Análisis 16s rRNA utilizando DADA2

Tu nombre

2023-11-06

Contents

Chapter 1

Introducción a DADA2

Repositorio de GitHub aquí

Cita:

> Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., y Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. Nature methods, 13(7), 581-583.

Descripción: DADA2 es un paquete de software de código abierto utilizado para modelar y corregir errores en secuencias de amplicones secuenciados con el protocolo Illumina. Infiere secuencias de una muestra de manera exacta y resuelve diferencias con una resolución de un nucleótido.

Los datos utilizados pueden ser consultados en el European Nucleotide Archive (ENA) bajo el nombre de proyecto PRJNA428495.

Descargamos los archivos FASTQ en la sección "Generated FASTQ files: FTP" y los guardamos en una carpeta.

Los binarios del paquete DADA2 están disponibles a través de Bioconductor

1.1 Instalación de DADA2 en R (versión 4.3)

Este bloque instala el paquete DADA2 a través de Bioconductor. Es importante tener instalado BiocManager para poder acceder a los paquetes de Bioconductor.

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("dada2")
library(dada2)
```

1.2 Configuramos el directorio en el que se ubican las reads

```
path <- "/Users/" # Cambiamos al directorio que contiene las reads
list.files(path)
```

1.3 Ordenamos los archivos y extraemos el nombre de las muestras

Este bloque ordena los archivos FASTQ y extrae el nombre de las muestras. Asegúrate de que los patrones de búsqueda (pattern="_1.fastq.gz" y pattern="_2.fastq.gz") coincidan con la nomenclatura de tus archivos.

```
fnFs <- sort(list.files(path, pattern="_1.fastq.gz"))
fnRs <- sort(list.files(path, pattern="_2.fastq.gz"))
sample.names <- sapply(strsplit(fnFs, "_"), '[', 1)
fnFs <- file.path(path, fnFs)
fnRs <- file.path(path, fnRs)
```

1.4 Establecer ruta para archivos filtrados

Define una ruta basada en la variable `sample.names` donde guardar los archivos filtrados. El código genera nombres de archivo con un sufijo ("_F_filt.fastq.gz" o "_R_filt.fastq.gz") para diferenciarlos.

```
filt_path <- file.path(path, "filtered")
filtFs <- file.path(filt_path, paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(filt_path, paste0(sample.names, "_R_filt.fastq.gz"))
print(fnFs)
print(fnRs)
print(filtFs)
print(filtRs)
```

1.5 Filtrar y recortar lecturas

La función filterAndTrim se utiliza para el filtrado y recorte de las lecturas, es decir, se eliminan las lecturas de baja calidad y las recorta a una longitud específica. Se especifican varios parámetros, como las longitudes de truncamiento, la calidad máxima permitida, la eliminación de secuencias de fagos (`rm.phix`), la compresión y el uso de múltiples hilos (`multithread`). El resultado se almacena en la variable `out`. Windows 10 permite el 'multi-theading', excepto en el comando `filterAndTrim`.

```
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(250, 200), maxN=0, maxEE=c(2,2), trun
```

1.6 Estimación de la tasa de error

Se estima la tasa de error para las lecturas adelante (errF) y las lecturas reversas (errR) utilizando la función `learnErrors`. La opción `multithread=TRUE` se utiliza para acelerar el proceso.

```
errF <- learnErrors(filtFs, multithread=TRUE)
errR <- learnErrors(filtRs, multithread=TRUE)
```

1.7 Dereplicar lecturas

Se realiza la eliminación de duplicados de lecturas (dereplicación) para las lecturas adelante (derepFs) y las lecturas reversas (derepRs) utilizando la función `derepFastq`. Se asignan nombres a las muestras con `sample.names`.

```
derepFs <- derepFastq(filtFs, verbose=TRUE)
derepRs <- derepFastq(filtRs, verbose=TRUE)
names(derepFs) <- sample.names
names(derepRs) <- sample.names
```

1.8 Inferencia de secuencias

Se utilizan las tasas de error estimadas en el paso anterior para la inferencia de secuencias únicas utilizando la función `dada` para las lecturas adelante (dadaFs) y las lecturas reversas (dadaRs).

```
dadaRs <- dada(derepRs, err=errR, multithread=TRUE)
dadaFs <- dada(derepFs, err=errF, multithread=TRUE)
```

1.9 Fusionar lecturas emparejadas

Aquí se fusiona las secuencias emparejadas utilizando la función `mergePairs`. Las secuencias únicas y las lecturas originales se utilizan para la fusión, y los resultados se almacenan en la variable mergers.

```
mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs, verbose=TRUE)
```

1.10 Crear tabla de secuencias

Se crea una tabla de secuencias utilizando la función `makeSequenceTable`. Esta tabla contendrá información sobre las secuencias y su abundancia.

```
seqtab <- makeSequenceTable(mergers)
```

1.11 Eliminar quimeras

Las quimeras son secuencias que resultan de la combinación de dos o más secuencias parentales diferentes durante el proceso de amplificación por PCR. Estas formaciones ocurren en ciclos posteriores de PCR cuando hay una alta concentración de cebadores parcialmente extendidos que compiten con los cebadores originales.

Se realiza la eliminación de quimeras utilizando la función `removeBimeraDenovo`. Se especifica el método de eliminación como “consensus”.

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbo
```

Las quimeras puede introducir artefactos¿¿‘¿’¿ en los análisis de datos, dando lugar a interpretaciones erróneas.

En el análisis de datos usando DADA2, las quimeras se eliminan utilizando el código removeBimeraDenovo. Esta función es una interfaz de conveniencia para la eliminación de quimeras. DADA2 ofrece varios métodos para identificar quimeras, como la identificación a partir de secuencias agrupadas y la identificación por consenso entre muestras. Al utilizar el método de consenso, por

ejemplo, las muestras en una tabla de secuencias se verifican independientemente en busca de quimeras, y se toma una decisión de consenso sobre cada variante de secuencia.

DADA2 es muy meticuloso al tratar con estas quimeras para asegurar que los datos analizados sean precisos. Por ejemplo, al utilizar removeBimeraDenovo con el método "pooled", todas las muestras en la tabla de secuencias se agrupan para la identificación de quimeras. Si se usa el método "consensus", las muestras en una tabla de secuencias se verifican independientemente en busca de quimeras, y se toma una decisión de consenso sobre cada variante de secuencia. Esto es vital ya que las quimeras pueden variar entre las muestras y es esencial asegurarse de que no afecten los resultados.

En conclusión, eliminar quimeras o bimeras es esencial para obtener una representación precisa de las comunidades microbianas en los análisis de datos. Si no se eliminan, podrían llevar a interpretaciones erróneas de la estructura y diversidad de la comunidad. DADA2 proporciona herramientas efectivas para identificar y eliminar estas quimeras, asegurando así la calidad y precisión de los resultados obtenidos.

1.12 Asignar taxonomía

Se asigna la taxonomía a las secuencias utilizando la función `assignTaxonomy`. Se proporciona un archivo de referencia para la asignación taxonómica, y se utiliza el `multithread` para acelerar el proceso. La base de datos taxonómica utilizada será un archivo fasta de entrenamiento derivado de la versión 138.1 del Proyecto Silva y formateado para su uso con DADA2. Este archivo puede ser descargado de Zenodo.

```
taxa <- assignTaxonomy(seqtab.nochim, "C:/Users/UBBBC/Desktop/Samuel/PRJNA428495/silva_nr99_v138.
```

1.13 Impresión de resultados

Finalmente, se imprime en la consola las primeras filas de la asignación taxonómica resultante.

```
unname(head(taxa))
```

Chapter 2

Cross-references

Cross-references make it easier for your readers to find and link to elements in your book.

2.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
    - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world` = `# Hello world {#hello-world}`.
    - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter 2.
    - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

2.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 2.1.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

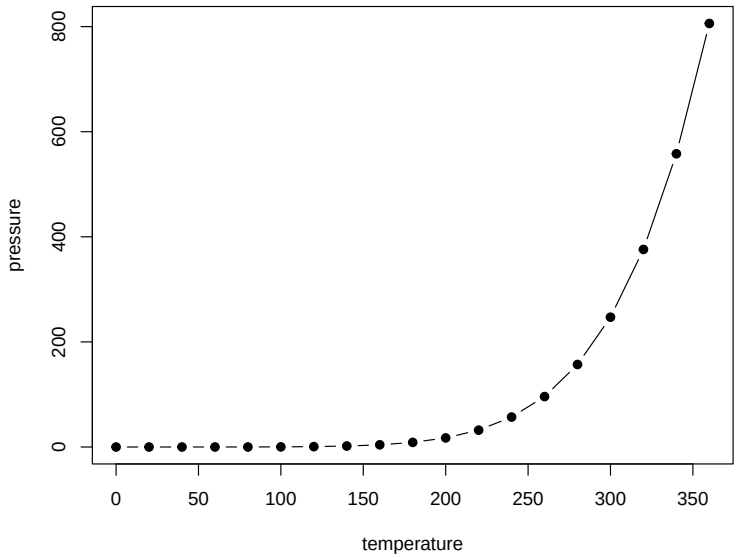

Figure 2.1: Here is a nice figure!

Don't miss Table 2.1.

```
knitr::kable(
  head(pressure, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

Table 2.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Chapter 3

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 4

Footnotes and citations

4.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

4.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [@R-bookdown] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [@xie2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 5

Blocks

5.1 Equations

Here is an equation.

$$f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 5.1.

Theorem 5.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

5.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 6

Sharing your book

6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```