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

Robert Scharpf

October 4, 2011

```
> require("crlmm")
> library(VanillaICE)
> ##library(HapmapCrlmmAffySet)
> library(RColorBrewer)
> ##if(!exists("hapmapSet")) data(hapmapSet)
> ##class(hapmapSet)
> ##dim(hapmapSet)
```

1 For datasets with more than 10 samples

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, which is why estimates from the linear model are available.

```
> data(sample.CNSet, 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 `oligoSnpSet`. If the `CNSet` object were very large, we might want to subset the `CNSet` by batch prior to subsetting. See the Section 1.1 for details. Here, the object is small by design and there is no need for additional subsetting. After coercion, the copy number estimates are saved on the log base 2 scale.

```
> sample.CNSet <- order(sample.CNSet)
> oligoSet <- as(sample.CNSet, "oligoSnpSet")
> range(copyNumber(oligoSet), na.rm=TRUE)
```

```
[1] -3.321928  3.000000
```

As BAFs may help improve the detection of hemizygous deletions and copy number duplications, we add estimates of the B allele frequencies to the `oligoSet` object as follows.

```
> baf.lrr <- calculateRBaf(sample.CNSet)
> ad <- assayDataNew(copyNumber=copyNumber(oligoSet),
+                    cnConfidence=cnConfidence(oligoSet),
+                    call=calls(oligoSet),
+                    callProbability=snpCallProbability(oligoSet),
+                    baf=baf.lrr[["baf"]])
> assayData(oligoSet) <- ad
> ls(assayData(oligoSet))
```

```
[1] "baf"             "call"            "callProbability" "cnConfidence"
[5] "copyNumber"
```

As in the VanillaICE vignette, we generate default parameters for the hmm using the HmmOptionList constructor.

```
> hmmOpts <- HmmOptionList(oligoSet, is.log=TRUE)
```

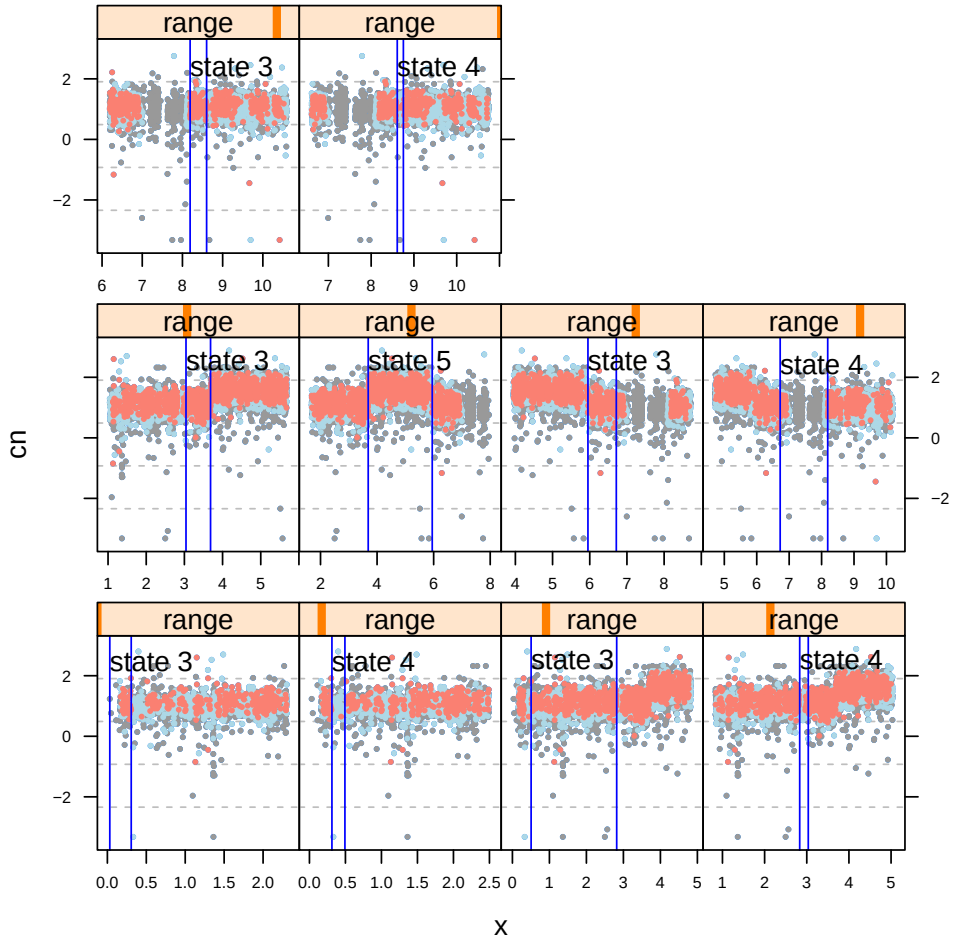
We now smooth the copy number estimates, integrating emission probabilities obtained from copy number and the emission probabilities obtained from the BAFs. See the documentation for cnEmission and bafEmission for details regarding the estimation of emission probabilities.

```
> fit <- hmm(oligoSet, hmmOpts, use.baf=TRUE)
```

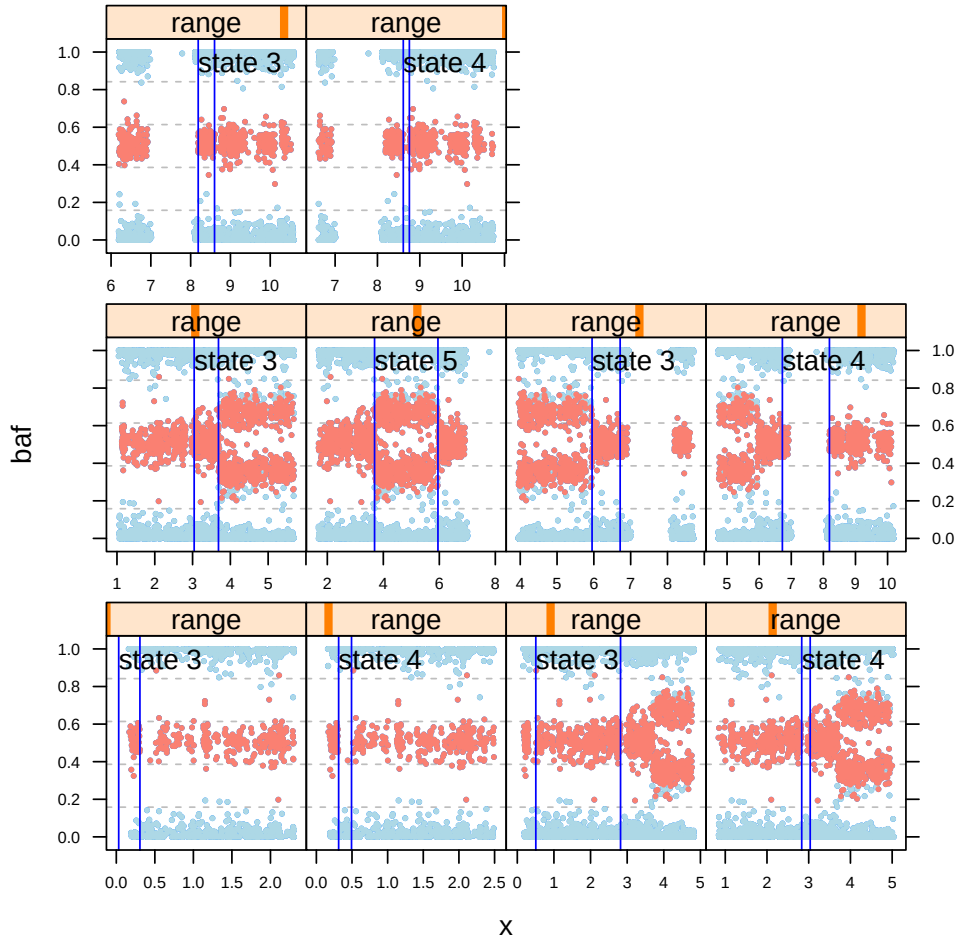
```
  |
  |                                                                      |   0%
  |
  |==================================                                    |  50%
  |
  |======================================================================| 100%
```

The fit object is an object of class RangedDataHMM. Several useful accessors are defined for this class, including sampleNames, state, and chromosome. In addition, findOverlaps methods defined for the class can be useful for identifying which markers in the original oligoSet 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. For example, in the following code chunk we plot the copy number estimates color coded by genotype for each of the range. Using the argument frame, we can include more or fewer markers to the left and right of the breakpoints. See the function xypanel for details on how to modify the appearance of the plotting symbols.

```
> cnfig <- xyplot(cn ~ x | range, data=oligoSet, range=fit[1:10, ], frame=2e6,
+                 panel=xypanel, cex=0.3, pch=21, border="blue",
+                 scales=list(x="free", cex=0.6),
+                 col.hom="lightblue", col.het="salmon", col.np="grey60", fill.np="grey60")
> print(cnfig)
```




```
> baffig <- xyplot(baf ~ x | range, data=oligoSet, range=fit[1:10, ], frame=2e6,
+                  panel=xypanel, cex=0.3, pch=21, border="blue",
+                  scales=list(x="free", cex=0.6),
+                  col.hom="lightblue", col.het="salmon", col.np="grey60", fill.np="grey60")
> print(baffig)
```



1.1 Large CNSet objects

Recall that any subset operation pulls the data from disk to RAM.

```
> sample.index.list <- split(seq_len(ncol(bigCnSet)), batch(bigCnSet))
> for(i in seq_along(sample.index.list)){
+         sample.index <- sample.index.list[[i]]
+         cnSet <- bigCnSet[, sample.index]
+         cnSet <- order(cnSet)
+         oligoSet <- as(cnSet, "oligoSnpSet")
+         ## and so on
+ }
```

2 Session Information

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

- R version 2.14.0 alpha (2011-10-04 r57166), 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 2.13.10, BiocInstaller 1.1.27, crlmm 1.11.51, IRanges 1.11.27, oligoClasses 1.15.55, RColorBrewer 1.0-5, VanillaICE 1.15.76
- Loaded via a namespace (and not attached): affyio 1.21.2, annotate 1.31.1, AnnotationDbi 1.15.28, Biostrings 2.21.11, bit 1.1-7, DBI 0.2-5, ellipse 0.3-5, ff 2.2-3, genefilter 1.35.0, grid 2.14.0, lattice 0.19-33, mvtnorm 0.9-9991, preprocessCore 1.15.0, RSQLite 0.10.0, SNPchip 1.17.0, splines 2.14.0, survival 2.36-10, tools 2.14.0, xtable 1.5-6, zlibbioc 0.1.8