Tarea 3 – Alineamiento lecturas cortas

Objetivos

1. Entender el proceso de alineamiento de lecturas a un genoma de referencia
2. Visualizar alineamientos
3. Practicar trabajo en el cluster

Antes de comenzar: Todos los archivos necesarios para este taller están en la carpeta `/hpcfs/home/bcom4006/` del cluster.

Parte 1. Alineamiento de lecturas en el cluster

1. Ingresar al cluster.

Desde windows descargar el ejecutable de PuTTY (archivo putty.exe en http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html) y escribir "magnus" en el campo llamado "hostname". PuTTY solicitará el usuario (bcom4006) y la clave.

Desde MAC o linux abrir una linea de comandos y ejecutar:

```
ssh bcom4006@magnus.uniandes.edu.co
```

La consola solicitará la clave.

En los dos casos, una vez se ingrese al nodo login del cluster, la linea de comandos debería verse así:

```
[bcom4006@magnus ~]$
```

2. Reservar un nodo interactivo. Escribir primero el comando:

```
[bcom4006@magnus ~]$ salloc --nodes=1 --ntasks-per-node=2 --mem=8G
--time=01:00:00
```

Este comando permite solicitar recursos para luego hacer uso de ellos. Genera la siguiente salida:

```
salloc: Granted job allocation 78239
salloc: Waiting for resource configuration
salloc: Nodes node-[2,22] are ready for job
```

El mensaje indica que los recursos solicitados fueron concedidos. Ahora debe conectarse al nodo indicado, el cual dispone de los recursos solicitados, para trabajar de manera interactiva. Por ejemplo:

[bcom4002@magnus ~]$ ssh node-22

La linea de comandos queda así:

```
[bcom4006@<NODO> ~]$
```

Donde NODO corresponde al nodo que ha sido reservado (cambia para cada persona).

Como estamos trabajando con el mismo usuario, debemos ir a carpetas distintas para no escribir todos el mismo archivo. Se debe ir a la carpeta correspondiente al número de estudiante que le ha sido asignado. Dado un número de estudiante <N>, escribir:

```
[bcom4006@<NODO> ~]$ cd estudiantes/estudiante<N>
```

Donde <N> es el número asignado.

3. Crear carpetas para este trabajo. Para que los archivos de la tarea queden bien organizados, crear las siguientes carpetas utilizando el comando mkdir:

```
mkdir Tarea3
mkdir Tarea3/reference
mkdir Tarea3/reads
mkdir Tarea3/mapping
```

4. Copiar archivos de entrada. En la carpeta `/hpcfs/home/bcom4006/data/yeast` están los archivos con el genoma de referencia y las lecturas para analizar en esta tarea. Como el genoma de referencia es un archivo pequeño en este caso (12Mb) se puede copiar a la carpeta de referencia de cada uno

```
cp /hpcfs/home/bcom4006/data/yeast/reference/yeastGenome.fa Tarea3/
reference/
```

A pesar de que los archivos de lecturas también se podrían copiar, normalmente estos archivos son mucho más grandes, entonces para esta tarea vamos a practicar creando enlaces simbólicos, de modo que parezca que cada uno tiene una copia de los archivos fastq:

```
ln -s /hpcfs/home/bcom4006/data/yeast/reads/*fastq Tarea3/reads/
```

5. Indexar genoma de referencia. Para alinear las lecturas al genoma de referencia vamos a utilizar la herramienta bowtie2. Los ejecutables de la versión 2.3.3 de está herramienta están disponibles en `/hpcfs/home/bcom4006/software/bowtie2-2.3.3/`. El primer paso es construir un índice FM para el genoma de referencia utilizando el comando bowtie2-build. El comando recibe 2 parámetros, la ruta al genoma de referencia y el prefijo para los archivos que constituyen el índice:

```
cd Tarea3/reference/
/hpcfs/home/bcom4006/software/bowtie2-2.3.3/bowtie2-build
yeastGenome.fa yeastGenomeFMIndex
```

Si todo sale bien, se deberían generar los siguientes 6 archivos:

```
yeastGenomeFMIndex.1.bt2
yeastGenomeFMIndex.2.bt2
yeastGenomeFMIndex.3.bt2
yeastGenomeFMIndex.4.bt2
yeastGenomeFMIndex.rev.1.bt2
yeastGenomeFMIndex.rev.2.bt2
```

Estos son los archivos que va a utilizar bowtie2 para alinear las lecturas en el siguiente paso.

6. Alinear lecturas al genoma. La operación principal de bowtie2 es el alineamiento de lecturas cortas al genoma de referencia. Para alinear las lecturas de la muestra Seg5, vaya a la carpeta mapping

```
cd ../mapping
```

y ejecute este comando:

```
/hpcfs/home/bcom4006/software/bowtie2-2.3.3/bowtie2 --rg-id Seg5 --
rg SM:Seg5 --rg PL:ILLUMINA -X 1000 -x
../reference/yeastGenomeFMIndex -1 ../reads/Seg5-1M_1.fastq -2
../reads/Seg5-1M_2.fastq -S Seg5_bowtie2.sam >& Seg5_bowtie2.log &
```

Los parámetros más importantes de este comando son -x, -1, -2 y -S. El parámetro -x permite indicar la ruta al índice FM del genoma de referencia generado en el paso anterior. Los parámetros -1 y -2 se usan cuando se hace un experimento de secuenciación que produce lecturas pareadas para indicar las rutas a los dos archivos que tienen la primera lectura y la segunda lectura de cada fragmento secuenciado (`../reads/Seg5-1M_1.fastq` y `../reads/Seg5-1M_2.fastq` en este caso). El parámetro -S permite indicar el archivo de salida, el cual estará en formato SAM. Los parámetros –rg-id y –rg permiten guardar en el encabezado el id de la muestra secuenciada. Aunque formalmente no son obligatorios, algunas herramientas de detección de SNPs exigen que estos encabezados esten presentes. Finalmente, la

opción -X indica que la longitud máxima de un alineamiento pareado válido va a ser de 1000bp y no de 500bp como sería el valor por defecto. Esta es una decisión conservadora que se utiliza cuando no se conoce el tamaño del inserto seleccionado durante la preparación de la librería (como en este caso).

Finalmente, si todo sale bien, el archivo Seg5_bowtie2.log indica varias estadísticas útilies incluyendo el número de fragmentos que aparecen en los archivos fastq. La cantidad que fueron alineados como pares consistentes, tanto de forma única como múltiples veces, la cantidad que fueron alineados como pares de manera inconsistente con el experimento (muy cerca, muy lejos o con orientación incorrecta) y de los que no fueron alineados como pares, las lecturas que se pudieron alinear como lecturas sencillas. Para la tarea se debe repetir este proceso para las otras 3 muestras disponibles en el directorio reads y reportar una tabla con las estadísticas que ofrece bowtie2.

PRECAUCION: El ampersand final del comando de alineamiento indica al nodo de trabajo, que este proceso se va a correr como un subproceso para que la consola quede disponible. Siempre que hagan esto con cualquier proceso asegurense antes que se encuentran en un nodo de trabajo y no en el nodo login. Se recomienda que solo trabajen así en casos en los que los procesos no demoren mucho porque estos procesos se eliminan una vez terminan la sesión en el nodo interactivo o se cae la sesión. Además en modo interactivo solo es recomendable lanzar un proceso a la vez. Para enviar procesos largos y/o en paralelo se recomienda aprender a lanzar procesos desde el nodo login utilizando sbatch.

7. Ordenar alineamientos respecto al genoma de referencia. Una vez terminado el alineamiento de lecturas, el archivo SAM resultante es un archivo de texto que se puede visualizar utilizando los comandos normales de linux (head, more, less, tail, etc) pero cuyo formato solo se puede entender leyendo la documentación (https://samtools.github.io/hts-specs/SAMv1.pdf). Por lo pronto, lo más importante para entender es que las lecturas en este archivo todavía están ordenadas de acuerdo a como aparecen en los archivos fastq. Para facilitar todos los procesos posteriores, el siguiente paso del proceso consiste en ordenar los alineamientos de acuerdo con el genoma de referencia, de tal manera que se vean primero las lecturas alineadas al primer cromosoma (chrI), luego al segundo (chrII) y así en adelante. Para esto vamos a utilizar la herramienta llamada “picard”. Como esta es una herramienta escrita en java, debemos cargar la última versión de java en la sesión de trabajo del cluster

```
module load jdk/1.8.0_121
```

El archivo jar que representa la aplicación picard está disponible en `/hpcfs/home/bcom4006/software/picard.jar`. Para ejecutarlo, escribir:

```
mkdir tmpdir_sort_Seg5
```

```
java -Xmx4g -jar /hpcfs/home/bcom4006/software/picard.jar SortSam SO=coordinate CREATE_INDEX=true TMP_DIR=tmpdir_sort_Seg5 I=Seg5_bowtie2.sam O=Seg5_bowtie2_sorted.bam >& Seg5_bowtie2_sort.log &
```

Este comando recibe como entrada el archivo SAM con los alineamientos y genera un archivo BAM (versión comprimida del formato SAM) con los alineamientos ordenados de acuerdo con el genoma de referencia. El directorio generado el la primera linea es un directorio de trabajo temporal que se puede eliminar al final. La opción "CREATE_INDEX=true" permite generar el archivo adicional Seg5_bowtie2_sorted.bai que es utilizado por varias herramientas como un índice para buscar rápidamente alineamientos en regiones específicas del genoma.

8. Estadísticas de calidad, cubrimiento y tamaño de fragmento. La información guardada en los archivos de alineamientos no se pueden visualizar completamente en una sola pantalla. Por esto, se deben utilizar diferentes estadísticas para resumir la información y poder entender los resultados del experimento. En este caso vamos a ejecutar dos estadísticas disponibles en el software NGSEP para visualizar la tasa de error de secuenciación y la distribución de profundidad por el genoma. La última versión de NGSEP está disponible en `/hpcfs/home/bcom4006/software/NGSEPcore_3.2.0.jar`. Para ejecutar las estadísticas de calidad, escribir:

```
java -Xmx1g -jar /hpcfs/home/bcom4006/software/NGSEPcore_3.2.0.jar QualStats ../reference/yeastGenome.fa Seg5_bowtie2_sorted.bam >& Seg5_bowtie2_readpos.stats &
```

Este comando genera un archivo de texto con 5 columnas:
1. Número de base en la lectura (5' a 3')
2. Número de diferencias con el genoma de referencia considerando todos los alineamientos
3. Número de diferencias con el genoma de referencia considerando solo alineamientos únicos
4. Total de alineamientos
5. Total de alineamientos únicos

Estas estadísticas se pueden cargar en excel. Para estimar la tasa de error, la mejor forma es dividir la columna 3 entre la 5. Esta es una sobreestimación porque algunas de las diferencias que se contabilizan en las columnas 2 y 3 son debidas a variación real. Sin embargo, sobretodo en WGS, la variación real debería generar una distribución uniforme para esta gráfica. Para la tarea, ejecutar este análisis para todas las muestras y reportar por cada muestra la curva que representa la ta tasa de error promedio, la tasa de error máxima y en qué base ocurre esta tasa máxima.

Para ejecutar las estadísticas de cubrimiento, ejecutar el siguiente comando:

```
java -Xmx1g -jar /hpcfs/home/bcom4006/software/NGSEPcore_3.2.0.jar CoverageStats Seg5_bowtie2_sorted.bam Seg5_bowtie2_coverage.stats >& Seg5_bowtie2_coverage.log &
```

Este comando produce un reporte con las siguientes 3 columnas:

1. Profundidad
2. Número de bases del genoma con la profundidad dada en la columna 1, teniendo en cuenta todos los alineamientos
3. Número de bases del genoma con la profundidad dada en la columna 1, teniendo en cuenta solo alineamientos únicos

Nuevamente este reporte se puede cargar en excel y se puede graficar cualquiera de las columnas 2 o 3 como gráfica de barras. Para la tarea, ejecutar estas estadísticas sobre todas las muestras y reportar las gráficas de profundidad que se obtienen y la moda de la distribución para cada muestra.

Finalmente, la herramienta picard tiene un comando llamado CollectInsertSizeMetrics que permite obtener la distribución de tamaños de fragmento que se pueden predecir a partir de los alineamientos, para compararla con el tamaño de fragmento escogido durante la preparación de la librería. Desafortunadamente esta utilidad algunas veces falla con algunos archivos BAM. Una forma relativamente sencilla de generar esta distribución se puede lograr combinando el comando view de la herramienta samtools con el comando awk. Para calcular esta distribución, podemos comenzar por cargar el módulo de samtools:

```
module load samtools/1.8
```

Luego, el comando para generar la distribución sería este:

```
samtools view -F 268 Seg5_bowtie2_sorted.bam | awk '{s=$9;if(s>=0){i=sprintf("%d",s/25)+1;if(i<100)a[i]++;else aM++}}END{for(i=1;i<100;i++)print (i-1)*25,a[i];print "More",aM}' > Seg5_bowtie2_insertLength.stats &
```

Para la tarea, generar la distribución de tamaño de fragmento para cada muestra y determinar qué tamaño escogerían como mínimo y como máximo si fuera necesario ejecutar de nuevo el alineamiento.

9. Terminar la sesión. Para salir del nodo de trabajo interactivo ejecutar el comando “exit”

[<USUARIO>@<NODO> ~]$ exit

y salir también del nodo de login utilizando nuevamente “exit”:

[<USUARIO>@magnus ~]$ exit

Parte 2: Visualización de alineamientos

Aunque los archivos BAM se pueden visualizar en consola utilizando el programa samtools, existen diferentes herramientas que permiten visualizar alineamientos de manera más gráfica e interactiva. Para esto vamos a utilizar la herramienta IGV, la cual se puede descargar e instalar a partir de este enlace:

http://software.broadinstitute.org/software/igv/download

Nota: Es recomendable para este y otros programas tener instalado el jdk, versión 1.8 o superior. Si no lo tienen instalado, lo pueden bajar e instalar de este enlace:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.htm

Para tener los archivos a visualizar, descargar el genoma de referencia y todos los archivos .bam y .bai generados en el cluster. Esto se puede hacer de manera gráfica utilizando herramientas como “FileZilla”. Una vez descargados los archivos, el genoma de referencia se puede cargar utilizando la opción del menu “Genome → Load genome from file”. Esta opción despliega un cuadro de dialogo donde se debe indicar el archivo yeastGenome.fa utilizado para alinear las lecturas.

Cada archivo BAM se puede cargar utlilizando la opción “File → Load from file”. Es necesario sin embargo que el archivo BAI que corresponde a cada archivo BAM esté disponible en la misma carpeta.

Una vez cargados los archivos, y al hacer zoom en una región del genoma, IGV muestra una vista de los alineamientos por cada muestra y una curva de la profundidad en cada punto del genoma. Las lecturas alineadas se representan por barras de diferentes colores. Para visualizarlas como pares, se puede hacer clic derecho en cualquier parte de la visualización de lecturas y seleccionar la opción “View as pairs”. Al pasar el ratón por cada alineamiento, se observa la información de la lectura correspondiente.

Para la tarea, investigar en la documentación de IGV y contestar las siguientes preguntas:

1. ¿Qué significan los diferentes colores asignados por IGV a los alineamientos?
2. ¿Cómo se ve una posición con un SNP? Determine 3 posiciones en las que visualmente parece haber un SNP e indique para cada muestra si el genotipo sería homocigoto o heterocigoto?

3. Mismas preguntas del punto anterior para 2 borrados y para 2 inserciones pequeñas.

Parte 3: Desarrollo de índices FM

El objetivo de esta parte es construir una implementación sencilla de un índice FM. Pueden utilizar el diseño que consideren más conveniente. Para trabajar en modo orientado por objetos, se recomienda construir la clase ArregloSufijos, la clase IndiceFM y una clase principal con un main que reciba el archivo fasta con la referencia y el archivo fastq con las lecturas a mapear y genere un archivo de texto con las coincidencias encontradas por cda lectura.

Primer paso. Construir un arreglo de sufijos
1. Leer una secuencia en formato fasta
2. Construir una lista de Strings con todos los sufijos de la secuencia original
3. Ordenar la lista (pueden utilizar Collections.sort)
4. Escribir una función que retorne un arreglo de enteros con las posiciones de inicio de los diferentes sufijos.

Segundo paso. Mapear lecturas al arreglo de sufijos.
1. Leer un archivo de lecturas en formato fastq
2. Por cada lectura, realizar búsqueda binaria para determinar en qué posiciones del arreglo de sufijos se encuentra cada cadena de búsqueda

Tercer paso. Construir un FM-index
1. Calcular la BWT, recorriendo el arreglo de sufijos y concatenando el carácter anterior a cada comienzo de sufijo
2. Recorrer la BWT para calcular la matriz de conteos (tally indexes)

Cuarto paso. Mapear lecturas utilizando el índice FM.
1. Implementar la función que para una posición del arreglo de sufijos calcula la posición anterior (operación LF)
2. Utilizar esta función para implementar la búsqueda de coincidencias de acuerdo con el algoritmo visto en clase. Utilizar el arrreglo de sufijos para saber las posiciones de inicio de cada coincidencia.

Bono [10%]. El problema principal de esta implementación es que generar la lista de Strings con todos los sufijos de la secuencia es muy ineficiente en espacio. Modificar la construcción del arreglo de sufijos para generar el arreglo ordenado de posiciones sin tener que calcular explicitamente los sufijos.