Kernel ocupado" se mostrará en la esquina superior derecha de la pantalla debajo del botón "Cerrar sesión").

In [ ]:
```
!sratoolkit.2.4.5-2-ubuntu64/bin/fastq-dump SRR172903
```

3.- Control de Calidad

Hay varios métodos para determinar la calidad de los datos de secuenciación que reunirá. Primero, uno puede mirar los puntajes de calidad de sus lecturas de secuenciación y, si lo desea, recortar las lecturas con puntajes de calidad que no sean suficientes para sus necesidades. Hay una gran cantidad de herramientas disponibles para realizar un recorte de calidad de las lecturas de secuenciación, incluidas herramientas con buenos tutoriales que incluyen FastX Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/) y (http://khmer-protocols.readthedocs.org/en/v0.8-1/metagenomics/1-quality.html (http://khmer-protocols.readthedocs.org/en/v0.8-1/metagenomics/1-quality.html)), FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) y http://ged.msu.edu/angus/tutorials-2013/short-read-quality-assessment.html (http://ged.msu.edu/angus/tutorials-2013/short-read-quality-assessment.html)) y Sickle (https://github.com/najoshi/sickle) y (http://2014-5-metagenomics-workshop.readthedocs.org/en/latest/assembly/qtrim.html (http://2014-5-metagenomics-workshop.readthedocs.org/en/latest/assembly/qtrim.html)).

El archivo de datos de secuencia que ha descargado es un archivo de texto "fastq", donde los datos que describen cada lectura de secuenciación se muestran en cuatro líneas. Echemos un vistazo rápido:

In [ ]:
```
!head -n 4 SRR172903.fastq
```

-Esta primera línea (que comienza con "@ SRR172903.1") es el identificador de lectura, por lo general muestra el ID de lectura, alguna información para la función de secuenciación sobre la ejecución en la que se obtuvo.
-La segunda línea es la secuencia de ADN.
-La tercera línea es la misma que la primera, pero reemplazando la "@" con un "+", a veces esto es solo un "+" en algunos conjuntos de datos.
-La cuarta línea le brinda información sobre el puntaje de calidad de cada par de bases para la secuencia de ADN. Tenga en cuenta que tiene la misma longitud que la secuencia de ADN y que las puntuaciones de calidad se basan en puntuaciones de caracteres ASCII (con una compensación determinada por la tecnología de secuenciación, Illumina tiene actualmente una compensación de 64, p. Ej., Código ASCII 64 = 0 puntuación Phred). El puntaje de calidad es igual a -10 * log (p), donde p es la probabilidad de que la base se llame incorrecta (por ejemplo, si Q = 20, p = 0.01, la base de probabilidad del 1% se llama incorrecta).

Para este tutorial, eliminaremos lecturas que tengan más del 50% de la longitud de lectura con una puntuación Phred de menos de 33. Usaremos Fastx-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/commandline.html) que se puede utilizar para muchos tipos de control de calidad (p. ej., recorte de adaptadores). Primero, descargaremos, descomprimiremos y luego instalaremos este programa.

In [ ]:
```
!wget https://github.com/agordon/fastx_toolkit/releases/download/0.0.14/fastx_toolkit-0.0.14.tar.bz2
!wget https://github.com/agordon/libgtextutils/releases/download/0.7/libgtextutils-0.7.tar.gz
```

In [ ]:
```
!tar -xvf fastx_toolkit-0.0.14.tar.bz2
!tar -xvf libgtextutils-0.7.tar.gz
```

In [ ]:
```
!bash fastx_install.sh
```

Ahora, realizaremos el filtrado de calidad guardando el archivo filtrado por calidad como SRR172903.qc.fastq:

Filtro de calidad FASTQ

$fastq_quality_filter -h usage: fastq_quality_filter [-h] [-v] [-q N] [-p N] [-z] [-i INFILE] [-o OUTFILE]

In [ ]:
```
version 0.0.6 [-h] = This helpful help screen. [-q N] = Minimum qualit
y score to keep. [-p N] = Minimum percent of bases that must have [-q]
quality. [-z] = Compress output with GZIP. [-i INFILE] = FASTA/Q input
file. default is STDIN. [-o OUTFILE] = FASTA/Q output file. default is
STDOUT. [-v] = Verbose - report number of sequences. If [-o] is specif
ied, report will be printed to STDOUT. If [-o] is not specified (and o
utput goes to STDOUT), report will be printed to STDERR.
```

In [ ]:
```
!fastq_quality_filter -q 33 -p 50 -i SRR172903.fastq > SRR172903.qc.fa
stq
```

4. Comprobación de la diversidad - ¿Cuál es la distribución de "¿Quién está ahí?"

Una ventaja de la secuenciación metagenómica es la capacidad de cuantificar la diversidad microbiana en un entorno sin la necesidad de cultivar células primero. Normalmente, la mayoría de los estudios acceden a la diversidad taxonómica (especialmente con el uso de la secuenciación dirigida del gen de rRNA 16S *). La diversidad también se puede medir en la representación de patrones de secuencia específicos en un metagenoma. Por ejemplo, se puede cuantificar la abundancia de "palabras" de nucleótidos únicos de longitud K, o k-mers, en un metagenoma. Estos k-mers también se pueden usar en el ensamblaje de metagenomas donde los k-mers superpuestos son indicativos de lecturas que deben conectarse entre sí. La diversidad de estos k-mers puede darle una idea de la diversidad de su muestra. Además, dado que el ensamblaje compara cada k-mer con todos los k-mer, un mayor número de k-mer presentes requerirá más memoria computacional. Una buena revisión sobre k-mers y ensamblaje es Miller et al.

(* Tenga en cuenta que la secuenciación de amplicones de ARNr 16S es un enfoque dirigido y no se considera metagenómica en esta revisión. La secuenciación metagenómica de escopeta utiliza ADN extraído de todas las células de una comunidad y secuenciado. La secuenciación dirigida amplifica un locus genómico específico y se secuencia de forma independiente. Una gran revisión en El análisis del metagenoma es Sharpton et al.)

Lo primero que haremos es instalar khmer (www.github.com/ged-lab/khmer); contiene un conjunto de herramientas de premontaje y khmer. Lo usaremos para el conteo de k-mer aquí. Una vez que ejecute el siguiente script, puede usar las muchas herramientas de khmer.

In [ ]:
```
!ls
```

In [ ]:
```
!bash khmer-install.sh
```

La siguiente secuencia de comandos está incluida en el paquete khmer y puede estimar el número único total de k-mers en su conjunto de datos. Los casos de uso para esto podrían ser a) determinar qué tan diverso se compara un metagenoma con, por ejemplo, un genoma bacteriano para el ensamblaje, b) comparar la diversidad de k-mer entre múltiples metagenomas, c) explorar los impactos de la elección de la longitud k para el ensamblaje.

A continuación, para estimar el número de k-mers únicos en los conjuntos de datos para múltiples k (17, 21, 25, 29, 33, 37), ejecute los siguientes scripts. La salida del script identificará a los k-mers únicos, pero también se guardará en un informe llamado unique_count. (Esto tarda unos 15 minutos en una instancia grande y de 8 a 10 minutos en una extra grande).

In [ ]:
```
!python unique-kmers.py -R unique_count -k 17 SRR172903.qc.fastq
!python unique-kmers.py -R unique_count -k 21 SRR172903.qc.fastq
!python unique-kmers.py -R unique_count -k 25 SRR172903.qc.fastq
!python unique-kmers.py -R unique_count -k 29 SRR172903.qc.fastq
!python unique-kmers.py -R unique_count -k 33 SRR172903.qc.fastq
!python unique-kmers.py -R unique_count -k 37 SRR172903.qc.fastq
```

Puede ver que este archivo ahora tiene en la primera columna la longitud k-mer y en la segunda columna el número estimado de palabras de longitud k en los metagenomas. Si tuviera varios genomas, podría comparar la diversidad de, por ejemplo, el número total de k-mers en los conjuntos de datos. Para ver los resultados del archivo, puede utilizar el programa / comando concatenado "cat".

In [ ]:
```
!cat unique_count
```

5. Obtener un perfil de cobertura de secuencia: ¿Qué genes están presentes en mi metagenoma?

La mayoría de los análisis metagenómicos requieren que uno estime la abundancia de genes de referencia (por ejemplo, que se originan a partir de genomas o del propio ensamblaje metagenómico). Este tutorial cubrirá ambos casos en los que las referencias están disponibles o no disponibles (que requieren ensamblaje de novo).

6. Caso I - Genomas de referencia disponibles.

Para el metagenoma simulado de HMP, el HMP ha secuenciado los genomas de los aislados utilizados para este conjunto de datos simulado. La lista de estos genomas se puede obtener en el sitio web de HMP, y la proporcionamos aquí en un repositorio de Github, una herramienta utilizada para compartir datos y código de forma colaborativa. El siguiente comando descargará datos para este tutorial.

In [ ]:
```
!cat ncbi_acc.txt
```

El siguiente comando descarga todos los genomas para cada ID en la lista anterior en un directorio llamado "genomas".

In [ ]:
```
!python fetch-genomes-fasta.py ncbi_acc.txt genomes
```

7. Estimación de la abundancia de contigs ensamblados

Para estimar la representación de genes o genomas de referencia en su metagenoma, puede alinear las lecturas con las referencias utilizando software de mapeo de lectura (por ejemplo, Bowtie2, BWA, etc.). En este tutorial, usaremos Bowtie2 que instalaremos en este servidor. Luego, mapearemos el metagenoma a un solo genoma de referencia (que descargamos arriba).

In [ ]:
```
!wget http://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.2.5/b
owtie2-2.2.5-linux-x86_64.zip
!unzip bowtie2-2.2.5-linux-x86_64.zip
```

Escribí un comando que asignará automáticamente un conjunto de lecturas a una referencia dada y generará un archivo que contiene el número de lecturas que se asignan a una referencia determinada. Para usar este comando, también necesitaremos instalar samtools. Un samfile es un archivo supercomprimido que almacena de manera eficiente información mapeada de los mapeadores. Samtools nos ayuda a interactuar con este archivo.

In [ ]:
```
!apt-get install samtools
```

Para asignar lecturas a una referencia, hemos proporcionado un programa fácil de usar.

Los pasos que realiza el programa son los siguientes:

-Indexe su genoma de referencia
-El mapa lee su genoma índice (con parámetros de pajarita predeterminados)
-Utilice Samtools para estimar el número de lecturas mapeadas, el número de lecturas no mapeadas y proporcione un archivo delimitado por tabulaciones con cada línea que consta de nombre de secuencia de referencia, longitud de secuencia, # lecturas mapeadas y # lecturas no mapeadas.

Esto tarda entre 8 y 10 minutos.

In [ ]:
```
!bash bowtie.sh genomes/NC_000913.2.fa SRR172903.qc.fastq
```

Podemos ver el número total de lecturas mapeadas y no mapeadas de nuestro metagenoma al genoma NC_000913.2.

También podemos obtener un archivo que muestre el nombre de la secuencia de referencia (primera columna), la longitud de la secuencia de referencia (segunda columna), # lecturas mapeadas (tercera columna) y # lecturas no mapeadas (última columna).

Las otras columnas contienen información que samtools puede usar para otras consultas, puede leer sobre samtools aqu (http://samtools.sourceforge.net/samtools.shtml.)

In [ ]:
```
!cat reads-mapped.count.txt
```

In [ ]:
```
!cat reads-unmapped.count.txt
```

In [ ]:
```
!cat reads.by.contigs.txt
```

Si desea un desafío, puede intentar mapear el metagenoma a todos los genomas de referencia proporcionados en la carpeta del genoma. Para hacerlo, intente concatenar todos los genomas en un archivo (usando este comando: "cat genomes / * fa >> all-genomes.fa") y ejecute el programa en all-genomes.fa en lugar de NC_000913.2.fa.

8. Caso II - Ensamblaje de novo de genes de referencia.

Montaje del metagenoma simulado de HMP

El ensamblaje es el proceso de fusionar lecturas metagenómicas superpuestas del mismo genoma en una secuencia más larga y continua (más comúnmente llamada contig). Es ventajoso porque proporciona longitudes más largas para las secuencias que luego pueden usarse como referencias (que pueden ser previamente desconocidas), reduce el tamaño del conjunto de datos para el análisis y proporciona referencias que no dependen de conocimientos previos.

La elección de qué ensamblador usar no es fácil y es un tema de debate ver la pagina (http://assemblathon.org). Es muy importante recordar que un ensamblado es una representación de consenso hipotética de su conjunto de datos. El montaje en sí es un paso inicial que debe ir seguido de una evaluación de su precisión y utilidad. Para la mayoría de los ensambladores, las entradas son lecturas secuenciales y parámetros para el software de ensamblaje. Para este tutorial, completaremos el ensamblaje con un ensamblador publicado en 2014 llamadoMegahit (Li et al., 2015) (https://github.com/voutcn/megahit). La revisión de Sharpton (Sharpton, 2014) también revisa bastante bien algunos de los muchos programas de ensamblaje y enfoques para el ensamblaje metagenómico.

Para reducir la memoria que se necesita, a menudo es ventajoso normalizar la distribución de k-mers en un metagenoma. La eliminación de información extraña que no es necesaria para el ensamblaje también elimina las lecturas que pueden contener errores y pueden mejorar el ensamblaje (http://arxiv.org/abs/1203.4802). Estos guiones y tutoriales están disponibles disponibles aquì (http://ged.msu.edu/angus/diginorm-2012/tutorial.html).

Para este tutorial, ensamblaremos nuestro metagenoma con el ensamblador Megathit, así que primero tenemos que instalarlo.

In [ ]:
```
!bash install-megahit.sh
```

Este ensamblaje tardará unos 15 minutos y guardará el ensamblaje en una carpeta con el nombre "megahit_assembly". Puede leer acerca de los parámetros de este programa, como --memory que especifica la memoria máxima que se puede usar en el repositorio de software megahit (https://github.com/voutcn/megahit).

In [ ]:
```
!megahit/megahit --memory 10e9 -l 250 --k-max 81 -r SRR172903.qc.fastq
--cpu-only -o megahit_assembly
```

Para echar un vistazo al ensamblaje, ejecutemos el programa de resumen del ensamblaje khmer en él, los contigs finales están en megahit_assembly/final.contigs.fa. Obtengamos estadísticas de todos los contigs mayores o iguales a 200 pb.

In [ ]:
```
!python khmer/sandbox/assemstats3.py 200
megahit_assembly/final.contigs.fa
```

9. Estimación de abundancias de contigs

Una vez finalizado el ensamblaje, tiene un conjunto de contigs de referencia que ahora puede estimar la abundancia del metagenoma. El enfoque para hacerlo es idéntico al que se muestra arriba, donde usa genomas de referencia.

Esto le llevará unos 20 minutos.

In [ ]:
```
!bash bowtie.sh megahit_assembly/final.contigs.fa SRR172903.qc.fastq
```

Puede echar un vistazo a los resultados del mapeo de manera muy similar a como lo hizo anteriormente cuando estábamos mapeando las lecturas del genoma del NCBI.

In [ ]:
```
!cat reads-mapped.count.txt
```

In [ ]:
```
!cat reads-unmapped.count.txt
```

In [ ]:
```
!cat reads.by.contigs.txt
```

10. Anotando los contigs ensamblados

La secuenciación se utiliza a menudo para determinar "quién" y / o "qué" hay en su muestra. En nuestro caso, sabemos que la comunidad simulada de HMP debería originarse a partir de un conjunto de genomas (que en realidad hemos descargado anteriormente). Una de las herramientas más populares para comparar una secuencia desconocida con una referencia conocida es la herramienta básica de búsqueda de alineación local (o BLAST). Para identificar el origen de nuestros contigs, alinearemos los contigs ensamblados con los genomas utilizados en la comunidad simulada de HMP.

Lo primero que haremos será descargar el software BLAST. Dado el creciente volumen de conjuntos de datos de secuenciación, también se pueden considerar nuevas herramientas para una anotación más eficiente que ahora están disponibles, como Diamond (https://github.com/bbuchfink/diamond/), y (http://dx.doi.org/10.1038/nmeth.3176 (http://dx.doi.org/10.1038/nmeth.3176)).

In [ ]:
```
!wget ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.30/ncbi-blast-2.2.30+-x64-linux.tar.gz
!tar -xvf ncbi-blast-2.2.30+-x64-linux.tar.gz
```

Ahora, crearemos una base de datos con capacidad de búsqueda para el software BLAST. Primero, concatenamos todos los genomas del directorio de genomas en un archivo.

In [ ]:
```
!cat genomes/*fa >> all-genomes.fa
!ncbi-blast-2.2.30+/bin/makeblastdb -in all-genomes.fa -dbtype nucl -out all-genomes
!ncbi-blast-2.2.30+/bin/blastn -db all-genomes -query megahit_assembly/final.contigs.fa -outfmt 6 -out contigs.x.all-genomes.blastnout
```

El comando anterior alinea cada consulta (cada secuencia en el archivo final.contings.fa ensamblado) con cada secuencia (por ejemplo, genoma en all-genomes.fa). -Outfmt le dice al programa que guarde los resultados en un formato delimitado por tabulaciones en el archivo -out contigs.x.all-genomes.blastnout.

Echemos un vistazo a las primeras 10 líneas de ese archivo. Verá la consulta (contig) y el hit (genoma) seguidos del porcentaje de identidad, la longitud de la alineación, los recuentos de discrepancias, los recuentos de espacios abiertos, la posición de inicio de la consulta, la posición de finalización de la consulta, la posición de inicio del sujeto, la posición de finalización del sujeto, E -valor y puntaje de bits.

In [ ]:
```
!head -n 10 contigs.x.all-genomes.blastnout
```

Dependiendo de su pregunta científica, puede ser más interesante tener marcos de lectura abiertos (ORF) anotados en lugar de secuencias contig. En este caso, existen múltiples llamadores ORF (por ejemplo, FragGeneScan (http://nar.oxfordjournals.org/content/early/2010/08/29/nar.gkq747.abstract) y Metagene (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1636498/)) que se pueden utilizar. Podemos llamar a ORF desde nuestros contigs usando FragGeneScan. Descargaremos, instalaremos y luego llamaremos a los ORF desde nuestros contigs de la siguiente manera:

In [ ]:
```
!wget http://downloads.sourceforge.net/project/fraggenescan/FragGeneScan1.19.tar.gz
!tar -xvf FragGeneScan1.19.tar.gz
```

In [ ]:
```
!bash fraggenescan-install.sh
```

Ejecutaremos FragGeneScan en los contigs ensamblados, asumiendo que se ajusta al perfil de entrenamiento de una secuencia de genoma "completa" (en su documentación, esto equivale a secuencias genómicas completas o lecturas de secuencia corta sin error de secuenciación).

In [ ]:
```
!FragGeneScan1.19/FragGeneScan -s megahit_assembly/final.contigs.fa -o
final.contigs.orfs.fa -w 1 -t complete
```

Los ORF llamados están contenidos como archivos FASTA en final.contigs.orfs.fa.faa (aminoácidos) y final.contigs.orfs.fa.ffn (nucleótidos). Puede anotarlos en una base de datos de su elección tal como se describe para los contigs arriba.

11.- Yendo adelante

Ahora tiene toda la información que necesita para producir la siguiente información:

Información de abundancia de secuencias: secuencia (p. Ej., Contig) y abundancia (p. Ej., Número de lecturas mapeadas).

Información de anotación de secuencia: secuencia (p. Ej., Contig) y genoma de NCBI

Notarás que esto es similar al análisis de amplicones de ARNr 16S en el que tendrías una tabla de abundancia de OTU y anotaciones de mejores resultados de OTU. Para el análisis metagenómico, esta información lo lleva a más paquetes de análisis y visualización como PhyloSeq en R (http://joey711.github.io/phyloseq/).

Una vez que haya leído este tutorial en este libro de trabajo, un buen ejercicio sería intentar ejecutar el ensamblado fuera del entorno de IPython Notebook. Para hacerlo, puede iniciar sesión en su instancia EC2, navegar hasta el directorio el directorio (cd/mnt/frontiers-review-2015) donde se almacenan estos datos y puede ejecutar todos los comandos de este cuaderno en la línea de comandos (con la excepción del "!" al comienzo de cada comando en el cuaderno. Además, tenga en cuenta que no tendrá que reinstalar el software.

Similar notebooks:
frontiers-nb-2015.bu
ChIP-Seq-I-InClass
ipyrad-manuscript-refmap-horserace
assessment
reference_assembly-checkpoint
example_usage
3 Validation Demultiplexing and Quality Control
03-integrateREwithMTs
run_ariba
02.Exploring_PacBio