HowTo: Creating HMM objects from BeadStudio-processed Illumina arrays

Robert B. Scharpf

April 8, 2008

This vignette describes how to create an instance of an `oligoSnpSet` from Illumina data. After reading this vignette, one should be able to fit a HMM to identify chromosomal alterations using the Illumina platform.

1 Reading in the data

To illustrate, an example of BeadStudio output obtained from the Pevsner website (`http://pevsnerlab.kennedykrieger.org/SNPtrio03.htm`) is included with this package. The Pevsner laboratory provide the following instructions for saving the data using Illumina's BeadStudio in the appropriate format:

1. select the "Full Data Table" tab
2. click on the "Column Chooser" icon
3. in the "Displayed Columns" area, keep "Name", "Chr" and "Position", hide the rest
4. the "Displayed Subcolumns" area, keep "GType" and "Log R Ratio", hide the rest
5. click on "Export Displayed Data to File" icon; finally, save the file

A subset of 1000 SNPs is included with this package and can be loaded by

```
> library(VanillaICE)
> pathToIlluminaData <- system.file("illumina", package = "VanillaICE")
> illuminaEx <- read.table(paste(pathToIlluminaData,
+     "/illuminaEx.txt", sep = ""), sep = "\t", as.is = TRUE)
```

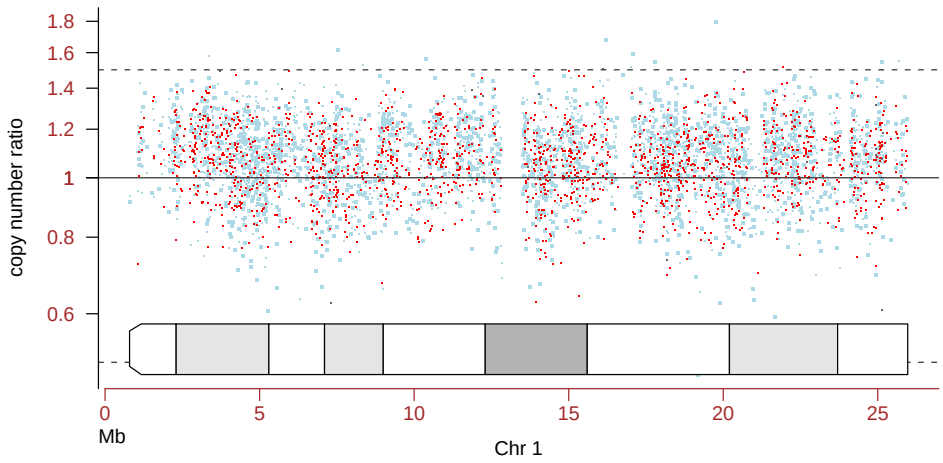
The following code converts this data.frame to an object of class `oligoSnpSet`:

```
> gt <- illuminaEx[, "S1135.GType", drop = FALSE]
> gt[gt == "AA"] <- 1
> gt[gt == "BB"] <- 3
> gt[gt == "AB"] <- 2
> gt[gt == "NC"] <- 4
> gt <- as.matrix(as.integer(gt[[1]]))
> ratio <- 2^as.matrix(as.numeric(illuminaEx[, "S1135.Log.R.Ratio"]))
> colnames(gt) <- colnames(ratio) <- "S1135"
> rownames(ratio) <- rownames(gt) <- illuminaEx[, "Name"]
> fd <- new("AnnotatedDataFrame", data = data.frame(position = illuminaEx[,
+     "Position"], chromosome = illuminaEx[, "Chr"],
+     stringsAsFactors = FALSE), varMetadata = data.frame(labelDescription = c("position",
+     "chromosome")))
> featureNames(fd) <- illuminaEx[, "Name"]
> ratioConfidence <- callsConfidence <- matrix(NA,
+     nrow = nrow(ratio), ncol = 1)
> colnames(ratioConfidence) <- colnames(callsConfidence) <- colnames(ratio)
> rownames(ratioConfidence) <- rownames(callsConfidence) <- rownames(ratio)
> snpset <- new("RatioSnpSet", ratio = ratio, ratioConfidence = ratioConfidence,
+     calls = gt, callsConfidence = callsConfidence,
+     featureData = fd, annotation = "Illumina550k")
> chrom <- chromosome2numeric(chromosome(snpset))
> snpset <- snpset[order(chrom, position(snpset)),
+     ]
> stopifnot(validObject(snpset))
```

We can now use methods from the R package *SNPchip* to plot the data:

```
> gp <- plotSnp(snpset)
> gp$cex <- 3
> gp$ylab <- "copy number ratio"
> gp$abline <- TRUE
> gp$abline.h <- c(0.5, 1, 3/2)
> gp$abline.col <- "grey20"
> gp$abline.lty <- c(2, 1, 2)


> show(gp)
```



Parameters for the HMM are generated by creating an instance of the `HmmParameter` class using the method `new`:

```
> featureData(snpset)$arm <- getChromosomeArm(snpset)
> options <- new("HmmOptions", snpset = snpset, states = c("D",
+     "N", "L", "A"), copyNumber.location = c(1/2,
+     1, 1, 3/2), probHomCall = c(0.99, 0.75, 0.99,
+     0.75))
> params <- new("HmmParameter", states = states(options))
> emission(params) <- copyNumber.emission(options)
```

```
[1] "Calculating emission probabilities on the log(copy number)"
```

```
> genomicDistance(params) <- exp(-2 * calculateDistance(options)/(100 *
+     1e+06))
> transitionScale(params) <- scaleTransitionProbability(options)
```

To fit the HMM,

```
> fit <- hmm(options, params)
```

```
[1] "Fitting HMM to sample 1"
```

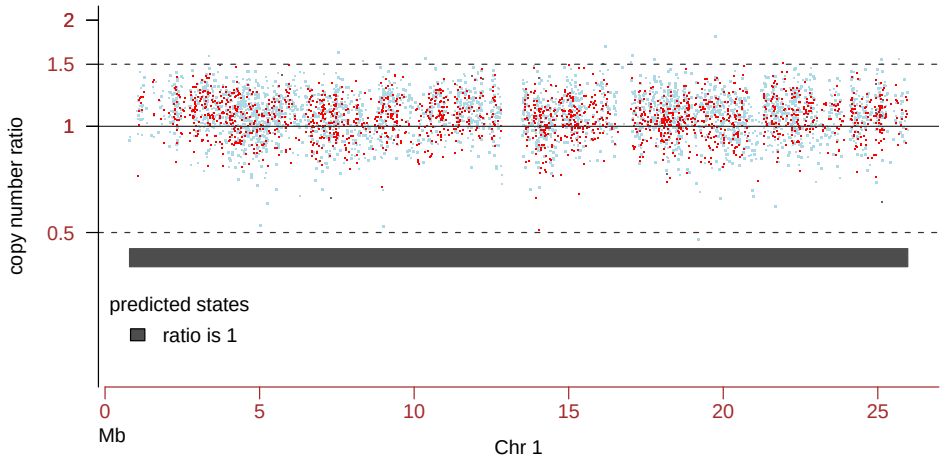
Here we visualize the data as well as the predicted states (in this example, the entire region is normal):

```
> gp <- plotSnp(options@snpset, fit)
```

```
[1] "col.predict not specified in list of graphical parameters. Using the following colors:"
[1] "#A6611A" "white"   "#80CDC1" "#018571"
```

```
> gp$add.cytoband <- FALSE
> gp$ylab <- "copy number ratio"
> gp$abline <- TRUE
> gp$abline.h <- c(0.5, 1, 3/2)
> gp$abline.col <- "grey20"
> gp$col.predict[states(fit) == "N"] <- "grey30"
> gp$abline.lty <- c(2, 1, 2)
> gp$hmm.ycoords <- c(0.4, 0.45)
> gp$ylim <- c(0.2, 2)
```

```
> show(gp)
> legend(0, 0.35, title = "predicted states", fill = "grey30",
+     legend = "ratio is 1", bty = "n")
```



Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 2.7.0 alpha (2008-04-06 r45127), `x86_64-unknown-linux-gnu`

- Locale: LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=C;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TE 8;LC_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: Biobase 1.99.0, oligoClasses 1.1.18, RColorBrewer 1.0-1, SNPchip 1.3.22, VanillaICE 1.1.18