

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 representation. The data used is derived from Nagalakshmi et al. [2]. This publication defined the term ‘RNA-seq’ to describe the use of high-throughput sequence methodologies 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
      <integer> <integer> <integer> <integer> <integer> <integer>
Q0010         0         0         5         0         0         1
Q0032         0         2         2         0         1         1
Q0055        101        217        284        104        130        129
Q0075         9         20         33         6         10         10
Q0080         1         1         8         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 samples 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] 6462    4

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

[1] 88
```

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*.

Solution:

```
> df1 <- 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
0.7154693 1.2069143 1.0252790 1.1668566

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

SRR002062 SRR002051 SRR002058 SRR002059
0.8055175 1.1362987 1.0221495 1.0360343
```

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 significance 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	Q0032	0.6285305	0.8285593	0.4285017	0.5171648	-0.9513041
2	Q0055	133.4524230	160.4817272	106.4231188	0.6631479	-0.5925974
3	Q0075	10.8931103	14.5751711	7.2110495	0.4947489	-1.0152317
	pval	padj	resVarA	resVarB		
1	0.68433920	1	0.7443564	0.46312092		
2	0.09488762	1	0.5451890	0.03941308		
3	0.07312592	1	0.2002860	0.19134132		

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

	id	baseMean	baseMeanA	baseMeanB	foldChange	log2FoldChange
4116	YLR154W-B	1146.689	19.25789	2274.121	118.08776	6.883716
4113	YLR154C-G	526.811	13.09803	1040.524	79.44129	6.311817
4118	YLR154W-E	706.113	17.29108	1394.935	80.67368	6.334026
	pval	padj	resVarA	resVarB		
4116	5.570068e-32	3.599378e-28	0.003747167	49.60941		
4113	1.048289e-30	3.073223e-27	0.004140151	44.16323		
4118	1.426752e-30	3.073223e-27	0.005967957	59.21823		

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 \log_2 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.