Segmentation demo

Wolfgang Huber

March 15, 2006

Contents

1 Introduction

This script presents a demo of the segmentation function on the *davidTiling* data.

```
> options(error = recover, warn = 0)
> graphics.off()
> library("tilingArray")
> library("davidTiling")
> if (!exists("davidTiling")) data("davidTiling")
> if (!exists("probeAnno")) data("probeAnno")
```

2 Normalization

Please see the vignette *Assessing signal/noise ratio before and after normalization* (`assessNorm.Rnw`) for explanation.

```
> xn = cache("xn", {
+     nc = as.integer(2560)
+     PMind = rep(seq(as.integer(1), nc - as.integer(3), by = as.integer(2)),
+         each = nc) * nc + (1:nc)
+     MMind = PMind + nc
+     ispm = rep(FALSE, nc * nc)
+     ispm[PMind] = TRUE
+     isbg = (probeAnno$probeReverse$no_feature == "no" & probeAnno$probeDirect$no_feature ==
```

```
+         "no" & ispm)
+     isRNA = davidTiling$nucleicAcid %in% c("poly(A) RNA", "total RNA")
+     isDNA = davidTiling$nucleicAcid %in% "genomic DNA"
+     stopifnot(sum(isRNA) == 5, sum(isDNA) == 3)
+     normalizeByReference(davidTiling[, isRNA], davidTiling[,
+         isDNA], pm = PMind, background = isbg)
+ })
> pData(xn)[, 2, drop = FALSE]

                               nucleicAcid
05_04_27_2xpolyA_NAP3.cel     poly(A) RNA
05_04_26_2xpolyA_NAP2.cel     poly(A) RNA
05_04_20_2xpolyA_NAP_2to1.cel poly(A) RNA
050409_totcDNA_14ug_no52.cel    total RNA
030505_totcDNA_15ug_affy.cel    total RNA
```

3 Segmentation

3.1 Selecting the probes in along-chromosome order

Extract for all probes that map to the "+" strand of chromosome 1 their start and end coordinate, and their index in the exprs(davidTiling) data matrix. Sort them by midpoint.

```
> chrstrd = "1.+"
> what = c("start", "end", "index", "unique")
> prbs = do.call("data.frame", mget(paste(chrstrd, what, sep = "."),
+     probeAnno))
> colnames(prbs) = what
> prbs$mid = (prbs$start + prbs$end)/2
> prbs = prbs[order(prbs$mid), ]
```

We throw out the missing (NA) values:

```
> numna = rowSums(is.na(exprs(xn)[prbs$ind, ]))
> stopifnot(all(numna %in% c(0, ncol(xn))))
> prbs = prbs[numna == 0, ]
```

3.1.1 Avoid oversampling

```
> sprb = prbs[sampleStep(prbs$mid, step = 7), ]
```

```
> par(mfrow = c(3, 1))
> hist(prbs$mid, col = "mistyrose", 100, main = "(a)")
> barplot(table(diff(prbs$mid)), main = "(b)")
> barplot(table(diff(sprb$mid)), main = "(c)")
```

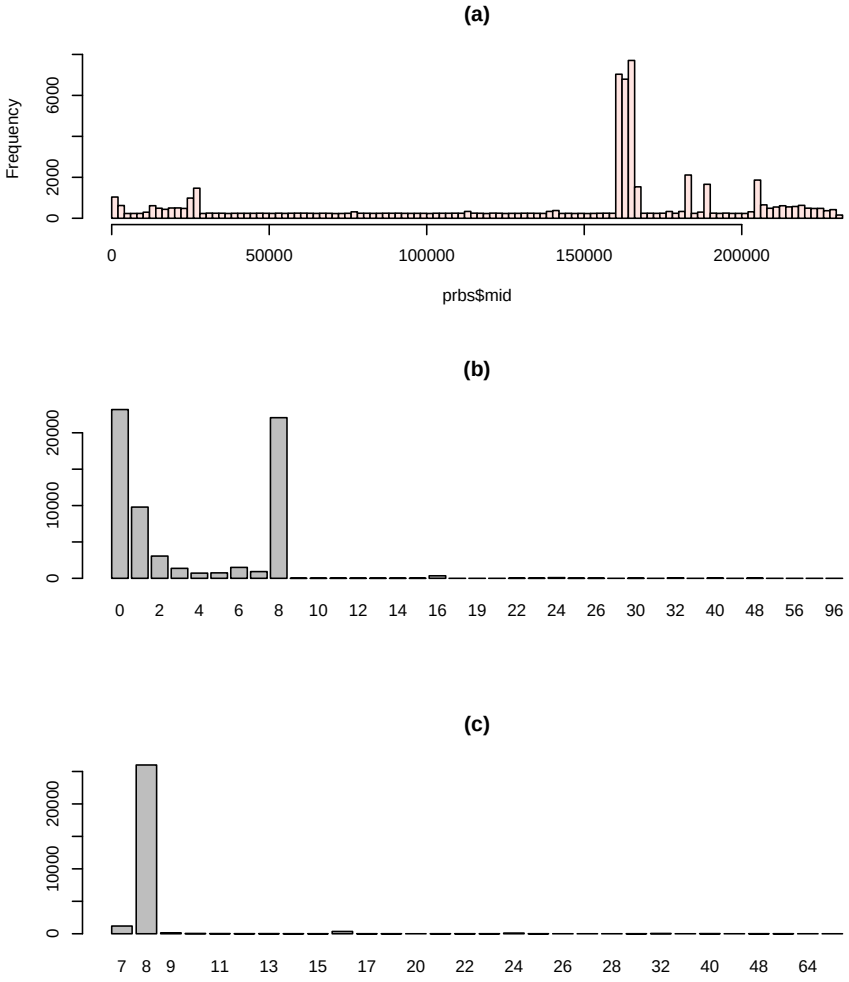

Figure 1: (a): Histogram of probe midpoints along the chromosome. There are some probe dense regions in particular around 160,000. The sequence of that region is repeated multiple times in the genome, and due to the chip design, there are also a lot of probes (more than necessary) for that region. (b): histogram of differences between probe midpoints (`prbs$mid`). The intention of the chip design was to have a regular spacing of 8 bases. In some cases, the spacing is wider, probably due to updates in the genome sequence between when the chip was designed and when probes were re-aligned. In many cases, it is tighter with multiple probes for the same target sequence, or only 1 or 2 bases offset. This occurs in the regions of duplicated sequence. (c): histogram of differences between probe midpoints after sampling (`sprb$mid`)

3.2 Call the segmentation algorithm

The segmentation algorithm needs two parameters, `maxseg`, the maximum number of segments that the algorithm is going to consider, and `maxk`, the maximum length per segment of individual segments. The algorithm will calculate all optimal segmentations with $1, 2, \ldots,$ maxseg segments. Note that `maxk` is measured in number of data points, not in genomic coordinates. The choice that we make here corresponds to an *average* length per segment of 1500 bases. Our choice of the parameter `maxk` corresponds to a maximum segment length of about $7.5 \times 3,000 = 22,500$ bases.

```
> nseg = round(sprb$end[nrow(sprb)]/1500)
```

```
[1] 153
```

```
> segw = cache("segw", {
+     y = exprs(xn)[sprb$ind, xn$nucleicAcid == "poly(A) RNA",
+         drop = FALSE]
+     segw = segment(y, maxseg = nseg, maxk = 3000)
+     segw@x = sprb$mid
+     segw@flag = sprb$unique
+     segw
+ })
```

We also add additional information to the object that was not used for the actual segmentation, but will be useful for the visualization: into the slot `x`, the x-coordinates of the probes, and into the slot `flag`, the uniqueness status of the probes (0 iff the probe has exactly one match in the genome).

Having to access the `x` and `flag` slots directly, in the code above, is a bit unelegant; I will probably provide accessor functions in subsequent versions of the package.

3.3 Calculate confidence intervals

This is simply a call to the `confint` method of the *segmentation* class.

```
> segwi = cache("segwi", confint(segw, parm = nseg))
```

Now we are ready to have a look at the result via the `plot` method of the *segmentation* class. The plot is shown in Figure 2.

```
> plot(segwi, nseg, pch = ".", xlim = c(0, 40000))
```

```
Note: slot 'y' has more than one column, calculating 'rowMeans'
```

3.4 Using the `plotAlongChrom` function for more elaborate displays.

Since the data in the *davidTiling* package is strand-specific, we can do the segmentation for the "-" strand as well and produce the along-chromosome plot shown in Figure 3. We run the same computations as above (code not shown) to obtain the object `segci`.

The function `plotAlongChrom` accepts an environment as its first argument which is expected to contain objects of class *segmentation* with names given by `paste(chr, c("+", "-"), sep=".")`, where `chr` is the chromosome identifier.

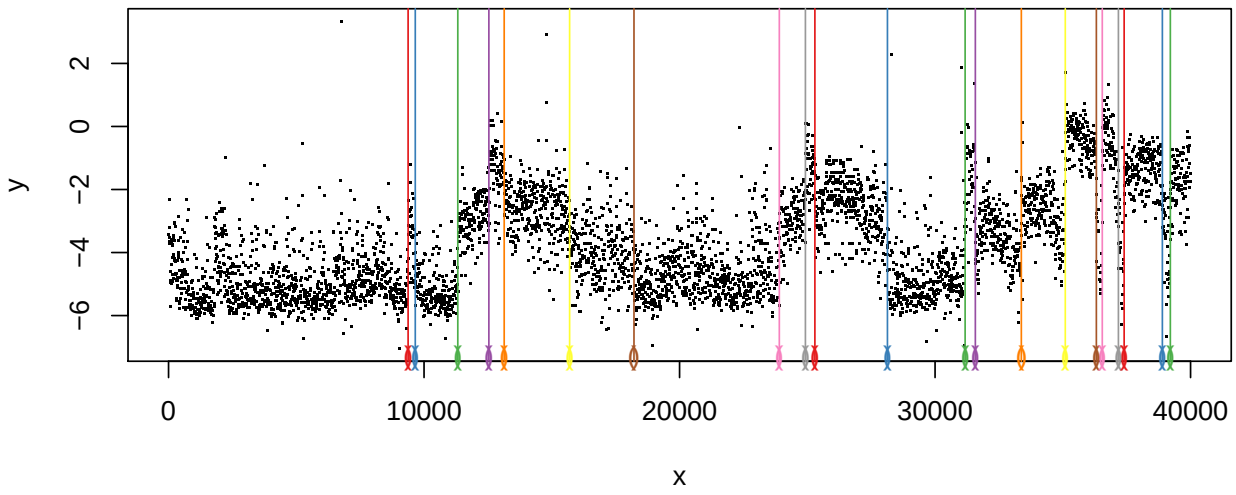

Figure 2: Segmentation with confidence intervals.

```
> data("gff")
> segObj = new.env(parent = baseenv())
> assign("1.+", segwi, segObj)
> assign("1.-", segci, segObj)
> plotAlongChrom(segObj, chr = 1, coord = 1000 * c(101.5, 115),
+     gff = gff, doLegend = FALSE)
```

References

[1] Lior David, Wolfgang Huber, Marina Granovskaia, Joern Toedling, Curtis J. Palm, Lee Bofkin, Ted Jones, Ronald W. Davis, and Lars M. Steinmetz A high-resolution map of transcription in the yeast genome. *PNAS*, 2006. 6

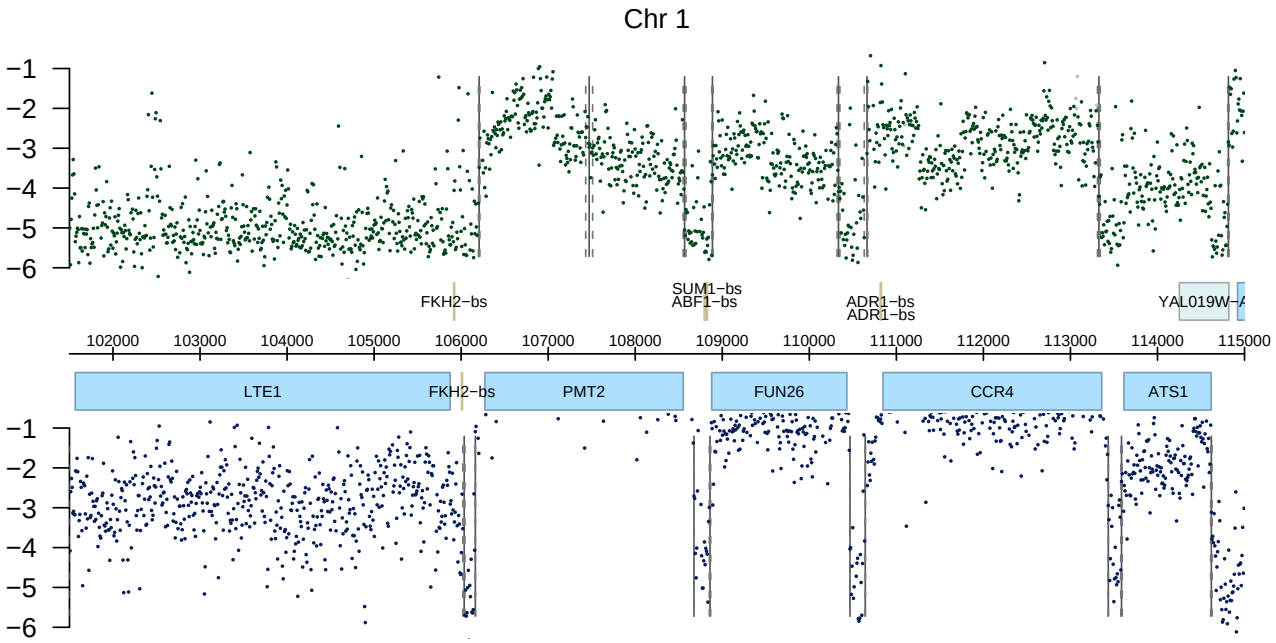

Figure 3: Along-chromosome plot similar to the one shown in the paper [1].