Microarray analysis with Bioconductor (R)
Outline

- Introduction to R and Bioconductor
- Analysis of two-color microarrays
- Analysis of single-color microarrays (Affymetrix)
- Advanced analysis of gene expression data
R environment

R version 2.6.0 (2007-10-09)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R ist freie Software und kommt OHNE JEGELICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.

Tippen Sie 'contributors()' oder 'citation()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beiträgern.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML-Browsereinschalttafel zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> load("H:\Computation_Biology\bioconductor\R.RData")
> marray

No documentation for 'marray' in specified packages and libraries:
you could try 'help.search("marray")'
> library(marray)
Lade nütziges Paket: limma

R:Computation_Biology\celsius_example.R R Editor

celsius=new("celsiusServer", celsiusURL="http://celsius.genomics.otd.ucla.edu/experimentation\makeExperiment(celsius, id="SN:1655595"), cver="c("TPATH:458")")
cvterm=c("q="adipo")
Installing Bioconductor (I)

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

- Installs the following packages:
  
  - affy, affydata, affyPLM, annaffy, annotate, Biobase, Biostrings, DynDoc, gcrma, genefilter, geneplotter, hgu95av2, limma, marray, matchprobes, multtest, reposTools, ROC, vsn, xtable

- Arguments
  - `destdir`: the directory where the downloaded packages will be stored.
  - `lib`: character vector giving the library directories under which packages may be installed. Recycled as needed.
  - `pkgs`: character vector of Bioconductor packages to install.

> biocLite("pamr")
Installing Bioconductor (II)

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

- Installs the default packages. For a different subset of packages:
  - "all": all packages (very large set!)
  - "affy" (affy, affycomp, affydata, affyPLM, annaffy, gcrma, makecdfenv, matchprobes, marray)
  - "default" (similar to biocLite())
  - "graph": graph, Rgraphviz, RBGL

- Other arguments
  - **destdir**: the directory where the downloaded packages will be stored
  - **lib**: character vector giving the library directories under which packages may be installed.
  - **pkg**: character vector of Bioconductor packages to install.
Installed Packages

• Show installed packages

    > library()

• Pre-installed Bioconductor packages:

    affy, affydata, affyPLM, affyQCReport, annaffy, annotate, Biobase, Biostrings, DynDoc, gcrma, genefilter, geneplotter, GOstats, hgu95av2, limma, marray, matchprobes, multtest, pamr, qvalue, ROC, topGO, vsn, xtable.

• Other Bioconductor and CRAN packages have to be installed at the users home directory (L:)
Bioconductor packages: pre-processing

- **affy, affycomp, affydata, affypdnn, affyPLM, gcrma, makecdfenv**: Diagnostic plots, expression measures, and normalization for Affymetrix chip data.

- **annaffy**: Functions for handling data from Bioconductor Affymetrix annotation data packages. Produces compact HTML and text reports including experimental data and URL links to many online databases.

- **marray**: Diagnostic plots and normalization for cDNA microarray data.

- **matchprobes**: Tools for sequence matching of probes on arrays

- **vsn**: Calibration and variance stabilizing transformations for both Affymetrix and cDNA array data.
Bioconductor packages: Analysis (I)

- **daMA**: Efficient design of factorial two-color microarray experiments and for the statistical analysis of factorial microarray data.

- **edd**: Expression density diagnostics: graphical methods and pattern recognition algorithms for distribution shape classification.

- **factDesign**: Analysis of data from factorial designed microarray experiments. Single outlier detection.

- **Genefilter**: Sequentially filtering genes using a wide variety of filtering functions (number of missing value, coefficient of variation of expression measures, ANOVA p-value, Cox model p-values).

- **globaltest**: Testing globally whether a group of genes is significantly related to some clinical variable of interest.
**Bioconductor packages: Analysis (II)**

- **gpls**: Classification using generalized partial least squares for two-group and multi-group
- **limma**: Linear models for microarray data
- **RMAGEML**: Used to handle MAGE-ML documents in Bioconductor
- **MeasurementError.cor**: Two-stage measurement error model for correlation estimation with smaller bias than the usual sample correlation
- **multtest**: Multiple testing procedures for controlling the family-wise error rate (FWER) and the false discovery rate (FDR).
- **pamr**: sample classification in microarrays
- **ROC**: Receiver Operating Characteristic (ROC) approach for identifying genes differentially expressed
- **siggenes**: Significance Analysis of Microarrays and the Empirical Bayes Analyses of Microarrays
- **splicegear**: A set of tools to work with alternative splicing
Bioconductor packages: Annotation

- **Annotate**: Associate experimental data in real time to biological metadata from web databases such as GenBank, LocusLink and PubMed. Process and store query results. Generate HTML reports of analyses.

- **AnnBuilder**: Assemble and process genomic annotation data, from databases such as GenBank, the Gene Ontology Consortium, LocusLink, UniGene, the UCSC Human Genome Project.

- **Data packages**: XML and R annotation data packages, providing mappings between different probe identifiers (e.g. Affy IDs, LocusLink, PubMed)

- **Resourcer**: This package allows user either to read an annotation data file from TIGR Resourcerer as a matrix or convert the file into a Bioconductor annotation data package using the AnnBuilder package.
Bioconductor packages: Ontologies

- **GOstats**: A set of tools for interacting with GO and microarray data. A variety of basic manipulation tools for graphs, hypothesis testing and other simple calculations.

- **goTools**: Wraper functions for description/comparison of oligo ID list using Gene Ontology database

- **ontoTools**: Tools for working with ontologies
Bioconductor packages

Database Interaction

- **Rdbi**: Generic framework for database access in R
- **RdbiPgSQL**: Provides methods for accessing data stored in PostgresSQL
- **SAGElyzer**: Locates genes based on SAGE tags

Proteomics

- **gpls**: Classification using generalized partial least squares for two-group and multi-group (more than 2 group) classification.
- **PROcess**: A package for processing protein mass spectrometry data.
- **apComplex**: This package contains functions to estimate a bipartite graph representing protein complex membership using data from AP-MS technology.
Bioconductor packages

Graphics

- **affylmGUI**: A Graphical User Interface for affy analysis using limma.
- **limmaGUI**: A Graphical User Interface for the limma package
- **geneplotter**: Graphical tools for genomic data, for example for plotting expression data along a chromosome or producing color images of expression data matrices.
- **hexbin**: Binning functions, in particular hexagonal bins for graphing.
- **tkWidgets**: Widgets in Tcl/Tk that provide functionality for Bioconductor packages.
- **webbioc**: An integrated web interface for doing microarray analysis using several of the Bioconductor packages. (Currently only Affymetrix oligonucleotide analysis is supported.)
- **widgetTools**: Tools for creating Tcl/Tk widgets, i.e., small-scale graphical user interfaces.
- **graph**: Classes and tools for creating and manipulating graphs within R.
- **RBGL**: A package that creates an interface between the graph package and the Boost graph libraries, allowing for fast manipulation of graph objects in R.
- **Rgraphviz**: Provides an interface with Graphviz for plotting graph objects in R.
- **SNAData**: Data from Wasserman & Faust (1999) "Social Network Analysis"
Some special features

- **Widgets**: Small, interactive, graphical devices to drive the analysis (user-friendly)

- **Vignette**: `openVignette()`
Differentiation of 3T3-L1 cell line

- **Preconfluent**
- + 12 hours
- + 3 days
- 0 day
- + 24 hours
- + 7 days
- + 6 hours
- + 2 days
- + 14 days
- + 7 days
- + 14 days
Microarray experiment

Reference (preconfluent)  Timepoint after induction

Isolate and pool total RNA

Label with fluorescent dyes

2 hybridizations with dye swap
Microarray Image

Hybridization:
Timepoint 7d: Cy5
Reference: Cy3

Scanparameter:
635nm: PMT=760V
532nm: PMT=610V
Power=100%
Resolution=10µm
Image analysis
### Result files (.gpr)

<table>
<thead>
<tr>
<th>Block</th>
<th>Column</th>
<th>Row</th>
<th>Name</th>
<th>ID</th>
<th>X</th>
<th>Y</th>
<th>Dia</th>
<th>F635 Med</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>At.LTP4</td>
<td>AF158801</td>
<td>2390</td>
<td>2460</td>
<td>130</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td>Ubiquinone</td>
<td>A836668</td>
<td>2560</td>
<td>2460</td>
<td>100</td>
<td>473</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>KIAA0184</td>
<td>A839716</td>
<td>2760</td>
<td>2460</td>
<td>110</td>
<td>1059</td>
<td></td>
</tr>
</tbody>
</table>
Analytical Pipeline
marray

- **marrayClasses**
  - Definition two-color microarray data objects
  - Methods for manipulation of these objects

- **marrayInput**
  - Reading-in intensity data (GenePix, Spot)
  - Widgets available

- **marrayPlots**
  - Diagnostic plots

- **marrayNorm**
marrayRaw class structure
Commands

● Get information on classes

  > openVignette()
  > getClassDef("marrayRaw")
  > showMethods(classes = "marrayLayout")

● Example data set swirl

  > data (swirl)
  > swirl
  > summary(swirl)
> summary(swirl)
Pre-normalization intensity data:  Object of class arrayRaw.

Number of arrays:  4 arrays.

A) Layout of spots on the array:
Array layout:  Object of class arrayLayout.
Total number of spots:  8448
Dimensions of grid matrix:  4 rows by 4 cols
Dimensions of spot matrices:  22 rows by 24 cols
Currently working with a subset of 8448 spots.

Control spots:
There are  2 types of controls:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7680</td>
<td>768</td>
</tr>
</tbody>
</table>

Notes on layout:
No Input File

B) Samples hybridized to the array:
Object of class arrayInfo.

<table>
<thead>
<tr>
<th>mlab</th>
<th>Names slide number experiment Cyt</th>
<th>experiment Cyt</th>
<th>date comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 swirl.1.spot swirl.1.spot</td>
<td>81 swirl wild type 2001/9/20</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>2 swirl.2.spot swirl.2.spot</td>
<td>82 wild type swirl 2001/9/20</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>3 swirl.3.spot swirl.3.spot</td>
<td>93 swirl wild type 2001/11/8</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>4 swirl.4.spot swirl.4.spot</td>
<td>94 wild type swirl 2001/11/8</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

Number of labels:  4
Dimensions of mInfo matrix:  4 rows by 5 columns

Notes:
C:/GNU/R/R-2.4.1/library/marray/swirldata/SlwirlSample.txt

C) Summary statistics for log-ratio distribution:

<table>
<thead>
<tr>
<th></th>
<th>Min. 1st Qu. Median Mean 3rd Qu. Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:/GNU/R/R-2.4.1/library/marray/swirldata/swirl.1.spot</td>
<td>4.73 -0.79 -0.58 -0.48 -0.29 4.42</td>
</tr>
<tr>
<td>C:/GNU/R/R-2.4.1/library/marray/swirldata/swirl.2.spot</td>
<td>-2.72 -0.15 0.03 0.03 0.21 2.35</td>
</tr>
<tr>
<td>C:/GNU/R/R-2.4.1/library/marray/swirldata/swirl.3.spot</td>
<td>-2.33 -0.75 -0.46 -0.42 0.42 2.65</td>
</tr>
<tr>
<td>C:/GNU/R/R-2.4.1/library/marray/swirldata/swirl.4.spot</td>
<td>-3.31 -0.48 -0.28 -0.27 -0.06 2.90</td>
</tr>
</tbody>
</table>

D) Notes on intensity data:
Spot Data
# Commands

- **Examples**

  ```
  > swirl[130:135,]  # info for spots 130-135 for all arrays
  > swirl[10:20,3]  # info for spots 10-20 for array 3
  > maRf(swirl[3:8,4])  # red (Cy5) foreground intensities
  > maGf(swirl[3:8,4])  # green (Cy3) foreground intensities
  > maRb(swirl[3:8,4])  # red (Cy5) background intensities
  > maGb(swirl[3:8,4])  # green (Cy3) background intensities
  > maLayout(swirl)  # Microarray layout
  > maGnames(swirl)[45:50]  # Gene names
  > maTargets(swirl)  # Sample description
  ```
read.marrayRaw package:marrayInput R Documentation

Create objects of class "marrayRaw"

Description:

This function reads in cDNA microarray data from a directory and creates objects of class "marrayRaw" from spot quantification data files obtained from image analysis software or databases.

Usage:

read.marrayRaw(fnames, path = ".", name.Gf, name.Gb=NULL, name.Rf, name.Rb=NULL, name.W=NULL, layout=NULL, gnames=NULL, targets=NULL, notes=NULL, skip=0, sep = " ", quote = "", ...)
marrayInput widget
Commands

● Examples

> swirl[130:135,]                # info for spots 130-135 for all arrays
> swirl[10:20,3]                  # info for spots 10-20 for array 3
> maRf(swirl[3:8,4])             # red (Cy5) foreground intensities
> maGf(swirl[3:8,4])             # green (Cy3) foreground intensities
> maRb(swirl[3:8,4])             # red (Cy5) background intensities
> maGb(swirl[3:8,4])             # green (Cy3) background intensities
> maLayout(swirl)                # Microarray layout
> maGnames(swirl)[45:50]         # Gene names
> maTargets(swirl)               # Sample description
Array quality assessment arrayQuality

> library("arrayQuality")
> maQualityPlots(swirl)
marrayNorm

- **maNorm()**: # within slide normalization

```
maNorm(mbatch, norm=c("printTipLoess", "none", "median", "loess", "twoD", "scalePrintTipMAD"), subset=TRUE, span=0.4, Mloc=TRUE, Mscale=TRUE, echo=FALSE, ...)
```

- mbatch (object of class marrayRaw)
- norm: normalization procedure
  - None: no normalization (n)
  - Median: global normalization (m)
  - Loess (l)
  - print-TipLoess (p)
  - scalePrintTipMAD (s)
- Subset: subset of points used for the norm.
- Span: smoothing function
- Mloc, Mscale: retain values or not
- echo: print index slide being normalized

Value: an object of class marrayNorm
marrayNorm

- **maNormScale()**  # between slide normalization

```r
maNormScale(mbatch, norm=c("globalMAD", "printTipMAD"), subset=TRUE, gec=TRUE, Mscale=TRUE, echo=FALSE)
```

- **maNormMain()**

```r
maNormMain(mbatch, f.loc=list(maNormLoess()), f.scale=NULL,
    a.loc=maCompNormEq(), a.scale=maCompNormEq(), Mloc=TRUE, Mscale=TRUE, echo=FALSE)
```
Normalizing the *swirl* experiment

\[
A = \frac{1}{2} \log_2 (R \times G) \\
M = \log_2 (R / G)
\]

> `swirl.norm <- maNorm(swirl, norm="p")`
> `summary(swirl.norm)`

> `maA(swirl.norm)[1:10,]`
> `maM(swirl.norm)[1:10,]`
> `boxplot(swirl.norm[,3])`
> `boxplot(swirl.norm)`
MA-plots of the *swirl* experiment
MA-plots before and after normalization
Boxplots
Data output

> write.marray(swirl)  # export normalized log-ratios (M) to text file oder Excel file

> library("convert")
> mdata<-as(swirl.norm, "exprSet")

Other packages for normalization of two-color microarrays

limma
vsn
Exercise

- Download the GenePix microarray data from the adipocyte study at http://icbi.at/bi/exercise2.html
- Perform a background correction and global median normalization
- Show boxplots and MA-plots before and after normalization
- Show xy-plot for log2-ratios of the dye-swap pairs
- Do a dye-swap normalization

\[
\text{normalized log}_2\text{-ratios } M_i = \hat{\mu}_i = \frac{1}{2} \left( \log_2 \frac{R_i}{G_i} + \log_2 \frac{G'_i}{R'_i} \right) = \log_2 \sqrt{\frac{R_i G'_i}{G_i R'_i}}
\]

- Provide a list of ESTs/genes, which were at least two-fold up or down-regulated in one of the two conditions (0h,7d)
Analysis of differentially expressed genes

- Cut-off (e.g. two-fold change)
- T-statistic, ANOVA
- Linear models for microarrays

  **Limma:**
  
  ```
  > fit <- lmFit(swirl, design=c(-1,1,-1,1))
  > fit <- eBayes(fit)
  > restable <- topTable(fit, n=30, adjust="fdr")
  > table2html(restable, disp="file")
  ```

- Significance analysis of microarrays (SAM)

  **Siggenes:**
  
  ```
  > library(siggenes)
  > sam()
  > sam.plot()
  ```
Affymetrix microarray analyses

- Read required packages
  ```r
circle6
library(affy); library(limma)
```

- Read targets (pheno data)
  ```r
circle6
targets <- ReadTargets("affy_targets.txt")
```

- Read .cel files
  ```r
circle6
data <- ReadAffy(filenames=targets$FileName)
```

- Normalization
  ```r
circle6
eset <- rma(data)
```

- List the file name and exports all expression values to text file.
  ```r
circle6
pData(eset); write.exprs(eset, file="affy_all.txt")
```

- Create appropriate design matrix
  ```r
circle6
design <- model.matrix(~ -1+factor(c(1,1,2,2,3,3)))
```

- Assign column names
  ```r
circle6
colnames(design) <- c("group1", "group2", "group3")
```
Affymetrix microarray analyses

- Fit a linear model for each gene based on the given series of arrays.
  \[
  \text{fit} \leftarrow \text{lmFit(eset, design)}
  \]

- Create appropriate contrast matrix to perform all pairwise comparisons.
  \[
  \text{contrast.matrix} \leftarrow \text{makeContrasts(group2-group1, group3-group2, group3-group1, levels=design)}
  \]

- Compute estimated coefficients and standard errors for a given set of contrasts.
  \[
  \text{fit2} \leftarrow \text{contrasts.fit(fit, contrast.matrix)}
  \]

- Compute moderated t-statistics and log-odds of differential expression by empirical Bayes
  \[
  \text{fit2} \leftarrow \text{eBayes(fit2)}
  \]

- Generates list of top 10 ('number=10') differentially expressed genes sorted by B-values ('sort.by=B') for first comparison group
  \[
  \text{topTable(fit2, coef=1, adjust="fdr", sort.by="B", number=10)}
  \]
Affymetrix microarray analyses

- Export complete limma statistics table for first comparison group.

```r
write.table(topTable(fit2, coef=1, adjust="fdr", sort.by="B", number=500), file="limma_complete.xls", row.names=F, sep="\t")
```

- Create venn diagram of all D.E. genes at the level of 0.05.

```r
results <- decideTests(fit2, p.value=0.05)
vennDiagram(results)

x <- topTable(fit2, coef=1, adjust="fdr", sort.by="P", number=50000)
y <- x[x$P.Value < 0.05,]
y <- x[x$P.Value < 0.01 & (x$logFC > 1 | x$logFC < -1) & x$AveExpr > 10,]
print("Number of genes in this list:")
length(y$ID)
```
Some useful packages for Genomic Data

- Gene Ontology (GO) analysis: GOstats; goCluster
- Chromosome maps: geneplotter
- Phylogenetic Analysis: ape
- Protein Structure Analysis: Bio3D
- Motif identification in promoter regions: COSMO
Hierarchical Clustering

- The basic hierarchical clustering functions in R are `hclust()` from the `stats` package, and `agnes()` and `diana()` from the `cluster` package.

- Compute distances between the rows of `x`.
  ```r
  Dist(x, method = "correlation") (amap)
  d1 <- Dist(t(mm), method = "correlation");
  hr <- hclust(d, method = "complete", members=NULL)
  ```

- The generated tree can be plotted with the `plot()` function or `cutplot.dendrogram` (Heatplus)

- Prints dendrogram structure as text.
  ```r
  str(as.dendrogram(hr))
  ```