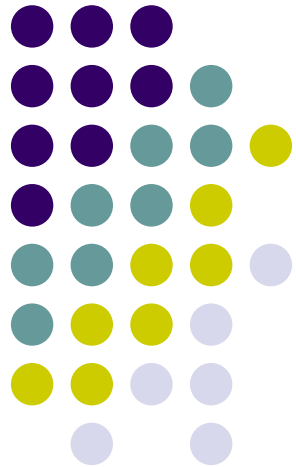
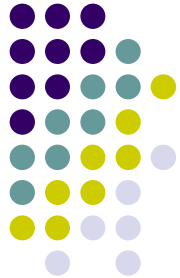
Introduction to Bioconductor

I. Overview of the Bioconductor Project



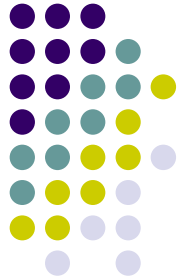
Bioinformatics and Biostatistics Lab.,
Seoul National Univ. Seoul, Korea
Eun-Kyung Lee

Outline



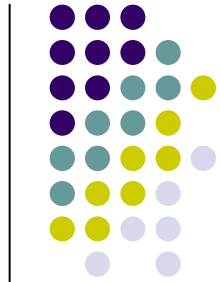
- What is R?
- Overview of the Biocondcutor Project
- Microarray
- Installing R and Bioconductor
- R package structure
- Vignette
- Sweave
- Bioconductor software design

What is R?



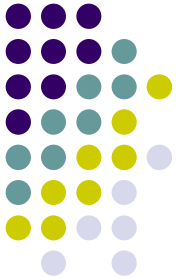
- A language and environment for statistical computing and graphics
- GNU project
- Combined Math and Stat library
- Fine graphics
- Easy and efficient handling of data
- Rich modern statistical routines

Strengths of R



- Available on a wide variety of UNIX platforms and similar systems, windows and MacOS
- Easy to produce well-designed publication quality plots, including math. Symbols and formulae
- For computationally-intensive tasks, C, C++ , and Fortran code can be linked and called at run time.
- Able to write C code to manipulate R objects directly

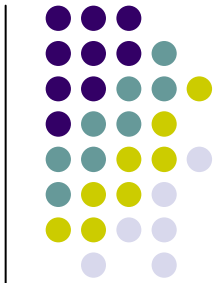
Differences between R and the other statistical SW



- R environment
  - characterize as a fully planned and coherent system
  - command line interface (CLI)
    - Preferred for power users
    - Intimidating for beginners
    - Longer learning curve
- Other SW
  - an incremental accretion of very specific and inflexible tools
  - Graphical user interface (GUI)

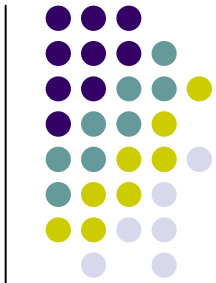
==> For Novice user, GUI is needed!

Bioconductor Project



- An open-source and open-development software project for the analysis of omics data
- R add-on packages
- Provide access to powerful statistical and graphical methods for the analysis
- Facilitate the integration of biological metadata from WWW
- Promote high-quality documentation and reproducible research

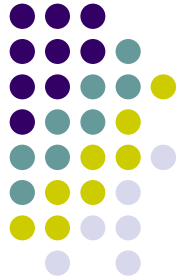
Bioconductor Packages



- Release 1.9 (Oct. 4, 2006) : 188 packages
- Designed for R 2.4.0

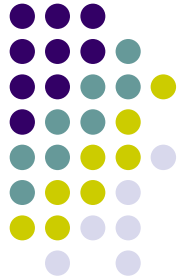
- Statistical method : cluster analysis, estimation and multiple testing for linear and non-linear models, resampling, visualization, etc
- Biological assays : cell-based assay, DNA microarray, proteomics, SAGE, SNP, etc
- Biological metadata from WWW : GenBank, GO, KEGG, PubMed, etc
- Interfaces with other languages : C, Java, per, Python, XML, etc..

Bioconductor Task View : SW



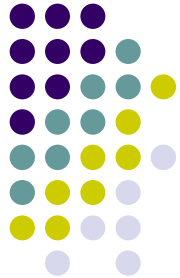
- Microarray
  - one channel, two channel, data input, quality control, preprocessing, transcription, DNA copy number, SNPs and genetic variability
  - **affy, marray, multtest, limma, etc.**
- Annotation
  - GO, Pathways, Proprietary platforms, Report writing
  - **annotate, AnnBuilder,etc.**
- Visualization
  - **chromoViz, arrayQCplot, etc**

Bioconductor Task View : SW



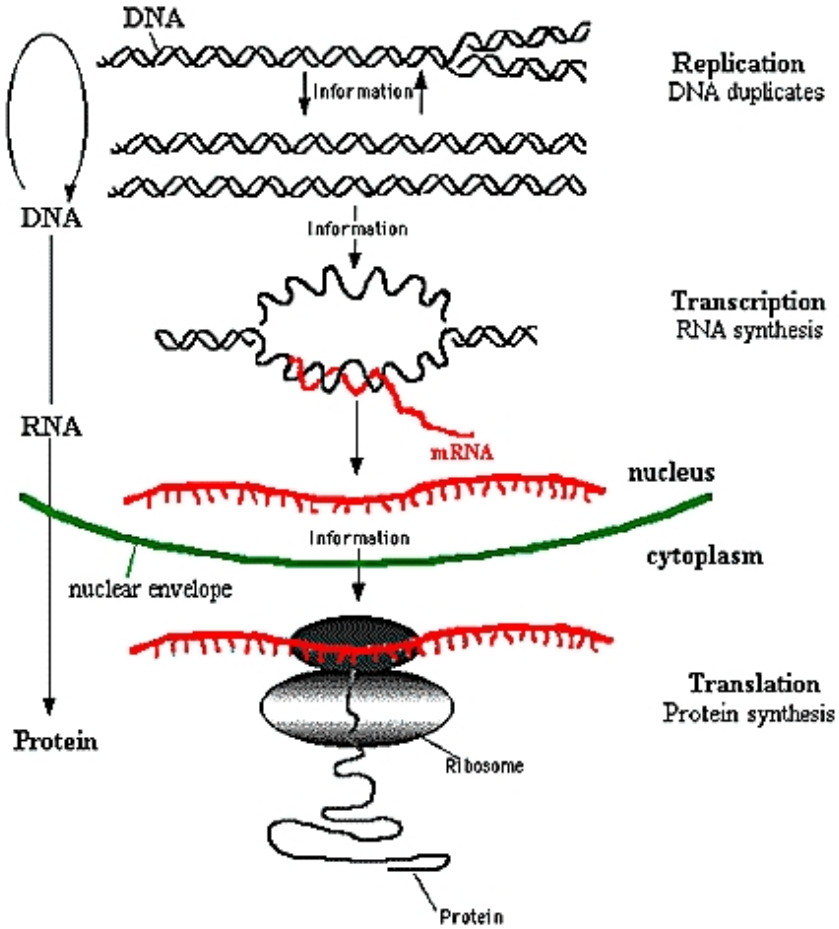
- Statistics
  - differential expression, clustering, classification, multiple comparison, time course, sequence matching
  - **limma, qvalue, affylmGUI, timecourse, etc**
- GraphAndNetworks
  - **GeneTS, Rgraphviz, etc.**
- Technology
  - microarray, proteomics, Mass spectrometry, SAGE, Cell based assays, genetics
- Intrastructure
  - **Biobase, Rdbi, tkWidgets, widgetTools, etc.**

Microarray



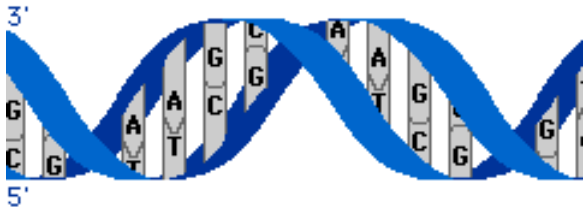
- DNA microarray chip
  - cDNA chip : usually custom based chip
    - Two channel
    - One channel
  - Oligonucleotide chip
    - Affymetrix
    - One channel eg) Agilent, Illumina

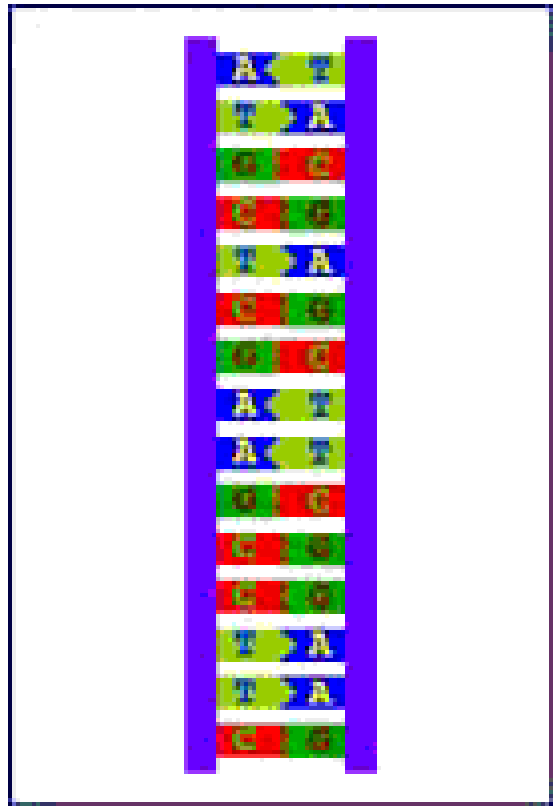
Central Dogma



The Central Dogma of Molecular Biology

cDNA Microarrays

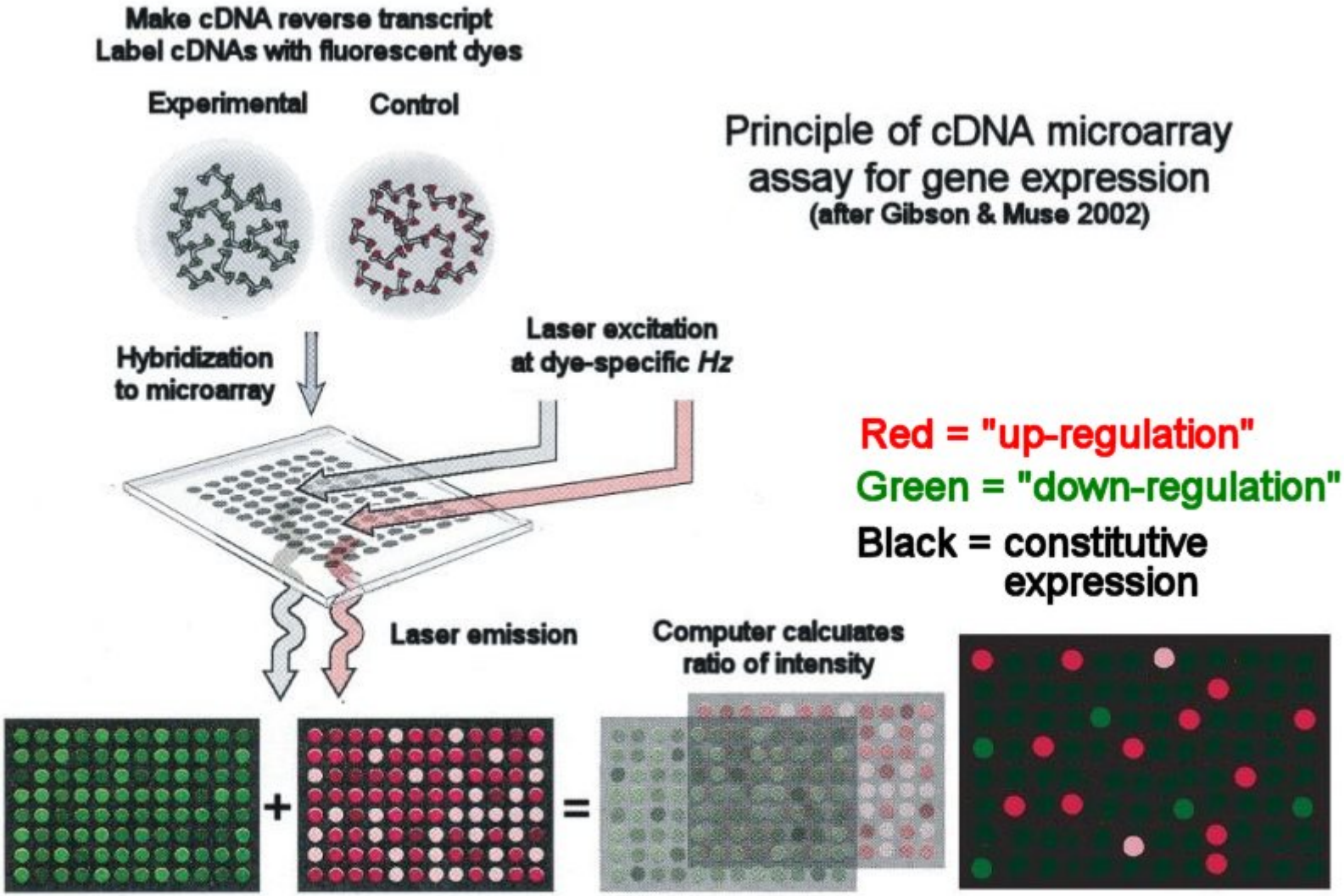




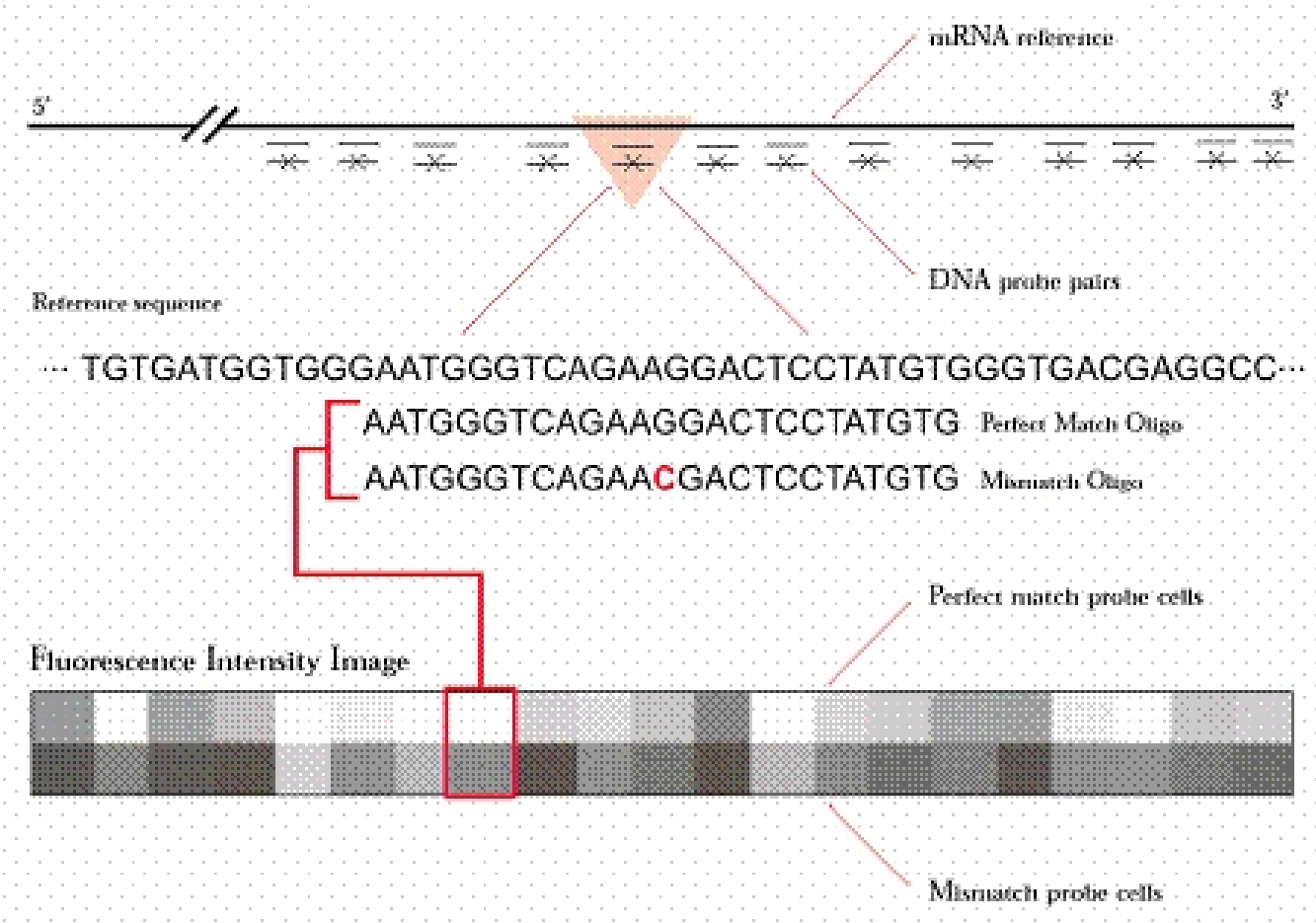
cDNA(complementary DNA)

A DNA molecule made in vitro using mRNA as a template and the enzyme reverse transcriptase. A cDNA molecule therefore corresponds to a gene, but lacks the introns present in the DNA of the genome.

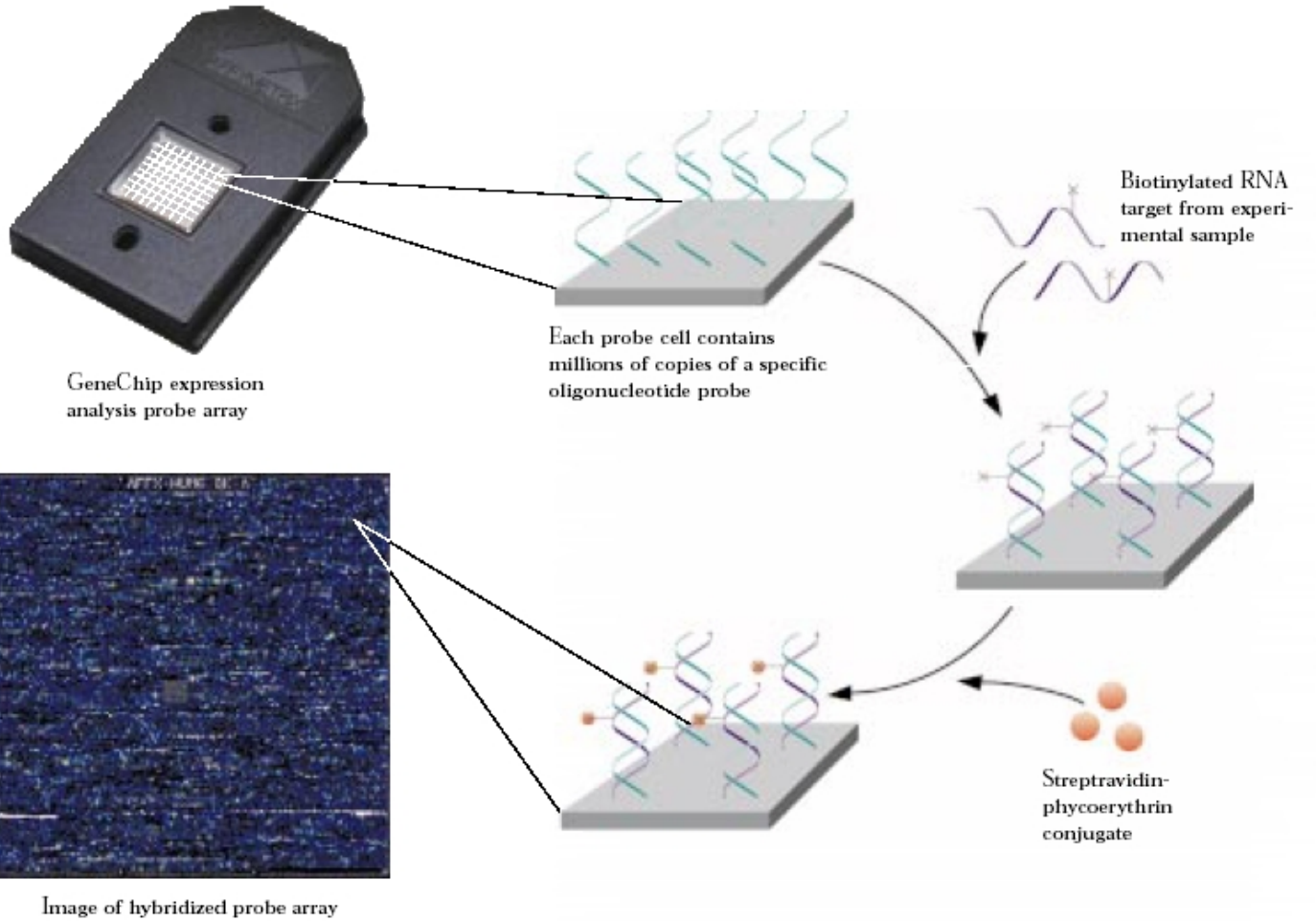
cDNA Microarrays



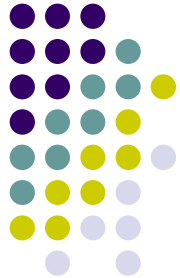
Affymetrix Microarrays



Affymetrix Microarrays



Microarray

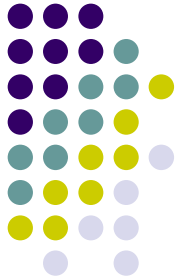


- Preprocessing
  - Background correction
  - Normalization
  - Summarization

Affimetrix : **affy**

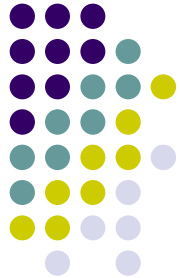
Two-channel cDNA : **marray**

Microarray



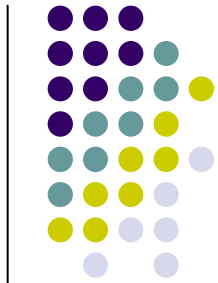
- **exprSet** : class for microarray data and methods for processing them
  - **exprs** : the observed expression levels.
  - **annotation** : character string identifying the annotation that may be used for the exprSet instance
  - **description**
  - **notes**
  - **phenoData** : containing the patient (or case) level data

Microarray



- Analysis
  - Differential expression
  - Graph and Networks
  - Clustering
  - Classification
  - Multiple comparison
  - Time course
  - Sequence matching

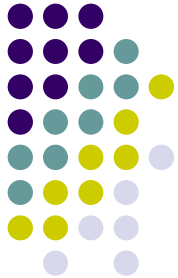
Installing R and Bioconductor



- R 2.4.0
  - Download from CRAN
    http://cran.r-project.org
- Bioconductor 1.9
  - Download from Bioconductor website or
    http://www.bioconductor.org
  - From R

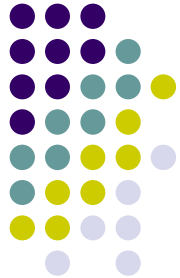
```
> source(http://www.bioconductor.org/getBioC.R)
> getBioC()
```

Starting and quitting R



- Start : R command
- Quit : **`q()`**
- Save : save current env. With **`save.image`**
- Working directory : **`getwd, setwd`**
- List objects : **`ls, objects`**
- Remove objects : **`rm, remove`**
- Search path : **`search, attach, detach`**
- Help **`: help(), ?`**

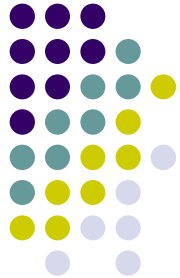
R package structure



- A structured collection of code, documentation and/or data
- files : DESCRIPTION, INDEX
- subdirectories
  - R
  - man
  - doc
  - src, data, demo, exec, inst

  * **`package.skeleton`**

How to make R package

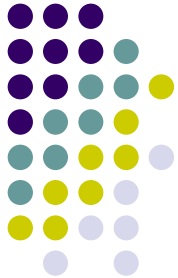


- Linux/Unix
  - **`R CMD check`**
  - **`R CMD build`**

- Windows (need to install a couple of SW)
  - **`Rcmd check`**
  - **`Rcmd build`**
  - **`Rcmd build --binary`**

  * http://cran.r-project.org/doc/contrib/cross-build.pdf

How to install and load R package



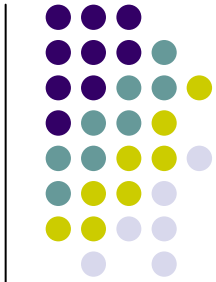
INSTALL

- Linux/Unix
  - **`R CMD INSTALL ---.tar.gz`**
- Windows
  - click

LOAD

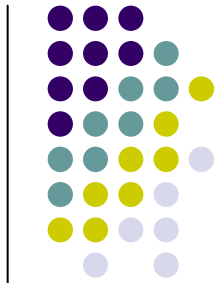
- **`library(package.name)`**
- **`packageDescription()`**
- **`.find.package()`**
- **`system.file()`**

Vignettes



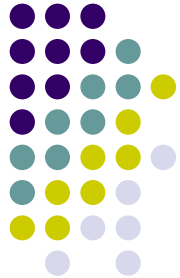
- new documentation paradigm
- an executable document consisting of a collection of code chunks and documentation text chunks
- provide dynamic, integrated, and reproducible statistical documents that can be automatically updated if either data or analyses are changed
- Vignettes can be generated using the “sweave” function from the R utils package

Vignettes



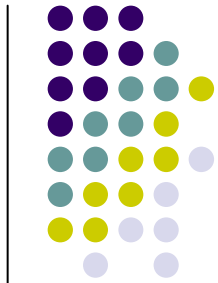
- Each Bioconductor package should contain at least one vignette, providing task-oriented descriptions of the package's functionality.
- located in the doc subdirectory of an installed package
- accessible from the help browser, via the help.start function
- available separately from the Bioconductor website

Vignettes



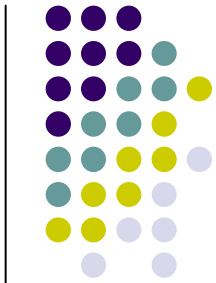
- **`Biobase`** package – **`openVignette`** function
  - menu of available vignettes and interface for viewing vignettes (PDF)
- **`tkWidgets`** package – **`vExplorer`** function
  - interactive use of vignettes, stepping through code chunks
- **`reposTools`** package

Sweave



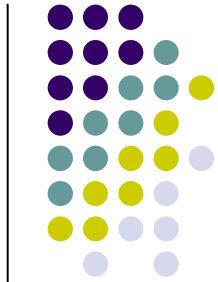
- allow the generation of dynamic, integrated and reproducible statistical documents, intermixing text, code, and code output(text and graphics)
- source file : an executable document consisting of a collection of code chunks and documentation text chunks.
- **`utils`** package : functions
- **`Sweave`**, **`Stangle`**

Sweave



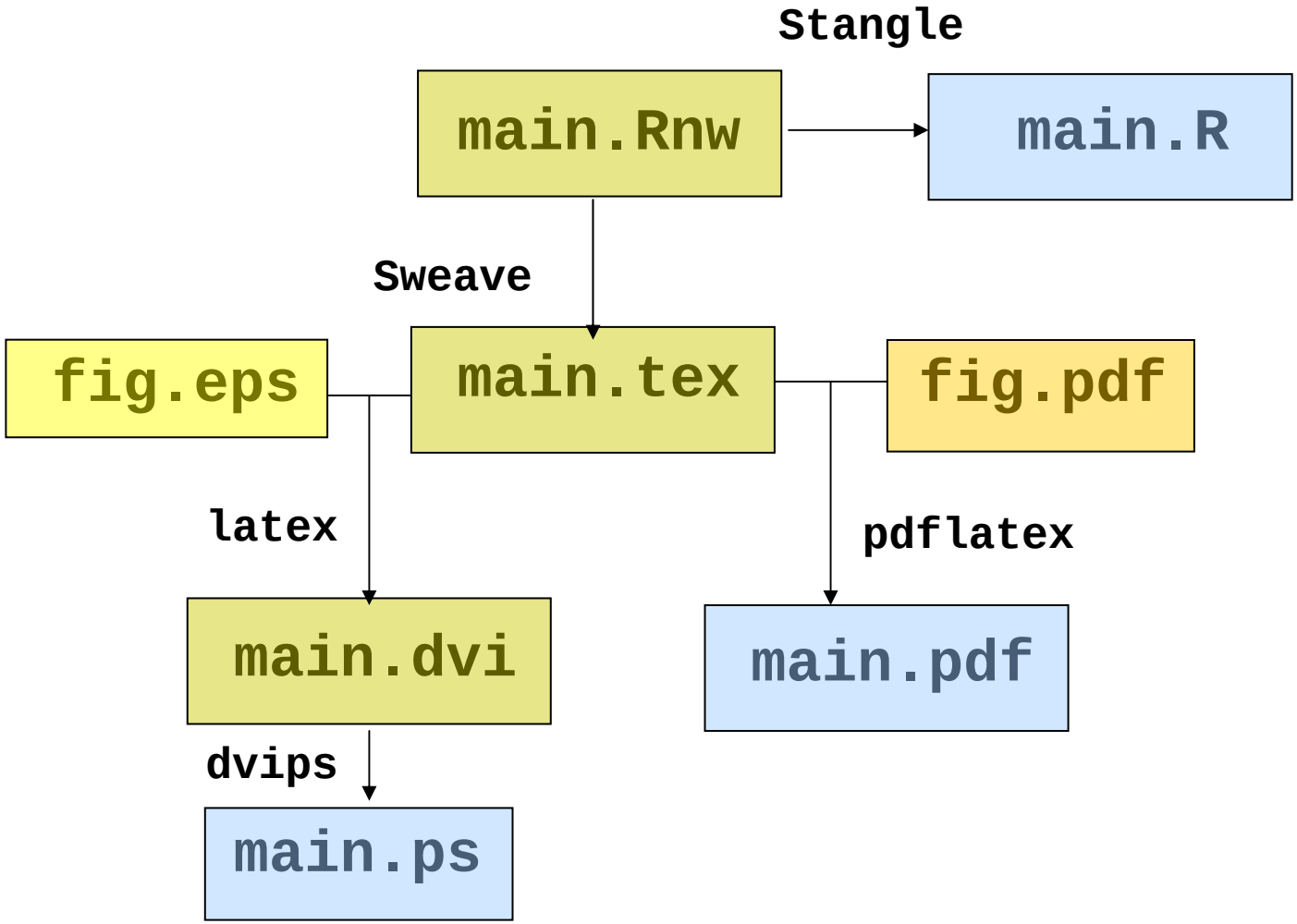
- Input (noweb file)
  - Documentation text chunks
    - start with @
    - text in a markup language like Latex
  - code chunks
    - start with <<name>>=
    - R code
  - file extension
    - .rnw, .Rnw, .snw, .Snw

Sweave

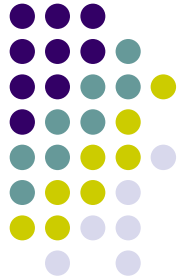


- **`Sweave`** function
  - extract the code chunks, run them and includes their output(text and graphs) in a .tex file and .ps or .PDF files
- **`Stangle`** function
  - concatenates all the code chunks into a .R file
- Output (.tex or .pdf)
  - a single document containing the documentation text, the R code, the code output (text and graphs)
  - automatically regenerated whenever the data, code or documentation text change

Sweave

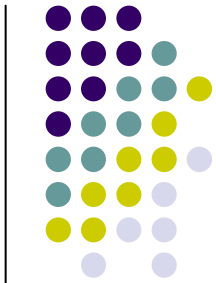

Stangle
main.Rnw
main.R
Sweave
fig.eps
main.tex
fig.pdf
latex
pdflatex
main.dvi
main.pdf
dvips
main.ps

Biocouductor SW design



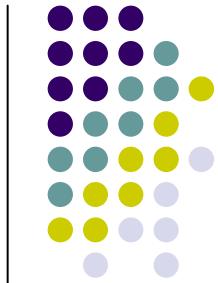
- programming approaches used in Biocouductor
  - Object-oriented S4 class/method framework
    - to deal with data complexity,
    - to represent and manipulate various data types
  - Environments
    - to provide mappings between different gene identifies in the annotation metadata packages
  - closures
    - for software modularity and extensibility

Experimental Metadata



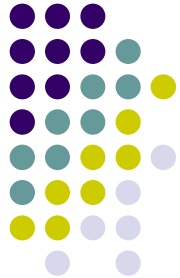
- gene expression measures
  - scanned image (TIFF)
  - image quantitation data (.gpr or .CEL)
  - normalized gene * array matrices of expression measures (log ratios or summary measures)
- reliability/quality information
- probe sequence information
- information on the target samples hybridized to the arrays : clinical covariates, experimental condition, etc.

Experimental Metadata



- Standard form
  - MIAME : minimum information about a microarray experiment
  - MAGE-ML : microarray gene expression

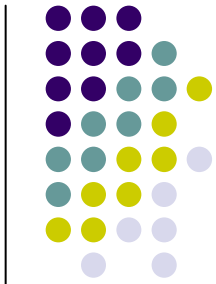
Annotation Metadata



- Biological attributes that can be applied to the experimental data
- for gene,
  - chromosome location
  - gene annotation (LocusLink, GO)
  - relative literature (PubMed)
- Biological metadata sets are large, of different types, evolving rapidly, and typically distributed via the WWW

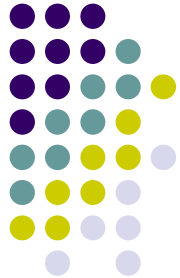
  `annotate`, **`annaffy`**, **`AnnBuilder`**

Data complexity



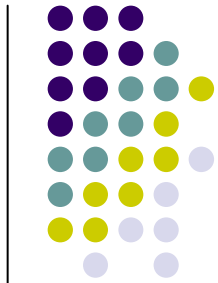
- large p, small n
- dynamic/evolving data
- multiple data source : WWW, in-house
- multiple data type
  - quantitative
  - qualitative
  - text, graphical
  - image, sound
- censored, missing, erroneous data
- various levels of processing

Object-Oriented Programming



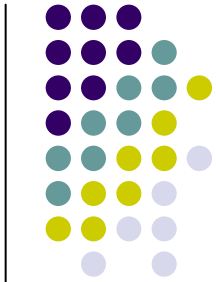
- adapt OOP paradigm in order to deal with the complexity of experimental and annotation metadata
- S4 class/method design allows efficient and reliable representation and manipulation of large and complex biological datasets of multiple types
- advantages of class/method design
  - keep all relevant information in one object
  - print, summary, accessor/assignment, subsetting, and more specialized methods
- Tools for programming using S4 : methods package

OOP : Classes



- provide a software abstraction of a real world object.
- It reflects how we think of certain objects and what information these objects should contain
- 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 and Documentation



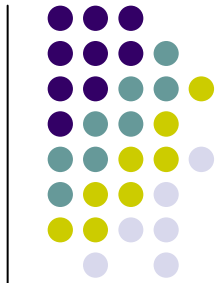
Methods

- function that performs an action on data
- define how a particular function should behave depending on the class of its arguments
- allow computations to be adapted to particular data types

Documentation

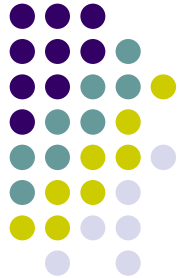
- special commands can be used to provide and access documentation for S4 classes and methods, using the type ? topic syntax.
- Methods available for a particular class are listed in the class help file

OOP : **exprSet** class



- defined in the **Biobase** package
- used to represent processed expression measures from either Affymetrix or two-color spotted microarrays
- slots
  - **exprs** : matrix of expression measures
  - **se.exprs** : matrix of SEs for expression measures
  - **phenoData** : sample level covariates and responses
  - **description** : MIAME information
  - **annotation** : name of annotation data
  - **notes** : any notes

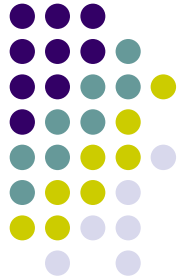
OOP : **phenoData** class



- defined in the **Biobase** package
- used to keep track of information on target samples hybridized to the microarray
  - **varLabels** : list of variable labels
  - **pData** : dataframe of sample level variables

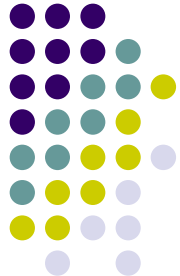
    arrays * variables

affy : Affymetrix Oligonucleotide chips



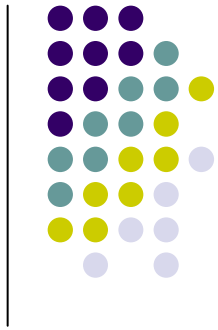
- **affy** package
  - class definitions for probe-level data and basic methods for manipulating microarray objects (printing, plotting, subsetting, class conversions, etc.)
  - **AffyBatch** : probe-level intensity data for a batch of arrays
  - **ProbeSet** : PM, MM intensities for individual probe-sets

marray: two-channel spotted microarrays



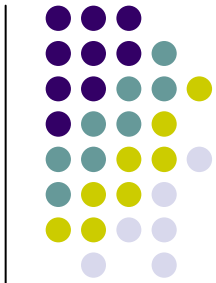
- **marray** package
  - class definitions for two-color spotted DNA microarray data and basic methods for manipulating microarray objects
  - **marrayLayout** : information on microarry layout
  - **marrayRaw** : pre-normalization intensity data for a batch of arrays (same layout)
  - **marrayNorm** : post-normalization intensity data for a batch of arrays.

pubMedAbst class



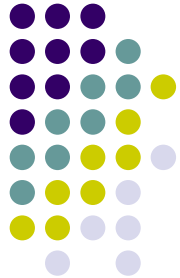
- **annotate** package
  - provide a **pubMedAbst** class for storing PubMed abstracts
  - **pmid**
  - **authors**
  - **abstText**
  - **articleTitle**
  - **journal**
  - **pubDate**

Annotation : matching IDs



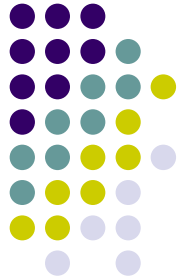
- Accessing annotation information from databases such as GeneBank, GO, or PubMed, presupposes the ability to perform the following essential bookkeeping task
- mapping between the different identifiers (IDs) for a given gene.
  - one GENENAME
  - one GenBank accession number
  - several different GO term IDs
  - several different PubMed IDs

R Environments



- provide key-value mappings
- similar to hash tables in other languages
- The term key refers to the name of variable, which can have different values in different environments.
- functions for working with environments include
  - **`ls`**
  - **`get`**, **`mget`**(base), **`multiget`**(Biobase)
  - **`assign`**(base), **`multiassign`**(Biobase)

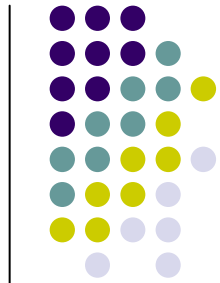
R Environments



- keys can be accessed using
  - **`ls(name of the environment)`**

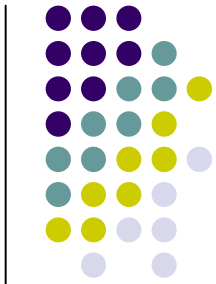
- Values can be accessed using
  - **`get(key, envir=name of the environment)`**
  - **`mget(keys, envir=name of the environment)`**

Closures



- consists of the body of the function along with an enclosing environment containing all variable bindings needed for evaluating the function.
- Closures facilitate software modularity and extensibility

Summary



- Overview of the Biocondcutor Project
- Microarray
- R package structure
- Vignette
- Sweave
- Bioconductor software design

- Next session : focusing on Microarray data analysis