Kollekolle 2003

- R/bioconductor overview
- VJ Carey <stvjc@channing.harvard.edu>
  - motivations/infrastructure
  - substantive efforts
  - illustrations of working with R
  - prelude to the lab
Why?

• statistical genomics and statistical computing generally need progress in
  • transparency of analyses
  • interoperability of implementations
  • awareness of and compliance with standards
  • reduced barriers to entry:
    • statisticians implement their ideas for use in the broadest conceivable context
    • biologists use state of the art inference tools no matter how obscure the experimental platform
A database map
If connected to web:

Predicament

- expanding diversity of technologies
- expanding volume of experimental results
- much commonality of scientific constraints and aims
- instability of data resource owing to iterative refinement of genomic resources/knowledge
- need to borrow strength across groups, disciplines, methods
Infrastructure efforts

- software development model
- software distribution
- integrated documentation
- GUI components; support for MIAME/MAGE standards
- formal (S4) classes and generics
- network structures and algorithms
Software development

• main portal: www.bioconductor.org

• Bioconductor core: statisticians, computer scientists, computational biologists with common commitment to software quality and transparency and methods supported by R

• independently verifiable (CMD check) components that interoperate

• CVS for concurrent version control
Software distribution

- *reposTools* copes with various branches, synchronization of versions, aim is to support decentralized repository protocol
- diminishes requirement for R expertise among users who need to use multiple packages
Documentation

- R’s packaging system based on stylized and unit-test equipped documentation
- Bioconductor has two new requirements
  - much work will involve several (perhaps many) packages
  - complex analyses need auditability/reproducibility support
- Sweave+Vignette+vExplorer address these issues
A Visit to vExplorer

- we will illustrate vExplorer in real time
GUI for MIAME compliance

- MIAME = minimum info about a uarray expt
GUI for MAGEML compliance

- MAGEML = markup language for MAGE-OM (object model)
Object-oriented programming

- Mainly for those interested in development
- To maximize efficiency of reuse and interoperation
  - formally specify classes of complex objects in terms of ‘slots’ and their classes (recursive)
  - generic functions like plot and show have specialized behaviors depending on class of instance supplied
- the end-user cares: much drudgery is hidden!
the exprSet class

- an early step in Bioc was formalization of a binding between expression array results and associated sample-level data (phenotype or treatment data)
- we don’t care about the underlying representation, but we have certain functional requirements
  - need to form arbitrary selections of genes and samples based on easily stated predicates
  - should be convenient to iterate over genes or samples
  - need rapid access to low-resolution metadata about the expression data under study
- exprSet is the source or target structure of a variety of processing tools in bioc: affy expresso, limma, vsn, so it is worthwhile to get familiar with it
Golub’s example data

- In a 1999 Science paper, Golub and colleagues described use of an Affymetrix™ chip to identify genomic signatures of two types of leukemia
  - ALL (acute lymphocytic leukemia)
  - AML (acute myelogenous leukemia)
- The analysis was performed in a split sample (training set/test set) design, and the expression values have been published on the web
an exprSet instance

> show(golubTrain)

Expression Set (exprSet) with
7129 genes
38 samples

phenoData object with 11 variables and 38 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
another instance

> set.seed(123)
> show(golubTrain[sample(1:7129, size = 100), ])

Expression Set (exprSet) with
  100 genes
  38 samples

  phenoData object with 11 variables and 38 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
phenotype data

```r
> print(table(dx <- pData(golubTrain)$ALL.AML))

ALL  AML
  27   11

> print(table(dx, pData(golubTrain)$T.B.cell))

dx    B-cell  T-cell
  ALL  19     8
  AML   0      0

> print(table(dx, pData(golubTrain)$Treatment))

dx   Failure  Success
  ALL   0      0
  AML   6      5
```
Idioms familiar in R have been reused in this application. If X is an instance of exprSet, then

- **X[v,]** is the restriction of X to genes identified by index v (which may be numeric, boolean, or character)
- **X[,w]** is restriction of X to samples identified by index w
- **pData(X)** accesses phenotype data only, in the form of a data.frame, and variables in the data.frame can be selected using the $ operator
Upshots

- the show method gives us the low-resolution summary
- the square brackets allow selection on the basis of arbitrary predicates
- the pData accessor gets access to the phenotype data component; the exprs accessor gets access to the expression data.
- Quiz: Create two exprSets based on the AML patients, one restricted to treatment successes, the other to treatment failures.
Solution

```r
> tx <- pData(golubTrain)$Treatment
> s1 <- golubTrain[, match(tx, "Success", 0)]
> s2 <- golubTrain[, match(tx, "Failure", 0)]
> print(dim(exprs(s1)))

[1] 7129  5

> print(dim(exprs(s2)))

[1] 7129  6

Why not tx == "Success"?
```
Benefits

- clear and familiar conventions allow useful reshaping of large complex data resource
- class is ‘closed under subsetting operations’
- guarantees: the class system checks that instance slots have appropriate class, and other validation methods can be invoked at appropriate times
  - author of a method to process exprSets need not perform tests on any features of the exprSet that is formalized
  - user of exprSet can rely on the formalized structure
inter-project relations

• infrastructure costs can be reduced by partnering with other open source projects
  • www.omegahat.org for inter-systems interfaces, XML, SOAP
  • www.ggobi.org for linked dynamic graphics
  • www.graphviz.org for network visualization
network data infrastructure

- packages *graph*, *RBGL* and *Rgraphviz* (not yet for windows) address interface, algorithm internals and visualization of network structures
- we will work with these tomorrow in connection with ontology concepts
graph viz example

\[
\begin{align*}
&\text{GPCR} \rightarrow \text{PKC} \rightarrow \text{PKA} \rightarrow \text{Rap1} \rightarrow \text{B-Raf} \rightarrow \text{MEK1/2} \rightarrow \text{ERK1/2} \rightarrow \text{RSK1} \rightarrow \text{CREB} \rightarrow \text{Elk-1} \\
&\text{Integrins} \rightarrow \text{src} \rightarrow \text{shc} \rightarrow \text{Grb2} \rightarrow \text{Akt/PKB} \rightarrow \text{p70S6K} \rightarrow \text{B-Raf} \rightarrow \text{MEK1/2} \rightarrow \text{ERK1/2} \\
&\text{RTK} \rightarrow \text{PI-3Ks} \rightarrow \text{PIP3} \rightarrow \text{PDK} \rightarrow \text{Akt/PKB} \\
&\text{Fas} \rightarrow \text{Grb2} \rightarrow \text{Akt/PKB} \rightarrow \text{FADD} \\

&\text{Translation} \nonumber \\
&\text{Cyto C release/mitoch} \\
&\text{Apoptosis} \\
\end{align*}
\]
network agenda

- representations: adjacency lists, hash tables, databases, GXL documents
- fixed contracts on all representations to provide information on structure and to facilitate iteration driven by generic methods (e.g., depth-first-search, shortest path, connectivity)
- data can be exported to ggobi for interactive visualization or rendered in SVG for dynamic document performance
- arc-wise specification of complex graphical structures in RDF (resource description framework) as a basis for inference on genomic network structure (last day)
recap on infrastructure

- even in a fairly circumscribed scientific domain, computational infrastructure demands are considerable
- investments should pay off here and in other domains
  - package distribution
  - advanced documentation
  - working with the S4 system
- yet more is needed!
  - security (be careful of what you download!)
  - multiple event-driven GUI components
  - integrated dynamic graphics ...
Substantive contributions

- Biobase resources: binding expression and phenotype/protocol data
- pre- and post-processing of diverse microarray platforms: affy, cDNA (marray*, limma)
- other technologies (SAGElyzer; various proteomic platforms forthcoming)
- significant annotation resources/curation for statistical use; ontology use
- standards: MIAME/MAGE
- various visualizations for genomics
- remaining lectures will confront a fraction of these activities
Preparing for the lab

- the lab includes basic overview of R primitives
- getting familiar with exprSet and its utilities for subsetting
- getting conversant with probeid to nomenclature facilities
- a primitive selection by ranking approach using R manually
- we will do some statistical estimation and testing both for gene-gene relations and gene-phenotype relations
- use a bioc:www facility to get literature on a discovered gene