HowTo: Creating HMM objects from BeadStudio-processed Illumina chips

Robert B. Scharpf

February 1, 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)
```

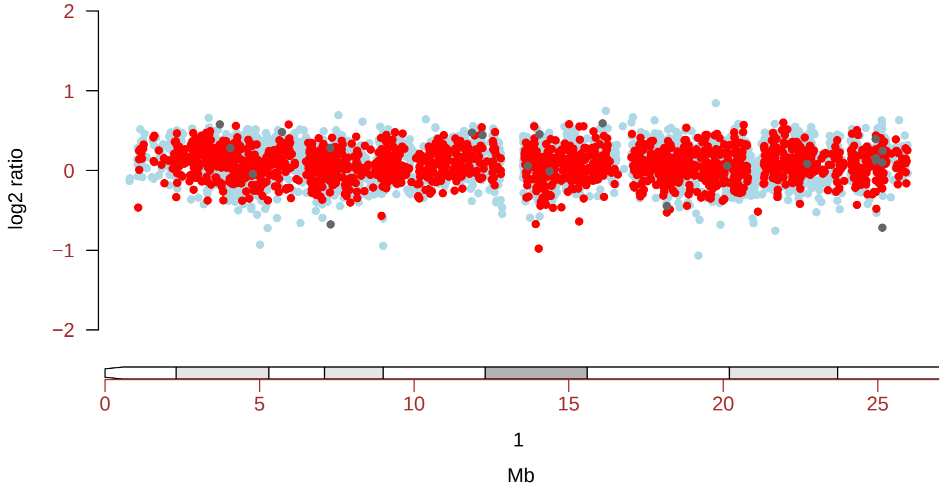
The following code converts this data.frame to an object of class `oligoSnpSet`:

```
> 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]]))
> cn <- as.matrix(as.numeric(illuminaEx[, "S1135.Log.R.Ratio"]))
> colnames(gt) <- colnames(cn) <- "S1135"
> rownames(cn) <- 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"]
> cnConfidence <- callsConfidence <- matrix(NA, nrow = nrow(cn),
+     ncol = 1)
> colnames(cnConfidence) <- colnames(callsConfidence) <- colnames(cn)
> rownames(cnConfidence) <- rownames(callsConfidence) <- rownames(cn)
> snpset <- new("oligoSnpSet", copyNumber = cn, cnConfidence = cnConfidence,
+     calls = gt, callsConfidence = callsConfidence,
+     featureData = fd, annotation = "Illumina550k")
> stopifnot(validObject(snpset))
```

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

```
> gp <- getPar(snpset)
> gp$pch <- 21
> gp$cex <- 0.8
> gp$ylim <- c(-2, 2)
> gp$ylab <- "log2 ratio"
> plotSnp(gp, snpset)

NULL
```



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

```
> params <- new("HmmParameter", snpset, states = c("D",
+     "N", "L", "A"), cn.location = log2(c(1/2, 1,
+     1, 3/2)))
```

```
[1] "Calculating emission probabilities for genotype calls.... "
[1] "Calculating emission probabilities for copy number estimates... "
```

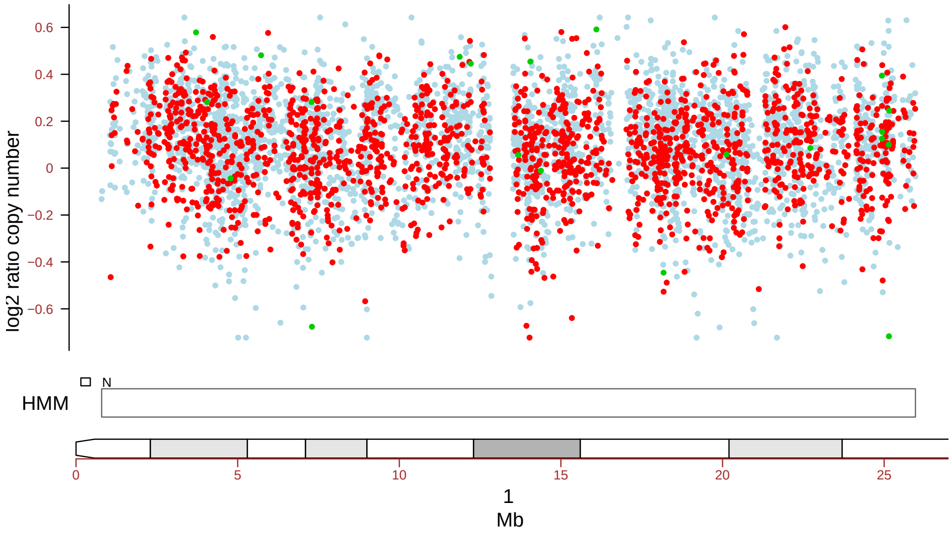
The HMM takes only two arguments: the parameter class and the data class:

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

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

```
> fn <- featureNames(params)
> idx <- match(fn, featureNames(snpset))
> snpset <- snpset[idx, ]
> gp <- getPar(fit, snpset)
> gp$cex <- 0.8
> gp$pch <- 21
> gp$ylab <- "log2 ratio copy number"
> plotSnp(object = gp, snpset = snpset, hmmPredict = fit)
```

```
[1] ""
```



Session Information

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

- R version 2.7.0 Under development (unstable) (2008-01-28 r44219), `powerpc-apple-darwin8.11.0`
- Locale: `C`
- Base packages: base, datasets, grDevices, graphics, methods, stats, tools, utils
- Other packages: Biobase 1.17.8, RColorBrewer 1.0-1, SNPchip 1.3.9, VanillaICE 1.1.5, oligoClasses 1.1.11