Differential Representation

Martin Morgan

Fred Hutchinson Cancer Research Center

9-10 December, 2010

This lab illustrates how *Bioconductor* tools can be used to investigate differential respresentation. The data used is derived from Nagalakshmi et al. [2]. This publication defined the term 'RNA-seq' to describe the use of high-througput sequence methodolgies for assessing differential mRNA abundance. The data were collected to explore technical, rather than biological, issues. The authors developed two different methods, *random hexamer* and *oligo(dT)*, to generate DNA from poly(A) RNA. Each method had an 'original', 'biological', and 'technical' replicate. Samples were run on an early-generation Solexa GAI; the reads are relatively short (33bp) and relatively limited in number (3.4-6.8 million reads per sample). Samples were from yeast strain BY4741; an interesting feature of this strain is that it has deletions in leu2, ura3, met15, and his3.

1 Exploration

Exercise 1
Review the pre-processing steps used to generate the counts present in `hitspergene`*. These steps are summarized in the* Appendix *vignette available in the SeattleIntro2010 package.*

Solution:

```
> browseVignette("SeattleIntro2010")
```

Exercise 2
Load the SeattleIntro2010 package and use the `data` *function to load the* `hitspergene` *object. Display it on the screen. Use* `class` *to determine the class of* `hitspergene`*. Use the help system to understand the class, its accessors, and how to manipulate it. In many respects, the object behaves much like a standard R* `data.frame`*.*

Solution:

```
> library(SeattleIntro2010)
> data(hitspergene)
> hitspergene[1:5,]

DataFrame with 5 rows and 6 columns
        SRR002062 SRR002051 SRR002064 SRR002058 SRR002059 SRR002061
        <numeric> <numeric> <numeric> <numeric> <numeric> <numeric>
R0010W        397       655      1190       518       785      1219
R0030W        146       291       470       167       273       429
R0020C        264       458       658       415       476       742
R0040C        209       374       677       354       443       692
YAL069W         0         0         0         0         0         1

> class(hitspergene)

[1] "DataFrame"
attr(,"package")
[1] "IRanges"
```

Exercise 3
It is sometimes necessary to convert objects from one representation to another. For instance, the lattice function `splom` *prints a 'scatter plot matrix', a matrix of scatter plots between the columns of a data.frame. Often, it is possible to coerce an object from one type to another using functions named like* `as.data.frame` *or* `as(hitspergene, "data.frame")`*. Coerce* `hitspergene` *to a* `data.frame`*, and plot the data frame using* `splom`*. Use the* `pch` *argument of* `splom` *to change the character used to plot. Transform all the values in* `hitspergene` *using the* `asinh` *function, and plot those.*

Solution:

```
> library(lattice)
> df1 <- as.data.frame(hitspergene)
> splom(df1, pch=".")

> df2 <- asinh(df1)
> splom(df2, pch=".")
```

Exercise 4
One feature of the DataFrame class is that the elements (columns) can retain 'metadata' describing their content. Extract the element metadata to discover which smaples correspond to each methodology and replicate. Subset `hitspergene` *so that it contains only columns corresponding to the 'Original' and 'Biological' replicate.*

Now remove rows for which none of the samples had any hits. Do this by converting `hitspergene` *to a data.frame using* `as.data.frame`*, and using* `rowSums` *to sum up the number of hits per row. Take a quick look at the distribution of the number of hits per row using the* `histogram` *function, perhaps transforming the data (e.g., using the* `asinh` *function). Finally, remove rows with zero counts using logical subsetting. How many rows have been removed*

Solution:

```
> elementMetadata(hitspergene)

DataFrame with 6 rows and 3 columns
                 Sample   Replicate         SRR
            <character> <character> <character>
SRR002062          dT  Biological   SRR002062
SRR002051          dT    Original   SRR002051
SRR002064          dT   Technical   SRR002064
SRR002058          RH  Biological   SRR002058
SRR002059          RH    Original   SRR002059
SRR002061          RH   Technical   SRR002061

> biolReps <- elementMetadata(hitspergene)$Replicate != "Technical"
> count <- hitspergene[, biolReps]
> df1 <- as.data.frame(count)
> hitsPerRow <- rowSums(df1)
> histogram(asinh(hitsPerRow))
> count <- count[hitsPerRow != 0,]
> dim(count)

[1] 6508    4

> sum(hitsPerRow==0)  ## number of rows removed

[1] 42
```

2 Differential Representation

Exercise 5
We'll look at differential representation using the DESeq package [1]. Load the package and take a few minutes to browse its vignette.

Solution:

```
> library(DESeq)
> browseVignettes("DESeq")
```

We'll take a 'fast track' through the analysis. There are four steps:

1. Convert our data into a format understood by *DESeq*
2. Calculate scale factors so that libraries can be treated as though they have comparable numbers of reads.
3. Estimate functions to describe the relationship between mean and variance
4. Evaluate significance of differences between the random hexamer and oligo(dT) methods.

Exercise 6
The DESeq package requires two pieces of information: a data.frame of counts, and a vector describing the treatments to which each column of the count data frame belongs. Extract this information from the `count` object created above, and create a CountDataSet using `newCountDataSet`. A slight transformation is to convert the numeric data in `counts` to integer, rounding up using the `ceiling` function.

Solution:

```
> df1 <- ceiling(as.data.frame(count))
> sample <- elementMetadata(count)$Sample
> cds <- newCountDataSet(df1, sample)
```

Exercise 7
Estimate the relative contributions of each sample with `estimateSizeFactors`. Read the help page for this function (with `?estimateSizeFactors`) to understand what it is doing. Extract the estimated size factors from the result with the `sizeFactors` function; how do the size factors compare with, say, a naive weight based on reads per sample? What factors are likely to contribute to this difference?

Solution:

```
> cds <- estimateSizeFactors(cds)
> sizeFactors(cds)

SRR002062 SRR002051 SRR002058 SRR002059
0.7389309 1.1877511 1.0004022 1.1686046

> colSums(df1) / mean(colSums(df1))

SRR002062 SRR002051 SRR002058 SRR002059
0.8321238 1.1143940 1.0179526 1.0355296
```

Exercise 8
Estimate the variance functions using `estimateVarianceFunctions`*; see its help page for details*

Solution:

```
> cds <- estimateVarianceFunctions(cds)
```

Exercise 9
Finally, perform a test of signficance between groups using `nbinomTest`*. Look at the first few rows of the result using* `head`*, and compare what you see with the description of return value on the* `nbinomTest` *help page. Use* `order` *to determine the row indexes that will place the rows of the result into increasing order, based on the adjusted p-value,* `padj`*.*

Solution:

```
> nbTopTable <- nbinomTest(cds, "dT", "RH")
> head(nbTopTable, 3)

      id baseMean baseMeanA baseMeanB foldChange log2FoldChange
1 R0010W 569.5645  544.3625  594.7665  1.0925928      0.1277558
2 R0030W 210.7821  221.2918  200.2724  0.9050151     -0.1439862
3 R0020C 391.2580  371.4378  411.0783  1.1067217      0.1462925
       pval padj     resVarA     resVarB
1 0.7516122    1 0.004249891 0.479626026
2 0.6463053    1 0.378153135 0.741546903
3 0.7538871    1 0.036932427 0.002542681

> o <- order(nbTopTable$padj)
> head(nbTopTable[o,], 3)

            id baseMean baseMeanA baseMeanB foldChange log2FoldChange
4017 YLR154W-B 5541.370  52.31132 11030.428  210.86122       7.720150
4323 YLR154C-G  924.925  23.74216  1826.108   76.91413       6.265177
4019 YLR154W-E 1903.130  43.99509  3762.265   85.51556       6.418115
             pval         padj      resVarA  resVarB
4017 8.406778e-32 5.471131e-28 0.0006851835 34.41416
4323 2.574797e-31 8.378390e-28 0.0051101685 49.36319
4019 2.448517e-29 5.311649e-26 0.0010866856 56.87386
```

3 Evaluation

This section is self-directed, asking you to use the facilities of *DESeq* to assess whether the steps we performed lead to reasonable results. In particular:

Exercise 10
Follow the vignette Analysing RNA-Seq data with the "DESeq" package *in the DESeq package to assess whether estimated variance functions are reasonable. Do this using the* `scvPlot` *function. Note that 'shot noise' dominates at low count number; what are the consequences of this for the results that can be inferred? Further explore your fit as described in the* DESeq *vignette.*

Exercise 11
Follow the vignette to explore your results. Plot the log2 fold change against the base mean, analogous to the `plotDE` *function defined in the vignette. Identify the genes with largest and smallest fold change.*

4 Resources

References

[1] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biol*, 11:R106, Oct 2010.

[2] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320:1344–1349, Jun 2008.