Introduction to Bioconductor:

Using R with high-throughput genomics

BaRC Hot Topics – Oct 2011

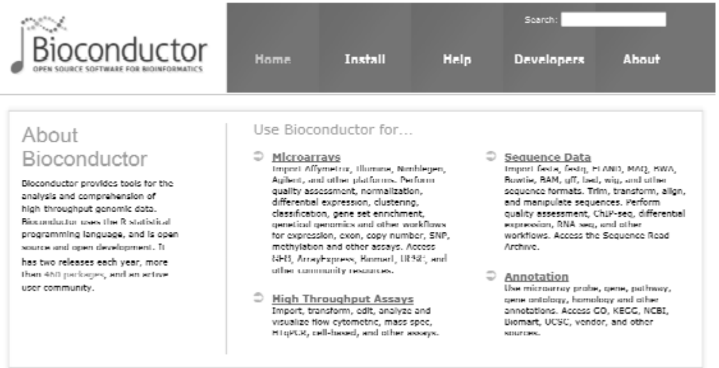
George Bell, Ph.D.

http://iona.wi.mit.edu/bio/education/R2011/

Topics for today

- Getting started with Bioconductor
- Expression microarrays
  - Normalization
  - Intro to differential expression
  - Using ‘limma’ for differential expression
- RNA-Seq
  - Preprocessing RNA-seq experiments
  - Intro to differential expression
  - Using edgeR/DESeq for differential expression

Intro to Bioconductor



About Bioconductor

Bioconductor provides tools for the analysis and comprehension of high throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, more than 460 packages, and an active user community.

Use Bioconductor for...

Microarrays
Import Affymetrix, Illumina, Nimblegen, Agilent, and other platforms. Perform quality assessment, normalization, differential expression, clustering, classification, gene set enrichment, genetical genomics and other workflows for expression, exon, copy number, SNP, methylation and other assays. Access GEO, ArrayExpress, Biomart, UCSC, and other community resources.

High Throughput Assays
Import, transform, edit, analyze and visualize flow cytometric, mass spec, HTqPCR, cell-based, and other assays.

Sequence Data
Import fasta, fastq, ELAND, MAQ, BWA, Bowtie, BAM, gff, bed, wig, and other sequence formats. Trim, transform, align, and manipulate sequences. Perform quality assessment, ChIP-seq, differential expression, RNA seq, and other workflows. Access the Sequence Read Archive.

Annotation
Use microarray probe, gene, pathway, gene ontology, homology and other annotations. Access GO, KEGG, NCBI, Biomart, UCSC, vendor, and other sources.

Getting started with Bioconductor

- Basic R installation includes no Bioconductor packages
- Install just what you want
- Steps:
  - Select BioC repositories
    `setRepositories()`
  - Install desired package(s) like
    `install.packages(limma)`
- See web page and local directory for vignettes
- After installing a package/library, you still need to load it, like
  `library(limma)`

Expression microarrays

- One color or two color
- Probes can be short (25-mer) or long (60-mer)
- A transcript may be represented by
  - One probe (Agilent)
  - Many probes (Affymetrix) grouped into a probeset
- Basic assumption: Intensity of color is correlated with gene-specific RNA abundance
- Today’s goals:
  - Measure relative RNA abundance
  - Identify genes that differ between samples

Preprocessing Affymetrix arrays

- Goals:
  - Normalize probes between arrays
  - Process mismatch probes (if present)?
  - Summarize probes into probeset values
- Common algorithms address these goals
  - MAS5 (original Affymetrix method)
  - RMA
  - GCRMA
- Choice of probeset definitions

Starting with Affy arrays in Bioc

- Install ‘affy’ and CDF (chip definition file) package for your array design
  - Example for U133 Plus 2.0 array:

```
install.packages(“affy”)
install.packages(“hg133plus2cdf”)
```

- Go to directory with CEL files (containing probe-level data) and read them

```
library(affy)
Data = ReadAffy()
```

- Preprocess into an expression set like

```
eset.mas5 = mas5(Data)
eset.rma = rma(Data)
```

Absent/present calls

- For Affymetrix arrays with mismatch probes too, they can be compared to perfect match probes
  - If the values are similar across the set, the probeset is called “absent”
- After reading a directory of CEL files as **`Data`**,

```
mas5calls = mas5calls(Data) # Do calls
# Get actual A/P matrix
mas5calls.calls = exprs(mas5calls)
write.table(mas5calls.calls, file=“APs.txt”,
  quote=F, sep="\t")
```

- You can choose if / how to use the calls.

Normalizing Agilent arrays

- Goal is do maximize biological signal and minimize technical “noise”
- Major comparisons to optimize
  - Within-array (red vs green channels)
  - Between-arrays (all arrays to each other)
- Other issues:
  - If / how to use background levels
  - If / how to add an offset to all values
- All methods rely on assumptions (expectations)
- Our favorite two-step method:
  - Use loess for within-array normalization
  - Use “Aquantile” normalization between arrays

2-color Agilent arrays in Bioc

- Read arrays

```
maData = read.maimages(dir(pattern = "txt"),
  source="agilent“)
```

- Background correct (or not)

```
maData.nobg.0 = backgroundCorrect(maData,
  method="none", offset=0)
```

- Normalize with loess

```
MA.loess.0 = normalizeWithinArrays(
  maData.nobg.0, method="loess")
```

- Normalize with Aquantile

```
MA.loess.q.0 = normalizeBetweenArrays(
  MA.loess.0, method="Aquantile")
```

Assaying differential expression

- Magnitude of fold change
- Magnitude of variation between samples
- Traditional statistical measures of confidence
  - T-test
  - Moderated t-test
  - ANOVA
  - Paired t-test
  - Non-parametric test (Wilcoxon rank-sum test)
- Other methods

Statistical testing with the t-test

- Considers mean values and variability
- Equation for the t-statistic in the Welch test:

$$t = \frac{mean_r - mean_g}{\sqrt{\frac{s_r^2}{n_r} + \frac{s_g^2}{n_g}}}$$

… and then a p-value is calculated

r ; g = data sets to compare

s = standard deviation

n = no. of measurements

- Disadvantages:
  - Genes with small variances are more likely to make the cutoff
  - Works best with larger data sets than one usually has

Statistics with limma

- Step 1: Fit a linear model for each gene
  - Starts with normalized expression matrix
  - Estimates the variability of the data
  - Based on experimental design
  - Includes effect of each RNA source
  - Command: **`lmFit()`**
- Step 2: Perform moderated t-test for each gene
  - Based on desired comparisons
  - Calculates A (mean level across all arrays) and M (log2 fold change)
  - T-test is moderated because variation is shared across genes
  - Command: **`eBayes()`**

Limma: describing your experiment

FileName	Target
GSM230387	OldSedentary
GSM230397	OldTrained
GSM230407	YoungSedentary
GSM230417	YoungTrained

FileName	Old sedentary	Old trained	Young sedentary	Young trained
GSM230387	1	0	0	0
GSM230397	0	1	0	0
GSM230407	0	0	1	0
GSM230417	0	0	0	1

- Limma gets this information in two ways:
  - Targets/design matrices: descriptions of RNA samples
  - Contrast matrix: list of desired comparisons

	OldTrained – OldSedentary	YoungTrained - YoungSedentary	TrainedVsSedentary
OldSedentary	-1	0	-0.5
OldTrained	1	0	0.5
YoungSedentary	0	-1	-0.5
YoungTrained	0	1	0.5

Multiple hypothesis testing

- When performing one moderated t-test per probe, we have to be careful of false positives
- Solution: Adjust/correct (increase) p-values to account for the high-throughput
- Most common method is False Discovery Rate
- Definition/example of FDR:
  - If you select a FDR-adjust p-value threshold of 0.05, then you can expect 5% of your list of differentially expressed genes to be false positives
- Do only as many statistical tests as necessary

RNA-Seq analysis basic steps

- Preprocessing:
  - Split by bar codes
  - Quality control (and removal of poor-quality reads)
  - Remove adapters and linkers
- Map to genome (maybe including gene models)
- Count genes (or transcripts)
- Remove absent genes
- Add offset (such as 1)
  - Prevent dividing by 0
  - Moderate fold change of low-count genes
- Identify differentially expressed genes

Counts-based statistics

- RNA-seq data representation is
  - Based on counts (integers), not continuous values
  - Different from expression array data
- Statistical test must be based on a corresponding distribution, such as the
  - Negative binomial
  - Poisson
- Expression data has the additional property of having more variability than expected for these distributions so is described as overdispersed

Assaying differential expression

- Robust and confident analysis requires replication!
- Different R packages are available for experiments
  - without replication (but don’t believe the statistics)
  - with replication
- With replication, BaRC has had success with
  - edgeR
  - DESeq
  - baySeq

Getting started in Bioc

- Input data: matrix of counts

	brain_1	brain_2	UHR_1	UHR_2
A1BG	46	65	96	107
A1CF	1	1	59	59

- Install package(s) [just the first time]
- Call package

```
Ex: library(DESeq)
```

- Read input matrix

```
counts = read.delim(counts.txt,
  row.names=1)
```

Intro to DESeq

- Requires raw counts, not RPKM values
- Takes sample depth into consideration using
  - Total read counts
  - Another more complex method
- Based on the negative binomial distribution
- Extends (and may slightly outperform) edgeR
- Calculates fold change and p-values

Quick start for DESeq

- Describe your samples (brain x2, UHR x2)

```
groups = c(rep("brain",2), rep("UHR", 2))
```

- Create a “count data set”

```
cds = newCountDataSet(counts, groups)
```

- Estimate effective library size

```
cds = estimateSizeFactors(cds)
```

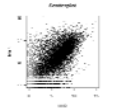
- Estimate variance for each gene (key step)

```
cds = estimateVarianceFunctions(cds)
```

- Run differential expression statistics (for brain/UHR)

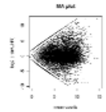
```
results = nbinomTest(cds, “UHR”, “brain”)
```

Helpful figures



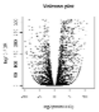
- Scatterplot:

  log2 RNA level 1 vs. log2 RNA level 1



- MA plot:

  log2 ratio vs. mean RNA level

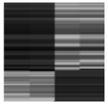


- Volcano plot

  -log10 (FDR) vs log2 ratio

- Heat map (selected genes) – Try Java Treeview

  RNA level vs reference (control or mean/median of all samples)



Local resources

- BaRC Standard Operating Procedures (SOPs)
- Previous Hot Topic:
  - Identifying and displaying differentially expressed genes
- Previous class:
  - Microarray Analysis (2007)
- R scripts for Bioinformatics
  - http://iona.wi.mit.edu/bio/bioinfo/Rscripts/
- We’re glad to share commands and/or scripts to get you started

For more information

- limma:

  Smyth GK. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol. 2004;3:Article 3.

- edgeR

  Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010 Jan 1;26(1):139-40.

- DESeq

  Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol. 2010;11(10):R106.

- baySeq

  Hardcastle TJ, Kelly KA. baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics. 2010 Aug 10;11:422.

Upcoming Hot Topics

- Unix, Perl, and Perl modules (short course in March)
- Quality control for high-throughput data
- RNA-Seq analysis
- Gene list enrichment analysis
- Galaxy
- Sequence alignment: pairwise and multiple

- See http://iona.wi.mit.edu/bio/hot_topics/
- Other ideas? Let us know.