Instructivo WORKSHOP 3CAB2C NGS: Ensamblado+RNA-Seq 2012

Convención: las líneas que comienzan con $ indican comandos, copiar y pegar generalmente no funcionará debido a diferencias en los caracteres "–" y "-" , así que CUIDADO! Obviamente, el carácter $ no forma parte del comando…

Control de calidad de secuencias

FASTQC

FastQC es un programa para visualizar estadísticas sobre una o más corridas de instrumentos de nueva generación para determinar posibles desviaciones de la información que evidencien problemas en el uso de los datos.

Para ejecutar, realizar en la línea de comandos:

$ fastqc&

Abrirá una ventana donde se pueden cargar los archivos en la interfaz gráfica.

Existe una carpeta llamada "calidad" en la localización /media/DATOS/calidad. En ella hay varios archivos FASTQ, para visualizar. En FASTQC, cargue las secuencias:

- bBT326E11_1.fastq (Illumina)
- Illumina36bp.fastq (Illumina)
- Tdemo.fastq (454)

Para el archivo bBT326E11_1.fastq:

Analizar:

Largo de lecturas

Calidad -> ¿Cómo se comportan las secuencias hacia el final?

Duplicación. ¿Qué indica este parámetro? Se utilizará en ensamblado

Para el archivo Illumina36bp.fastq

Analizar:

Largo de lecturas

Calidad -> ¿Cómo se comportan las secuencias hacia el final?

Contenido de secuencias por base

Secuencias sobrerepresentadas. ¿Cuáles son las secuencias que se encuentran en mayor número? ¿Tiene esto sentido?

Para el archivo Tdemo.fastq

Analizar:

Largo de lecturas. ¿Cómo cambia con los ejemplos anteriores?

Calidad -> ¿Cómo se comportan las secuencias hacia el final?

Contenido de secuencias por base y GC por base. Observe el comienzo de las secuencias ¿Puede encontrar alguna razón para el resultado obtenido?

PRINSEQ

PRINSEQ es un programa similar a FastQC, que realiza un análisis de los datos, y permite además filtrar las secuencias utilizando diferentes controles de calidad. Fue especialmente diseñado para lecturas de Roche 454, pero puede utilizarse con todas las tecnologías.

Está disponible una versión web (http://edwards.sdsu.edu/prinseq_beta/) que puede utilizarse online o descargar y utilizar en un servidor propio. Además, existe una versión "*lite*" que permite su uso en la línea de comandos (es especialmente recomendable cuando se trabaja con archivos muy grandes).

prinseq-lite y prinseq-graphs son programas de Perl que pueden correrse localmente. Utilizando la opción –h pueden verse las opciones correspondientes (que son muchas…). Para correrlo, primero se utiliza prinseq-lite, que genera las estadísticas, y prinseq-graph genera luego los gráficos.

$ prinseq-lite.pl -verbose -fastq Tdemo.fastq -graph_data Tdemo.gd -out_good null -out_bad null

$ prinseq-graphs.pl –I Tdemo.gd –png_all –o resultado

Esto generará una serie de archivos de imágenes que contienen los gráficos:

resultado_cd.png	resultado_dm.png	resultado_or.png	resultado_qd3.png
resultado_ce.png	resultado_gc.png	resultado_pm.png	resultado_qd.png
resultado_df.png	resultado_ld.png	resultado_pv.png	resultado_td3.png
resultado_dl.png	resultado_ns.png	resultado_qd2.png	resultado_td5.png

También puede generarse una salida en formato HTML que contendrá los resultados para ver con un navegador:

$ prinseq-graphs.pl -i Tdemo.gd -html_all -o resultadoHTML

Puede abrir los resultados con Firefox:

$ firefox resultadoHTML.html

De la misma forma que con FASTQC, analizar los resultados, haciendo incapié en la distribución del largo de lecturas, la calidad y el comienzo de las mismas

Ensamblado de un BAC

Datos

Se cuenta con una corrida de Illumina de un BAC. Se trata de lecturas pareadas con inserto de 400pb, de 100pb de largo. El largo total del BAC es aproximadamente 155 kb.

El nombre del BAC, es bBT336E11, y los archivos provistos son:

bBT326E11.1.fastq

bBT326E11.2.fastq

Cada uno conteniendo las lecturas pareadas para cada inserto.

Actividades

En un terminal, localizarse en la carpeta: /media/DATOS/workshop, para eso:

$ cd /media/DATOS/workshop

En esa carpeta, existe una carpeta denominada BAC, entrar en esa carpeta:

$ cd BAC

En la carpeta se encuentran los archivos anteriormente nombrados, puede visualizarlos con: ls

$ ls

Si en su lugar hace:

$ ls –l

Le mostrará además información de los archivos, entre lo que puede visualizar el tamaño de los mismos.

Analizar el contenido de los archivos fastq, para eso, utilizar more, less, head y tail. Por ejemplo:

$ more bBT326E11.1.fastq

Le mostrará el contenido del archivo en pantalla.

Para contar cuántas secuencias hay en el archivo, puede realizar:

$ grep -c "@HS24_07969" bBT326E11.1.fastq

Dado que el BAC tiene aproximadamente 160 kb, estime la cobertura en función del largo de las lecturas y el total de lecturas con que se cuenta

¿Cómo podría evaluar esta cobertura? ¿Sería correcto trabajar con estos datos?

Lo que sigue puede tomarse como opción, y decida si trabaja con todas las secuencias o no…

Para obtener un subgrupo de las secuencias, puede usarse un programa que pertenece al paquete de Celera Assembler (wgs-assembler) llamado fastqSample.

Corra fastqSample sin opciones para ver cuáles son las opciones disponibles

$ fastqSample

Para obtener un subgrupo que de una cobertura de 200X el tamaño del fragmento, correr:

$ fastqSample -g 160000 -l 100 -c 200 -n bBT326E11

Esto generará dos archivos: bBT326E11.200x.1.fastq y bBT326E11.200x.2.fastq

A partir de ahora, para trabajar con velvet, utilizar este subgrupo (o no… es su elección…)

Contar ahora nuevamente el número de secuencias.

Calcular nuevamente la cobertura.

Velvet

Para realizar un ensamblado con velvet, existen dos programas, velveth y velvetg. El primero de ellos ordena las lecturas, kmers, construye hash, etc; y el segundo construye y analiza el grafo de de Bruijn como vimos anteriormente.

$ velveth

Correr el programa sin opciones, muestra una gran parte de las disponibles. Es muy recomendable (en todos los casos) leer el manual para poder hacer un uso correcto del software.

En primer lugar, se hará un ensamblado en el cual no se tiene en cuenta las lecturas pareadas, esto es, como si se tratara de fragmentos. Para esto, generar un archivo que contenga todas las secuencias juntas, haciendo:

$ cat bBT326E11.200x* > bBT326E11.200x.ALL.fastq

Contar ahora nuevamente el número de secuencias.

Una vez analizados los parámetros que puede utilizar velvet, preparar las secuencias corriendo:

$ velveth fragmentos 47 -short -fastq bBT326E11.200x.ALL.fastq

En este caso se le indica que se cuenta con lecturas cortas, en formato fastq, el archivo que las contiene y que se trabajará en la carpeta fragmentos y se utilizará un tamaño de kmer 47.

Ahora, correr velvetg sin opciones y ver las variantes.

Correr velvetg indicando solo la carpeta que contiene los resultados de velveth

$ velvetg fragmentos

Analizar la carpeta fragmentos y los archivos generados, en particular, el archivo stats.txt.

Para hacer un análisis gráfico, puede correrse el script stats-analysis, el cual genera 4 gráficos.

$ stats-analysis

Este programa, es un script de R que genera histogramas (e histogramas ponderados sobre el archivo stats.txt. El contenido del script es básico y se muestra a continuación

```
#!/usr/local/bin/Rscript
library(plotrix)
data = read.table("stats.txt", header=TRUE)
png("hist_short.png")
hist(data$short1_cov, xlim=range(0,200), breaks=100000)
png("hist_long.png")
hist(data$long_cov, xlim=range(0,200), breaks=100000)
png("whist_short.png")
weighted.hist(data$short1_cov, data$lgth, breaks=seq(0,200,by=2))
png("whist_long.png")
weighted.hist(data$long_cov, data$lgth, breaks=seq(0,200,by=2))
```

Si se obtiene un error en la corrida, es porque falta el módulo “plotrix” en R. Para instalarlo en R, ejecutar lo siguiente:

$ sudo R

El password para el usuario “alumno” es “alumno”

Cuando entre en la línea de comandos de R, ejecutar:

> install.packages(“plotrix”)

> q()

Ingresar “n” para no guardar.

Luego, el script debería funcionar sin problemas.

Los gráficos generados son los archivos hist_long.png, hist_short.png, whist_long.png y whist_short.png. Para visualizarlos, haga:

$ eog hist_short.png

Analice los archivos correspondientes a lecturas cortas, dado que se trabajó con este tipo de lecturas.

Analizar el significado de los parámetros exp_cov y cov_cutoff.

En función de los resultados obtenidos, ¿qué tipo de cobertura espera y qué valores utilizaría?

Existe una alternativa, y es dejar que velvet decida cuáles son los mejores parámetros, para lo cual, puede utilizarse –exp_cov auto y cov_cutoff auto. Vuelva entonces a correr velvetg:

$ velvetg fragmentos –exp_cov auto –cov_cutoff auto

Analizar el significado de los resultados.

¿Cuál es el tamaño del contig más grande?

¿Cuántos contigs se obtuvieron?
Cuente los contigs del archivo contigs.fa utilizando grep –c ">" contigs.fa

Analice el tamaño de los contigs, haciendo grep ">" contigs.fa

¿Cuál es el N50?

Ahora, se analizarán estos datos, teniendo en cuenta que se tratan de lecturas pareadas, para lo cual, se debe especificar esto en velveth:

$ velveth pareadas 47 -shortPaired -fastq -separate bBT326E11.200x.1.fastq bBT326E11.200x.2.fastq

Se indica en este caso que son lecturas pareadas, en archivos separados, para que se tenga en cuenta esta información.

Vuelva a correr velvetg, ahora sobre la carpeta pareadas:

$ velvetg pareadas

Y utilice luego los parámetros –exp_cov auto y cov_cutoff auto

$ velvetg pareadas –exp_cov auto –cov_cutoff auto

Analizar el significado de los resultados.

¿Cuál es el tamaño del contig más grande?

¿Cuántos contigs se obtuvieron?
Cuente los contigs del archivo contigs.fa utilizando grep –c ">" contigs.fa

Analice el tamaño de los contigs, haciendo grep ">" contigs.fa

¿Cuál es el N50?

¿El archivo contigs.fa contiene contigs o scaffolds?

Para realizar una visualización de los resultados, deben volver a correrse velvetg, pero con el siguiente comando:

$ velvetg pareadas –exp_cov auto –cov_cutoff auto –read_trkg yes –amos_file yes

Las opciones agregadas hacen que, en primer lugar, se conserve la ubicación de las lecturas en el ensamblado (-read_trkg yes) y, segundo, se genere un archivo AFG que permitirá la visualización del ensamblado (-amos_file yes).

Para visualizar el ensamblado, puede utilizarse el paquete AMOS, que cuenta con la aplicación Hawkeye para visualizar contigs y scaffolds. Existen otras opciones, como el programa tablet.

Para correr hawkeye, hacer:

$ hawkeye&

Cargar en el mismo el archivo velvet_asm.afg para tener una visualización del los scaffolds y contigs.

Debido a que velvet renombra las lecturas, se pierde la información de lecturas pareadas, por lo que para incorporar esta información, una de las opciones es mapear todas las lecturas contra el scaffold formado, es decir las secuencias del archivo contigs.fa.

Para realizar esto, correr:

$ bowtie-build contigs.fa indice

$ bowtie -S -q -I 150 -X 500 indice -1 ../bBT326E11.200x.1.fastq -2 ../bBT326E11.200x.2.fastq mapeo_scaffold.sam

$ samtools view -bT contigs.fa mapeo_scaffold.sam > mapeo_scaffold.bam

$ samtools sort mapeo_scaffold.bam mapeo_scaffold_ordenado

$ samtools index mapeo_scaffold_ordenado.bam

Estos pasos utilizan Bowtie, un programa de mapeo de lecturas cortas a una referencia. Los pasos generan en un primer momento un archivo SAM, el cual es transformado en BAM, el cual se ordena e indexa.

Así se genera un archivo en formato BAM ordenado e indexado, el cual puede ser visualizado de diferentes maneras.

Una de ellas es utilizar Gap5 del paquete STADEN. Para ello, se debe convertir utilizando:

$ tg_index –b mapeo_scaffold_ordenado.bam

Se generan así las bases de datos de Gap5, las cuales se pueden visualizar mediante:

$ gap5 mapeo_scaffold_ordenado.0&

Visualizar mediante “Template Display”, y en el botón Template, seleccionar Stacking y Reads

Busque las regiones en las cuales se encuentran las Ns

¿Existe evidencia que afirme que esos contigs con contiguos?

¿Cuál sería una forma para cerrar esos gaps?

Ensamblado de un genoma bacteriano

Datos

La carpeta Bceti, contiene una corrida de Roche 454 con lecturas pareadas del genoma de *Brucella ceti*. El archivo es Bceti.sff (archivo de salida de 454) y además un archivo que se generó a partir de este último llamado Bceti.fastq.

Por los tiempos de corrida, y por las capacidades computacionales, ya se realizó el ensamblado de las secuencias con Celera Assembler (wgs-assembler) y Newbler. Los resultados se encuentran en las carpetas "celera" y "newbler" respectivamente.

Se realizará el ensamblado de las lecturas correspondientes a las corridas utilizando los archivos FASTQ utilizando velvet.

¿Cuántas secuencias contiene?

Si se supone un genoma de 3.3 Mb, ¿cuál sería la cobertura con la que se cuenta?

Para realizar el ensamblado con velvet, revisar los parámetros utilizados anteriormente. Para preparar las secuencias, correr:

$ velveth sinfiltro 47 –long –fastq Bceti.fastq

$ velvetg sinfiltro

Determine exp_cov utilizando stats-analysis

Vuelva a correr con un –exp_cov de 12:

$ velvetg sinfiltro –exp_cov 12 –cov_cutoff auto –read_trkg yes –amos_file yes

Determine nodos, contigs, N50 y contig más largo para luego comparar

Analice el archivo Bceti.fastq con FASTQC para ver la calidad de las secuencias. Verá que es necesario realizar algún corte por largo y calidad…

Un ejemplo de filtro y corte a utilizar es el que se da aquí:

$ prinseq-lite.pl -fastq Bceti.fastq -out_format 3 -min_len 50 -max_len 300 -trim_qual_right 25

En este caso, prinseq-lite filtrará aquellas secuencias de largo menor a 50 pb y mayor a 300, y cortará en el 3' las lecturas cuando la calidad caiga por debajo del valor phred 20 y guardará el

resultado como un archivo FASTQ. Una vez corrido, el programa mostrará estadísticas sobre los cortes.

Analizar la salida del programa (archivo Bceti_prinseq_good_XXXX.fastq con FASTQC para comparar con el anterior

Realizar nuevamente el ensamblado con velvet, pero en este caso, utilizar el archivo de lecturas cortadas:

$ velveth filtradas 47 –long –fastq Bceti_prinseq_good_XXXX.fastq

$ velvetg filtradas –exp_cov 12 –cov_cutoff auto –read_trkg yes –amos_file yes

Comparar los resultados obtenidos con los obtenidos anteriormente

En este caso, no se tuvieron en cuenta las lecturas pareadas, por lo que el scaffolding no se realizó. Se analizarán ahora ensamblados en los que estos datos fueron tenidos en cuenta utilizando otros programas.

Ensamblado con Celera Assembler:

Abrir mediante hawkeye el ensamblado de Bceti, entrando en la carpeta “celera” y haciendo:

$ cd celera

$ hawkeye Bceti.bnk&

En este caso sí se cuenta con información de lecturas pareadas, por lo cual se pueden visualizar la relación entre los fragmentos.

Determinar número de scaffolds, contigs, N50, para luego comparar con los demás ensambladores.

Ensamblado con Newbler

En la carpeta “newbler”, analizar el archivo 454NewblerMetrics.txt

$ less 454NewblerMetrics.txt

Determinar número de scaffolds, contigs, N50, para luego comparar con los demás ensambladores.

Se puede visualizar el archivo ACE con hawkeye, pero el mismo pierde la información de scaffolds y lecturas pareadas.

$ hawkeye&

Importar el archivo 454Contigs.ace

¿Cuál de los ensamblados le parece mejor?

RNA-Seq

Datos

Se cuenta con lecturas correspondientes a un experimento de RNA-Seq de *Saccharomyces cerevisiae*, en el cual se trabajó en dos medios de cultivo diferente. Las lecturas se encuentran en la carpeta RNASeq, y se denominan

MEDIO1.fastq

MEDIO2.fastq

¿Cuántas secuencias hay y de qué tamaño son las lecturas?

Analice con FastQC la calidad de la corrida

En realidad, se trata de un set recortado a 10.000.000 de lecturas… ya que originalmente contiene aproximadamente 25.000.000…

NOTA: Si la corrida de los comandos demora mucho tiempo, o si no se cuenta con mucho tiempo, la carpeta RESULTADOS, contiene el resultado de todos ellos.

Mapeo de lecturas

Las lecturas se mapearán al genoma de *Saccharomyces cerevisiae*, el cual está en formato fasta en el archivo Saccharomyces_cerevisiae.fasta. Además, se cuenta con la anotación del genoma, es decir los genes y sus posiciones en el archivo Saccharomyces_cerevisiae.gtf.

Para mapear las lecturas, se utiliza tophat, que a su vez utiliza Bowtie, para lo cual se debe construir un índice de Bowtie.

Construcción de índice de Bowtie (en este caso debe utilizarse la versión 2 de bowtie!)

$ bowtie2-build Saccharomyces_cerevisiae.fasta SCindex

El índice se denomina SCindex y es el que utilizará tophat.

Se realizarán dos tipos de mapeo. En primer lugar se realizará un mapeo que tiene en cuenta la anotación del genoma de Saccharomyces, y luego un mapeo que no lo tenga en cuenta. Esto se hará con cada una de las condiciones del experimento.

Para el mapeo de las lecturas en el MEDIO1:

$ tophat -p 6 -o medio1_guiado -G Saccharomyces_cerevisiae.gtf --no-novel-juncs SCindex MEDIO1.fastq

Las opciones indican que use 6 procesadores, la salida será guardada en la carpeta medio1_guiado, usará la anotación de Saccharomyces_cerevisiae.gtf, no propondrá nuevas uniones, usará el índice creado SCindex y las lecturas corresponden al archivo MEDIO1.fastq

Lo mismo se realiza con las lecturas correspondientes al MEDIO2:

$ tophat -p 6 -o medio2_guiado -G Saccharomyces_cerevisiae.gtf --no-novel-juncs SCindex MEDIO2.fastq

Para correr sin la guía de la anotación, al igual que antes:

$ tophat –p 6 –o medio1_singuia SCindex MEDIO1.fastq

$ tophat –p 6 –o medio2_singuia SCindex MEDIO2.fastq

Para ordenar los resultados (y facilitar el uso), se renombran los archivos:

$ mv medio1_guiado/accepted_hits.bam medio1_guiado/medio1_guiado.bam

$ mv medio2_guiado/accepted_hits.bam medio2_guiado/medio2_guiado.bam

$ mv medio1_singuia/accepted_hits.bam medio1_singuia/medio1_singuia.bam

$ mv medio2_singuia/accepted_hits.bam medio2_singuia/medio2_singuia.bam

Para poder visualizar los resultados, se indexan los archivos BAM:

$ samtools index medio1_guiado/medio1_guiado.bam

$ samtools index medio2_guiado/medio2_guiado.bam

$ samtools index medio1_singuia/medio1_singuia.bam

$ samtools index medio2_singuia/medio2_singuia.bam

Visualización

Abrir IGV mediante:

$ igv&

Se importará el genoma y su anotación para visualizar los genes y sus localizaciones. Para esto, en el menú File—Import Genome… Completar el formulario, teniendo en cuenta que lo importante es indicar el "FASTA file" que contiene el genoma (está en la carpeta RNASeq y se denomina Saccharomyces_cerevisiae.fasta) y el archivo de anotación en donde indica "Gene File" denominado Saccharomyces_cerevisiae.gtf

Así cargará la información del genoma en el Browser.

Luego, se pueden cargar los resultados de los alineamientos (deben estar indexados) mediante: File—Load from File…

Cargar así los resultados del MEDIO1 y MEDIO2 (guiados o sin guiar).

Puede además cargar cualquiera de los archivos .bed (junctions, deletions, insertions) de las carpetas creadas

Visualice las lecturas y navegue por diferentes regiones del genoma.

Ensamblado de transcriptos - Niveles de expresión

Se utilizará Cufflinks para el ensamblado de isoformas y la cuantificación del nivel de expresión de los transcriptos. Así pueden identificarse nuevos transcriptos. Se trabajará con los resultados que no tienen en cuenta la guía, ya que la idea es ensamblar los transcriptos…

Para ello, a partir de los resultados de tophat (es decir, el mapeo de las lecturas contra el genoma de referencia):

$ cufflinks –p 6 –o cufflinks_medio1 medio1_singuia/medio1_singuia.bam

$ cufflinks –p 6 –o cufflinks_medio2 medio2_singuia/medio2_singuia.bam

Se generarán las carpetas cufflinks_medio1 y cufflinks_medio2.

Ingrese a una de las carpetas y observe los archivos generados

Abra el archivo genes.fpkm_tracking con alguna planilla de cálculo para simplificar su visualización (por ejemplo oocalc, puede ser una opción en Linux)

Abra el archivo transcripts.gtf, el mismo contiene los transcritos ensamblados, en un formato que puede visualizarse en IGV

Una vez ensamblados los transcriptos, ya que en este caso todas las condiciones contribuyen a la identificación de los mismos, los resultados se mezclan para obtener un solo archivo de anotación.

Para ello, crear un archivo de texto que contenga los resultados anteriores (transcripts.gtf). Generar el archivo: lista_cufflinks.txt que contenga las líneas:

cufflinks_medio1/transcripts.gtf

cufflinks_medio2/transcripts.gtf

Para mezclar ahora las anotaciones:

$ cuffmerge –s Saccharomyces_cerevisiae.fasta lista_cufflinks.txt

Se creará un archivo dentro de la carpeta merged_asm/merged.gtf

El mismo puede visualizarse en IGV, y compararse visualmente con la anotación de referencia

OPCIONAL…

Una de las opciones es, además, comparar la anotación obtenida con una anotación de referencia (ya que en este caso se cuenta con una). Para ello se usa cuffcompare de la siguiente manera:

$ cuffcompare –s Saccharomyces_cerevisiae.fasta –r Saccharomyces_cerevisiae.gtf merged_asm/merged.gtf

Es decir, se brinda la anotación conocida y se la compara con la generada en el paso anterior.

Ver los archivos resultantes

Expresión diferencial

Para determinar la expresión diferencial entre dos condiciones, se utiliza Cuffdiff (del paquete Cufflinks). Cuffdiff puede utilizarse con los archivos generados por tophat, o los generados por cufflinks. Es decir, el paso de ensamblado de transcriptos puede realizarse o no, dependiendo si lo que se quiere es el descubrimiento o no de isoformas y/o nuevos genes, o simplemente tomar la anotación de la referencia.

Para cuffdiff los parámetros incluyen un archivo de anotación y los resultados del mapeo en formato BAM. En el caso que se cuente con réplicas, deben separarse con comas, y diferentes condiciones con un espacio (OJO!). Como en este ejemplo no hay réplicas, esto no se tiene en cuenta.

En este caso, si se utilizan la anotación generada en el paso anterior (ensamblado de transcriptos), se correría como sigue:

$ cuffdiff –p 6 –o cuffdiff_singuia merged_asm/merged.gtf medio1_singuia/medio1_singuia.bam medio2_singuia/medio2_singuia.bam

En el caso que se corra con respecto a la anotación de referencia:

$ cuffdiff –p 6 –o cuffdiff guiado Saccharomyces_cerevisiae.gtf medio1_guiado/medio1_guiado.bam medio2_guiado/medio2_guiado.bam

Ingrese a la carpeta cuffdiff_guiado observe los archivos generados

Abra el archivo gene_exp.diff con alguna planilla de cálculo para simplificar su visualización