

Introduction to Bioconductor and Annotation

in
Master in Omic data analysis
Universitat Vic (Uvic)

Juan R Gonzalez
(jrgonzalez@creal.cat)

BRGE - Bioinformatics Research Group in Epidemiology
<http://www.creal.cat/brge>
Barcelona Institute for Global Health (ISGlobal)
Departament of Mathematics, Universidad Autonoma de Barcelona (UAB)



1 Bioconductor

2 Data bases and annotation

Outline

- 1 Bioconductor
- 2 Data bases and annotation

What is Bioconductor?

- Software project for analysis of genomic data (and related tools)
- **Open source** and **Open development**
- **Free**
- www.bioconductor.org

Bioconductor basics

- Begun in 2001, based at Harvard and now FHCRC (Seattle)
- A large collection of R packages
- **Book:** Bioinformatics and Computational Biology Solutions Using R and Bioconductor. R Gentleman et al. Springer

Bioconductor basics

Two step installation

- Main R software: download from CRAN (www.cran.r-project.org)
- Bioconductor packages: download from Bioconductor website

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

install some basic libraries like affy affydata
affyPLM annotate Biobase limma multtest ROC xtable
... then you use `library(affy)` to load a given
library. `openVignette("limma")` opens a worked example
- very helpful introduction

Bioconductor basics


- To get other packages, use e.g. `biocLite("SNPchip")`
- **Do not** need to type `biocLite()` after you install (even in a new R session)
- This would install everything again (slow)

Bioconductor basics

Documentation and Help

- `help.start()`
- `help(lm)`
- `? mean`
- `apropos(clust)`
- `example(hclust)`
- `demo(image)`
- **Vignettes**
- **Google!!!**

Bioconductor basics



Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS


[Home](#)
[Install](#)
[Help](#)
[Developers](#)
[About](#)


About Bioconductor


Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, more than [460 packages](#), and an active user community.

Use Bioconductor for...

- [Microarrays](#)
Import Affymetrix, Illumina, Nimblegen, Agilent, and other platforms. Perform quality assessment, normalization, differential expression, clustering, classification, gene set enrichment, genetical genomics and other workflows for expression, exon, copy number, SNP, methylation and other assays. Access GEO, ArrayExpress, Biomart, UCSC, and other community resources.
- [Sequence Data](#)
Import fasta, fastq, ELAND, MAQ, BWA, Bowtie, BAM, gff, bed, wig, and other sequence formats. Trim, transform, align, and manipulate sequences. Perform quality assessment, ChIP-seq, differential expression, RNA-seq, and other workflows. Access the Sequence Read Archive.
- [Annotation](#)
Use microarray probe, gene, pathway, gene ontology, homology and other annotations. Access GO, KEGG, NCBI, Biomart, UCSC, vendor, and other sources.
- [High Throughput Assays](#)
Import, transform, edit, analyze and visualize flow cytometric, mass spec, HTqPCR, cell-based, and other assays.


[Mailing Lists](#)


[Events](#)


[News](#)

[Re: Illumina array analysis](#)
about 2 hours ago

[Bioconductor Basics](#)
24 - 25 October 2011 — Boston, MA, USA

[Advanced R Programming](#)
28 - 29 November 2011 — Heidelberg

[BioC 2011 conference material](#)
BioC 2011 conference material is now available.

Bioconductor basics

[Home](#)
[Install](#)
[Help](#)
[Developers](#)
[About](#)

Search:

[Home](#) » [Bioconductor 2.8](#) » 2.8 Software Packages

Bioconductor Software Packages

Bioconductor version: Release (2.8)

Package	Maintainer	Title
a4	Tobias Verbeke	Automated Affymetrix Array Analysis Umbrella Package
a4Base	Tobias Verbeke	Automated Affymetrix Array Analysis Base Package
a4Classif	Tobias Verbeke	Automated Affymetrix Array Analysis Classification Package
a4Core	Tobias Verbeke	Automated Affymetrix Array Analysis Core Package
a4Preproc	Tobias Verbeke	Automated Affymetrix Array Analysis Preprocessing Package
a4Reporting	Tobias Verbeke	Automated Affymetrix Array Analysis Reporting Package
ABarray	Yongming Andrew Sun	Microarray QA and statistical data analysis for Applied Biosystems Genome Survey Microarray (AB1700) gene expression data.
aCGH	Peter Dimitrov	Classes and functions for Array Comparative Genomic Hybridization data.
ACME	Sean Davis	Algorithms for Calculating Microarray Enrichment (ACME)
ADaCGH2	Ramon Diaz-Uriarte	Analysis of data from aCGH experiments using parallel computing and ff objects
adSplit	Claudio Lottaz	Annotation-Driven Clustering
affxparser	Kasper Daniel Hansen	Affymetrix File Parsing SDK

Bioconductor Release »

Packages in the stable, semi-annual release:

- [BioViews](#) package discovery
- [Software](#)
- [Metadata](#) (Annotation, CDF and Probe)
- [Experiment Data](#)

Bioconductor is also available as an [Amazon Machine Image](#) (AMI).

Development Version»

Bioconductor packages under development:

- [BioViews](#) package discovery
- [Software](#)
- [Metadata](#) (Annotation, CDF and Probe)
- [Experiment Data](#)

Developer Resources:

- [Subversion Log](#)
- [New Packages](#)

Bioconductor basics

- Provide access to powerful **statistical and graphical methods** for the analysis of biomedical and genomic data
- Facilitate the integration of **biological metadata** from WWW in the analysis of experimental data (e.g., GenBank, GO, LocusLink, PubMed)
- 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

- Analysis packages: `annotate`, `affy`, `marray`, `multtest`
- Data packages:
 - Biological metadata: mapping between different gene identifiers (e.g., AffyID, GO ID, PMID) [`hgu95av2`, `GO`, `KEGG`]
 - Experimental data: code, data and documentation for specific experiments or projects
 - `ALL`: Chiaretti et al. (2004) ALL dataset
 - `golubEsets`: Golub et al. (2000) ALL/AML dataset
 - `yeastCC`: Spellman et al. (1998) yeast cell cycle dataset
- Course packages: code, data, documentation and labs for the instruction of a particular course (e.g. `EMB003` package)

Outline

- 
- 1 Bioconductor
 - 2 Data bases and annotation

Types of databases

- **Translations of names:** Affy probe 32972_at is the gene **NADPH oxidase 1** with symbol **NOX1** and Ensembl gene id **ENSG00000007952**
- **Location:** NOX1 is on Xq22.1, from 99984969 to 100015990, coded on the negative strand. There are 120 known polymorphisms (SNPs or indels) in this range.
- **Homology:** The mouse version of NOX1 is also on the X chromosome, starting at 130621066 (and called Nox1)
- **Structure and function:** **NOX1** is a membrane protein (location), involved in voltage-gated ion channel activity (molecular function), and involved in signal transduction (biological process)

Annotate

Bioconductor distributes annotation packages for a wide range of gene expression microarrays and RNA-seq data. The `annotate` package is one way to use this annotation information.

```
library("annotate")  
library("hgu95av2.db")  
library("GO.db")  
library(biomaRt)
```

loads the *annotate* package and the databases for the Gene Ontology and one of the Affymetrix human microarray chips.

Lookups

The databases are queried with `get()` or `mget()` for multiple queries:

```
get("32972_at", envir=hgu95av2GENENAME)
```

```
## [1] "NADPH oxidase 1"
```

```
go <- get("738_at", envir=hgu95av2GO)
names(go)
```

```
## [1] "GO:0006195" "GO:0016310" "GO:0016311" "GO:0017144" "GO:0046040"
## [6] "GO:0046085" "GO:0005829" "GO:0000166" "GO:0008253" "GO:0046872"
## [11] "GO:0050146"
```

```
mget(c("738_at", "40840_at", "32972_at"),
     envir=hgu95av2GENENAME)
```


Lookups

```
get("GO:0009117", envir=GOTERM)

## GOID: GO:0009117
## Term: nucleotide metabolic process
## Ontology: BP
## Definition: The chemical reactions and pathways involving a
##             nucleotide, a nucleoside that is esterified with
##             (ortho)phosphate or an oligophosphate at any hydroxyl group on
##             the glucose moiety; may be mono-, di- or triphosphate; this
##             definition includes cyclic nucleotides (nucleoside cyclic
##             phosphates).
## Synonym: nucleotide metabolism
```

Biomart

BioMart (www.biomart.org) is a query-oriented data management system developed jointly by the European Bioinformatics Institute (EBI) and Cold Spring Harbor Laboratory (CSHL). *biomaRt* is an R interface to BioMart systems, in particular to Ensembl (www.ensembl.org). Ensembl is a joint project between EMBL - European Bioinformatics Institute (EBI) and the Wellcome Trust Sanger Institute (WTSI) to develop a software system which produces and maintains automatic annotation on selected eukaryotic genomes.

biomaRt

- biomaRt enables easy retrieval of large amounts of data
- Access to Ensembl, COSMIC, Uniprot, HGNC, Gramene, Wormbase, dbSNP...

```
library(biomaRt)
head(listMarts())
```

```
##              biomaRt              version
## 1 ENSEMBL_MART_ENSEMBL      Ensembl Genes 86
## 2  ENSEMBL_MART_MOUSE      Mouse strains 86
## 3    ENSEMBL_MART_SNP      Ensembl Variation 86
## 4 ENSEMBL_MART_FUNCGEN      Ensembl Regulation 86
## 5    ENSEMBL_MART_VEGA              Vega 66
```

Selecting dataset

After selecting a database (e.g., *ensembl*) we select a dataset:

```
mart <- useMart(biomart="ensembl")
listDatasets(mart)[1:10,]
```

##	dataset		
## 1	oanatinus_gene_ensembl		
## 2	cporcellus_gene_ensembl		
## 3	gaculeatus_gene_ensembl		
## 4	itridecemlineatus_gene_ensembl		
## 5	lafricana_gene_ensembl		
## 6	choffmanni_gene_ensembl		
## 7	csavignyi_gene_ensembl		
## 8	fcatus_gene_ensembl		
## 9	rnorvegicus_gene_ensembl		
## 10	psinensis_gene_ensembl		
##		description	version
## 1	Ornithorhynchus anatinus genes	(OANA5)	OANA5
## 2	Cavia porcellus genes	(cavPor3)	cavPor3
## 3	Gasterosteus aculeatus genes	(BROADS1)	BROADS1
## 4	Ictidomys tridecemlineatus genes	(spetri2)	spetri2
## 5	Loxodonta africana genes	(loxAfr3)	loxAfr3
## 6	Cheleponus heffmanni genes	(cheHef1)	cheHef1

Selecting attributes

After selecting a dataset (e.g., *hsapiens_gene_ensembl*) we select the attributes we are interested in:

```
mart <- useMart(biomart="ensembl", dataset="hsapiens_gene_ensembl")  
listAttributes(mart)[1:10,]
```

##	name	description	page
## 1	ensembl_gene_id	Ensembl Gene ID	feature_page
## 2	ensembl_transcript_id	Ensembl Transcript ID	feature_page
## 3	ensembl_peptide_id	Ensembl Protein ID	feature_page
## 4	ensembl_exon_id	Ensembl Exon ID	feature_page
## 5	description	Description	feature_page
## 6	chromosome_name	Chromosome Name	feature_page
## 7	start_position	Gene Start (bp)	feature_page
## 8	end_position	Gene End (bp)	feature_page
## 9	strand	Strand	feature_page
## 10	band	Band	feature_page

NOTE: sometimes the host is not working. If so, try
`host="www.ensembl.org"` in the `useMart` function.

Querying

After selecting the dataset we can make different types of queries:

Query 1: We could look for all the transcripts contained in the gene 7791 (entrez id):

```
tx <- getBM(attributes="ensembl_transcript_id",
            filters="entrezgene",
            values="7791", mart=mart)
```

```
tx
```

```
##      ensembl_transcript_id
## 1      ENST00000322764
## 2      ENST00000449630
## 3      ENST00000468083
## 4      ENST00000457235
## 5      ENST00000354434
## 6      ENST00000392910
## 7      ENST00000477373
## 8      ENST00000436448
## 9      ENST00000446634
## 10     ENST00000497119
```

Querying

Query 2: We could look for chromosome, position and gene name of a list of genes (entrez id):

```
genes <- c("79699", "7791", "23140", "26009")
tx <- getBM(attributes=c("chromosome_name", "start_position",
                        "hgnc_symbol"),
            filters="entrezgene",
            values=genes, mart=mart)
```

tx

##	chromosome_name	start_position	hgnc_symbol
## 1	17	4004445	ZZEF1
## 2	1	77562416	ZZZ3
## 3	7	143381080	ZYX
## 4	1	52726467	ZYG11B

Querying

Query 3: We could look for chromosome, position and gene name of a list of genes (ENSEMBL):

```
genes <- c("ENSG000000074755")
tx <- getBM(attributes=c("chromosome_name", "start_position",
                        "hgnc_symbol"),
            filters="ensembl_gene_id",
            values=genes, mart=mart)
```

tx

```
## chromosome_name start_position hgnc_symbol
## 1              17          4004445      ZZEF1
```


Homology

getLDS() combines two data marts, for example to homologous genes in other species. We can look up the mouse equivalents of a particular Affy transcript, or of the NOX1 gene.

```
human <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
mouse <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")
getLDS(attributes = c("hgnc_symbol", "chromosome_name",
  "start_position"),
  filters = "hgnc_symbol", values = "NOX1",
  mart = human,
  attributesL = c("external_gene_name", "chromosome_name",
  "start_position"),
  martL = mouse)
```

##	HGNC.symbol	Chromosome.Name	Gene.Start..bp.	Associated.Gene.Name
## 1	NOX1	X	100843324	Nox1
##	Chromosome.Name.1	Gene.Start..bp..1		
## 1	X	134086421		

The mouse gene name is the same as the human one apart from capitalisation

Homology

The `getSequence` function looks up DNA or protein sequences by chromosome position or gene identifiers

```
agt<-getSequence(id="AGT", type="hgnc_symbol",  
                 seqType="peptide", mart=mart)  
agt  
  
##  
## 1 MRKRAPQSEMAPAGVSLRATILCLLAWAGLAAGDRVYIHPFHLVIHNESTCEQLAKANAGPKDPTFIPAPIQA  
## hgnc_symbol  
## 1 AGT
```

Example: finding SNPs

We had a set of 100 SNPs without chromosome and genomic position information. We need to know the gene that those SNPs belong to.

```
load("data/snpsList.Rdata")
head(snpsList)
```

```
## [1] "rs1932919" "rs6849766" "rs11246756" "rs7184849" "rs308857"
## [6] "rs16990309"
```

A hand-search (Genome Browser - <http://genome.ucsc.edu/>) would be easy but tedious, so we want an automated approach

Types of databases

NOTE: Use `listAttributes` function to get valid attribute names. Use `listFilters` function to get valid filter names.

```
snpmart <- useMart("ENSEMBL_MART_SNP", dataset = "hsapiens_snp")
snpInfo <- getBM(c("refsnp_id", "chr_name", "chrom_start",
                  "allele"), filters = c("snp_filter"),
                 values = snpsList, mart = snpmart)

head(snpInfo)
```

##	refsnp_id	chr_name	chrom_start	allele
## 1	rs1000669	8	106264762	T/C
## 2	rs10084057	18	40285960	C/T
## 3	rs1010795	8	22862510	G/A
## 4	rs1037488	11	68809483	A/G
## 5	rs10496205	2	77266581	C/T
## 6	rs10506568	12	69649610	T/C