reb package - Identification of Regional Expression Biases

Karl Dykema

February 13, 2006

Laboratory of Computational Biology
Van Andel Research Institute
Grand Rapids, MI

1 Overview

This package contains functions to analyze gene expression data with the aim of identifying regional expression biases. To make use of these tools you will need to have Bioconductor installed along with the *ideogram* package. These must be installed in your version of R and when you start R you must load them with the `library` command. The *ideogram* package duplicates some of the functionality found in the `plotChr` function of the *geneplotter* library. Please examine both packages and determine which one best suits your analysis requirements.

This vignette is divided into two parts, basic and advanced reb prediction. Under the "Basic" section the techniques described are computationally faster, yet more rudimentary. Typically, the resolution of this approach is at the level of chromosome arms. Under the "Advanced" section, the techniques described are computationally slower but the resolution achieved can be higher.

```
> library(reb)

Loading required package: Biobase

Welcome to Bioconductor

        Vignettes contain introductory material.
        To view, simply type 'openVignette()' or start with 'help(Biobase)'.
        For details on reading vignettes, see the openVignette help page.

Loading required package: ideogram
Loading required package: annotate

> library(ideogram)
```

2 Basic Regional Expression Bias (REB) predictions

The presence of a regional expression bias often indicates an genomic amplification or deletion has occured. However, other genomic features, such as dramatic changes in gene density, can also lead to expression biases. Generally, regional expression biases are identified by isolating a gene expression values from a defined region (such as all the gene expression values mapping to a single chromosome) and applying any number of tests to determine if a bias exists in the expression values. Gene expression values are separated into subsets using a `chromLocation` object. The `chromLocation` object can be built using the `buildChromLocation` function or the `buildChromMap` function. The `buildChromLocation` function is typically used to subset the gene expression data based on whole chromosomes, while the `builChromMap` function can be used to subset the gene expression data into regions corresponding to cytogenetic bands. For example, buildChromMap can be used to produce a 'chromLocation' object composed of the genes that populate human chromosome 1p and chromosome 1q.

`summarizeByRegion` is most straightforwardly used to identify regional gene expression biases when a test sample is compared to a biologically meaningful reference sample; data that is typically generated from two-color gene expression data but also can be generated from one-color expression data. The gene expression values from each of these regions are extracted from the `exprSet` and a summary statistic is computed for each region. A number of simple tests (`t.test`,etc) can be used to determine if a genomic region contains a disproportionate number of positive or negative log transformed gene expression ratios.

If multiple clones map to the same genomic locus the `aggregate.by.loc` argument can be used to replace multiple overlapping values with a single representative value. For example, if 50 copies of the actin gene are on a particular array and actin changes expression under a given condition, it may appear as though a regional expression bias exists as 50 values in a small region change expression. Using the `aggregate.by.loc` argument can partially correct this situtation.

The mcr.eset is a two-color gene expression `exprSet` object with cytogenetically complex (MCR) and normal control (MNC) samples that were profiled against a pooled-cell line reference. For specifics, see PMID: 15377468

Please note that we are summarizing this data by chromosome. The function `buildChromMap` can be used to build a `chromLocation` object where the gene expression values can separated by more refined cytogenetic mapping information, for example by chromosome arm (1p,1q, 2p, 2q, etc) or chromosome band (1p1, 1p2, 1p3). Hs.arm and Hs.cytoband are data structures that contain chromosome arm and band information for the human genome and are included in this package and ideogram, respectively.

For example: "affy.arms <- buildChromMap("hgu133plus2",Hs.arms)"

```
> data(mcr.eset)
```

```
> data(ideogramExample)
> ref.ix <- grep("MNC", colnames(mcr.eset@exprs))
> mcr.sum <- summarizeByRegion(mcr.eset, vai.chr, ref = ref.ix)
```

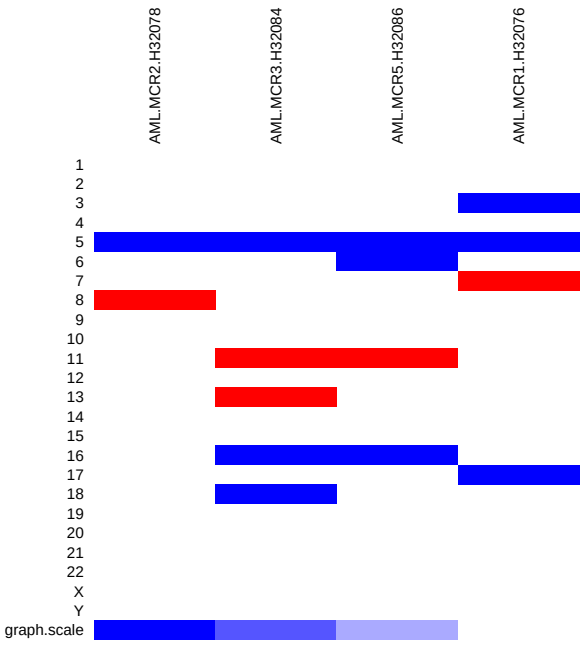
```
Creating ratios...
```

For one-color data (or more complex two-color data), the `ref` argument in the `summarizeByRegion` can be used to specify which sample(s) should be used as the reference sample(s).

2.1 Displaying basic reb predictions

`regmap` or `heatmap` can used to display the results of a `summarizeByRegion` call. `regmap` is a simple wrapper around the `image` function to display genome region summary statistics. It includes a scale argument that is a two-element vector which provides a floor and ceiling for the matrix and allows a crude scale bar to be included on the lower potion of the graph. The `regmap` includes a default red-white-blue color scheme called `.rwb`. For other colors consider using the *gtools* `colorpanel` function (ex. "regmap(m,col=colorpanel(30,"blue","white","red"))") or other color schemes (ex. `topo.colors`).

```
> complex.ix <- grep("MCR", colnames(mcr.eset@exprs))
> regmap(mcr.sum[, complex.ix], col = .rwb, scale = c(-3, 3))
```



Notice the diminished expression present in all the samples in the region of chromosome 5.

3 Advanced reb prediction

The function `smoothByRegion` (or `reb`) allows the user to smooth the gene expression data using a subset of smoothing functions. Visualization of the resultant gene expression data can also be a useful way to identify regional expression biases. The default smoothing function is a multiple span moving binomial test. Emperically, we have found that this test identifies regional expression biases quite well. Other functions such as `supsmu` and `lowess` can also be used.

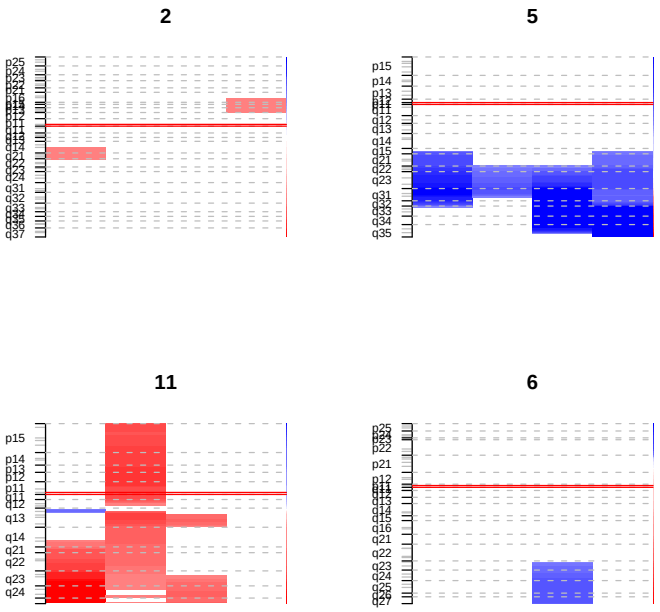
```
> mcr.cset <- smoothByRegion(mcr.eset, vai.chr, ref = ref.ix)

Creating ratios...
```

The `smoothByRegion` returns an `exprSet` object that contains the smoothed gene expression data. The smoothed gene expression data can be plotted using the *ideogram* package. Please note that in an effort to make this vignette of a reasonable size, we have not used the mideogram function. In this situation it could be used like this: mideogram(mcr.cset@exprs[,complex.ix],vai.chr,method="i",col=.rwb)

Also note that we are using an empirically derived mask of 1.96 when plotting this data. This not only filters out possible noise, but makes the image files smaller for this vignette.

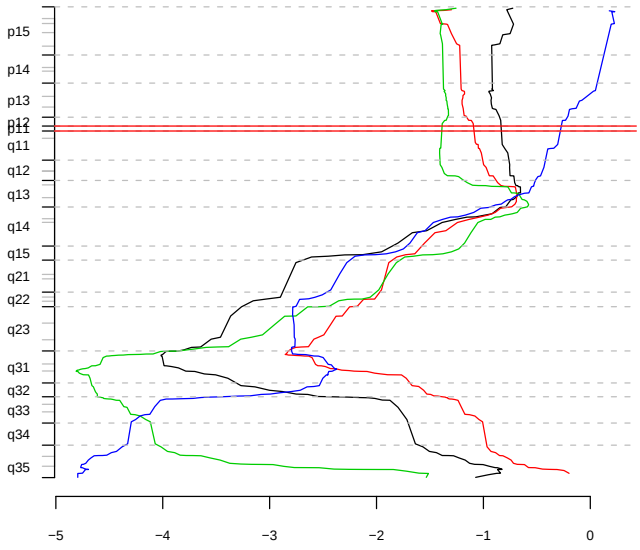
```
> data <- mcr.cset@exprs[, complex.ix]
> data[abs(data) < 1.96] <- NA
> op <- par(no.readonly = TRUE)
> layout(rbind(c(1, 2), c(3, 4)))
> for (i in c("2", "5", "11", "6")) ideogram(data, vai.chr, method = "i",
+     col = .rwb, dlim = c(-4, 4), chr = i)
> par(op)
```



A closer look at a single chromosome in the samples with complex karyotypes. Notice that the region of diminished expression that is consistant across all of the samples.

```
> ideogram(mcr.cset@exprs[, complex.ix], vai.chr, method = "m",
+     chr = 5, dlim = c(-5, 5), type = "l", lty = 1)
```

5


p15
p14
p13
p12
p11
q11
q12
q13
q14
q15
q21
q22
q23
q31
q32
q33
q34
q35
−5
−4
−3
−2
−1
0