Babraham Bioinformatics

Exercises:
Advanced R

Version 2018-10

Licence

Exercise 1: Using R Notebooks

- Create a new R notebook for the course and save it as a .Rmd file into the Advanced R Data folder which contains the data you're going to use.
- In the Advanced R Data folder, you should have 2 tab delimited text files
  - `expression.txt` contains RPKM expression values
  - `methylation.txt` contains %methylation for gene bodies and promoters
- Load both of these datasets into separate data frames using `read.delim` and display them (just put the variable name as a stand-alone statement)
- Use the `hist` function to draw a histogram of the `Promoter_meth` column of the methylation dataset.

Exercise 2: Filtering and Deduplicating

- In the methylation dataset remove any rows where the promoter methylation level is -1. Break this operation down into sections:
  - Extract the column containing the promoter methylation
  - Apply a test for a value equal to (remember to use **2** equals signs) -1
  - Invert the logical vector using a logical NOT operation (an `!`)
  - Use this logical vector to select which rows to keep when performing a selection on the methylation dataset. You can either save the results to a new variable name or simply overwrite the existing methylation variable.
  - Re-draw the histogram of promoter methylation values to check that the values below zero have actually gone.
- The methylation data is already deduplicated, but you should deduplicate the expression data. Again, think of this in steps
  - Extract the `Probe` column from `expression`
  - Run `duplicated` on this vector and use `sum` to find out how many duplicates there are
  - Use a logical NOT (an `!`) on the results of `duplicated` to get a logical vector which says which genes are NOT duplicated, and use this to select which rows to keep when making a selection on expression
  - Again, you can either save the results under a new name, or you can overwrite the existing `expression` variable.
- Use the `sum` function with the `%in%` operator to find out how many Probes in `expression` are also in `methylation`.

Exercise 3: Text Manipulation

- Use `grep` with the `value=TRUE` option to find the probe names in the expression data set come from olfactory receptors (they contain "Olfr" in their names).
- In the expression data make a vector of chromosome names which have "chr" in front of them (so chrX instead of just X). To do this:
  - Extract the `Chromosome` column from `expression`
  - Use the `paste` function to join the fixed string `"chr"` to the vector of chromosome names. Set the `sep` parameter to an empty string to join them directly together.

Exercise 4: Extending and Merging

- Make a new column in `methylation` called `difference` which is the `Gene_body_meth` values minus the `Promoter_meth` values.
- Use `hist` to plot a histogram of the difference values
- Merge the expression and methylation data together. You want to merge based on the Probe column. As this column is already named the same between the two data frames you can just use `merge` with no additional options to merge the datasets together into a new combined data frame. Save the results into a new variable called `merged`. Display this new variable in your notebook so you can check that everything worked.

Exercise 5: Looping

- Use an `apply` statement to calculate the `range` of values in the expression gene body methylation and promoter methylation columns in your merged dataset (probably columns 6,7 and 8). Do a selection on `merged` to get out just the columns you want, and then use this filter result to construct the `apply` command. The second argument to apply will need to be `2` since you want to loop over columns (`1` would loop over rows), and the function to call is `range`.
- Use `tapply` to calculate the `mean` gene body methylation for genes on each chromosome. The data will be the gene body methylation column, and the groups will be the chromosomes. The function you need will simply be `mean`.
    - [Optional] Plot these results as a barplot.

Exercise 6: Functions

- There is no direct function in R to calculate the standard error of the mean (SEM). This value is simply the standard deviation divided by the square root of the number of observations. Write a function called `sem` to perform this calculation.
- Add error checking to your function so that it will emit a suitable warning message and return an `NA` value if passed in a vector which contains an `NA` value.
- Rerun the `tapply` calculation from the end of exercise 5 to calculate the SEM of the gene body methylation levels.

Exercise 7: Packages

- Locate the package you'd use for constructing bean plots (like an improved version of a boxplot). Install and load this package. Read the documentation to find out how it works.
- Construct bean plots for the two Methylation columns in your merged dataset. Extract the two columns and pass them to `beanplot`. You can tidy it up by passing `what=c(0,1,0,0)` as an extra option to the `beanplot` function.

Exercise 8: Notebooks

- Save the `.Rmd` Notebook file you've been writing throughout the course and compile it into an HTML document. Look at it in a web browser.