RNA Sequencing Guide

Milena Wünsch and Pat Callahan

2023-07-02

Contents

Chapter 1

About

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports; for example, a math equation $a^2 + b^2 = c^2$.

1.1 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

1.2 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select "All formats" if you'd like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

1.3 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in "Preview book", or from the R console:

```
bookdown::serve_book()
```

Part I

Common Processing Steps

Chapter 2

Prefiltering

2.1 Libraries

2.1.1 Install Libraries

All necessary packages are available on Bioconductor, and should be installed from there if not already available on your machine. The code below will install {BiocManager} from CRAN, and you can then use this package to install PADOG, tweeDEseqCountDatam, and KEGGREST to your system.

```
install.packages("BiocManager")
BiocManager::install("PADOG")
BiocManager::install("tweeDEseqCountData")
BiocManager::install("KEGGREST")
```

2.1.2 Load Libraries

Note that loading these libraries will mask many functions from base R packages. If you run into unexpected errors on functions you're using, it is recommended to use namespacing to explicitly clarify the package from which you need a given function. (I have suppressed the library() loading messages from this document, however.)

```
library(PADOG)
library(tweeDEseqCountData)
library(KEGGREST)
```

Provide a brief note on what these libraries are for?

2.1.3 PADOG

The PADOG library does XYZ. You can find more information about it online at LINK.

2.1.4 tweeDEseqCountData

2.1.5 KEGGREST

2.2 Load Data

This section will change substantially when I reconfigure the project as an R package/book.

For now, place any data you need into the `./data` directory.

```
# we load the voom-transformed Pickrell data set
load("data/expression_data_voomtransformed_Entrez.Rdata")

# alternatively: load the gene expression measurements that have been transformed usin
load("data/expression_data_vsttransformed_Entrez.Rdata")

# additionally, we load the pickrell data set so that we can access the sample conditi
data(pickrell)
```

The sample conditions (i.e. phenotype labels) of the pickrell data set can be accessed using

```
pickrell.eset$gender
#>  [1] male   male   female male   female male   female male
#>  [9] female male   female male   female male   female male
#> [17] female female male   female male   female female male
#> [25] female male   female female male   female female male
#> [33] female male   female female female female male   female
#> [41] male   male   female female male   female female male
#> [49] female female male   female male   male   female female
#> [57] male   female male   female male   female female male
#> [65] female female female male   female
#> Levels: female male
```

We proceed with the voom-transformed pickrell data set and the corresponding phenotype labels

```
# gene expression measurements (transformed)
# note: you can also proceed with the vst-transformed gene expression measurements
expression_data_transformed <- expression_data_voomtransformed_Entrez
# sample conditions
sample_conditions <- pickrell.eset$gender
```

2.3 Prepare Sample Conditions

First, we inspect the form of the initial (raw) sample conditions

```
## look at the class:
class(sample_conditions)
#> [1] "factor"
# -> the sample labels are already coded as factor

# the current levels are:
levels(sample_conditions)
#> [1] "female" "male"
```

PADOG requires character vector with class labels of the samples. It can only contain “c” for control samples or “d” for disease samples

```
# prepare sample conditions
# we want to convert
# (i) "female" to "c"
# (ii) "male" to "d"
sample_conditions_prep <- factor(sample_conditions,
                                 levels=c("female","male"),
                                 labels=c("c","d"))
```

2.4 Run PADOG

It is recommended to set a seed to ensure exact reproducibility of the results if the code is run at multiple time points

you can specify any integer number as the seed. It is VERY IMPORTANT to choose the seed arbitrarily and WITHOUT INSPECTING the results the seed should NEVER be specified based on which value yields the most preferable results.

```
# run PADOG:
 PADOG_results <- padog(esetm = as.matrix(expression_data_transformed),
                        group = sample_conditions_prep,
                        dseed = 1)
```

arguments:

- `esetm`: matrix that contains the expression measurements
  - note: since the expression data is initially stored in a data frame, we transform it to a matrix when running PADOG
- `group`: sample conditions (has values "c" and "d")
- `dseed`: seed for random number generation (used in the process of phenotype permutation)

additional arguments:

- `paired`: indicates whether the samples in both groups are paired
- `block`: if the samples are paired (i.e. argument paired = TRUE), then the paired samples must have the same block value
- `gslist`: gives instructions on how to cluster the genes into gene sets
  - gslist = "KEGGRESTpathway": gene sets correspond to KEGG pathways
  - alternative: provide a user-defined list of gene sets
- `organism`: organism from which the gene expression measurements are taken
  - for human, set organism = "hsa"
  - the required character value for other organisms can be extracted from the KEGGREST package:
- `obtain` required organisms from column organism
- `annotation`: required if gslist is set to "KEGGRESTpathway" and the rownames of esetm are probe IDs
- can be set to NULL of gslist is set to "KEGGRESTpathway" and the rownames of esetm are in the Entrez gene ID format
- if rownames are other gene IDs, then sett annotation = NULL and make sure that the rownames are elements of gslist (and unique!)
- `gs.names`: contains names of gene sets -> character vector
  - must have the same length as gslist
- `NI`: number of phenotype permutations employed in the assessment of the significance of a given gene set

2.5 Adjust for Multiple Testing

2.6 Interpretation of Results

Chapter 3

GSEA Preranking

Skip to Chapter 4 Differentiaon Expression Analysis if conversion of gene IDs is not needed.

Chapter 4

Gene ID Duplicate Removals

Chapter 5

Differential Expression Analysis

Chapter 6

RNA-Seq Transformation

Branching point: continue to the next chapter if conversion of gene IDs is not necessary. However, if conversion was/is necessary, proceed to Chapter ?? Cluster Profiler.

Part II

No Gene ID Conversion

Chapter 7

Go SEQ

Chapter 8

Cluster Profiler: GSEA GO

Chapter 9

DAVID

Chapter 10

GSEA WEB

Part III

With Gene ID Conversion

Chapter 11

Cluster Profiler: GSEA KEGG

Chapter 12

Cluster Profiler: ORA

Chapter 13

PADOG