BioC for HTS - PDCB topic
Bioconductor 01

LCG Leonardo Collado Torres
lcollado@wintergenomics.com – lcollado@ibt.unam.mx

September 8th, 2010

Quick review

- Main site: `http://www.bioconductor.org/`
- Finding packages: BiocViews and/or Workflows
- Installing BioC:

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

- Installing a specific BioC package:

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

- Browsing your local Vignettes:

```
> browseVignettes(package = "PkgName")
```

- So, how many vignettes do you have locally?

Vignettes

- Just use the browseVignettes function without any arguments:

  ```
  > browseVignettes()
  ```

- The result is a html page with links to all the PDFs and R files.
- The whole idea behind a vignette is to exemplify how you can combine multiple functions from the same package.

Experimental Data Pkgs

- Using the BioCViews, which experimental data packages are related to high throughput sequencing?
- Having a broader diversity of exp. data pkgs has been one of the goals for some time: **you can contribute!**

Motivation

Why learn BioC from a Dev's point of view?

- You will know where to find every piece of help :)
- Using personal pkgs means that you can share your work easily with colaborators
- Emphasis on reproducibility
- Useful documentation framework: help files and vignettes

BioC Dev page

Main site:
`http://www.bioconductor.org/developers/index.html`

- Devolper Wiki
- Dev Mailing List
- Subversion repository access
- Daily build system reports

Subversion repository

- Subversion is a version control system.
  Basically, it's useful when several developers work on the same code. More info on its home site and a quick intro at BioC.
- Get into the BioC repository: `https://hedgehog.fhcrc.org/bioconductor/trunk/madman/Rpacks`
  Username and passwd are **readonly**
- We now have access to the latest BioC :)

Package structure: Rsamtools

Lets browse the Rsamtools structure:

- **DESCRIPTION**
  It contains most of the info that appears on a pkg's homesite or when you use `help(package = pkgName)`.
  What is Rsamtools used for?
  What is the difference between importing or depending on or suggesting a package? Any clues?
- **NAMESPACE**
  This file basically specifies which methods or classes or functions or packages Rsamtools imports and exports.
- **NEWS**
  Description of new features, bug fixes and other changes from version to version.

Package structure: Rsamtools

- **R/**
  Here we can find R script files that define functions, methods and/or classes.
- **inst/**
  This is the main documentation directory.
  - *doc/*
    Here you'll find the raw vignette files (Rnw)
  - *extdata/*
    Small example data sets including some Rda or Rdata files
  - *scripts/*
    Some example scripts
  - *unitTests/*
    R scripts that test functions from the package

Package structure: Rsamtools

- **man/**
  Home to the Rd files: the function help files :)
- **src/**
  Home to external code like C libraries (.h) and code (.c).
- **tests/**
  Some basic tests in R code.
  What does Rsamtools test?

Build reports

- To ensure that R and BioC changes don't affect users, the BioC team builds all packages every night.
- Building a package checks that all files are in the correct format, that the structure is correct, that the tests pass, and that the package can be installed in Mac, Windows and Linux.
- The daily results are available through `http://bioconductor.org/checkResults/`
- From the current devel, are there any packages that cannot be built?
- Are all packages supported on all OS?
- When was the last change to the GenomeGraphs and GenomicRanges packages?

Dev Wiki

`http://wiki.fhcrc.org/bioc/DeveloperPage/`

- Includes notes on dev meetings and discussions
- Has a section of guidelines for packages
- HowTo section is useful for any new dev :)

Dev Mailing List

`https://stat.ethz.ch/mailman/listinfo/bioc-devel`

- A must for all BioC package developers
- Low traffic and great for learning details on package dev
- Here is where you'll ask questions about pkg dev that you **cannot** find by using Google (and similars).

BioC requirements

- Good to know before hand :)
- Check `http://www.bioconductor.org/developers/package-submission/`
  What is the max pkg size allowed? Max time to build?
- A key difference between BioC and CRAN is that packages get reviewed in BioC. The goal is that submitting a pkg will be similar to submitting a paper.
- Can you include static vignettes? Why yes/no?

Pkg guidelines

`http://www.bioconductor.org/developers/package-guidelines/`

- There you can find all the specific reqs
- Example: specify 1 or more biocViews categories
- Use vectorized calculations
- Pkg versions follow the x.y.z system where:
  - x is normally 0 if the pkg hasn't been released
  - y is even of release versions, odd for devel
  - z changes every time you commit changes
- Please avoid duplicating efforts! (pkgs, classes, containers, ...)

Lets start!

- We now know alot about BioC packages, but lets start by making a simple one.
- First, lets define a function:

```
> randomSeq <- function(n, prob = rep(0.25,
+     4), alphabet = c("A", "C",
+     "T", "G")) {
+     res <- sample(alphabet, n,
+         replace = TRUE, prob = prob)
+     res <- paste(res, collapse = "")
+     return(res)
+ }
```

- What does *randomSeq* do?

```
> randomSeq(20)
```

Lets start!

```
[1] "CTCTGTGAGGAGTATTTTAA"
```

- Now lets create a second function:

```
> calcGC <- function(x) {
+     seq <- toupper(strsplit(x,
+         "")[[1]])
+     GC <- length(which(seq == "G")) +
+         length(which(seq == "C"))
+     return(GC/length(seq) * 100)
+ }
```

- Need a volunteer to explain *calcGC*
- Now lets make a small test:

```
> seq <- randomSeq(10)
> seq
```

Lets start!

```
[1] "GGGTTTCAGG"
```

```
> calcGC(seq)
```

```
[1] 60
```

- Did we get the correct GC%?

Using our functions for tons of sequences

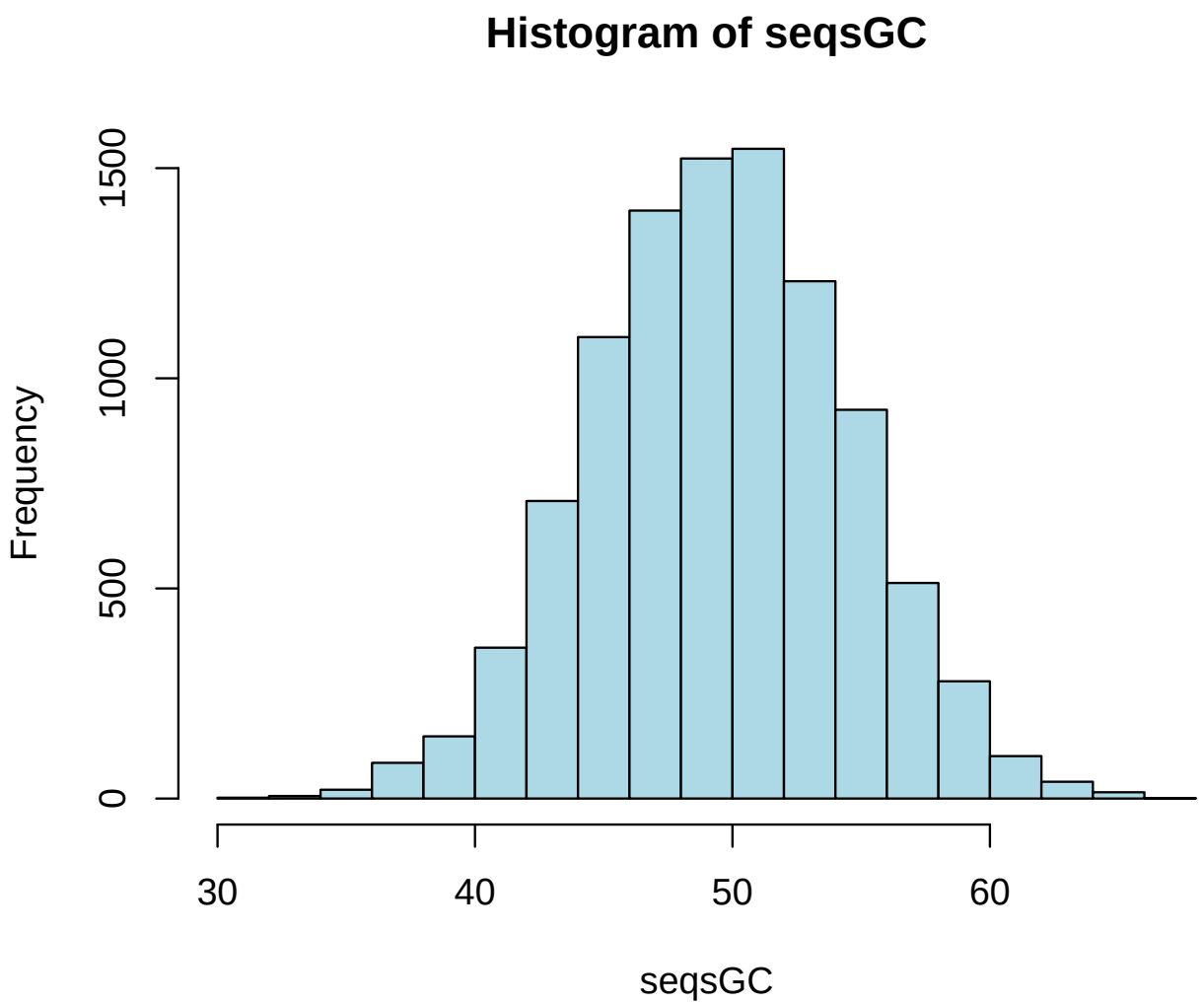
- What does the following code do?

```
> seqs <- lapply(1:10000, function(x) randomSeq(n = 100
> seqsGC <- unlist(lapply(seqs, calcGC))
```

Histogram of the GC from 10k seqs

```
> hist(seqsGC, col = "light blue")
```

Histogram of the GC from 10k seqs



package.skeleton

- We now have two functions and a set of objects

```
> ls()
```

```
[1] "calcGC"    "randomSeq" "seq"
[4] "seqs"      "seqsGC"
```

- Check the help for the function package.skeleton
- Now we can start to build our package *GCcalc*:

```
> package.skeleton(name = "GCcalc",
+     list = c("seqs", "seqsGC",
+         "randomSeq", "calcGC"),
+     path = ".", namespace = TRUE)
```

Neat? :)

- What files did it create?
- Check the *Read-and-delele-me* file
- The prompt function is useful for creating Rd files as well :)

Rd files

`http://cran.fhcrc.org/doc/manuals/R-exts.html#Rd-format`

- The **Writing R Extensions** manual is helpful for developing packages, hence it is long!
- Rd files, or the help files for methods / functions have a *funny* syntax. Basically, it's close to LaTeXbut it can include R code.
- Lets open the Rd file for our GCcalc package.
- Sections are divided by backslashes and reys. For example, what is the docType?
- The output of package.skeleton is very helpful as you just need to edit the files :)

calcGC.Rd

- Open the calcGC help file.
- What differences do you note between this file and the package help file?

Some of the differences

- Usage tag: very important as this is where you specify how to use the function
- Details tag: it is meant to describe every argument
- Value tag: explain the output :)
- See also tag: link to other Rd files
- All examples here will be used for checking the package. They should be simple

Exercise

- Modify the randomSeq and calcGC functions to include parameter checks using conditionals (if, ...), and functions such as `stop` and `warning`
- Test the functions for cases with Ns and/or Us. Do they affect the distribution of GC?
- Create again the package GCcalc and add simple descriptions to all the help files.
- Try building and checking the package (R CMD build and R CMD check)[1]. Do they work?

[1] Only available through command line

Please install Bioconductor

- Besides the basic Bioconductor, we'll need the following packages:

```
> biocLite(c("ShortRead", "Rsamtools",
+     "GenomicRanges", "IRanges",
+     "GenomeGraphs", "biomaRt",
+     "GenomicFeatures", "ChIPpeakAnno",
+     "edgeR", "DESeq", "baySeq",
+     "chipseq", "Biostrings", "SRAdb",
+     "DEGseq", "MotIV", "BayesPeak",
+     "ChIPsim"))
```

- We might use others along the way, but I'll try to notify you.

Session Information

```
> sessionInfo()

R version 2.11.1 (2010-05-31)
x86_64-pc-linux-gnu

locale:
 [1] LC_CTYPE=en_US.utf8
 [2] LC_NUMERIC=C
 [3] LC_TIME=en_US.utf8
 [4] LC_COLLATE=en_US.utf8
 [5] LC_MONETARY=C
 [6] LC_MESSAGES=en_US.utf8
 [7] LC_PAPER=en_US.utf8
 [8] LC_NAME=C
 [9] LC_ADDRESS=C
[10] LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.utf8
[12] LC_IDENTIFICATION=C

attached base packages:
```

Session Information

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