Datos de RNA-Seq

Guillermo Ayala Gallego

2024-04-02

Table of contents

Introducción

Flujo de trabajo

1. Diseño experimental.
2. Protocolos de extracción del RNA.
3. Preparación de las librerías. Se convierte el RNA en cDNA y se añaden los adaptadores para la secuenciación.
4. Se secuencian las lecturas cDNA utilizando una plataforma de secuenciación.
5. **Alineamiento de las lecturas secuenciadas a un genoma de referencia.**
6. **Resumen del número de lecturas alineadas a una región.**
7. **Normalización de las muestras para eliminar diferencias técnicas en la preparación.**
8. **Estudio estadístico de la expresión diferencial incluyendo en lo posible un modelo.**
9. Interpretación de los resultados desde el punto de vista biológico.

Objetivos

1. ¿Dónde podemos obtener datos de secuenciación? De otro modo: ¿qué repositorios podemos utilizar?
2. ¿Cómo podemos realizar búsquedas en los metadatos con objeto de obtener experimentos que tenga que ver con lo que me interesa?
3. ¿Cómo bajarlos?
4. ¿Cómo contar las lecturas alineadas sobre regiones genómicas (genes, exones o ...)?

Repositorios

1. NCBI SRA
2. EBI ENA
3. DDBJ

Formato FASTA

- El formato **fasta** está basado en texto.
- Se utiliza representar secuencias bien de nucleótidos bien de aminoácidos.
- Tanto unos como otros son representados por una sola letra.
- También tiene símbolos para representar un hueco (gap) o parada en la traducción o bien que no se sabe el nucleótido o aminoácido.

- Tiene una línea que comienza con el símbolo > al que sigue una descripción de la secuencia.
- En la siguiente línea empieza la secuencia de bases o aminoácidos. Se recomienda que no tener más de 80 columnas y se pueden tener todas las filas que se precisen.
- Un ejemplo

```
>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken     MADQLTEEQIAEFKEAFSLFDKI
PEFLTMMARKMKDTDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGEKLTDEEVDEMIREADI*
```

Formato FASTQ

- El formato **fastq** es el más popular para datos de secuencias.
- Consiste de cuatro líneas por lectura.
- La primera que comienza con el carácter `@` y contiene el **nombre de la secuencia** con alguna descripción opcional de la misma.
- La segunda línea contiene la secuencia con las letras que correspondan dependiendo del tipo (nucleótidos, aminoácidos).
- La tercera línea que comienza con + contiene información opcional sobre la secuencia.
- La cuarta línea cuantifica la confianza o calidad en la determinación de cada base recogida en la segunda línea (**Phred**).

```
@SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
ACAGGGACGCCATCGAATCCGGATCNTNNNNNNNNNNNNANNNNNNNNNN
+SRR1293399.1 ILLUMINA-545855_0026_FC629BG:6:1:1022:5049 length=50
dee\edYcdc`bbY`S]bb_]]Ua^BBBBBBBBBBBBBBBBBBBBBBBBB
```

Phred

- ¿Cómo se cuantifica la confianza o precisión?

1. Phred asigna los picos de fluorescencia a una de las cuatro bases: **base call**.
2. P: probabilidad para una base dada de ser mal asignada o clasificada
3. Se muestra:
$$Q = -10 \log_{10} P.$$
4. Una probabilidad P muy pequeña de clasificación incorrecta se traduce en un valor grande de Q.
5. Se muestra el caracter ASCII que ocupa la posición $33 + Q$.

Phred

- Simulamos las probabilidades de clasificación incorrecta y generamos al azar las bases.

```
x = sample(Biostrings::DNA_ALPHABET[1:4],10,replace=TRUE)
y = runif(10,min=0,max=.001)
names(y) = x
y
```

```
           C            C            C            G            T            T
6.932435e-04 3.477133e-04 9.631243e-04 2.477561e-05 2.620529e-05 9.133127e-04
           G            C            A            G
6.893515e-04 7.800213e-04 4.801834e-05 4.233442e-04
```

- Calculamos los `Q` valores.

```
(Q = round(-10 * log10(y)))
```

```
 C  C  C  G  T  T  G  C  A  G
32 35 30 46 46 30 32 31 43 34
```

- Codificación Sanger

```
intToUtf8(Q+33)
```

```
[1] "AD?OO?A@LC"
```

Alineamiento de lecturas cortas sobre genoma de referencia

- Wikipedia

- Tomamos una lectura o secuencia corta.
- Pretendemos encontrar en una secuencia larga donde es similar, en que punto hay una mayor similitud respecto de la secuencia larga o de referencia.

```
Read:      GACTGGGCGATCTCGACTTCG
           |||||  |||||||||| |||
Reference: GACTG--CGATCTCGACATCG
```

-

- Alienamiento global o local.
- Utilizamos : Bowtie2 o STAR.

Nuestro flujo de trabajo

¿Qué necesitamos?

- Mucha paciencia.
- Asegurarse de que tenemos (muchísima) memoria disponible en nuestro disco.
- SRA-Toolkit si vamos a bajar el experimento de la base de datos NCBI-SRA.

Experimento

- Elegimos un experimento en NCBI SRA.
- Hemos seleccionado PRJNA218851.

Obtención de los identificadores, datos y metadatos

1. En esta página tenemos el menú `SRA Run Selector`
2. En nuestro caso corresponde con este enlace
3. Aquí tenemos información sobre el experimento.
4. Con la pestaña `Select` podemos seleccionar información de toda la muestra o bien de una parte.
5. En `Accession List` generamos un fichero texto `SRR_Acc_List.txt` en donde aparecen los nombres de los `Run`. Lo guardamos, por ejemplo, con dicho nombre.
6. En la misma página elegimos en la pestaña `Metadata` el fichero `SraRunTable.txt`. Contiene las variables fenotípicas o covariables para cada una de las muestras. Lo bajamos y guardamos con el mismo nombre.
7. Bajamos los datos con `prefetch` de `SRA-Toolkit`.

```
prefetch --option-file SRR_Acc_List.txt
```

8. Los ficheros con las lecturas los guarda por defecto en `~/ncbi/public/sra/`.
9. En mi caso los he movido (nada de copiar) al directorio `PRJNA218851/sra`.

De SRA a FASTQ

1. Utilizamos la función `fastq-dump` de `SRA-Toolkit`.

```
fastq-dump -I --split-files nombre_fichero.sra
```

2. Y lo repetimos para cada fichero SRA.
3. Una opción que nos lo hace para todos es

```
cat files | parallel -j 7 fastq-dump --split-files {}.sra
```

donde `files` es el fichero `SRR_Acc_List.txt`, es decir, los nombres de los ficheros sin la extensión.

4. Puede ser útil `tami::StarSamtools_sh()`.

Control de calidad

- Más adelante

Alineamiento con STAR

1. **Generación de fichero de índices**
   En el siguiente código hemos de sustituir `genomeDir0` por el directorio donde queremos guardar el fichero de índices y `GRCh38.p13.genome.fa` por el nombre del fichero `fastq` donde tenemos el genoma de referencia.

```
STAR --runMode genomeGenerate --runThreadN 10 --genomeDir genomeDir0 --genomeFastaFiles GR
```

2. **Alineamiento de las secuencias**

   1. El siguiente ejemplo supone lecturas apareadas en los ficheros `f_1.fastq` y `f_2.fastq` y `name_file` es el nombre del fichero de salida.

```
STAR --genomeDir genomeDir0 --runThreadN 14 starIndex --readFilesIn f_1.fastq f_2.fastq --
```

Alineamiento con Bowtie2

1. **Fichero de índices**

   1. Bajamos el fichero de índices.
   2. Aquí los tenemos para distintas especies.
   3. Descomprimimos el fichero de índices y sacamos los ficheros.

4. Supongamos que lo hemos colocado en el directorio `indexDir0` y queremos alinear el fichero `name_file.fastq`.

```
bowtie2 -x indexDir0 -U name_file.fastq -S name_file_1.sam
```

SAM a BAM con samtools

1. Ejecutamos para cada fichero `name_file`.

```
samtools view -S -b name_file.sam > name_file.bam
```

2. Ordenamos las lecturas: sustituimos `name_file` por cada nombre original.

```
    samtools sort name_file.bam -o name_file_sort.bam
```

3. Puede ser útil `tami::StarSamtools_sh()`.

Conteo

- Podemos usar el siguiente código de R adaptado para nuestros datos.

```
pacman::p_load(Rsamtools,GenomicFeatures,GenomicAlignments,org.Hs.eg.db)
gtfFile = "GRCh38.p13.genome/gencode.v37.annotation.gff3"
txdb = makeTxDbFromGFF(gtfFile, format="gff3")
genes = exonsBy(txdb, by="gene")
indir = getwd()
files = list.files(indir, pattern = '*sort.bam')
bamLst = BamFileList(files, index=character(),obeyQname=TRUE)
PRJNA411984 = summarizeOverlaps(features = genes,
                                read=bamLst,
                                mode="Union",
                                singleEnd=FALSE,
                                ignore.strand=TRUE,
                                fragments=FALSE)
save(PRJNA411984,file="PRJNA411984.rda")
```

Ejemplos

- `tamidata::PRJNA266927`
- `tamidata::PRJNA297664`
- `tamidata::PRJNA297798`

`SummarizedExperiment`

- Cargamos el paquete.

```
library(SummarizedExperiment)
```

- Leemos unos datos de `tamidata`

```
data(PRJNA297664,package = "tamidata")
```

- ¿Qué clase es?

```
class(PRJNA297664)
```

```
[1] "RangedSummarizedExperiment"
attr(,"package")
[1] "SummarizedExperiment"
```

- ¿Cuántos genes y muestras tenemos?

```
dim(PRJNA297664)
```

```
[1] 7126    6
```

- La matriz de conteos es

```
head(assay(PRJNA297664))
```

- ¿Qué devuelve `assay`?

```
class(assay(PRJNA297664))
```

```
[1] "matrix" "array"
```

```
head(which(apply(assay(PRJNA297664),1,sum) > 10))
```

```
  ICR1   LSR1   NME1 RDN5-1 RDN5-6 RNA170
     4      5      6     42     47     50
```

- Variables fenotípicas

```
colData(PRJNA297664)
```

```
DataFrame with 6 rows and 4 columns
   SampleName         Run      treatment replication
  <character> <character>       <factor>   <numeric>
1  GSM1900735  SRR2549634 Wild                     1
2  GSM1900737  SRR2549636 Wild                     3
3  GSM1900739  SRR2549638 SEC66 deletion           2
4  GSM1900736  SRR2549635 Wild                     2
5  GSM1900738  SRR2549637 SEC66 deletion           1
6  GSM1900740  SRR2549639 SEC66 deletion           3
```

- ¿Y es?

```
class(colData(PRJNA297664))
```

```
[1] "DFrame"
attr(,"package")
[1] "S4Vectors"
```

- Nombres de las variables fenotípicas.

```
names(colData(PRJNA297664))
```

```
[1] "SampleName"  "Run"         "treatment"   "replication"
```

- Y accedemos a los valores de la variable `treatment` con

```
colData(PRJNA297664)[,"treatment"]
```

```
[1] Wild           Wild           SEC66 deletion Wild           SEC66 deletion
[6] SEC66 deletion
Levels: Wild SEC66 deletion
```

- Datos relativos a las filas o características o genes.

```
head(rownames(PRJNA297664))
```

```
[1] "15S_rRNA" "21S_rRNA" "HRA1"     "ICR1"     "LSR1"     "NME1"
```

- ¿Hay más? Pues sí.

```
head(rowData(PRJNA297664))
```

```
DataFrame with 6 rows and 4 columns
                 ORF         SGD    ENTREZID     ENSEMBL
         <character> <character> <character> <character>
15S_rRNA    15S_rRNA          NA          NA          NA
21S_rRNA    21S_rRNA          NA          NA          NA
HRA1            HRA1  S000119380     9164866          NA
ICR1            ICR1  S000132612     9164906          NA
LSR1            LSR1  S000006478     9164871          NA
NME1            NME1  S000007436     9164967          NA
```

- ¿Algo más? De hecho, mucho más.

```
rowRanges(PRJNA297664)[345]
```

```
GRangesList object of length 1:
$YBR031W
GRanges object with 1 range and 2 metadata columns:
      seqnames        ranges strand |   exon_id   exon_name
         <Rle>     <IRanges>  <Rle> | <integer> <character>
  [1]       II 300166-301254      + |       224   YBR031W.1
  -------
  seqinfo: 17 sequences (1 circular) from an unspecified genome; no seqlengths
```

Normalización

Efectos a corregir

- Profundidad de secuenciación o tamaño de la librería
- Composición de RNA
- Contenido GC
- Longitud del gen

Método TMM: Media ajustada de M-valores

- Y_{ij} (y_{ij}) el conteo aleatorio (observado) del gen i en la muestra j.
- Denotamos L_i la longitud del gen i y m_j es el total de lecturas de la librería j (o tamaño de la librería j).
- En (Robinson and Oshlack 2010) se asume que la media de Y_{ij} verifica

$$E\Big[Y_{ij}\Big] = \frac{\mu_{ij} L_i}{c_j} m_j,$$

  siendo $c_j = \sum_{i=1}^{N} \mu_{ij} L_i$. Notemos que c_j nos está representando el total de RNA en la muestra.
- La producción total de RNA, c_j, no es conocida.
- Podemos estimar el cociente de estos valores para dos muestras

$$f_j = \frac{c_j}{c_{j'}}$$

- Elegimos una muestra como muestra de referencia. Por ejemplo, la muestra r denota a partir de ahora la muestra tomada (arbitrariamente) como de referencia.
- Nos fijamos en la muestra j y vamos a determinar la constante por la que multiplicaremos los conteos originales.
- En lo que sigue tanto j como r son fijos y las cantidades definidas dependen de i que denota el gen.
- Se define

$$M_{ij}^{(r)} = \log_2 \frac{y_{ij}/m_j}{y_{ir}/m_r} = \log_2(y_{ij}/m_j) - \log_2(y_{ir}/m_r),$$

  y

$$A_{ij}^{(r)} = \frac{1}{2}\Big(\log_2(y_{ij}/m_j) + \log_2(y_{ir}/m_r) \Big)$$

- Se eliminan los valores extremos tanto de los $M_{ij}^{(r)}$ como de los $A_{ij}^{(r)}$.
- En concreto eliminamos un porcentaje de los M_i más pequeños y el mismo porcentaje de los más grandes.
- Lo mismo hacemos para los valores A_i.
- También eliminamos aquellos índices i tales que \$y_{ij} =0 \$ o bien $y_{ir} = 0$.
- El conjunto de índices i restante lo denotamos por G^*.
- Finalmente, el factor de normalización sería

$$\log_2(TMM_j^{(r)}) = \frac{\sum_{i \in G^*} w_{ij}^{(r)} M_{ij}^{(r)}}{\sum_{i \in G^*} w_{ij}^{(r)}}$$

con

$$w_{ij}^{(r)} = \frac{m_j - y_{ij}}{m_j y_{ij}} + \frac{m_r - y_{ir}}{m_r y_{ir}}.$$

- La muestra de referencia r es fija y lo que acabamos de calcular es el factor por el que multiplicamos los conteos originales de la muestra j. Este factor viene dado por $TMM_j^{(r)}$.
- Obviamente tenemos que $TMM_r^{(r)} = 1$ y esta muestra de referencia no se normaliza.

Mediana de los cocientes

- Se propuso en Anders and Huber (2010).
- Se consideran los factores de normalización

$$s_j = mediana_{\{i:\tilde{m}_i \neq 0\}} \frac{y_{ij}}{\tilde{m}_i}$$

siendo

$$\tilde{m}_i = \left(\prod_{j=1}^{n} y_{ij} \right)^{\frac{1}{n}}.$$

- La idea es normalizar para cada gen diviendo el conteo por la correspondiente media geométrica de los distintos conteos para un mismo gen.
- Una vez que tenemos los datos normalizados intra gen buscamos una constante de normalización por muestra como la correspondiente mediana en la muestra.
- Estas constantes son utilizadas no para normalizar directamente el conteo sino para incorporarlo como factor que multiplica a la media del conteo en los métodos DESeq y DESeq2.

Bibliografía

Anders, Simon, and Wolfgang Huber. 2010. "Differential Expression Analysis for Sequence Count Data." *Genome Biology* 11 (10): R106. https://doi.org/10.1186/gb-2010-11-10-r106.

Robinson, M. D., and A. Oshlack. 2010. "A Scaling Normalization Method for Differential Expression Analysis of RNA-Seq Data." *Genome Biol* 11 (3): R25. https://doi.org/10.1186/gb-2010-11-3-r25.