Seminar III: R/Bioconductor

Leonardo Collado Torres
lcollado@lcg.unam.mx
Bachelor in Genomic Sciences
www.lcg.unam.mx/~lcollado/

August - December, 2009

Bioconductor and Documentation

Bioconductor

Reproducible Research

Exercises/Homework

Intro

- It's the largest repository of genomic related packages for `R` available at `http://bioconductor.org`.
- BioC was founded in 2001 and here you can find the core developers. Just like `R`, it follows a 6 month release cycle.
- I highly *recommend* you to visit the basic introduction here[1].
- It's open source and open development initiative! *You* can contribute to BioC!

[1] Scroll down to the *What is Bioconductor?* section

Getting started

- In R, the basic function to install a packages is without much surprise install.packages()
- For Bioconductor, use the biocLite script. You might find this guide useful :)

  ```
  > source("http://bioconductor.org/biocLite.R")
  > biocLite()
  ```

- Using biocLite without any arguments downloads a basic set of packages for your appropiate R version and plataform.

Browising for packages

- If you are looking for a package that might help you with your work, I recommend these two options:
    1. While very new, the `biocViews` taxonomy browser is very promising and easy to browse: software 2.5 biocViews and biocViews categories
    2. Currently, the most complete option is to simply browse the download section. For example, software for the current dev version (BioC 2.5).
- A package can *depend*, *import* and *suggest* other packages.
    1. Depend: end user can see the functions
    2. Import: the package uses but does not let the end user see
    3. Suggest: useful for some expanded workflows
- On which packages does chipseq depend on?
- What is the 5th most downloaded Bioconductor package?

Viewing a package

- As for any package you've installed, you can view a basic description, the list of functions and methods with the following syntax:

  ```
  > help(package = pkgname)
  ```

- Who is the maintainer of the `Biostrings` package?
- Her or his email?
- How is it licensed?

Package documentation

- A BIG difference between Bioconductor packages and regular CRAN packages is that Bioconductor packages are documented with a *vignette* file and a reference manual.
- A vignette is a document that contains both text (explanations) and `R` code that exemplify how to use the functions from a given package.
- The reference manual lists all the functions/methods with some examples but can be harder to understand.

Finding vignettes

- While the pdf files are normally built on your machine, you can also download them by browsing the download section.
  - For example look here for the chipseq vignette[2].
- Inside R, you can also find the list of available vignettes by typing:

  ```
  > vignette(package = "pkgname")
  ```

- Note: if you are using the dev version (such as us), checking the `Bioconductor Changelog` for a package can be informative!
- What kind of bug did they fix on August 4th?

[2] More exactly, a workflow.

Expert help

- If you have explored every way to find help, there is a way to get expert help!
- Have you really, really, yes . . . really explored all the options? Obviously including a google search. Reading the posting guide is a must!
- Then, simply send your question to the Bioconductor Mailing List. There are three flavors:
  1. General bioconductor list
  2. BioC-devel list
  3. High throughput sequencing list

Registering to the list

- At least during this semester, I will require all of you to register to the BioC mailing list.
- As you could see on the syllabus, from next class on forth, I will ask some of you to present interesting topics from the discussions of that week.
- So, go to this URL: `https://stat.ethz.ch/mailman/listinfo/bioconductor`
- Enter your information and I highly recommend you to choose "Yes" for the option: `Would you like to receive list mail batched in a daily digest?`

Extra

- Feel free to register to the other two mailing lists:
- `https://stat.ethz.ch/mailman/listinfo/bioc-devel`
- `https://stat.ethz.ch/mailman/listinfo/bioc-sig-sequencing`
- You may decide to *filter* the emails into a specific folder in your mail :)

Workshops

- In accordance with the open source nature of Bioconductor, you can find presentations, talks, labs and much more on the Workshops page.
- `http://bioconductor.org/workshops/`
- If you browse to 2008 and 2009, you'll notice some familiar courses :)
- For the curious ones, the BioC workshops such as BioC2008 and BioC2009 have very interesting labs. A lab is a practical session.

Workflows

- Although partially contained on the workshops section, Bioconductor has a set of freely available workflows.
- `http://bioconductor.org/docs/workflows/`
- For example, there are workflows for `Affymetrix SNP arrays`, `Illumina Expression Microarrays`, etc.

Books

- Finally, but not least important, there is a section for Bioconductor related publications:
- `http://bioconductor.org/pub/`
- We already ordered some of those books and you can also find the reference on the supporting material for this course.
- Note that we DO have access to some of these books on pdf format through our `Springer` trial subscription.
- I encourage you to read the following New York Times articles on Bioconductor.

The core

- Biobase is the main package for Bioconductor, specially if you are working with microarrays.
- It defines the *ExpressionSet* class which was constructed to organize large amounts of biological data.
  1. `experimentData` to describe the experiment
  2. metadata such as `annotation`, information on the chip technology in `featureData` and info on the samples in `phenoData`
  3. tips on how to access the data values[3] as `assayData`

[3] As its meant for microarrays, the data values are normally expression data.

More

- Biobase has other handy functions, such as biocReposList in case that you want to use the `install.packages` function. The reference manual is rather long!

  ```
  > library(Biobase)
  > biocReposList()
  ```

```
                                                      bioc
          "http://bioconductor.org/packages/2.5/bioc"
                                                     aData
"http://bioconductor.org/packages/2.5/data/annotation"
                                                     eData
"http://bioconductor.org/packages/2.5/data/experiment"
                                                     extra
         "http://bioconductor.org/packages/2.5/extra"
```

More

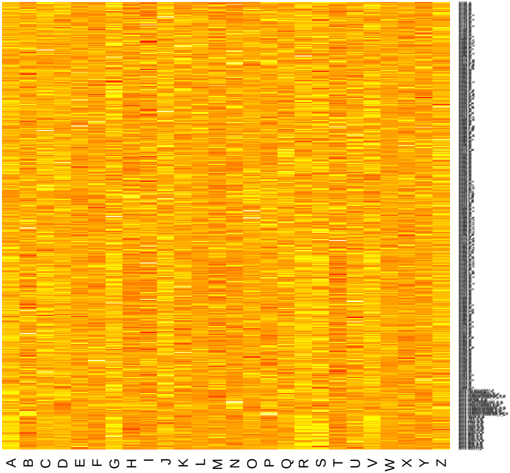
```
                               brainarray
"http://brainarray.mbni.med.umich.edu/bioc"
                                     cran
                     "http://cran.fhcrc.org"
```

Heatmap

- Lets view a more complicated version of the image function. Biobase has a data set called geneData. What are the dimensions?

```
> data(geneData)
> heatmap(geneData, Rowv = NA, Colv = NA,
+      cexRow = 0.2)
```

Heatmap



Like `image`?

- What does heatmap do to our data before plotting it?

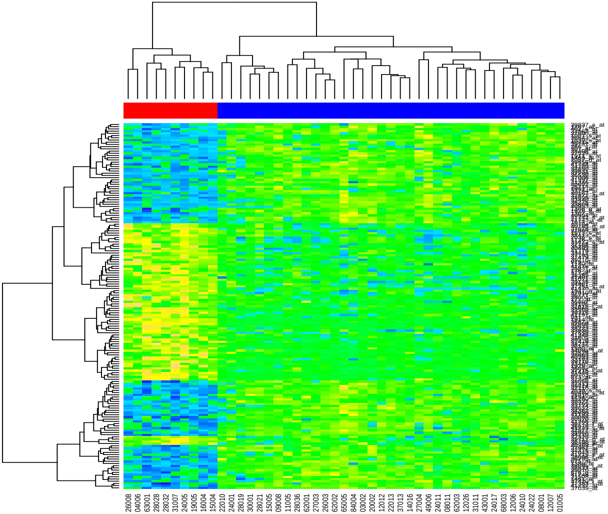
  ```
  > `?`(heatmap)
  ```

- Play around with the previous plot:
  1. Delete the `Colv` argument
  2. Delete the `Rowv` argument while keeping `Colv`
  3. Delete both and only keep `cexRow`
- Are all the heatmaps equal? If not, what changes?

Quick heatmap explanation

- We won't get into the details, but heatmap with the default parameters re-orders the rows and the columns and creates groups (clusters) determined by euclidean distance.
- At some point in the course you'll be able to do heatmaps just like the following one.

A full heatmap



What is it?

- The goal is simple: to enable others to reproduce your results.
- But, isn't research supposed to be reproducible in order to be published? What about supp. material?
- *Discussion:* Is it to use the exact same scripts/programs with the same parameters? Or is it to follow the same workflow even if you re-write the scripts?

Discussion cont.

- If you don't get the same results using the same scripts/programs and parameters, then something is seriously wrong! Or you are not using the same input files; could be a version issue.
- Whom do you *trust*? The one who did the original work or the one who re-wrote the scripts/programs to match the same workflow?

Discussion cont.

- Everyone and anyone makes simple mistakes: typos, starting from 0 instead of 1, positive as negative and vice versa, etc.
- You can *inherit* problems! Simple enough, you are using data from a previous work and the data has some errors.

Forensic bioinformatics

- No, it's not to figure out who was the murderer in a crime scene.
- Its deciphering someone's code when its messy and the code doesn't match the written description of the algorithm/workflow.
- If you aren't careful, you might end up doing forensic bioinformatics with your own code!!! I do recommend using a version control system such as subversion for your scripts.[4]

[4] RapidSVN is a simple GUI if you want to avoid the command line

Some extreme cases

- Keith Baggerly gave an excellent talk on the subject at BioC2009. Find it through the workshops site.
- 1 to 0 mistakes, adding 1 to names, inverting positive and negative responses, wrong association between names and data, manual input of the biological relevant genes, and overall a big mess!!
- Magazines didn't seem to care much as no *fe de erratas* was published. Keeping themselves "clean" on public eyes.
- Funding agencies see these events *frequently* and they do care more. However, on Baggerly's case study, the scientists are proceeding to experiment with humans. . .

So. . .

- So, in R, how do you do reproducible research?
- An excellent practice would be to develop a experimental data package and submit it every time you publish your work.
- You might just use the package to share the data with your lab members or collegues.
- Vignettes! (Sweave is behind)

TEH solution

- Developed by Friedrich Leisch[5], Sweave is an R function that evaluates R code chunks and parses the output into LaTeXformat.
- LaTeXfiles look like a mix between a script and a plain text file. You can turn LaTeXfiles into PDF files, just like this presentation and the vignette files!
- The workflow is basically:
  1. Create a `.Rnw` file in LaTeXformat with some R code specified as such.
  2. Transform your `.Rnw` file into a `.tex` file using `Sweave`.
  3. Create the final `.pdf` file from the `.tex` file.

[5] He is a BioC core dev.

Commands in Unix

- `R CMD Sweave file.Rnw`
- `R CMD Stangle file.Rnw`[6]
- `pdflatex file.tex`
- `pdftalex file.tex`[7]
- If you wish, you can then remove some of the files using `rm`. To avoid typing, it's very useful to create a general shell script :)

[6] Stangle extracts the R code pieces and creates a `.R` file with the `R` code

[7] Yes, two times. You need to do so for structures such as the outline.

In Windows

- You will need to install Miktex. The first time you use pdflatex, Miktex will download some LaTeXpackages.
- The commands themselves change such as `R.exe -e "Sweave('file.Rnw')"` and `pdflatex.exe file.tex`[8]
- `www.johndcook.com/troubleshooting_sweave.html` is very useful for Windows users.

[8] You might need to modify your PATH environment variable to include the R and R/lib folders

User guides

- We won't go deep in class time into LaTeXnor Beamer[9], but I have cited some very good pdf manuals on the supporting material of this course.
- The Not so Short guide to LaTeXis very complete :) Check it out for tips on typesetting text and mathematical formulae as well as for a LaTeXintroduction.
- There is a second PDF specialized on symbols. . . and there are LOTS.
- Finally, the Beamer User Guide has all you need to know about Beamer and has a funny tutorial.

[9] It's used to make presentations such as this one

Exploring a Rnw file

- Now, I got started by comparing the `Rnw` files with the pdf files from James Bullard course. And if I had a question, I would check the pdf guides.
- To understand more about Sweave, lets check a Rnw file. Open `www.lcg.unam.mx/~lcollado/B/quizes/01_answer/`
- You'll notice the `Sweave.sty` file, which you normally need on every sweave working directory.

Top of the Rnw file

- Open the Rnw file. The % symbol is used to comment lines in LaTeX, so which is the first un-commented line?
- Next we load some LaTeXpackages, define some commands, set the page style and bibliography style.
- What do you think the `SweaveOpts` line does?

R code chunks

- To avoid spamming our folder, we save the images on the plots folders with the name starting by `fig`.
- I do not recommend having multiple Rnw files on the same working directory. I sometimes use 2 but I need to be careful and specify different figure surnames.
- Then we have our first R command:

```
> options(width = 40)
```

- As you can see, an R code chunk starts with a line `<<eval=TRUE, echo=TRUE>>=`. Then you can put any R code, and you end the chunk with the symbol `@`.

The rest of the doc

- Next, On this Rnw file you'll find information on the title, the author, the start of the document, how to make the title, some line escapes, notes and the abstract.
- A file can be divided into sections and subsections.
- Check out the special syntax to include `R` figures.
- Remember that for every `begin` there must be an `end` or it'll crash.
- The rest should be self explanatory including when the document ends.

Workspace

- Be careful with your workspace when using Sweave.
- If you have saved a workspace on your current working directory, when you use Sweave it'll be loaded automatically.
- You can always add this code line to avoid inheriting workspace issues:

  ```
  > rm(list = ls())
  ```

A Sweave complement

- On Bioconductor you can find the weaver package.
- It was designed to help you when your document is large and/or you have time consuming computations that you don't want to repeat every time you change a detail on your Rnw file.
- Quite helpful for writing a thesis or some other long project.
- Install it with `biocLite` and check out the vignettes; specially the *howto*.

Weaver R chunks

- To use weaver, you'll need to load it at the beginning.

  ```
  > library(weaver)
  ```

- Then, your R code chunks will start with:
- `<<eval=TRUE, echo=TRUE, cache=TRUE>>=`

Part 1: template

Create your own template Sweave document.

- Title: course name, homework number
- Author: name, email, include a link to your personal academic webpage if you have one.[10]
- Abstract: short description on the homework and any notes you might want to add
- A sample homework solution: meaning a short description and some code. For example, how to sum $2 + 3$.

[10] You will probably make one this semester on the PHP course.

Part II: *ALL* dataset

- You'll have to explore the ALL dataset[11] and create your first homework as a vignette document.
- Install the ALL package and explore the `ALL` object.

```
> library(ALL)
> data(ALL)
```

- Select the samples from the B-cell tumors.
- Select those of molecular type `BCR/ABL` or `NEG`.
- Combine the previous two subsets and keep the *intersect*ion
- Eliminate unused factor levels on your resulting subset.
- Use the nsFilter function from the `genefilter` package to keep those with *entrez* ID, *GOBP*, remove duplicate *entrez* and the following arguments:

Part II: *ALL* dataset

```
> nsFilter(var.fun = IQR, var.cutoff = 0.5,
+     feature.exclude = "^AFFX")
```

- Meaning that we'll use the interquantile range with a variance cutoff of 0.5 to eliminate those with small variation and by excluding `AFFX` we'll take out the controls AFFY probes.
- How many:
  1. duplicates were removed?
  2. control features were excluded?
  3. had low variance (small variation)?
  4. had no GO?
  5. had no entrez ID?

[11] John Quackenbush mentioned it on Monday as the most studied dataset.

Session Info

```
> sessionInfo()

R version 2.10.0 Under development (unstable) (2009-07-25 r48998)
i686-pc-linux-gnu

locale:
 [1] LC_CTYPE=en_US.UTF-8
 [2] LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=C
 [6] LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8
 [8] LC_NAME=C
 [9] LC_ADDRESS=C
[10] LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8
[12] LC_IDENTIFICATION=C

attached base packages:
```

Session Info

```
[1] stats     graphics  grDevices
[4] utils     datasets  methods
[7] base

other attached packages:
[1] Biobase_2.5.5

loaded via a namespace (and not attached):
[1] tools_2.10.0
```