Bioconductor tutorial

Adapted by Alex Sanchez from tutorials by
(1) Steffen Durinck, Robert Gentleman and Sandrine Dudoit
(2) Laurent Gautier
(3) Matt Ritchie
(4) Jean Yang
Outline

- The Bioconductor Project
- OOP in R and Bioconductor
- Start-up: Installation, Courses, Vignettes
The Bioconductor Project
Bioconductor

- Bioconductor is an open source and open development software project for the analysis of biomedical and genomic data.
- The project was started in the Fall of 2001 and includes more than 25 core developers in the US, Europe, and Australia.
- Releases
  - v 1.0: May 2\textsuperscript{nd}, 2002, 15 packages.
  - v 1.1: November 18\textsuperscript{th}, 2002, 20 packages.
  - v 1.2: May 28\textsuperscript{th}, 2003, 30 packages.
  - v 1.9: October 4, 2006, 188 packages.
Goals

• Provide access to powerful statistical and graphical methods for the analysis of genomic data.
• Facilitate the integration of biological metadata (GenBank, GO, Entrez, PubMed) in the analysis of experimental data.
• Allow the rapid development of extensible, interoperable, and scalable software.
• Promote high-quality documentation and reproducible research.
• Provide training in computational and statistical methods.
Bioconductor packages

• **R** and the **R package system** are used to design and distribute Bioconductor software.

• An **R package** is a structured collection of code (R, C, or other), documentation, and/or data for performing specific types of analyses.

• Different types of packages
  – Code
  – Data (metadata or experimental data)
  – Other (course packages …)
Bioconductor packages

- **Code packages** provide implementations of specialized statistical and graphical methods.

- **Data packages:**
  - Biological metadata: mappings between different gene identifiers (e.g., AffyID, GO, Entrez, PMID), CDF and probe sequence information for Affy arrays.
    E.g. hgu133plus2, GO, KEGG.
  - Experimental data: code, data, and documentation for specific experiments or projects.
    - golubEsets: Golub et al. (2000) ALL/AML data.

- **Course packages**: code, data, documentation, and labs for the instruction of a particular course. E.g. EMBO03 course package.
Some packages arranged by their functionality

Pre-processing

- CEL, CDF
  - affy
  - vsn
- .gpr, .Spot, MAGEML
  - marray
  - limma
  - vsn

ExpressionSet

Differential expression
- edd
genefilter
limma
multtest
ROC
+ CRAN

Graphs & networks
- graph
- RBGL
- Rgraphviz

Cluster analysis
- CRAN
  - class
  - cluster
  - MASS
  - mva

Prediction
- CRAN
  - class
e1071
ipred
LogitBoost
MASS
nnet
randomForest
rpart

Annotation
- annotate
- annaffy
+ metadata packages

Data
- estrogen
- AMLL
OOP in Bioconductor and R
The Bioconductor project has adopted the object-oriented programming (OOP) paradigm proposed in J. M. Chambers (1998). *Programming with Data.*

This object-oriented class/method design allows efficient representation and manipulation of large and complex biological datasets of multiple types.

Tools for programming using the class/method mechanism are provided in the R `methods` package.

OOP: classes

• A class provides a software abstraction of a real world object. It reflects how we think of certain objects and what information these objects should contain.
• Classes are defined in terms of slots which contain the relevant data.
• An object is an instance of a class.
• A class defines the structure, inheritance, and initialization of objects.
OOP: methods

• A method is a function that performs an action on data (objects).
  – Methods define how a particular function should behave depending on the class of its arguments.
  – Methods allow computations to be adapted to particular data types, i.e., classes.

• A generic function is a dispatcher, it examines its arguments and determines the appropriate method to invoke.
  – Examples of generic functions in R include `plot`, `summary`, `print`. 
It is important to realize that when calling a generic function (such as `plot`), the actions performed depend on the class of the arguments.

Methods define how a particular function should behave depending on the class of its arguments.

Methods allow computations to be adapted to particular data types, i.e., classes.
**ExpressionSet class**

**Processed Affymetrix or spotted array data**

- **assayData**: Matrix of expression measures, genes x samples
- **phenoData**: Sample level covariates, instance of class annotatedDataFrame
- **annotation**: Name of annotation data source (annotation package)
- **annotation**: Name of features = data identifiers
- **description**: MIAME information
  - Use of object-oriented programming to deal with data complexity.
  - S4 class/method mechanism (methods package).
- **notes**: Any notes
### AffyBatch class

**Probe-level intensity data for a batch of arrays (same CDF)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cdfName</code></td>
<td>Name of CDF file for arrays in the batch</td>
</tr>
<tr>
<td><code>nrow</code></td>
<td>Dimensions of the array</td>
</tr>
<tr>
<td><code>ncol</code></td>
<td></td>
</tr>
<tr>
<td><code>exprs</code></td>
<td>Matrices of probe-level intensities and SEs</td>
</tr>
<tr>
<td><code>se.exprs</code></td>
<td></td>
</tr>
<tr>
<td><code>phenoData</code></td>
<td>Sample level covariates, instance of class <code>phenoData</code></td>
</tr>
<tr>
<td><code>annotation</code></td>
<td>Name of annotation data</td>
</tr>
<tr>
<td><code>description</code></td>
<td>MIAME information</td>
</tr>
<tr>
<td><code>notes</code></td>
<td>Any notes</td>
</tr>
</tbody>
</table>
Getting Started
Installation

1. **Main R software**: download from CRAN ( cran.r-project.org ), use latest release.

2. **Bioconductor packages**: download from Bioconductor (www.bioconductor.org), use latest release.

Available for Linux/Unix, Windows, and Mac OS.
Installation

- After installing R, install Bioconductor packages using `getBioC` install script.

- From R
  ```r
  > source("http://www.bioconductor.org/biocLite().R")
  > biocLite()
  ```

- In general, R packages can be installed using the function `install.packages`.

- In Windows, can also use “Packages” pull-down menus.
Installing vs. loading

• Packages only need to be installed once.
• But … packages must be loaded with each new R session.
• Packages are loaded using the function library. From R
  > library(Biobase)
  or the “Packages” pull-down menus in Windows.
• To update packages, use function update.packages or “Packages” pull-down menus in Windows.
• To quit:
  > q()
Documentation and help

- **R manuals and tutorials**: available from the R website or on-line in an R session.

- **R on-line help system**: detailed on-line documentation, available in text, HTML, PDF, and LaTeX formats.

```
> help.start()
> help(lm)
> ?hclust
> apropos(mean)
> example(hclust)
> demo()
> demo(image)
```
Short courses

• Bioconductor short courses
  – modular training segments on software and statistical methodology;
  – lectures notes, computer labs, and course packages available on WWW for self-instruction.
Vignettes

• Bioconductor has adopted a new documentation paradigm, the vignette.
• A vignette is an executable document consisting of a collection of code chunks and documentation text chunks.
• Vignettes provide dynamic, integrated, and reproducible statistical documents that can be automatically updated if either data or analyses are changed.
• Vignettes can be generated using the Sweave function from the R tools package.
Vignettes

- Each Bioconductor package contains at least one vignette, providing task-oriented descriptions of the package's functionality.
- Vignettes are located in the `doc` subdirectory of an installed package and are accessible from the help browser.
- Vignettes can be used interactively.
- Vignettes are also available separately from the Bioconductor website.
Vignettes

• Tools are being developed for managing and using this repository of step-by-step tutorials.

• Some packages have functions to access or browse vignettes:
  - **Biobase:**
    • `openVignette` – Menu of available vignettes and interface for viewing vignettes (PDF).
    • `browseVignette()`–
  - **tkWidgets:**
    • `vExplorer` – Interactive use of vignettes.
Vignettes

- HowTo’s: Task-oriented descriptions of package functionality.
- Executable documents consisting of documentation text and code chunks.
- Dynamic, integrated, and reproducible statistical documents.
- Can be used interactively – vExplorer.
- Generated using Sweave (tools package).
References

• **R** [www.r-project.org](http://www.r-project.org), [cran.r-project.org](http://cran.r-project.org)
  – software (CRAN);
  – documentation;
  – newsletter: R News;
  – mailing list.

• **Bioconductor** [www.bioconductor.org](http://www.bioconductor.org)
  – software, data, and documentation (vignettes);
  – training materials from short courses;
  – mailing list.
Exercises

- Install Bioconductor using BiocLite()
- Load the Biobase package
- Browse which vignettes are available using the openVignette() function
- Browse the vignettes and carefully read and try the exercises in # 1: *An Introduction to Biobase and ExpressionSets*