Introduction To Bioconductor

Robert Gentleman
Sandrine Dudoit
Denise Scholtens

Computational and Statistical Aspects of Microarray Analysis
University of Milan
May 26-30, 2003

© Copyright 2003, all rights reserved
Bioconductor Basics

- Bioconductor (www.bioconductor.org) is a software project aimed at providing high quality, innovative software tools appropriate for computational biology
- We rely mainly on R (www.r-project.org) as the computational basis
- We welcome contributions
Some basics

• for microarray data analysis we have assembled a number of R packages that are appropriate to the different types of data and processing

• some issues:
  – data complexity
  – data size
  – data evolution
  – meta-data
Software Design

• to overcome complexity we use two strategies: Abstract Data Types and object oriented programming
• to deal with data evolution we have separated the biological meta-data from the experimental data
Pedagogy

- among the many choices we made in the Bioconductor project is to try and develop better teaching materials
- in large part this is because we are between two disciplines (Biology and Statistics) and most users are familiar with only one of these
Vignettes

• we have adopted a new type of documentation: the *vignette*

• a vignette is an integrated collection of text and code – the code is runnable and using Sweave it is possible to replace the code with its output

• these documents are short and explicit directions on how to perform specific tasks
Vignettes – HowTo’s

• a good way to find out how to use Bioconductor software is to read the relevant Vignette
• then extract the code (tangleToR) and examine it
• HowTo documents are shorter (one or two pages)
• please write and contribute these
Vignettes

• in Bioconductor 1.1 we introduced two new methods to interact with Vignettes
  • openVignette() – gives you a menu to select from
  • vExplorer() – our first attempt at turning Vignettes into interactive documents
Bioconductor packages
Release 1.1, Nov. 18, 2002

- General infrastructure:
  - Biobase, rhdf5, tkWidgets, reposTools.

- Annotation:
  - annotate, AnnBuilder \rightarrow data packages.

- Graphics:
  - geneplotter, hexbin.

- Pre-processing for Affymetrix oligonucleotide chip data:
  - affy, CDF packages, vsn.

- Pre-processing for cDNA microarray data:
  - marrayClasses, marrayInput, marrayNorm, marrayPlots, vsn.

- Differential gene expression:
  - edd, genefilter, multtest, ROC.
Outline

• Biobase and the basics

• annotate and AnnBuilder packages

• genefilter package

• multtest package

• R clustering and classification packages
**Biobase**: `exprSet` class

- **exprs**: Matrix of expression measures, genes x samples
- **se.exprs**: Matrix of SEs for expression measures
- **phenoData**: Sample level covariates, instance of class `phenoData`
- **annotation**: Name of annotation data
- **description**: Object of class MIAME
- **notes**: Any notes
Typing the name of the data set produces this output

> golubTest

Expression Set (exprSet) with
7129 genes
34 samples

phenoData object with 11 variables and 34 cases

varLabels
Samples: Sample index
ALL.AML: Factor, indicating ALL or AML
BM.PB: Factor, sample from marrow or peripheral blood
T.B.cell: Factor, T cell or B cell leuk.
FAB: Factor, FAB classification
Date: Date sample obtained
Gender: Factor, gender of patient
pctBlasts: pct of cells that are blasts
Treatment: response to treatment
PS: Prediction strength
Source: Source of sample
• the set is closed under subsetting operations (either $x[,1]$ or $x[1,]$) both produce new exprSets
• the first subscript is for genes, the second for samples
• the software is responsible for maintaining data integrity
exprSet: accessing the phenotypic data

- phenotypic data is stored in a special class: phenoData
- this is simply a dataframe and a set of associated labels describing the variables in the dataframe
Meta-data packages

• One of the largest challenges in analyzing genomic data is associating the experimental data with the available metadata, e.g. sequence, gene annotation, chromosomal maps, literature.

• The annotate and AnnBuilder packages provides some tools for carrying this out.

• These are very likely to change, evolve and improve, so please check the current documentation - things may already have changed!
Meta-data packages

• meta-data packages;
• Matching IDs using environments;
• Searching and processing queries from WWW databases
  – LocusLink,
  – GenBank,
  – PubMed;
• HTML reports.
WWW resources

- Nucleotide databases: e.g. GenBank.
- Gene databases: e.g. LocusLink, UniGene.
- Protein sequence and structure databases: e.g. SwissProt, Protein DataBank (PDB).
- Literature databases: e.g. PubMed, OMIM.
- Chromosome maps: e.g. NCBI Map Viewer.
- Pathways: e.g. KEGG.
- Entrez is a search and retrieval system that integrates information from databases at NCBI (National Center for Biotechnology Information).
NCBI Entrez

**annotate**: matching IDs

Important tasks

• Associate manufacturers probe identifiers (e.g. Affymetrix IDs) to other available identifiers (e.g. gene symbol, PubMed PMID, LocusLink LocusID, GenBank accession number).

• Associate probes with biological data such as chromosomal position, pathways.

• Associate probes with published literature data via PubMed.
<table>
<thead>
<tr>
<th><strong>annotate:</strong> matching IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Affymetrix identifier</strong></td>
</tr>
<tr>
<td><strong>HGU95A chips</strong></td>
</tr>
<tr>
<td><strong>LocusLink, LocusID</strong></td>
</tr>
<tr>
<td><strong>GenBank accession #</strong></td>
</tr>
<tr>
<td><strong>Gene symbol</strong></td>
</tr>
<tr>
<td><strong>PubMed, PMID</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Chromosomal location</strong></td>
</tr>
</tbody>
</table>
Annotation data packages

- The Bioconductor project has started to deploy packages that contain only data. E.g. `hgu95av2` package for Affymetrix HGU95A GeneChips series, also, `hgu133a`, `hu6800`, `mgu74a`, `rgu34a`.

- These data packages are built using `AnnBuilder`.

- These packages contain many different mappings to interesting data.

- They are available from the Bioconductor website and also using `update.packages2`. 
Meta-data packages

- Maps to GenBank accession number, LocusLink LocusID, gene symbol, gene name, UniGene cluster.
- Maps to chromosomal location: chromosome, cytoband, physical distance (bp), orientation.
- Maps to KEGG pathways, enzymes, Gene Ontology Consortium (GO).
- Maps to PubMed PMID.
- These packages will be updated and expanded regularly as new or updated data become available.
### hu95av2 data package (1.2.6)

A data package containing annotation data for hu95av2

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hgu95av2</td>
<td>Genomic Annotation data package built with ArniBuilder</td>
</tr>
<tr>
<td>hgu95av2ACCCNUM</td>
<td>Annotation data file for ACCNUM in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2CHR</td>
<td>Annotation data file for CHR in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2CHRLOC</td>
<td>Annotation data file for CHRLOC in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2CHRORI</td>
<td>Annotation data file for CHIORI in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2ENZYME</td>
<td>Annotation data file for ENZYME in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2ENZYME2PROBE</td>
<td>Annotation data file for ENZYME2PROBE in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2GENENAME</td>
<td>Annotation data file for GENENAME in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2GO</td>
<td>Annotation data file for GO in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2GO2ALLPROBES</td>
<td>Annotation data file for GO2ALLPROBES in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2GO2PROBE</td>
<td>Annotation data file for GO2PROBE in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2GRIF</td>
<td>Annotation data file for GRIF in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2LOCUSID</td>
<td>Annotation data file for LOCUSID in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2MAP</td>
<td>Annotation data file for MAP in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2PATH</td>
<td>Annotation data file for PATH in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2PATH2PROBE</td>
<td>Annotation data file for PATH2PROBE in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2PMID</td>
<td>Annotation data file for PMID in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2PMID2PROBE</td>
<td>Annotation data file for PMID2PROBE in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2QC</td>
<td>Quality control information for hgu95av2</td>
</tr>
<tr>
<td>hgu95av2SUMFUNC</td>
<td>Annotation data file for SUMFUNC in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2SYMBOL</td>
<td>Annotation data file for SYMBOL in the hgu95av2 package</td>
</tr>
<tr>
<td>hgu95av2UNIGENE</td>
<td>Annotation data file for UNIGENE in the hgu95av2 package</td>
</tr>
</tbody>
</table>
**annotate**: matching IDs

- Much of what `annotate` does relies on matching symbols.
- This is basically the role of a hash table in most programming languages.
- In R, we rely on environments (they are similar to hash tables).
- The annotation data packages provide R environment objects containing key and value pairs for the mappings between two sets of probe identifiers.
- Keys can be accessed using the R `ls` function.
- Matching values in different environments can be accessed using the `get` or `multiget` functions.
annotate: matching IDs

E.g. hgu95av2 package.

- To load package `library(hgu95av2)`
- For info on the package and list of mappings available
  
  ```
  ? hgu95av2
  hgu95av2()
  ```

- For info on a particular mapping
  
  ```
  ? hgu95av2PMID
  ```
annotate: matching IDs

> library(hgu95av2)
> get("36823_at", env = hgu95av2ACCNUM)
[1] "AF055026"
> get("36823_at", env = hgu95av2LOCUSID)
[1] "10900"
> get("36823_at", env = hgu95av2SYMBOL)
[1] "RPIP8"
> get("36823_at", env = hgu95av2GENENAME)
[1] "RaP2 interacting protein 8"
> get("36823_at", env = hgu95av2SUMFUNC)
[1] "Rap2 interacting protein 8; interacts with the activated form of Rap2|Proteome"
> get("36823_at", env = hgu95av2UNIGENE)
[1] "Hs.6755"
**annotate**: matching IDs

```r
> get("36823_at", env = hgu95av2CHR)
[1] "17"
> get("36823_at", env = hgu95av2CHRLOC)
  17
 42396727
> get("36823_at", env = hgu95av2CHRORI)
  17
  "+"
> get("36823_at", env = hgu95av2MAP)
[1] "17q21.31"
> get("36823_at", env = hgu95av2PMID)
[1] "9523700" "9110174" "8619474"
> get("36823_at", env = hgu95av2GO)
  P            P
"GO:0005083"  "GO:0007264"
```
annotate: database searches and report generation

• Provide tools for searching and processing information from various biological databases.
• Provide tools for regular expression searching of PubMed abstracts.
• Provide nice HTML reports of analyses, with links to biological databases.
**annotate: WWW queries**

- Functions for querying WWW databases from R rely on the `browseURL` function

```r
browseURL("www.r-project.org")
```
annotate: GenBank query

- Given a vector of GenBank accession numbers or NCBI UIDs, the `genbank` function
  - opens a browser at the URLs for the corresponding GenBank queries;
  - returns an `XMLdoc` object with the same data.

`genbank("AF055026", disp="browser")`

`genbank("3005754", disp="data", type="uid")`
annotate: LocusLink query

www.ncbi.nlm.nih.gov/LocusLink/

- **locuslinkByID**: given one or more LocusIDs, the browser is opened at the URL corresponding to the first gene.

  ```
  locuslinkByID(“10900”)
  ```

- **locuslinkQuery**: given a search string, the results of the LocusLink query are displayed in the browser.

  ```
  locuslinkQuery(“Rap2”)
  http://www.ncbi.nih.gov/LocusLink/list.cgi?Q=Rap2&ORG=Hs&V=0
  ```
annotate: PubMed query


• For any gene there is often a large amount of data available from PubMed.
• The annotate package provides the following tools for interacting with PubMed
  – pubMedAbst: a class structure for PubMed abstracts in R.
  – pubmed: the basic engine for talking to PubMed.
• WARNING: be careful you can query them too much and be banned!
annotate: pubMedAbst class

Class structure for storing and processing PubMed abstracts in R

- authors
- pmid
- abstText
- articleTitle
- journal
- pubDate
- abstUrl
annotate: high level tools for PubMed query

- **pm.getabst**: download the specified PubMed abstracts (stored in XML) and create a list of `pubMedAbst` objects.
- **pm.titles**: extract the titles from a list of PubMed abstracts.
- **pm.abstGrep**: regular expression matching on the abstracts.
**annotate**: PubMed example

```r
pmid <- get("36823_at", env=hgu95av2PMID)
pubmed(pmidx, disp="browser")


absts <- pm.getabst("36823_at",
     base="hgu95av2")
pm.titles(absts)
pm.abstGrep("GTP",absts[[1]])
```
annotate: PubMed example

> pm.titles(absts)

[[1]]
(1) "Identification of a specific effector of the small GTP-binding protein Rho2."
(2) "Large-scale concatenation cDNA sequencing."
(3) "A "double adaptor" method for improved shotgun library construction."
annotate: data rendering

• A simple interface, **1l.htmlpage**, can be used to generate an HTML report of your results.

• The page consists of a table with one row per gene, with links to LocusLink.

• Entries can include various gene identifiers and statistics.
# BioConductor Gene Listing

Golub et al. data, genes with permutation maxT adjusted p-value < 0.01

## Locus Link Genes

<table>
<thead>
<tr>
<th>LocusID</th>
<th>Gene name</th>
<th>Chromosome</th>
<th>ALL mean</th>
<th>AML mean</th>
<th>t-statistic</th>
<th>raw p-value</th>
<th>adj p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3729</td>
<td>X96725_at</td>
<td>7</td>
<td>0.295</td>
<td>1.59</td>
<td>-0.6</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>1471</td>
<td>M27391_at</td>
<td>20</td>
<td>1.81</td>
<td>2.08</td>
<td>-9.78</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>3184</td>
<td>M53330_at</td>
<td>15</td>
<td>0.488</td>
<td>1.24</td>
<td>8.03</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>4029</td>
<td>M10039_at</td>
<td>8</td>
<td>2.28</td>
<td>1.1</td>
<td>9.98</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5334</td>
<td>L06206_s_at</td>
<td>11</td>
<td>1.62</td>
<td>1.36</td>
<td>7.97</td>
<td>2e-05</td>
<td>2e-04</td>
</tr>
<tr>
<td>6929</td>
<td>M16139_at</td>
<td>19</td>
<td>1.852</td>
<td>1.391</td>
<td>7.53</td>
<td>2e-05</td>
<td>2e-04</td>
</tr>
<tr>
<td>9206</td>
<td>X74262_at</td>
<td>2</td>
<td>0.669</td>
<td>0.565</td>
<td>7.42</td>
<td>2e-05</td>
<td>0.00076</td>
</tr>
<tr>
<td>6155</td>
<td>Z15313_at</td>
<td>3</td>
<td>1.84</td>
<td>1.945</td>
<td>7.38</td>
<td>2e-05</td>
<td>0.001</td>
</tr>
<tr>
<td>56929</td>
<td>L47739_at</td>
<td>5</td>
<td>0.784</td>
<td>0.779</td>
<td>1.21</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>4922</td>
<td>U2276_a_at</td>
<td>6</td>
<td>1.86</td>
<td>0.294</td>
<td>7.28</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>6512</td>
<td>HG1612-H16112_at</td>
<td>1</td>
<td>1.81</td>
<td>0.889</td>
<td>7.11</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>21</td>
<td>H49142_at</td>
<td>1</td>
<td>0.431</td>
<td>0.771</td>
<td>7.08</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5923</td>
<td>L41871_at</td>
<td>13</td>
<td>0.438</td>
<td>1.3</td>
<td>7.08</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>648</td>
<td>U73968_at</td>
<td>NA</td>
<td>0.097</td>
<td>1.07</td>
<td>7.07</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>7230</td>
<td>X1521_at</td>
<td>5</td>
<td>1.82</td>
<td>1.07</td>
<td>7.06</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>4036</td>
<td>U30136_s_at</td>
<td>5</td>
<td>0.71</td>
<td>1.51</td>
<td>6.97</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5971</td>
<td>Y12070_at</td>
<td>1</td>
<td>0.157</td>
<td>0.892</td>
<td>6.96</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>7293</td>
<td>X74601_at</td>
<td>1</td>
<td>0.611</td>
<td>0.183</td>
<td>6.95</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5936</td>
<td>M00769_at</td>
<td>4</td>
<td>0.371</td>
<td>0.52</td>
<td>6.87</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>6729</td>
<td>D5S243_at</td>
<td>9</td>
<td>0.413</td>
<td>0.982</td>
<td>6.68</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>1723</td>
<td>U26266_at</td>
<td>10</td>
<td>0.209</td>
<td>1.16</td>
<td>6.55</td>
<td>4e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5305</td>
<td>U82799_at</td>
<td>7</td>
<td>0.64</td>
<td>0.504</td>
<td>6.82</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>5331</td>
<td>M23971_at</td>
<td>19</td>
<td>0.981</td>
<td>0.554</td>
<td>6.79</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>1559</td>
<td>M18353_at</td>
<td>11</td>
<td>1.21</td>
<td>2.12</td>
<td>6.77</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>6953</td>
<td>M19399_at</td>
<td>14</td>
<td>1.13</td>
<td>1.32</td>
<td>6.76</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>957</td>
<td>X62654_at</td>
<td>12</td>
<td>0.513</td>
<td>1.33</td>
<td>6.66</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>3141</td>
<td>X07745_at</td>
<td>2</td>
<td>1.939</td>
<td>0.553</td>
<td>6.74</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>14245</td>
<td>M63121_at</td>
<td>12</td>
<td>0.106</td>
<td>0.953</td>
<td>6.71</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>7356</td>
<td>U02136_at</td>
<td>8</td>
<td>0.153</td>
<td>0.92</td>
<td>6.68</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>3366</td>
<td>X15946_at</td>
<td>4</td>
<td>1.935</td>
<td>1.33</td>
<td>6.61</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
<tr>
<td>9265</td>
<td>FMR224_at</td>
<td>14</td>
<td>0.626</td>
<td>0.305</td>
<td>4.54</td>
<td>2e-05</td>
<td>2e-05</td>
</tr>
</tbody>
</table>
annotate: chromLoc class

Location information for one gene

- **chrom**: chromosome name.
- **position**: starting position of the gene in bp.
- **strand**: chromosome strand +/-.
annotate: chromLocation class

Location information for a set of genes

- **organism**: organism that the genes correspond to.
- **dataSource**: source of the gene location data.
- **chromLocs**: location of the genes on the chromosome.
- **probesToChrom**: mapping from probes to chromosome.
- **chromInfo**: information about names, lengths etc.
- **geneSymbols**: mapping from probes to symbols.
geneplotter: cPlot
geneplotter: alongChrom
geneplotter: alongChrom
Gene filtering

• A very common task in microarray data analysis is gene-by-gene selection.
• Filter genes based on
  – data quality criteria, e.g. absolute intensity or variance;
  – subject matter knowledge;
  – their ability to differentiate cases from controls;
  – their spatial or temporal expression pattern.
• Depending on the experimental design, some highly specialized filters may be required and applied sequentially.
Gene filtering

• *Cohort/Clinical trial*. Filter genes based on association with survival, e.g. using a Cox model.

• *Factorial experiment*. Filter genes based on interaction between two treatments, e.g. using ANOVA.

• *Time-course experiment*. Filter genes based on periodicity of expression pattern, e.g. using Fourier transform.
The `genefilter` package provides tools to sequentially apply filters to the rows (genes) of a matrix.

There are two main functions, `filterfun` and `genefilter`, for assembling and applying the filters, respectively.

Any number of functions for specific filtering tasks can be defined and supplied to `filterfun`.

E.g. Cox model p-values, coefficient of variation.
**genefilter**: separation of tasks

1. Select/define functions for specific filtering tasks.
2. Assemble the filters using the `filterfun` function.
3. Apply the filters using the `genefilter` function → a logical vector, `TRUE` indicates genes that are retained.
4. Apply that vector to the `exprSet` to obtain a microarray object for the subset of interesting genes.
**genefilter**: supplied filters

Filters supplied in the package
- **kOverA** – select genes for which k samples have expression measures larger than A.
- **gapFilter** – select genes with a large IQR or gap (jump) in expression measures across samples.
- **ttest** – select genes according to t-test nominal p-values.
- **Anova** – select genes according to ANOVA nominal p-values.
- **coxfilter** – select genes according to Cox model nominal p-values.
**genefilter**: writing filters

- It is very simple to write your own filters.
- You can use the supplied filtering functions as templates.
- The basic idea is to rely on **lexical scope** to provide values (bindings) for the variables that are needed to do the filtering.
**genefilter: How to?**

1. First, build the filters
   ```r
   f1 <- anyNA
   f2 <- kOverA(5, 100)
   ```
2. Next, assemble them in a filtering function
   ```r
   ff <- filterfun(f1,f2)
   ```
3. Finally, apply the filter
   ```r
   wh <- genefilter(exprs(golubTest),
                    ff)
   ```
4. Use `wh` to obtain the relevant subset of the data
   ```r
   mySub <- golubTest[wh,]
   ```
Differential gene expression

- Identify genes whose expression levels are associated with a response or covariate of interest
  - clinical outcome such as survival, response to treatment, tumor class;
  - covariate such as treatment, dose, time.
- **Estimation**: estimate effects of interest and variability of these estimates.
  E.g. slope, interaction, or difference in means in a linear model.
- **Testing**: assess the statistical significance of the observed associations.
Acknowledgements

- **Bioconductor core team**
  - Ben Bolstad, Biostatistics, UC Berkeley
  - Vincent Carey, Biostatistics, Harvard
  - Francois Collin, GeneLogic
  - Leslie Cope, JHU
  - Laurent Gautier, Technical University of Denmark, Denmark
  - Yongchao Ge, Statistics, UC Berkeley
  - Robert Gentleman, Biostatistics, Harvard
  - Jeff Gentry, Dana-Farber Cancer Institute
  - John Ngai Lab, MCB, UC Berkeley
  - Juliet Shaffer, Statistics, UC Berkeley
  - Terry Speed, Statistics, UC Berkeley
  - Yee Hwa (Jean) Yang, Biostatistics, UCSF
  - Jianhua (John) Zhang, Dana-Farber Cancer Institute
- Spike-in and dilution datasets:
  - Gene Brown’s group, Wyeth/Genetics Institute
  - Uwe Scherf’s group, Genomics Research & Development, GeneLogic.
- GeneLogic and Affymetrix for permission to use their data.