Bioconductor for high-throughput genomic analysis

Martin Morgan (mtmorgan@fhcrc.org)
Fred Hutchinson Cancer Research Center
Seattle, WA, USA

February 19, 2010

Acknowledgments

- Robert Gentleman (*R*, *Bioconductor*) Vince Carey, Rafael Irizzary, Wolfgang Huber
- Patrick Aboyoun, Marc Carlson, Jerry Davison, Seth Falcon, Nishant Gopalakrishnan, Hervé Pagès, Chao-Jen Wong, Zizhen Yao
- Tony Chiang, Brig Mecham, Noah Hoffman

A short history

S: An environment for quantitative computation and visualization.

- Late 1970s; John Chambers and colleagues at Bell Labs.
- Ideas from *awk*, *lisp*, *APL*, . . .
- ‘A breath of fresh air’ (paraphrasing).

R: A language ‘not unlike S’.

- R an independent open source version.
- Originally: Ross Ihaka, Robert Gentleman at University of Auckland. Now: *R* core.
- CRAN: contributed package repository.

Why success? Open development; early converts – domain experts; visionary.

R

- Interpreted, dynamic; ‘vectorized’.
- Copy-on-change semantics; implicit memory management.
    - Friendly to non-programmers.
- Column-oriented – data-intensive task.

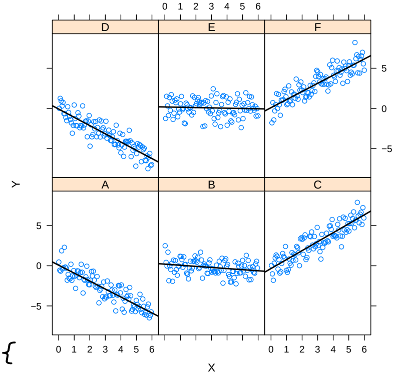
```
> x0 <- (1:600)/100
> x1 <- x0 * c(-1, 0, 1)
> df <- data.frame(X = x0, Y = x1 + rnorm(length(x0)),
+     Group = LETTERS[1:6])
> search()

[1] ".GlobalEnv"         "package:stats"
[3] "package:graphics"   "package:grDevices"
[5] "package:utils"      "package:datasets"
[7] "package:methods"    "Autoloads"
[9] "package:base"
```

Uses

- Applied statistical analysis
- Visualization – e.g., *lattice*, *ggplot2*
- Domain-specific analysis
  - Econometrics, finance
  - High-throughput biological assays: *Bioconductor*
- Academic statistics

```
> library(lattice)
> xyplot( Y ~ X | Group, df,
+   panel=function(x, y, ...) {
+       panel.xyplot(x, y, ...)
+       panel.lmline(x, y, lwd=2)
+   })
```



Bioconductor

- Focus
  - Expression and other microarray; flow cytometry.
  - High-throughput sequencing.
- Themes
  - Open source – algorithms are complicated and nuanced, there is often no ‘correct’ implementation.
  - Code reuse – *R* statistics and visualization; domain-specific applications, e.g., *limma*.
  - Interoperable – data reuse, e.g., *biomaRt*, *GEOquery*, *rtracklayer*.
  - Reproducible – objects self-describing; complex work flows captured in *vignettes*; data bundled with analyses in *R* packages.
- Success: > 350 packages; $> 50,000$ unique IP downloads per year; very active mailing list; conferences and courses.

Microarrays

Technology

- Short (25-60) DNA nucleotide ‘probes’ attached to surface.
- Hybridize processed, florescent cDNA.
- Measure florescence intensity.

Biological questions

- Originally: expression, e.g., in ‘cancer’ vs. ‘normal’ tissue across 30k genes.
- Copy number variation, methylation, single nucleotide polymorphism.

Overall work flow.

1. Experimental design.
2. Technology preparation & assay.
3. Pre-processing.
4. Statistical analysis.

Analysis work flows (psuedo-code)

```
> library(affy)
> phenoData <-
+   read.AnnotatedDataFrame(
+     "sample-descr.csv")
> eset <-
+   justRMA("/celfile-dir",
+     phenoData=phenoData)
> library(limma)
> design <-
+   model.matrix(~ Disease,
+      pData(eset))
> fit <- lmFit(eset, design)
> efit <- eBayes(fit)
> topTable(efit)
```

1. Quality Assessment.
2. Pre-processing: background correct; normalize; summarize.
3. Explore & visualize
4. Differential expression
    - Gene-centric
5. Gene set enrichment / pathways / . . .

Object representation: *ExpressionSet*

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 12625 features, 128 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 01005, 01010, ..., LAL4  (128 total)
  varLabels and varMetadata description:
    cod:  Patient ID
    diagnosis:  Date of diagnosis
    ...: ...
    date last seen:  date patient was last seen
    (21 total)
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 14684422 16243790
Annotation: hgu95av2
```

Short read: context

Technology.

- Many short (80-500bp) DNA fragments.
- Amplified (current) or single-molecule (tomorrow) sequencing.

Biological questions.

- ChIP-seq; SNP discovery; digital gene expression; metagenomics; RNA-seq; de novo assembly.

Overall process – Illumina Genome Analyzer II.

1. Biological preparation, e.g., ChIP.
2. ‘Sequencing‘: library preparation, cluster generation, sequencing. 20M reads / lane, 8 lanes / flow cell.
3. Primary analysis: alignment, quality assessment.
4. Domain-specific analysis.

Bioconductor tools

Data representation and manipulation

- *IRanges*: range-based calculations, infrastructure, ...
- *Biostrings*: string manipulation, pattern matching, ...
- *ShortRead*: I/O, quality assessment; *Rsamtools*: I/O ...
- *rtracklayer*: browser integration; *GenomicFeatures*: transcript-level annotation.
- *BSgenome*: genome-scale data representations

Analysis

- *chipseq*, *ChIPseqR*, *CSAR*, *ChIPsim*, *ChIPseqAnno*.
- *edgeR*, *baySeq*, *DEGseq* *DESeq*.
- *Genominator*

Case study: digital gene expression

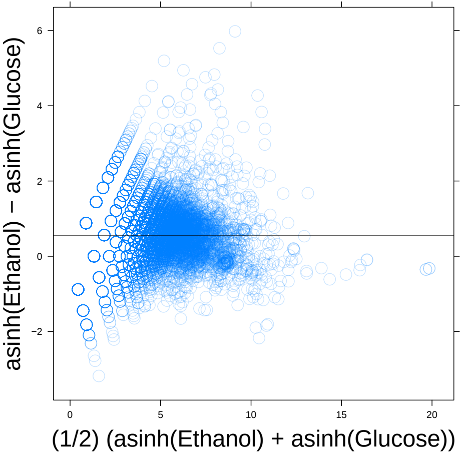
- Bloom et al., 2009: two strains of yeast under two different growth conditions – factorial experiment, though no replication
- Parallels previous microarray differential expression study, Smith & Kruglyak, 2008.
- Early ‘Solexa’ experiments; short (32bp) and not too many (4-5M) reads per sample.
- Original analysis required many hand-crafted tools, e.g., finding reads overlapping genes.

We start by loading additional libraries

```
> library(ShortRead)
> library(org.Sc.sgd.db)
```

Analysis work flow

1. Quality assessment.
2. Alignment.
3. Counts per region of interest, e.g., gene coding sequence.
4. Differential expression.
5. Annotation.



Analysis in detail I

- Input

```
> aln <- readAligned(filePath, type = "Bowtie")
```

…some tidying, then…

- Regions of interest – also USCS, Biomart, …

```
> library(org.Sc.sgd.db)
> tbl <- merge(toTable(org.Sc.sgdCHRLOC),
+              toTable(org.Sc.sgdCHRLOCEND))

> ranges <-
+     with(tbl, IRanges(abs(start), abs(stop)))
> regions <- RangedData(ranges,
+     space=tbl[["Chromosome"]],
+     id=tbl[["systematic_name"]])
```

Analysis in detail II

- Counts

```
> query <- as(aln, "RangesList")
> qlen <- sapply(query, length)
> olaps <- findOverlaps(query, regions)
> counts <- tabulate(subjectHits(olaps), qlen)
```

- Annotation

```
> anno <- org.Sc.sgdDESCRIPTION[["YNL117W"]]
> noquote(strwrap(anno, 40))
```

```
[1] Malate synthase, enzyme of the
[2] glyoxylate cycle, involved in
[3] utilization of non-fermentable carbon
[4] sources; expression is subject to
[5] carbon catabolite repression; localizes
[6] in peroxisomes during growth in oleic
[7] acid medium
```

Rigor

- Differential expression as linear model
- Appropriate error model (*edgeR*: Poisson; *DESeq*: negative binomial); ‘borrowing’ information across regions.
- ‘Dependent’ variable is estimated (alignments) rather than given (probes)
- Poorly characterized contributions to error
  - Amplification bias, e.g., coverage in GC-rich regions
  - Base calls: position- and sequence-dependent
  - Alignment: ‘mappable genome’

Case study: human microbiomes

Experiment

- 16S rRNA bacterial sequences sampled from individuals with and without bacterial vaginosis over a (short) time series.
- Roche / 454 sequences – 100's of thousands of 200-300bp,
- Biological samples PCR amplified, bar-coded.

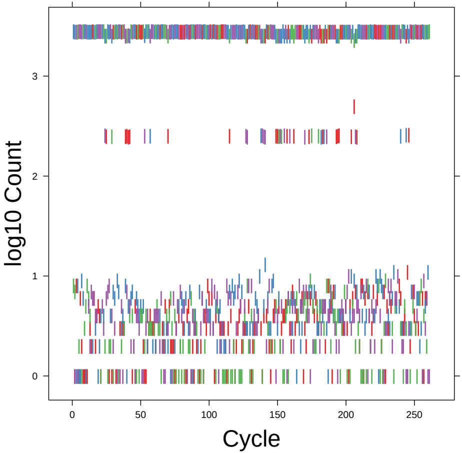
Analysis work flow

Analysis: pre-processing

1. Bin by bar code
2. Remove PCR primers
3. Remove low quality reads

Subsequent

- Phylogenetic placement (*pplacer*)
- Community composition change over time.



Analysis in detail

1. Input (valid code, when appropriate input available)

```
> bar <- read454(filePath, "1.*fna", "1.*qual")
```

2. Group by bar code, trim bar code (and 3 trailing nucleotides)

```
> code <- narrow(sread(bar), 1, 8)
> aBar <- bar[code == "AAGCGCTT"]
> noBar <- narrow(aBar, 11, width(aBar))
```

3. Remove PCR primer

```
> pcrPrimer <- "GGACTACCVGGGTATCTAAT"
> trimmed <-
+     trimLRPatterns(pcrPrimer, noBar,
+         Lfixed=FALSE)
```

Rigor

- Error model, e.g., indel PCR artifacts
- Phylogenetic placement
- Multivariate analysis – time series, count data, uncertain assignment
- Greatly facilitated by *R* functions and additional packages..

Reflections

Reproducibility

- Scripting, package structure, versioned software, common data structures all facilitate reproducible research.

Object representation

- *ExpressionSet* coordinates data in a reproducible way.
- *AnnDbBimap* accessibly re-interprets SQL. Trade-off between 'current' and reproducible annotations.
- *RangedData* shifts attention from gene-centric to coordinate-centric queries.

Knowledge as data base

- Traditional resources, e.g., ENSEMBL
- Experiment repositories, e.g., GEO, ArrayExpress.
- Consortium studies, e.g., HapMap, TCGA, 1000 genomes.

Opportunities & challenges

Integrative analysis: *Bioconductor* strength

- Pre-processing (e.g., RMA) and domain-specific analysis.
- Annotation & data base access.
- Statistical integration.

Range-based algorithms.

- Fine structure, e.g. transcripts
- Regulatory elements

Graph representations over diverse scales

- Transcript assembly
- Copy number variants
- Whole genome ‘reference set’

Academic research and the edge of ignorance

Resources

Bioconductor: `http://bioconductor.org`
Package installation

```
source("http://bioconductor.org/biocLite.R")
biocLite()              # core packages
biocLite('ShortRead') # specific package
```

References

- Bloom et al., 2009. BMC Genomics 10:221.
- Smith & Kruglyak, 2008. PLOS Biology 6:e83.
- Hahne et al., 2009. Bioconductor Case Studies, Springer.
- Gentleman, 2009, R Programming for Bioinformatics, Chapman & Hall.