Introduction to R and Bioconductor

18 November, 2009

Contents

1 Introduction

This portion of the course is designed to provide you with basic skills for working with R and Bioconductor. In the exercises that follow, you will install R and Bioconductor packages, learn how to use the R help system, and familiarize yourself with typical R operations.

2 Installing R

To reduce download time, we have put R installers and other software on an internal server. Normally, you would download the latest version of R from a CRAN mirror site listed on `http://cran.r-project.org/mirrors.html`.

Exercise 1
If you haven't already installed R 2.10 on your computer, point your browser to `http://wilson2/course/` *and select the R installer package appropriate for your operating system. Please wait to use a wired connection.*

Exercise 2
Verify that you are running R version 2.10 by checking the output of the the `sessionInfo` *function. If you are not running version 2.10, please repeat the previous exercise to obtain the correct R version or ask a course assistant for help.*

3 Installing Packages with `biocLite`

The Bioconductor project provides a helper function called `biocLite` that we recommend you use to install Bioconductor and CRAN packages. To provide faster downloads during the course, we have setup a special version of `biocLite` that makes use of internal servers.

Exercise 3
Load the special version of `biocLite` *into a new R session using the following call to* `source`*:*

```
> source("http://wilson2/biocLite.R")
```

After the course, you can obtain the latest version of the `biocLite` function by executing the following:

```
> ## don't use this one during the course, please
> source("http://bioconductor.org/biocLite.R")
```

The `biocLite` function expects a character vector of packages that you wish to install. Internally, `biocLite` calls `install.packages` and any additional named parameters will be passed on to `install.packages`. In this way you can control the fine details of package installation.

Exercise 4
How would you call `biocLite` *if you wanted to control where the downloaded packages will be stored? Hint: read the help page for* `install.packages`*.*

Exercise 5
Install the collection of packages we will use in the rest of the course using the special version of `biocLite`*. You can find a list of packages to copy/paste here:* `http://wilson2/course/install-day1.txt`*. Please wait for a wired connection; do not use the wifi network to install packages. While the software is downloading and installing, you may be able to continue with other parts of the lab.*

4 Bioconductor Resources on the Web

You may find the following resources available from the Bioconductor website useful:

- Common workflows: http://bioconductor.org/docs/workflows
- Presentations and lab excercises from previous courses: http://bioconductor.org/workshops
- A taxonomical view of Bioconductor packages: http://bioconductor.org/packages/release/Software.html
- Details on Bioconductor's mailing lists: http://bioconductor.org/docs/mailList.html

Exercise 6
Use the GMANE search interface to find the posting that announced this course on the Bioconductor mailing list.

Exercise 7
Find the six packages in the latest Bioconductor release that are listed in the SequenceMatching task view.

Exercise 8
Use the BiocViews taxonomy to find all annotation data packages for Mus musculus.

5 Getting Help in R

Learning to use R's help system effectively will make it easier to use new Bioconductor packages and accomplish analysis tasks. Two essential help commands are `help` and `help.start`. The `help` function allows you to quickly display the manual page for a specified topic (usually the name of a function). Try these:

```
> help("help")
> ?c
```

If you are trying to remember the arguments to a function you can use the `args` function.

```
> args(update.packages)
```

R also provides an HTML-based help system that you can access locally using your web browser. To start the help system enter:

```
> help.start()
```

5.1 Searching

There is a link "Search Engine & Keywords" on the start page of `help.start` that allows you to query the help system for a topic of interest. You can also search for help using `help.search` and `RSiteSearch`, the latter will search R mailing lists in addition to documentation. Finally, `apropos` is useful for finding functions that are in the current search path.

Exercise 9
Find the function for performing a Mann-Whitney test.

Exercise 10
Find all the functions on your search path that have a name consisting of a single character.

5.2 Vignettes

Many R packages come with a *vignette*, a short document providing a detailed example of how to make use of the package's functionality. Vignettes contain executable R code that allow you to step through the examples as you read the document. You can use the `browseVignettes` function to explore the vignettes available on your system.

Exercise 11
Use `browseVignettes` to see the list of vignettes available on your system.

6 Exploring R objects

Every object in R has an associated class which you can determine using the `class` function. This is often a good way to begin exploring an unfamiliar object. Other functions useful for exploring are `length`, `dim`, `summary`, and `str`.

Exercise 12
Execute the following call and then determine what type of object is stored in `pkgs`.

```
> pkgs <- installed.packages()
```

Exercise 13
What do the `length` and `dim` functions return for `pkgs`? Can you reconcile the answers given by these two functions?

Exercise 14
Display a subset of `pkgs` consisting of the last two rows and the seventh column.

Exercise 15
List the column names of `pkgs`.

Exercise 16
See what happens when you subset `pkgs` *as if it was a simple vector.*

6.1 S4 Classes and Methods

Understanding packages that use S4 classes and generic functions is made easier by a few utility functions described below.

Consider the ShortRead package as an example:

```
> suppressMessages(library("ShortRead"))
```

The function `getClasses` can be used to list all classes defined in a package.

Exercise 17
Use `getClasses` *to list the classes defined in the ShortRead package.*

Given an instance of an S4 class "X", you might like to know what methods are available that operate on objects of class "X". You can use the `showMethods` function for this purpose as follows:

```
> showMethods(classes = "X", where = getNamespace("SomePackage"))
```

Exercise 18
Pick a class from the ShortRead package and use `showMethods` *to list methods that operate on it.*

7 R Operations in Practice: Package Dependencies

The following exercises will give you a chance to apply your R knowledge to gain some understand of package dependencies. To begin, create a matrix of the packages available via Bioconductor and CRAN.

```
> contrib <- contrib.url(biocinstallRepos())
> pkgs <- available.packages(contrib)
```

Exercise 19
Each package's dependencies are listed in the "Depends" column of `pkgs`*, but the listing is not easy to compute on since the dependencies are listed in a comma separated string instead of a character vector with one element per dependency. Create a list named by the packages in* `pkgs` *such that each element is a character vector of the named package's dependencies. Hint: work with a subset of the data to refine your solution.*

Exercise 20
Which package has the most dependencies? Which package is listed as a dependency of other packages most often?

8 Overview of high throughput sequencing packages

The Bioconductor contains over 25 packages related to the analysis of high throughput sequence data. While these packages can be categorized many different ways, a useful grouping is as follows:

SNP Associations : *snpMatrix*

Differential Expression of Tag Counts : baySeq, DEGseq, edgeR

Peak Analysis : ChIPpeakAnno, *chipseq*, ChIPseqR

Alignment I/O & QA : *Rsamtools*, *ShortRead*

Solexa Base Calling & QA : Rolexa

Read Simulation : ChIPsim

Annotation I/O : *biomaRt*, *rtracklayer*

Annotation Packages : *BSgenome.**, *GenomicFeatures.**, *org.*.db*, *SNPlocs.**

Annotation Infrastructure : *AnnotatinDbi*, *BSgenome*, *GenomicFeatures*

Sequence Visualizations : *GenomeGraphs*, HilbertVis, HilbertVisGUI

Sequence Infrastructure I : *Biostrings*, *IRanges*

Sequence Infrastructure II & Analysis : *Genominator*

Interval Alternative : genomeIntervals

Those packages printed in slanted text will be used throughout this course.

On the first day of this course we will be discussing the ShortRead package, which contains the functions `readAligned` and `report` for reading in alignments from popular high throughput sequencing aligners and creating QA reports respectively. From there we will take a look at the infrastructure and basic operations for sequences using the packages IRanges (intervals), Biostrings (DNA sequences), and BSgenome (genomes).

On the second day of this course we will begin the day using the chipseq package, which provides a workflow for ChIP-seq analyses. Its `estimate.mean.fraglen` provides multiple ways for estimating the length of the fragment captured during the chromatin immunoprecipitation step of the experiment so the alignments can be extended prior to coverage calculations. It also contains methods for peak calling and finding genomic context from those peaks. The second half of day two will be focused on rare variants and using Rsamtools to read in alignments; SNPlocs.*, GenomicFeatures.* and org.* packages to provide annotation context; and the snpMatrix package to examine SNP calls.

And on the last day of this course we will be turning our attention to RNA-seq experiments and the Genominator and GenomeGraphs packages. Genominator is a comprehensive tool for both representing and analyzing high throughput sequencing data. It uses a separate, yet complementary, approach for representing and analyzing sequencing data than is presented during the first two days.

9 Session information

- R version 2.10.0 Patched (2009-11-17 r50467), `i386-apple-darwin10.2.0`
- Locale: `C/C/C/C/C/en_US.utf-8`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: BSgenome 1.14.1, Biostrings 2.14.5, IRanges 1.4.6, ShortRead 1.4.0, lattice 0.17-26
- Loaded via a namespace (and not attached): Biobase 2.6.0, grid 2.10.0, hwriter 1.1