Introduction of Bioinformatics

Fall 2007

Part I: Introduction of R & Bioconductor

Websites

- R: www.r-project.org
  - Software
  - Documentation
  - RNews
- Bioconductor: www.bioconductor.org
  - software, data, and documentation
  - training materials from short courses
  - mailing list

Introduction of R

What is R?

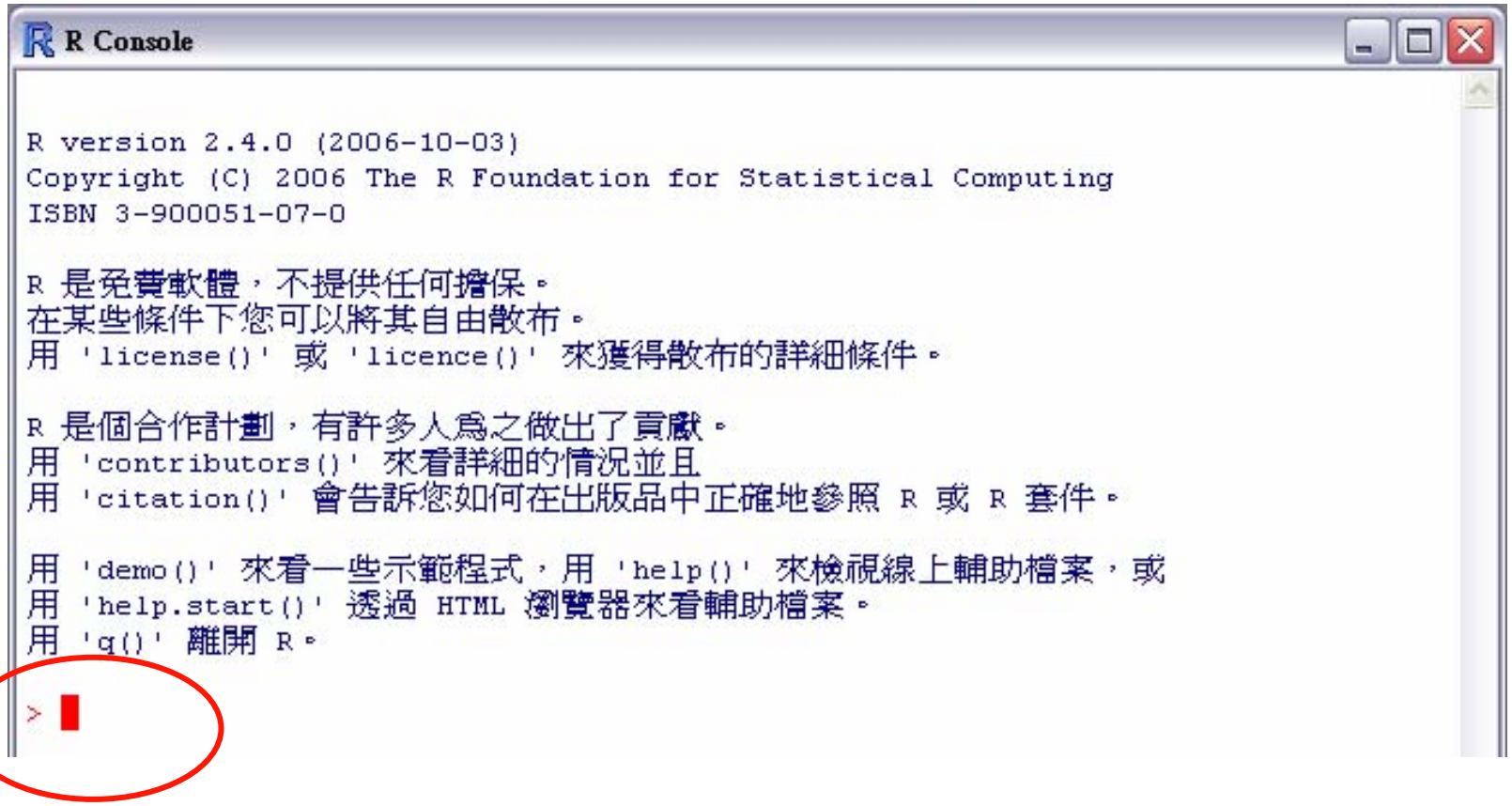
- R 並非專用統計軟體, 而是可用來執行統計分析的環境:
  - 匯入適當的 package (套件)
  - 應用套件內提供之 function (函式)
- Packages 由許多熱心人士編寫並免費提供學術使用。

  You can make your own contribution in the future.

R的優缺點

- 優點:
  - 免費軟體
  - 完善的說明文件與討論區
  - 漂亮的圖型介面
  - 程式容易根據使用者需求做修改
- 缺點:
  - 並無 user friendly 之使用者介面
  - 需詳知函式名稱與程式編寫邏輯
  - 說明文件與討論區使用英文

R Console


R Console
R version 2.4.0 (2006-10-03)
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
R 是免費軟體，不提供任何擔保。
在某些條件下您可以將其自由散布。
用 'license()' 或 'licence()' 來獲得散布的詳細條件。
R 是個合作計劃，有許多人爲之做出了貢獻。
用 'contributors()' 來看詳細的情況並且
用 'citation()' 會告訴您如何在出版品中正確地參照 R 或 R 套件。
用 'demo()' 來看一些示範程式，用 'help()' 來檢視線上輔助檔案，或
用 'help.start()' 透過 HTML 瀏覽器來看輔助檔案。
用 'q()' 離開 R。
>

R 的提示符號: > 與 +

- “>” 為提示符號; 當提示符號出現時表示R正在待命中, 可以隨時鍵入下一個命令。當提示符號為“+”時, 表示程式正在執行中，或在等待未完成的指令。例如:

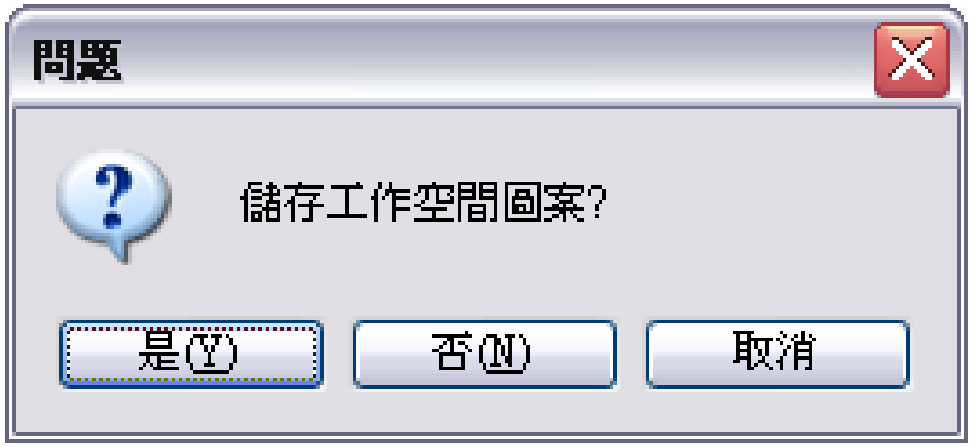
```
> (1.36 + 0.7
+ )
>
```

  按“ESC”可強制退出未完成的工作。

- 可利用鍵盤上下鍵重複輸入指令或做小幅度修改

離開 R

- Method 1: File -> Exit
- Method 2: > q()
- Method 3: R 視窗上方 



Getting help from R

- Getting help from R
  - ?*command* and help(*command*): 查詢特定命令 (名稱已知且package已載入) 的使用
  - Search help…: 查詢未知名稱的命令(package未載入但已安裝) →help.search("*keyword* ")
  - search.r-project.org: 廣義搜尋

利用R進行簡單計算

```
> 2 + 3
[1]  5
> sqrt(3/4)/(1/3-2/pi^2)
[1]  6.626513
> exp(3.72)
[1]  41.26439
> sin(pi); log(10); log2(10); log10(10)
```

指定變數名稱

- 名稱 = 物件 或 名稱 ← 物件
  - 可由英文字母、數字、英文句點(.)組成。
  - 英文大小寫有所區別 (ab 與 Ab 可分別代表不同變數) 。
  - 變數名稱須起始於英文字母。
  - 有些英文單字具有特殊意義，不能當做變數名稱: return, break, if, TRUE, FALSE, (T, F).

Some Simple Examples

- 單一數值或字元:
  > x1 = 3.0
  > x2 = “NTU”
  > x3 = TRUE
- 計算結果:
  > y1 = exp(3.72)
  > y2 = y1*3
- 其它:
  > out = lm(y~x+z)
  > y = NA

R 函式 (function)

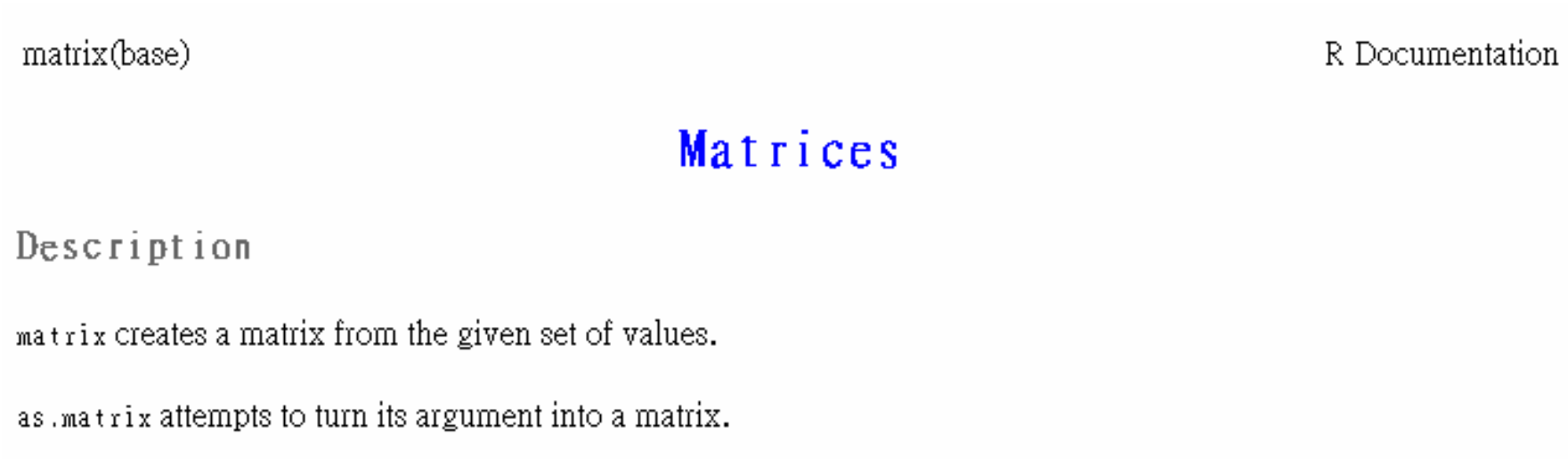
- R是由“變數”與“函式”組成。前面幾張 slide 已用的 function: **c, lm**, **seq**, **matrix**, **exp**, etc.
- 基本語法:

  funcname(參數)

- 查詢function的使用方法:
  - help or ?

    Example: > ?lm # help(lm)

Help Menu



matrix(base) R Documentation

Matrices

Description

`matrix` creates a matrix from the given set of values.

`as.matrix` attempts to turn its argument into a matrix.

- Description
- Details
- See Also
- Usage
- Value
- Examples
- Arguments
- References

apropos

- apropos(“matrix”)

```
> apropos("matrix")
 [1] ".__C__matrix"                  "model.matrix"
 [3] "model.matrix.default"          "model.matrix.lm"
 [5] "head.matrix"                   "tail.matrix"
 [7] "as.data.frame.matrix"          "as.data.frame.model.matrix"
 [9] "as.matrix"                     "as.matrix.data.frame"
[11] "as.matrix.default"             "as.matrix.noquote"
[13] "as.matrix.POSIXlt"             "data.matrix"
[15] "determinant.matrix"            "duplicated.matrix"
[17] "is.matrix"                     "isSymmetric.matrix"
[19] "matrix"                        "prmatrix"
[21] "subset.matrix"                 "summary.matrix"
[23] "unique.matrix"
>
```

只查閱函式的參數時

> args(“matrix”)

```
> args("matrix")
function (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
NULL
>
```

- 參數分為必要與非必要兩種
- 順序不對調時, 參數名稱可不給定:

  > matrix(x, 3, 2)

- 順序對調時參數命稱必須指定:

  > matrix(nrow=3, ncol=2, data=x)

R 函式 (function)

- 函數也可由使用者自行定義

```
> my.add <- function(x){x+3}
> my.add(1:3)
[1] 4 5 6
```

資料輸入: c

- 輸入少量資料最簡單的方法: c function.

Example: 西元 1861 – 1870 年間重大的科學發現或發明數:

3 0 2 0 3 2 3 6 1 2

```
> nod = c(3, 0, 2, 0, 3, 2, 3, 6, 1, 2)
> nod
[1] 3 0 2 0 3 2 3 6 1 2
```

資料輸入: c

- 前述指令指定一組數據給名為 nod 之變數; 以 “=” 或 “<-” 進行指定的工作。
- 指定變數 nod 後, 其數值不會自動出現在螢幕上; 在提示符號後輸入變數名稱, 才會顯示其數值。
- 螢幕顯示

  [1] 3 0 2 0 3 2 3 6 1 2

  表示此變數為一**向量(vector)**。

Data is a vector!

資料是以“向量”或“矩陣”型態組成, 元素可用其相對位置做為指標:

```
> length(x)   # how many elements
> x[2]        # the 2nd element
> x[1:5]      # the first 5 elements
> x[c(1,2,5)] # specific elements
> x[x>3]      # all greater than 3
> x[x < -2 | x > 2]
> which(x==5)   # which indices are equal to 5
> c(x,48,49,51,50,49)  # append values to x
```

Example

```
> nod[2]
[1] 0
> nod[-4]
[1] 3 0 2 3 2 3 6 1 2
> nod[c(1,2,5)]
[1] 3 0 3
> nod[nod > 1]
[1] 3 2 3 2 3 6 2
> which(nod > 1)
[1] 1 3 5 6 7 8 10
```

Matrix operation is similar!

```
> y2 = matrix(c(1:10), nrow = 2, ncol = 5)
> y2[2,1]
> y2[1,]
> y2[,4]
> y2[,-2]
```

變數名稱[列指標, 行指標]

```
> z2 = matrix(c(11:25), nrow = 3, ncol = 5)
> rbind(y2,z2)
> cbind(y2,z2)   # error
> cbind(t(y2),t(z2))
```

Other Methods to Input Data

- 類似c的輸入方式 (以空格分格, 空行結束):

  > x <- scan() # input numbers

  > x <- scan(what="") # input characters

  Example: 利用 scan 產生

  nod = c(3, 0, 2, 0, 3, 2, 3, 6, 1, 2)

```
> nod = scan()
1: 3 0 2 0 3 2 3 6 1 2
11:
Read 10 items
>
```

Other Methods to Input Data

- Read from files:

```
> x <- scan("test.txt")   # numbers
> x <- scan("testc.txt", what="")
                                  # characters
```

- 讀取數字與文字混合的資料: read.table, read.csv, etc

Read Excel Files

- xls → csv (保留每一行的標題)
- read.csv(file = "*filename*")

Example: M.L. Bittner, P. Meltzer, Y. Chen, et al., "Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling," *Nature*, 2000, vol. 406, pp. 536-540.

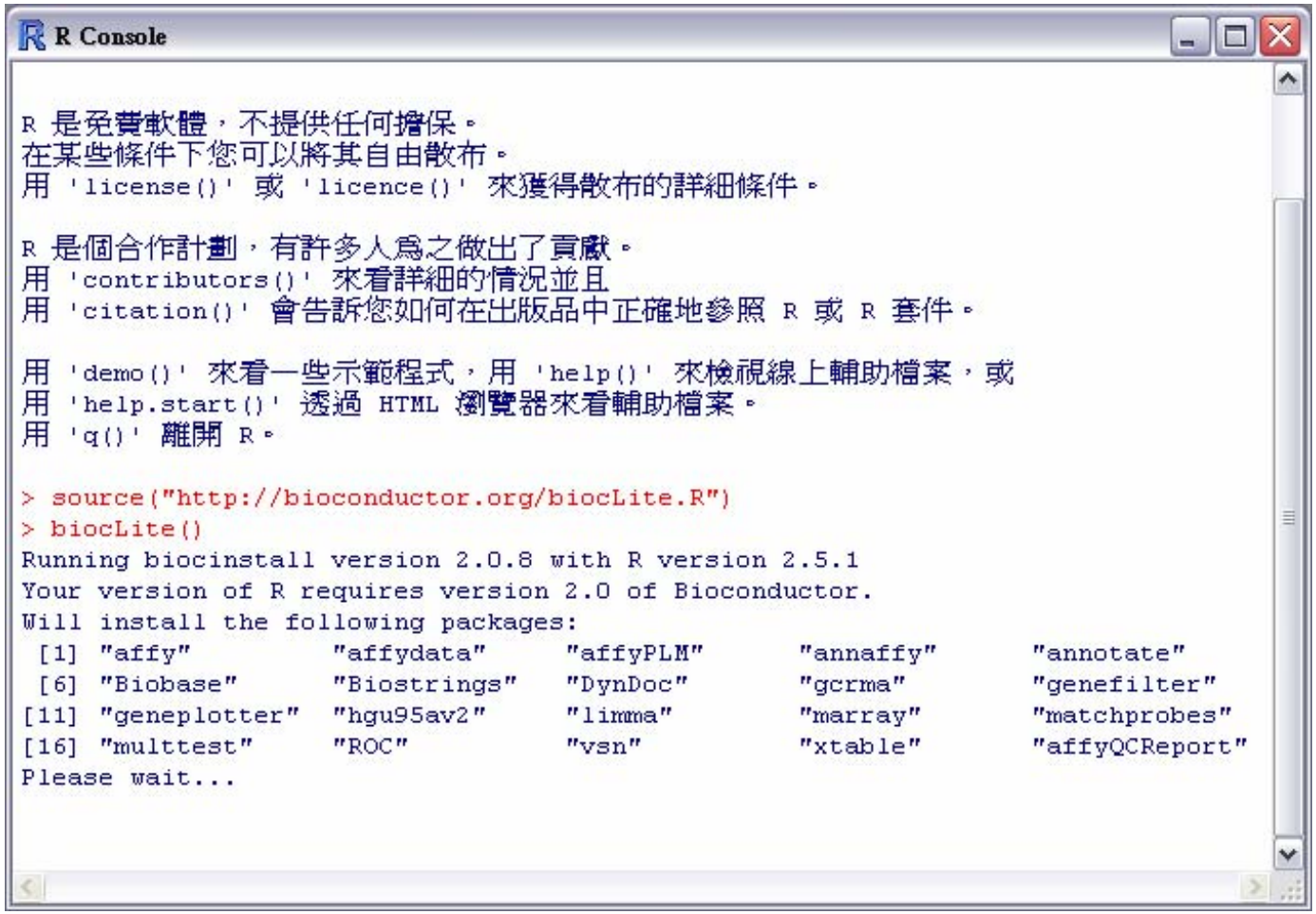
```
> dd = read.csv("melanoma.csv")
> dd[,1]
> dd$PlateLoc
```

Introduction of Bioconductor

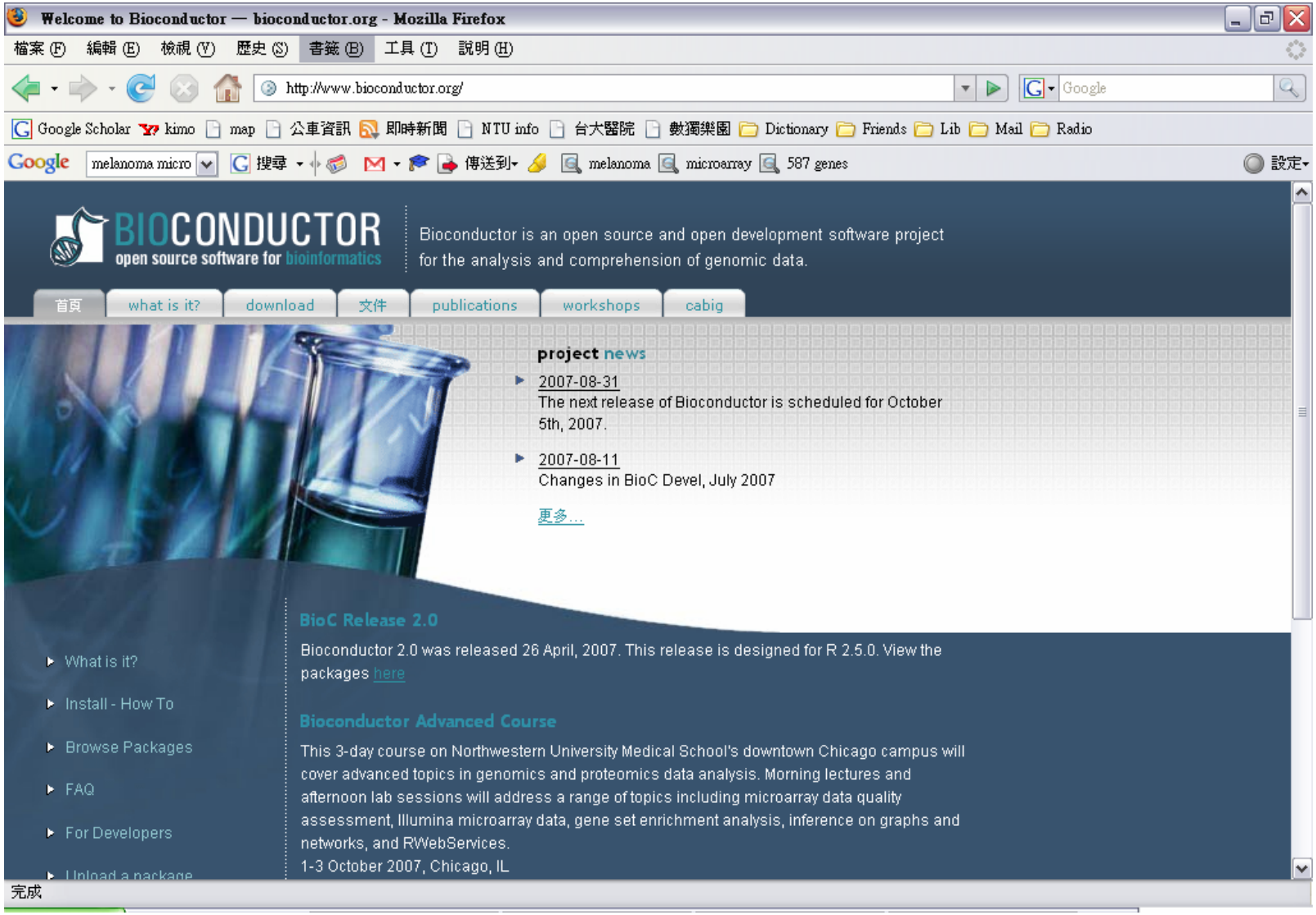
Overview

- R software project for the analysis of biomedical and genomic data
  - Microarrays
  - Genome sequence data
  - Pathway graphs
- Started in 2001 by Robert Gentleman. Additional packages developed and contributed by the research community
- Tools for integrating biological metadata from the web (annotation, literature)

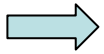
Installation of Bioconductor



Bioconductor Website



Bioconductor Packages



- What is it?
- Install - How To
- Browse Packages
- FAQ
- For Developers
- Upload a package
- Join mailing list

BioC Release 2.0

Bioconductor 2.0 was released 26 April, 2007. This release is designed for R 2.5.0. View the packages here

Bioconductor Advanced Course

This 3-day course on Northwestern University Medical School's downtown Chicago campus will cover advanced topics in genomics and proteomics data analysis. Morning lectures and afternoon lab sessions will address a range of topics including microarray data quality assessment, Illumina microarray data, gene set enrichment analysis, inference on graphs and networks, and RWebServices.
1-3 October 2007, Chicago, IL
Details **(CURRENTLY FULL!)**

BioC Release 2.1

The next release of Bioconductor is scheduled for October 5th, 2007. For details, see the
BioC 2.1 Release Schedule

Ph.D.-Course: Statistical Analysis of Microarray Expression Data with R and Bioconductor

The course aims to give Ph.D.-students in statistics as well as other Ph.D.-students a good introduction to microarray data analysis using R and Bioconductor. This is achieved by inviting some of the leading researchers in statistical analysis of microarray data and developers of R-packages to give the main lectures and combine this with hands-on computer exercises.
5-9 November 2007, Copenhagen, DK
Details

Bioconductor Packages

Bioconductor Task View: BiocViews

Subviews

- Software
- AnnotationData
- ExperimentData

Packages in view

No packages in this view

Bioconductor Packages -- Software

- Microarray:
  - OneChannel
  - TwoChannel
  - DataImport
  - QualityControl
  - Preprocessing
  - Transcription
  - DNACopyNumber
  - SNPsAndGeneticVariability
- Annotation:
  - GO
  - Pathways
  - ProprietaryPlatforms
  - ReportWriting
- Technology:
  - Microarray
  - Proteomics
  - MassSpectrometry
  - SAGE
  - CellBasedAssays
  - Genetics
- Statistics:
  - DifferentialExpression
  - Clustering
  - Classification
  - MultipleComparisons
  - TimeCourse
  - SequenceMatching
- Visualization
- GraphsAndNetworks
- Infrastructure

Bioconductor Packages -- Others

- AnnotationData: These packages provide annotation on the genes on microarrays. This resource can be searched by organism, chip manufacturer, chip name etc.
- ExperimentData: They contain published data pre-prepared for Bioconductor. Many of the tutorial s(vignettes) in Bioconductor use these data in exercises.

Vignettes

- Each Bioconductor package contains at least one vignette, a document that provides a task-oriented description of package functionality. Vignettes contain executable examples and are intended to be used interactively.

```
> library(affy)
Loading required package: Biobase
Loading required package: tools

Welcome to Bioconductor

  Vignettes contain introductory material. To view, type
  'openVignette()'. To cite Bioconductor, see
  'citation("Biobase")' and for packages 'citation(pkgname)'.

Loading required package: affyio
> openVignette()
Please select a vignette:

 1: affy - 1. Primer
 2: affy - 2. Built-in Processing Methods
 3: affy - 3. Custom Processing Methods
 4: affy - 4. Import Methods
 5: affy - 5. Automatic downloading of CDF packages
 6: Biobase - An introduction to Biobase and ExpressionSets
 7: Biobase - Bioconductor Overview
 8: Biobase - esApply Introduction
 9: Biobase - Notes for eSet developers
10: Biobase - Notes for writing introductory 'how to' documents

選擇: q
Enter an item from the menu, or 0 to exit
選擇: 0
>
```

Classes and Methods

- In order to deal with the complexity of microarray data, the Bioconductor packages have adopted the class/method object-oriented programming (OOP) paradigm
  - Classes are objects that follow a particular format.
  - Methods are functions, such as plot, that behave differently depending on class

```
> library(affydata)
> data(Dilution)
> class(Dilution)
[1] "AffyBatch"
attr(,"package")
[1] "affy"
> slotNames(Dilution)
[1] "cdfName"           "nrow"              "ncol"
[4] "assayData"         "phenoData"         "featureData"
[7] "experimentData"    "annotation"        ".__classVersion__"
>
```