Bioconductor for Sequence Analysis

Martin T. Morgan[1]

27-28 February 2014

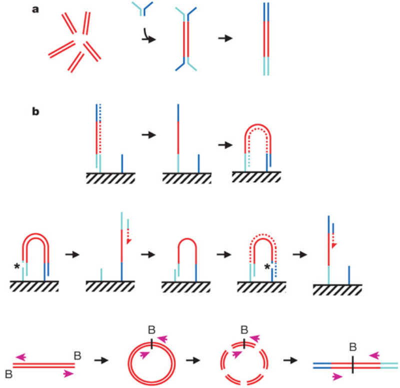
[1] mtmorgan@fhcrc.org

Introduction: What is *Bioconductor* good for?

- Sequencing: RNA-seq, ChIP-seq, called variants, . . .
  - Especially *after* assembly / alignment
- Annotation: genes, pathways, gene models (exons, transcripts, etc.), . . .
- Microarrays: expression, copy number, SNPs, methylation, . . .
- Flow cytometry, proteomics, image analysis, high-throughput screens, . . .

Sequencing: Work flows

1. Experimental design
2. 'Wet lab' sample prep
3. Sequencing
    - 100's of millions of reads
    - 30-150 nucleotides
    - Single and paired-end
    - Bar codes, lanes & flow cells
4. Alignment
5. Analysis: DNA, RNA, epigenetics, integrative, microbiome, . . .



Bentley et al., 2008, Nature 456: 53-9

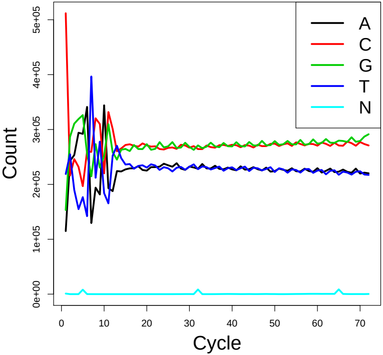
```
@ERR127302.1703 HWI-EAS350_0441:1:1:1460:19184#0/1
CCTGAGTGAAGCTGATCTTGATCTACGAAGAGAGATAGATCTTGATCGTCGAGGAGATGCTGACCTTGACCT
+
HHGHHGHHHHHHHHDGG<GDGGE@GDGGD<?B8??ADAD<BE@EE8EGDGA3CB85*,77@>>CE?=896=:
@ERR127302.1704 HWI-EAS350_0441:1:1:1460:16861#0/1
GCGGTATGCTGGAAGGTGCTCGAATGGAGAGCGCCAGCGCCCCGGCGCTGAGCCGCAGCCTCAGGTCCGCCC
+
DE?DD>ED4>EEE>DE8EEEDE8B?EB<@3;BA79?,881B?@73;1?########################
@ERR127302.1705 HWI-EAS350_0441:1:1:1460:13054#0/1
AAAACACCCTGCAATCTTTCAGACAGGATGTTGACAATGCGTCTCTGGCACGTCTTGACCTTGAACGCAAAG
+
EEDEE>AD>BBGGB8E8EEEGBGGGGBGGGGG3G>E3*?BE??BBC8GB8??:??GGDGDDD>D>B<GDDC8
@ERR127302.1706 HWI-EAS350_0441:1:1:1460:14924#0/1
CACCCAGTGGGGTGGAGTCGGAGCCACTGGTCCTGCTGCTGGCTGCCTCTCTGCTCCACCTTGTGACCCAGG
+
HHHHHGEEGEEADDGDBG>GGD8EG,<6<?AGGADFEHHC@>D@<@G@>AB@B?8AA>CE@D8@B=?CC>AG
@ERR127302.1707 HWI-EAS350_0441:1:1:1461:6983#0/1
CGACGCTGACACCGGAACGGCAGCAGCAGCAGGACGATTAAGACAAGGAGGATGGCTCCACAGACGCTCATG
+
GEEGEGE@GGGGGGEGGGGGBB>G3?33?8*;;79?<9@?DD8@DDEE888;-BB?.A##############
@ERR127302.1708 HWI-EAS350_0441:1:1:1461:10827#0/1
AAAGAAGGTCCTTGCAATAGACTGCCTCTGCTTGAGAACTTATGATGTAATTATTGCATGCTGCTAATATAC
+
GGGGGDDEBFGGGGGBE,DAGDDGGGEEEG<EEFDECFFEEEDE@<>ACEBEFDEEFE<EDC@E<EECCBEB
@ERR127302.1709 HWI-EAS350_0441:1:1:1461:7837#0/1
CAGCCACAGAACCACGGCACGGAAGACATGAGGCAGCATGCTCACGAGAGAGGTGAGGGTCTCCCCTCCAGG
+
HHGHHHH>DH:@.7@49;88G8>G>DDG@D>D@G@GE>@DDBDDG<A82?######################
```

Sequencing: The *ShortRead* package

```r
## Use the 'ShortRead' package
library(ShortRead)
## Create an object to represent a sample from a file
sampler <- FastqSampler("ERR127302_1.fastq.gz")
## Apply a method to yield a random sample
fq <- yield(sampler)
## Access sequences of sampled reads using `sread()`
## Summarize nucleotide use by cycle
## 'abc' is a nucleotide x cycle matrix of counts
abc <- alphabetByCycle(sread(fq))
## Subset of interesting nucleotides
abc <- abc[c("A", "C", "G", "T", "N"),]
```

Sequencing: The *ShortRead* package

```
## Create a plot from a
## matrix
matplot(t(abc), type="l",
  lty=1, lwd=3,
  xlab="Cycle",
  ylab="Count",
  cex.lab=2)
## Add a legend
legend("topright",
  legend=rownames(abc),
  lty=1, lwd=3, col=1:5,
  cex=1.8)
```



Sequencing: Essential packages and classes

- *Biostrings* and *DNAStringSet*
- *GenomicAlignments* and *GAlignments*
- *GenomicRanges* and *GRanges*
- *GenomicFeatures* and *TranscriptDb*
- *VariantAnnotation* and *VCF*
- Input and output: *rtracklayer* (WIG, BED, etc.), *Rsamtools* (BAM), *ShortRead* (FASTQ) file input

Reads

Data Short reads and their qualities

Tasks Input, quality assessment, summary, trimming, . . .

Packages *ShortRead*, *Biostrings*

Functions

- `readFastq`, `FastqSampler`, `FasqtStreamer`.
- `qa`, `report`.
- `alphabetFrequency`, `alphabetByCycle`, `consensusMatrix`.
- `trimTails`, `trimLRPatterns`, `matchPDict`, . . .

Alignments

Data BAM files of aligned reads

Tasks Input, BAM file manipulation, pileups

Packages *GenomicAlignments*, *Rsamtools* (also: *GenomicRanges*)

Functions

- `readGAlignments`
- `BamFile`, `BamFileList`
- `scanBam`, `ScanBamParam` (select a subset of the BAM file)
- `asBam`, `sortBam`, `indexBam`, `mergeBam`, `filterBam`
- `BamSampler`, `applyPileups`

Ranges

Data Genomic coordinates to represent data (e.g., aligned reads) or annotation (e.g., gene models).

Tasks Input, counting, coverage, manipulation, . . .

Packages *GenomicRanges*, *IRanges*

Functions

- `readGAlignments`, `readGAlignmentsList`
- Many intra-, inter-, and between-range manipulating, e.g., `narrow`, `flank`, `shift`, `intersect`, `findOverlaps`, `countOverlaps`

Variants

Data VCF (Variant Call Format) file

Tasks Calling, input, summary, coding consequences

Packages *VariantTools* (linux only), *VariantAnnotation*, *ensemblVEP*

Functions

- `tallyVariants`
- `readVcf`, `locateVariants`, `predictCoding`
- Also: SIFT, PolyPhen data bases

Annotations

Data Gene symbols or other identifiers

Tasks Discover annotations associated with genes or symbols

Packages *AnnotationDbi* (*org.**, *GO.db*, . . .), *biomaRt*

Functions

- Discovery: `columns`, `keytype`, `keys`
- `select`, `merge`
- *biomaRt*: `listMarts`, `listDatasets`, `listAttributes`, `listFilters`, `getBM`

Features

Data Genomic coordinates

Tasks Group exons by transcript or gene; discover transcript / gene identifier mappings

Packages *GenomicFeatures* and *TxDb.** packages (also: *rtracklayer*)

Functions

- `exonsBy`, `cdsBy`, `transcriptsBy`
- `select` (see Annotations, below)
- `makeTranscriptDb*`

Genome annotations

Data
: FASTA, GTF, VCF, . . . from internet resources

Tasks
: Define regions of interests; incorporate known features (e.g., ENCODE marks, dbSNP variants) in work flows

Packages
: *AnnotationHub*

Functions
: - `AnnotationHub`, `filters`
  - `metadata`, `hub$<tab>`

Sequences

Data Whole-genome sequences

Tasks View sequences, match position weight matricies, match patterns

Packages *Biostrings*, *BSgenome*

Functions

- `available.genomes`
- `Hsapiens[["chr3"]]`, `getSeq`, `mask`
- `matchPWM`, `vcountPattern`, `...`
- `forgeBSgenomeDataPkg`

Import / export

Data Common text-based formats, `gff`, `wig`, `bed`; UCSC tracks

Tasks Import and export

Packages *rtracklayer*

Functions

- `import`, `export`
- `browserSession`, `genome`

And. . .

Data representation: *IRanges*, *GenomicRanges*, *GenomicFeatures*, *Biostrings*, *BSgenome*, *girafe*. Input / output: *ShortRead* (fastq), *Rsamtools* (bam), *rtracklayer* (gff, wig, bed), *VariantAnnotation* (vcf), *R453Plus1Toolbox* (454). Annotation: *GenomicFeatures*, *ChIPpeakAnno*, *VariantAnnotation*. Alignment: *Rsubread*, *Biostrings*. Visualization: *ggbio*, *Gviz*. Quality assessment: *qrqc*, *seqbias*, *ReQON*, *htSeqTools*, *TEQC*, *Rolexa*, *ShortRead*. RNA-seq: *BitSeq*, *cqn*, *cummeRbund*, *DESeq*, *DEXSeq*, *EDASeq*, *edgeR*, *gage*, *goseq*, *iASeq*, *tweeDEseq*. ChIP-seq, etc.: *BayesPeak*, *baySeq*, *ChIPpeakAnno*, *chipseq*, *ChIPseqR*, *ChIPsim*, *CSAR*, *DiffBind*, *MEDIPS*, *mosaics*, *NarrowPeaks*, *nucleR*, *PICS*, *PING*, *REDseq*, *Repitools*, *TSSi*. Motifs: *BCRANK*, *cosmo*, *cosmoGUI*, *MotIV*, *seqLogo*, *rGADEM*. 3C, etc.: *HiTC*, *r3Cseq*. Copy number: *cn.mops*, *CNAnorm*, *exomeCopy*, *seqmentSeq*. Microbiome: *phyloseq*, *DirichletMultinomial*, *clstutils*, *manta*, *mcaGUI*. Work flows: *ArrayExpressHTS*, *Genominator*, *easyRNASeq*, *oneChannelGUI*, *rnaSeqMap*. Database: *SRAdb*. . . .

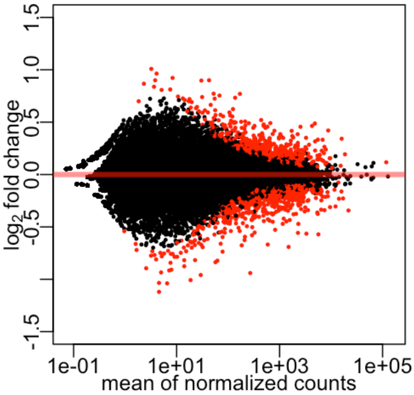
Exemplars: Algorithms to action

1. Batch effects
2. Methylation
3. RNA-seq Differential Representation
4. Visualization

Exemplar: Differential Representation

Haglund et al., 2012 J Clin Endocrin Metab

- Scientific finding: identify genes whose expression is regulated by estrogen receptors in parathyroid adenoma cells
- Statistical challenges: between-sample normalization; appropriate statistical model; efficient estimation; . . .



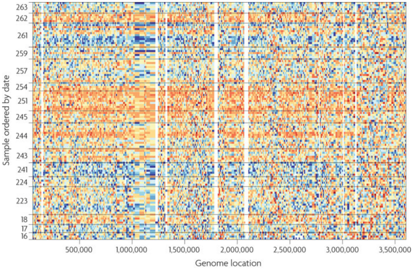
Bioconductor support: *DESeq2*, *edgeR*, many statistical 'lessons learned' from microarrays; extensive integration with down-stream tools

Exemplar: Batch Effects

Leek et al., 2010, Nature Reviews Genetics 11, 733-739, Leek & Story PLoS Genet 3(9): e161

- Scientific finding: pervasive batch effects
- Statistical insights: surrogate variable analysis: identify and build surrogate variables; remove known batch effects
- Benefits: reduce dependence, stabilize error rate estimates, and improve reproducibility

Bioconductor support: *sva*



Nature Reviews | Genetics

HapMap samples from one facility, ordered by date of processing. From

Exemplar: Batch Effects

Leek et al., 2010, Nature Reviews Genetics 11, 733-739, Leek & Story PLoS Genet 3(9): e161

- Scientific finding: pervasive batch effects
- Statistical insights: surrogate variable analysis: identify and build surrogate variables; remove known batch effects
- Benefits: reduce dependence, stabilize error rate estimates, and improve reproducibility

Bioconductor support: *sva*

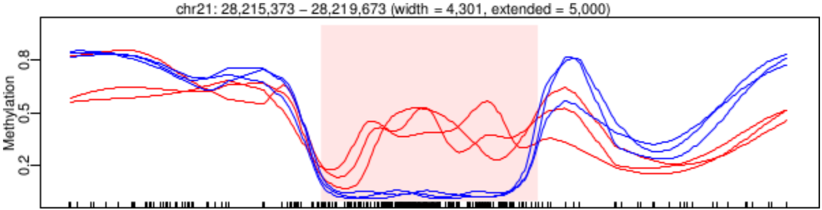
1. Remove signal due to variable(s) of interest
2. Identify subset of genes driving orthogonal signatures of EH
3. Build a surrogate variable based on full EH signature of that subset
4. Include significant surrogate variables as covariates

EH: expression heterogeneity

Exemplar: Methylation

Hansen et al., 2011, Nature Genetics 43, 768-775

- Scientific finding: stochastic methylation variation of cancer-specific de-methylated regions (DMR), distinguishing cancer from normal tissue, in several cancers.
- Statistical challenges: smoothing, non-specific filtering, *t* statistics, find DMRs

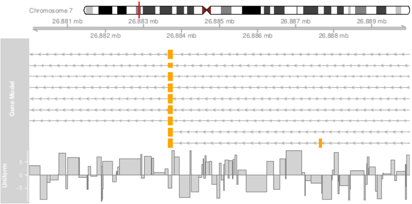


Bioconductor support: whole-genome (*bsseq*) or reduced representation (*MethylSeekR*) bisulfite sequencing; Illumina 450k arrays (*minfi*)

Exemplar: Visualization

Gviz

- Track-like visualizations
- Data panels
- Fully integrated with *Bioconductor* sequence representations



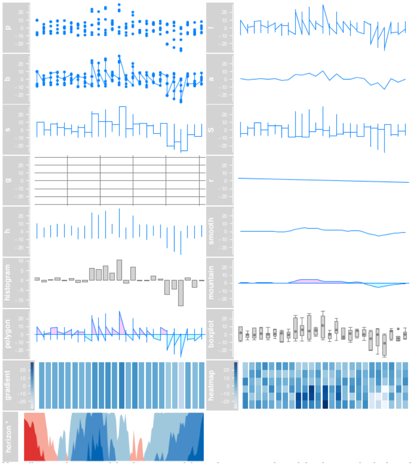
ggbio
epivizr

Exemplar: Visualization

Gviz

- Track-like visualizations
- Data panels
- Fully integrated with *Bioconductor* sequence representations

ggbio
epivizr



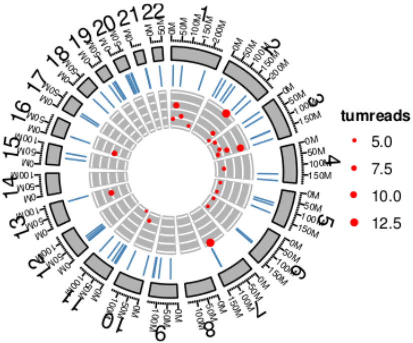
Exemplar: Visualization

Gviz

ggbio

- Comprehensive visualizations
- `autoplot` file and data types
- Fully integrated with *Bioconductor* sequence representations

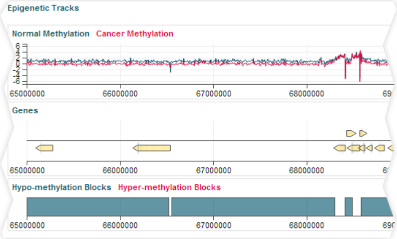
epivizr



Exemplar: Visualization

Gviz
ggbio
epivizr

- Genome browser with socket communication to *R*
- Fully integrated with *Bioconductor* sequence representations



Principles: Some key points

- *R* is a high-level programming language, so lots can be accomplished with just a little code
- Packages such as *ShortRead* provide a great way to benefit from the expertise of others (and to contribute your own expertise back to the community!)
  - The path from 'user' to 'developer' is not that long, and has been taken by many!
- Objects and methods such as *data.frame*, *ShortReadQ* and `alphabetByCycle()`) help to manage complicated data
  - Reducing possibility for clerical and other mistakes
  - Facilitating inter-operability between different parts of an analysis
- Scripts make work flows reproducible
- Visualizing data is an important part of exploratory analysis

Principles: Successful computational biology software

1. Extensive: software, annotation, integration
   - 750 inter-operable *Bioconductor* packages
2. Statistical: volume, technology, experimental design
   - *R* a 'natural' for statistical analysis
3. Reproducible: long-term, multi-participant science
   - Objects, scripts, vignettes, packages, . . . encourage reproducible research
4. Leading edge: novel, technology-driven
   - Packages and user community closely track leading edge science
5. Accessible: affordable, transparent, usable
   - *Bioconductor* is free and open, with extensive documentation and an active and supportive user community

Case study: differential expression of known genes; see also reproducible research lecture.

Challenges & Opportunities

- Big data – transparent management within *R*, facile use of established resources
- Developer and user training

Resources

- `http://r-project.org`, *An Introduction to R* manual; Dalgaard, *Introductory Statistics with R*; *R* for Dummies
- `http://bioconductor.org/`
- `http://rstudio.org`
- StackOverflow, *Bioconductor* mailing list