A hidden Markov model for SNP arrays processed with *crlmm*

Robert Scharpf

July 5, 2012

```
> require("crlmm")
> library(VanillaICE)
> library(RColorBrewer)
```

For datasets with more than 10 samples processed in a batch, copy number estimation using the linear model described in Scharpf et al, 2010 is feasible. Following the vignettes for copy number analysis in the *crlmm* package, one obtains an object of class `CNSet`. Here we describe how to smooth the copy number estimates integrating information from the B allele frequencies. We begin with a `CNSet` object containing the information on chromosome 8 for two samples. These samples were processed as part of a larger batch, making estimates from the linear model available.

```
> data(cnSetExample, package="crlmm")
```

We begin by ordering the `CNSet` object by chromosome and physical position, then coercing the ordered `CNSet` object to an object of class `BafLrrSetList`.

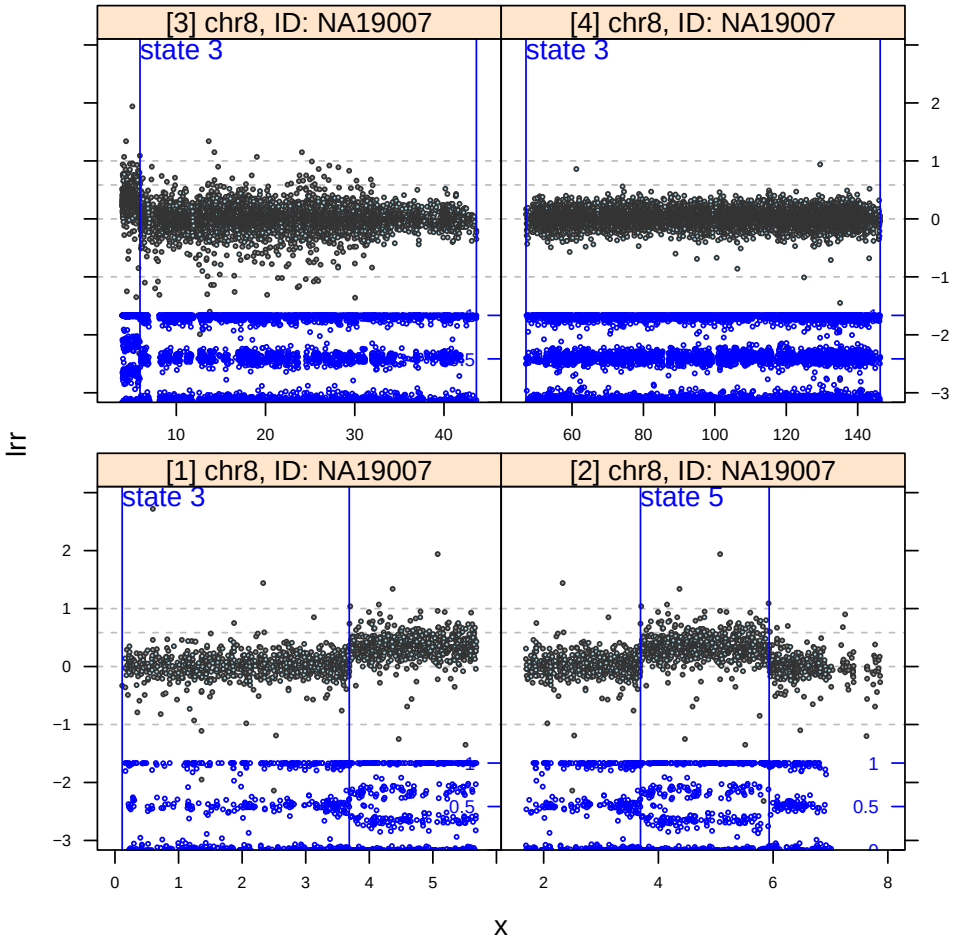
```
> cnSetExample <- chromosomePositionOrder(cnSetExample)
> brList <- constructBafLrrSetListFrom(cnSetExample)
```

We now smooth the copy number estimates, integrating emission probabilities obtained from log R ratios and BAFs. As a first step, we fit the HMM to a single sample / chromosome and visually inspect the inferred states to verify that the default settings are suitable.

```
> fit1 <- hmm(brList[[1]][, 1])
```

The `fit` object is an object of class `GRangesList`. Each element in the list is a `GRanges` object for one sample that provides the start and stop positions of the inferred copy number state. The *IRanges* function `findOverlaps` can be useful for identifying which markers in the original `BafLrrSetList` object lie within a particular range. Methods for visualizing the low level summaries along with the inferred breakpoints for the copy number states make use of the `findOverlaps`. In the following code chunk we use the function `xyplotLrrBaf` to plot the log R ratios and B allele frequencies for the genomic intervals in the `GRanges` object. We plot a 2 megabase window framing the genomic intervals by passing the argument `frame=2e6`. See the function `xypanelBaf` for details on how to modify the appearance of the plotting symbols.

```
> library(SNPchip)
> brSet <- brList[chromosome(brList) == 8][[1]]
> rd.sample1 <- fit[[1]]
> rd <- rd.sample1[chromosome(rd.sample1)=="chr8", ]
> cnfig <- xyplotLrrBaf(rd, brSet[,1], frame=2e6,
+                       panel=xypanelBaf, cex=0.3, pch=21, border="blue",
+                       scales=list(x="free", cex=0.6))
> print(cnfig)
```



As the HMM states for this sample appear reasonable, one can fit the HMM to all samples and all chromosome by

```
> fit2 <- hmm(brList)
```

Since there are only 2 samples in the `brList` object, the length of fit2 is only two.

```
> length(fit2)
```

```
[1] 2
```

The `GRangesList` object can be stacked to create a `GRanges` object as follows:

```
> library(GenomicRanges)
> gr <- stack(fit2)
> gr
```

```
GRanges with 12 ranges and 4 elementMetadata cols:
       seqnames               ranges strand |  sample numberProbes     state
          <Rle>            <IRanges>  <Rle> |   <Rle>    <integer> <integer>
   [1]     chr8 [  113368,   3685172]      * | NA19007          831         3
   [2]     chr8 [ 3687921,   5930859]      * | NA19007          776         5
   [3]     chr8 [ 5934100,  43630633]      * | NA19007         5893         3
   [4]     chr8 [47105586, 146293414]      * | NA19007         7500         3
```

```
  [5]     chr8 [  113368,   3773935]      * | NA19003          854         3
  [6]     chr8 [ 3787377,   3788239]      * | NA19003            4         2
  [7]     chr8 [ 3789986,   7186620]      * | NA19003         1087         3
  [8]     chr8 [ 7214656,   7787950]      * | NA19003           43         5
  [9]     chr8 [ 7809879,  24972187]      * | NA19003         3515         3
 [10]     chr8 [24974565,  24984144]      * | NA19003            6         2
 [11]     chr8 [24991115,  43630633]      * | NA19003         1991         3
 [12]     chr8 [47105586, 146293414]      * | NA19003         7500         3
             LLR
       <numeric>
  [1]    0.000000
  [2] 1812.851325
  [3]    0.000000
  [4]    0.000000
  [5]    0.000000
  [6]    7.083952
  [7]    0.000000
  [8]   64.787870
  [9]    0.000000
 [10]   16.569297
 [11]    0.000000
 [12]    0.000000
 ---
 seqlengths:
       chr8
  146364022
```

1 Troubleshooting

Missing values are permissable for the LRRs and BAFs:

```
> set.seed(1)
> brSet <- brList[[1]]
> lrr(brSet)[sample(1:nrow(brSet), 50), 1] <- NA
> baf(brSet)[sample(1:nrow(brSet), 50), 1] <- NA
> fit2 <- hmm(brSet)
> all.equal(state(fit2[[1]]), state(fit[[1]]))
```

```
[1] TRUE
```

While permissable, a large number of NA's may indicate problems with the preprocessing or the `crlmmCopynumber` step described in the `copynumber` vignette in the *crlmm* package. First, verify that the signal to noise ratio (SNR) is in an acceptable range. For Affymetrix, the SNR should be above 5 and for Illumina the SNR should be above 25.

```
> isTRUE(all(cnSetExample$SNR[] > 5))
```

```
[1] TRUE
```

Next, check that the percentage of missing values is reasonably low:

```
> r <- lrr(brList[[1]])[,]
> isTRUE(all(apply(is.na(r), 2, sum) < 0.01))
```

```
[1] TRUE
```

```
> snp.index <- which(isSnp(brList[[1]]))
> b <- baf(brList[[1]])[snp.index,]
> isTRUE(all(apply(is.na(b), 2, sum) < 0.01))
```

```
[1] TRUE
```

If, for example, all the log R ratios / BAFs are missing, this indicates that the `crlmmCopynumber` step was either not performed or unsucessful. Specifically, the following unevaluated step is needed:

```
> crlmmCopyNumber(cnSetExample)
```

where `cnSetExample` is an object of class `CNSet`.

2 Session Information

```
> toLatex(sessionInfo())
```

- R version 2.15.1 Patched (2012-07-01 r59713), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.iso885915`, `LC_NUMERIC=C`, `LC_TIME=en_US.iso885915`, `LC_COLLATE=en_US.iso885915`, `LC_MONETARY=en_US.iso885915`, `LC_MESSAGES=en_US.iso885915`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.iso885915`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: Biobase 2.16.0, BiocGenerics 0.2.0, BiocInstaller 1.4.7, crlmm 1.15.11, GenomicRanges 1.8.7, IRanges 1.14.4, oligoClasses 1.19.32, RColorBrewer 1.0-5, SNPchip 2.3.7, VanillaICE 1.19.17
- Loaded via a namespace (and not attached): affyio 1.24.0, annotate 1.34.1, AnnotationDbi 1.18.1, Biostrings 2.24.1, bit 1.1-8, codetools 0.2-8, compiler 2.15.1, DBI 0.2-5, ellipse 0.3-7, ff 2.2-7, foreach 1.4.0, genefilter 1.38.0, grid 2.15.1, iterators 1.0.6, lattice 0.20-6, msm 1.1.1, mvtnorm 0.9-9992, preprocessCore 1.18.0, RSQLite 0.11.1, splines 2.15.1, stats4 2.15.1, survival 2.36-14, tools 2.15.1, XML 3.9-4, xtable 1.7-0, zlibbioc 1.2.0