Homework Week 8

Friederike Dündar and Luce Skrabanek

ANGSD Course 2020

Inspecting BAM files (**1pt**)

1. Download the Integrative Genomics Viewer from the Broad Institute. Download 2 BAM files from the Gierlinski data set to your computer and load them into the IGV Browser. Find a gene with at least one intron and take a Sashimi plot that you include in your html file.

Understanding read counts (**9pts**)

1. Describe 2 properties of RNA-seq read count data that researchers have identified – i.e. what are some of the reasons why we cannot take the read counts at face value? Please also describe the adjustments that need to be made to use them as more reliable proxies for gene expression evaluations. (**4pts**)
2. What is the main difference between exploratory analyses and the tests for differential gene expression in terms of the types of questions that can be addressed with either one? (**1pt**)
3. Exploratory analyses of `DESeq.ds` objects: generate a **dendrogram** and a **PCA plot** of our 10 samples and briefly explain the major insights you can derive from them. You may use the code detailed in https://physiology.med.cornell.edu/faculty/skrabanek/lab/angsd/lecture_notes/08_practical_exploratory.pdf, but make sure to explain what every (!) function does and what type of output each function generates. Can you infer what the `pcaExplorer` plots of the "Top Loadings" represent? (**4pts**)

Understanding `DESeq2` S4 objects. (**5.5pts total**)

As we briefly discussed during class, `DESeq2` (and many other bioconductor packages) often use very specific and elaborate R objects to combine observations (e.g. counts, normalized expression values, etc.) with corresponding meta data (e.g., gene IDs, sample IDs, sample conditions, gene ranges etc.). These objects are usually based off the `S4` convention and if you are interested in the nitty-gritty details, we can recommend the bioconductor team's write-up. If you do not know what type of object you are dealing with, `class(object)` is the function

that releases that information. If the result isn't one of the base R classes you learnt about during Luce's class (e.g. `data.frame`, `matrix`, `list`), it is highly likely an S4 object. To get more background information, e.g. about how to interact or generate a certain object, you can always resort to the generic help call that you should be familiar with for functions, such as `?list` or `?data.frame`. To explore an S4 object more, you can use the function `showClass`, which expects the result of `class(object)` as an input (e.g. `class(object) %>% showClass`). You can think of S4 objects as sophisticated lists insofar as they allow the combination of different types of objects. But where the content of lists can be accessed via the same accessors that are used for `data.frames` (e.g. `list$stored_object1` or `list[["stored_object1"]]`, S4 objects use different accessors called `slots`, whose content you can retrieve using the `@` symbol or via specific accessor function provided by the parent package, e.g. `DESeqObject@counts` or `counts(DESeqObject)`. Which slots are present in a given S4 object can be determined via `slotNames(object)`.

Use the functions we've described to **inspect the objects that are returned by `rlog()` and `DESeqDataSetFromMatrix`**, respectively. You can also try to just type their name in the console and hit return and see what happens.

- Describe at least 2 similarities and 2 differences between them in regards to their content and the downstream functions you may have to use to interact with them. Feel free to use any additional documentation available, just let us know how you found each answer. (*4pts*)
- How can you extract the expression values stored in either object? (*1pt*)
- How can you add an additional matrix named "my_personal_normalization" to either object? (You do not need to make up new values for this; just use the same expression values already stored in either object and assign it using a new name.) (*0.5pt*)

Inspecting the source code of R functions (**1.5pts**)

Read the instructions here. Note that the `getMethods()` described there has been deprecated and has been replaced by `findMethods()`, which does not require you to specify the object type for which you're retrieving the code.

Include the source code of the following `DESeq2` functions in your homework's html:

- `rlog()`
- `estimateDispersions()`
- `rlogData()` – this is a non-exported function from `DESeq2`

Understanding DE analysis (**4pts** (plus 2 pts for extra credit))

- In your own words, describe how `DESeq2` calculates a p-value that you can use to decide whether a given gene shows different expression values when comparing two conditions of interest. Start from what types of values are used as the measurements. You may consult the document on our website that contains a verbose summary of what we talked about in class. (**3pts**)
- Despite the fact that we are testing each gene individually, which calculations/values are influenced by the values of the other genes in the same matrix? Explain at least one. (**1pt**)
- *Bonus question* (not required): From the source code of `nbinomWaldTest()`, identify the relevant line where the Wald statistic is calculated. Explain what the objects contain that are used for the calculation. (**2pts** for extra credit)

Compile the `.Rmd` file and send both the `.Rmd` and the `HTML` files to **angsd_wmc@zohomail.com** by Saturday night. If you need support, get in touch with Merv on Thursday, 3-4pm.
(**21pts**)