Tools for visualization of processed Affymetrix SNP chip data

Robert Scharpf, Jonathan Pevsner, Jason Ting, and Ingo Ruczinski

October 24, 2006

1 Introduction

SNPscan makes genome-wide plots of copy number and genotype calls from Affymetrix SNP chips.

Simple Usage

Getting the data

First, we load a list of matrices obtained from normal subject in the Hapmap project (Need ref) and processed by CRLMM (Need ref). For purposes of illustration, the hapmap data shown here only contains every 10th SNP from the Xba 50k chip.

```
> library(SNPscan)
```

```
KernSmooth 2.22 installed
Copyright M. P. Wand 1997
```

```
> data(hapmap)
```

Each matrix in the list contains probeset summaries (rows) by column (samples). It is important that the rownames of the above matrices are labeled by the Affymetrix probeset id. For instance,

```
> rownames(hapmap$calls)[1:5]
```

```
[1] "SNP_A-1747057" "SNP_A-1642733" "SNP_A-1650180" "SNP_A-1735038"
[5] "SNP_A-1711738"
```

To utilize the plotting methods in the *SNPscan*, we need to convert the above matrices to the classes of oligoSnpSet defined in *oligo* (Need Ref). An object of class `oligoSnpSet` can be obtained when both calls and copyNumber estimates are available. To begin, we create a `phenoData` object (in this case, we define all samples to be normal):

```
> df <- data.frame(rep(0, dim(hapmap$calls)[2]), row.names = colnames(hapmap$calls))
> colnames(df) <- c("normal")
> varMetadata <- data.frame("normal Refset", row.names = "normal")
> colnames(varMetadata) <- "labelDescription"
> ad <- new("AnnotatedDataFrame", data = df, varMetadata = varMetadata)
> snpset <- new("oligoSnpSet", phenoData = ad, calls = hapmap$calls,
+     callsConfidence = hapmap$callsConfidence, cnConfidence = hapmap$callsConfidence,
+     copyNumber = hapmap$copyNumber, annotation = "mapping100k")
> snpset
```

```
Instance of oligoSnpSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features  5850            5850         5850       5850
Samples      5               5            5          5

phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames:
  varLabels and descriptions:

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation [1] "mapping100k"
```

Converting output from Affymetrix CNAT software to objects of class oligoSnpSet is shown here:

```
> fname <- list.files()[1]
> cnat <- read.table(fname, as.is = TRUE, sep = "\t", header = TRUE,
+     row.names = 1, skip = 0)
> cn <- as.matrix(cnat[, grep("SPA_CN", colnames(x))])
> calls <- cnat[, grep("_Call", colnames(x))]
> calls[calls == "AA"] <- 1
> calls[calls == "AB"] <- 2
> calls[calls == "BB"] <- 3
> calls[calls == "NoCall"] <- 4
> calls <- matrix(as.integer(as.matrix(calls)), nc = dim(calls)[2],
+     byrow = FALSE)
> cnConfidence <- as.matrix(cnat[, grep("SPA_pVal", colnames(cnat))])
> callsConfidence <- as.matrix(cnat[, grep("LOH", colnames(cnat))])
> rownames(calls) <- rownames(cn) <- rownames(cnConfidence) <- rownames(callsConfidenc
> colnames(cn) <- colnames(calls) <- colnames(callsConfidence) <- colnames(cnConfidenc
+     1, 7)
> pdata <- data.frame(1)
> colnames(pdata) <- "family"
> rownames(pdata) <- colnames(calls)
> vmd <- data.frame("trio variable")
> rownames(vmd) <- colnames(pdata)
> colnames(vmd) <- "labelDescription"
> ad <- new("AnnotatedDataFrame", data = pdata, varMetadata = vmd)
> trios <- new("oligoSnpSet", calls = calls, copyNumber = copyNumber,
+     callsConfidence = callsConfidence, cnConfidence = cnConfidence,
+     phenoData = ad, annotation = "mapping100k")
```

Note the annotation slot contains information on whether the chip was 10k, 100k, or 500k – this information is used to load the proper annotation. The *SNPscan* plotting methods rely on annotation of the Affymetrix probeset identifiers. Annotation packages for SNP chip data are currently being developed at Bioconductor, but as a temporary solution we have posted static files here http://biostat.jhsph.edu/ iruczins/publications/sm/2006.scharpf.bioinfo/mapp The annotation slot of SnpSet ensures that the appropriate annotation file is downloaded and converted to an R object, but one could also do this manually. This may take several minutes depending on your internet connection.

```
> load(url("http://biostat.jhsph.edu/~iruczins/publications/sm/2006.scharpf.bioinfo/ma
```

This data should then be placed in the featureData slot of oligoSnpSet.

```
> tmp <- addFeatureData(snpset, path = "~/projects/software/snpscan2/")
> snpset <- addFeatureData(snpset)
> annSnpset <- as(snpset, "AnnotatedSnpSet")
> data(chromosomeAnnotation)
> chromosomeAnnotation(annSnpset) <- chromosomeAnnotation
```

```
> data(annSnpset)
> annSnpset
```

```
Instance of SnpCallSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features  5850            5850         5850       5850
Samples      5               5            5          5

phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames: 501741, 384421, 449441, ..., 432551, 432871 (5850 total)
  varLabels and descriptions:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
```

```
chr2        91748045      94748045      243018229
...
```

Mean-center copy number:

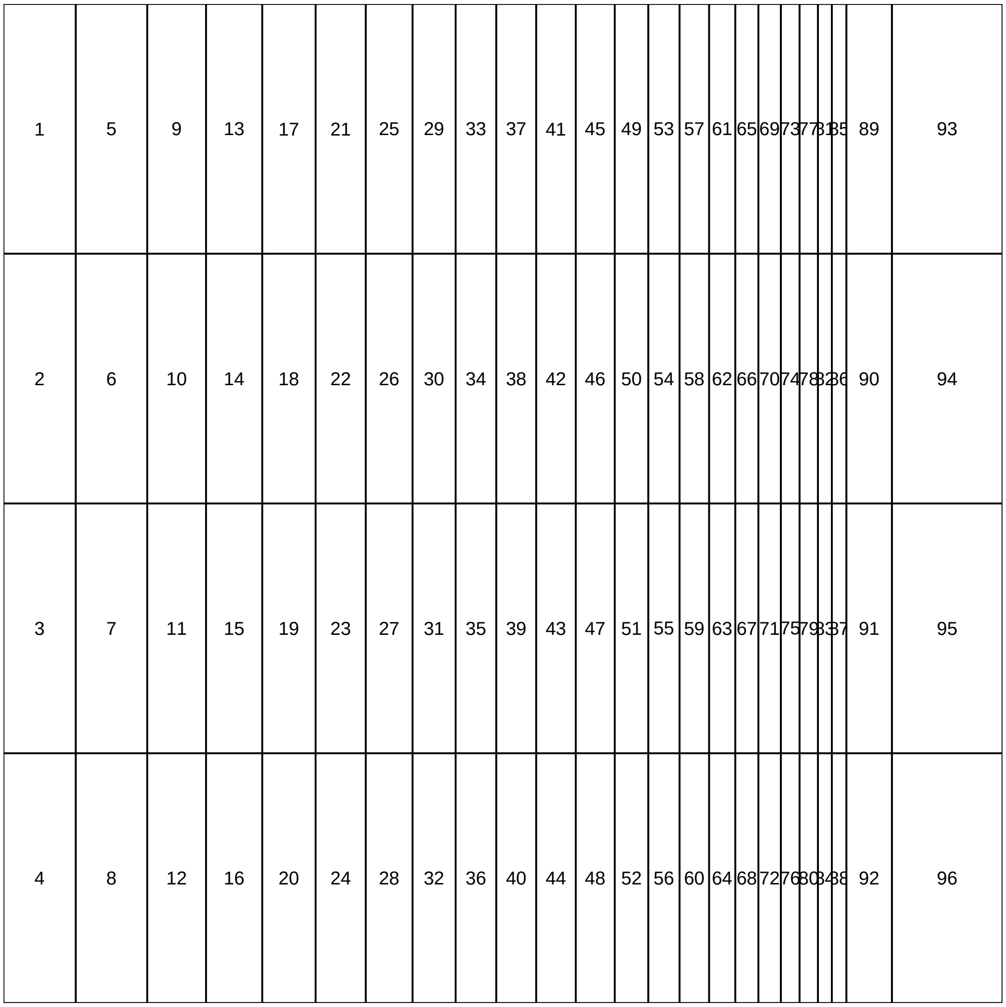
```
> copyNumber(annSnpset) <- base::scale(copyNumber(annSnpset), scale = FALSE)
```

Plotting the data

Plots of copy number versus physical position can be made for 1 or more chromosomes and one or more samples in the AnnotatedSnpSet object using the method `plotSnp`. Before making a plot of copy number versus physical position for all chromosomes and samples in the `AnnotatedSnpSet`, it is worthwhile to preview the layout for the graph. This can be done by setting the argument `plotIt` to FALSE. For instance,

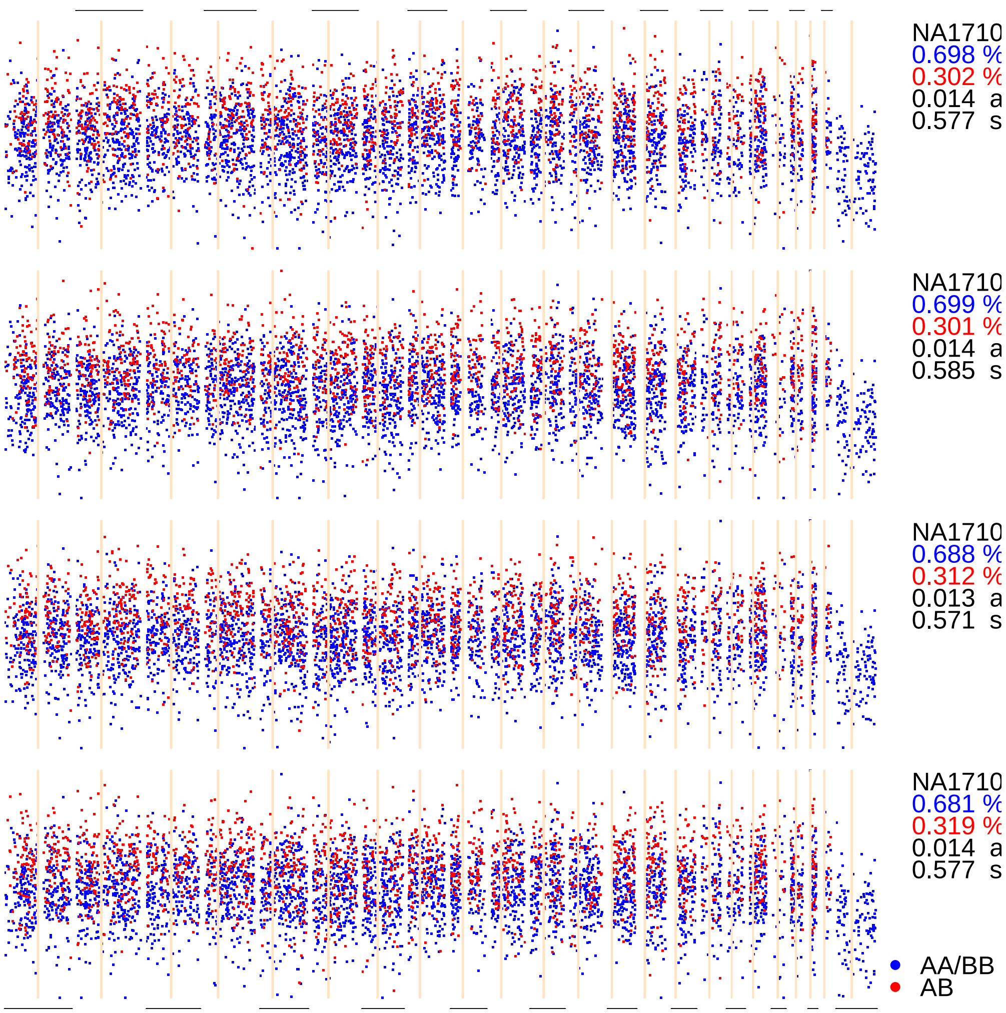
```
> plotSnp(chromosomes = 1:23, object = annSnpset, samples = 1:4,
+     oma = rep(0, 4), mar = rep(0.1, 4), width.left = 1.5, width.right = 8,
+     height.bottom = 0.8, cexAA = 2, cexAB = 2, plotIt = FALSE,
+     lwdChr = 1, cexChr = 1.1, summaryPanel = TRUE, cex.legend = 1.2)
```

```
NULL
```



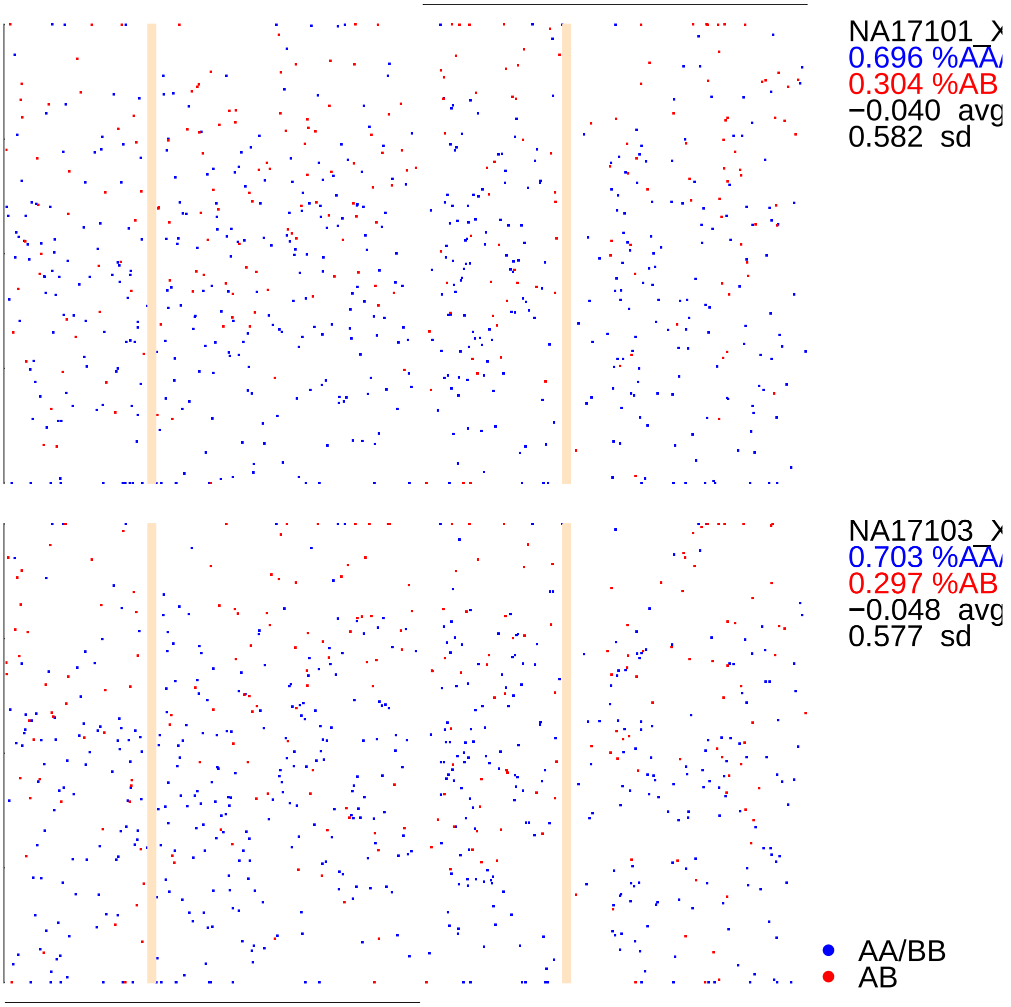
width.left specifies the size of the y-axis relative to the size of the smallest chromosome plotted. width.right specifies the size of the summary panel (if summaryPanel = TRUE) relative to the size of the smallest chromosome. height.bottom specifies the height of the x-axis at the bottom of the plot relative to the height of the samples. Hence, height.bottom = 1 gives the same space for the x-axis as for the samples (plotted by row).
To plot all chromosomes for the first 4 samples,

```
> plotSnp(annSnpset, 1:23, 1:4, oma = rep(0, 4), mar = rep(0.1,
+     4), width.left = 3, width.right = 9, height.bottom = 0.8,
+     cexAA = 2, cexAB = 2, plotIt = TRUE, lwdChr = 1, cexChr = 1.1,
+     summaryPanel = TRUE, cex.legend = 1)
```



To plot chromosomes 6 and 7 for samples 1 and 3.

```
> plotSnp(chromosomes = 6:7, object = annSnpset, samples = c(1,
+     3), oma = rep(0, 4), mar = rep(0.1, 4), width.left = 0.5,
+     width.right = 0.5, height.bottom = 0.1, cexAA = 2, cexAB = 2,
+     plotIt = TRUE, lwdChr = 1, cexChr = 1.1, summaryPanel = TRUE,
+     cex.legend = 1.2)
```



Summary

For each chromosome in the `AnnotatedSnpSet`, `summary` calculates the average and standard deviation of the copy number estimates, as well as the % homozygous and heterozygous calls. In addition, summary calculates the average copy number, standard deviation, % homozygous and heterozygous across all autosomes in the `AnnotatedSnpSet`. The dimensions of the four matrices are S x C + 1, where S is the number of samples and C is the number of chromosomes in the `AnnotatedSnpSet`.

```
> x <- summary(annSnpset)
> str(x)
```

```
List of 2
 $ chromosome:List of 4
  ..$ avgCopyNumber: num [1:5, 1:24] 0.0134 0.0281 0.0109 0.0111 0.0130 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "N
  .. .. ..$ : chr [1:24] "chr1" "chr2" "chr3" "chr4" ...
  ..$ sdCopyNumber : num [1:5, 1:24] 0.551 0.573 0.563 0.572 0.562 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "N
  .. .. ..$ : chr [1:24] "chr1" "chr2" "chr3" "chr4" ...
  ..$ propNoCalls  : num [1:5, 1:24] 0 0 0 0 0 0 0 0 0 0 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "N
  .. .. ..$ : chr [1:24] "chr1" "chr2" "chr3" "chr4" ...
  ..$ propHo       : num [1:5, 1:24] 0.702 0.704 0.706 0.665 0.700 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "N
  .. .. ..$ : chr [1:24] "chr1" "chr2" "chr3" "chr4" ...
 $ overall   : num [1:4, 1:24] 0.01530 0.00906 0.00000 0.69528 0.01352 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "overall mean" "sd of means" "avg prop no calls" "avg prop AA/BB am
  .. ..$ : chr [1:24] "chr1" "chr2" "chr3" "chr4" ...
```

Smoothing example

For further statistical analysis of copy number and genotype data, it is often convenient to work with AnnotatedSnpSets of chromosomes. `AnnotatedSnpSetList` is a list of `AnnotatedSnpSet`'s. This can be useful to produce smoothed estimates of copy number by applying a loess smoother to each chromosome, or each element in the `AnnotatedSnpSetList`. The following code chunk first assigns heterozygous calls to the integer 1 and homozygous calls to the integer zero. In this way, regions of deletions will have homozygous calls of zero. The following code chunk first simulated a deletion of 50 consecutive SNPs and then converts the `AnnotatedSnpSet` to an object of class `AnnotatedSnpSetList`.

```
> chrom <- paste("chr", 1:5, sep = "")
> sim <- annSnpset[chromosome(annSnpset) %in% chrom, 1:3]
> sim

Instance of SnpCallSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features  2215            2215         2215       2215
Samples      3               3            3          3
```

```
phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames: 501741, 384421, 449441, ..., 271691, 271851 (2215 total)
  varLabels and descriptions:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
...

> calls(sim) <- ifelse(calls(sim) == 2, 1, 0)
> copyNumber(sim)[101:150, 1] <- copyNumber(sim)[101:150, 1] -
+     1
> calls(sim)[101:150, 1] <- 0
> sim.list <- as(sim, "AnnotatedSnpSetList")
> snpSetList(sim.list)[1]

[[1]]
Instance of SnpCallSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features   466              466          466        466
Samples      3                3            3          3

phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames: 501741, 384421, 449441, ..., 350401, 353201 (466 total)
  varLabels and descriptions:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
...
```

We can now do the smoothing over all chromosomes and samples in the object as follows:

```
> smoothChromosome <- function(obj, span) {
+     loessX <- function(X, location, span) {
+         fit <- loess(X ~ location, span = span)$fitted
+         return(fit)
+     }
+     cn.smooth <- apply(copyNumber(obj), 2, loessX, position(obj),
+         span = span)
+     call.smooth <- apply(calls(obj), 2, loessX, location = position(obj),
+         span = span)
+     copyNumber(obj) <- cn.smooth
+     calls(obj) <- call.smooth
+     obj
+ }
> obj.list <- snpSetList(sim.list)
> obj.list[1]

[[1]]
Instance of SnpCallSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features   466             466          466        466
Samples      3               3            3          3

phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames: 501741, 384421, 449441, ..., 350401, 353201 (466 total)
  varLabels and descriptions:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation [1] "mapping100k"

chromosomeAnnotation
```

```
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
...

> sim.smooth <- sim.list
> sim.smooth@snpSetList <- lapply(obj.list, smoothChromosome, span = 1/10)
> smooth.obj <- as(sim.smooth, "AnnotatedSnpSet")
> smooth.obj

Instance of SnpCallSet

assayData
  Storage mode: lockedEnvironment
  Dimensions:
         calls callsConfidence cnConfidence copyNumber
Features  2215            2215         2215       2215
Samples      3               3            3          3

phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and descriptions:
    normal: normal Refset

featureData
  rowNames: 501741, 384421, 449441, ..., 271691, 271851 (2215 total)
  varLabels and descriptions:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position

Experiment data
  Experimenter name:
  Laboratory:
  Contact information:
  Title:
  URL:
  PMIDs:
  No abstract available.

Annotation character(0)

chromosomeAnnotation
```

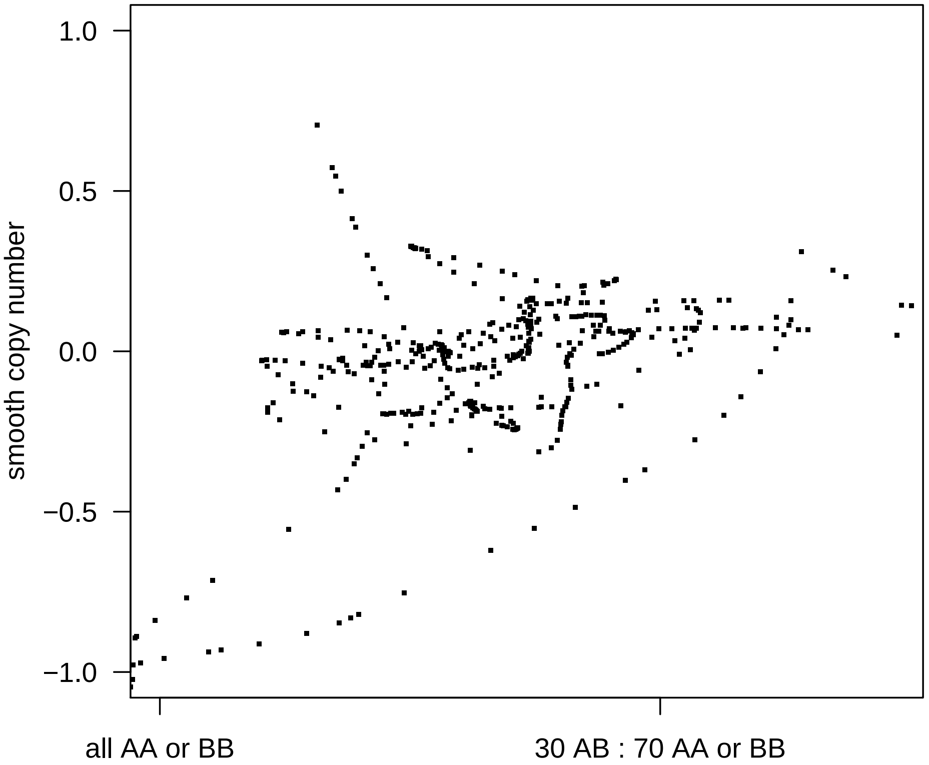
```
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
...
```

The methods smoothSnp takes an object of class AnnotatedSnpSet and does the above automatically.

```
> smooth.obj2 <- smoothSnp(chromosomes = 1:5, object = sim, samples = 1:3)
> identical(copyNumber(smooth.obj), copyNumber(smooth.obj2))
```

A plot of the smoothed calls versus copynumber can be used to visualize the deletion and deciding on a threshold for calling deletions.

```
> par(las = 1, mfrow = c(1, 1))
> plot(calls(smooth.obj)[chromosome(smooth.obj) == "chr1", 1],
+     copyNumber(smooth.obj)[chromosome(smooth.obj) == "chr1",
+         1], ylim = c(-1, 1), pch = ".", cex = 3, xlab = "", ylab = "smooth copy numb
+     xaxt = "n", xlim = c(0, 30/70 + 0.2))
> axis(side = 1, at = c(0, 30/70), labels = c("all AA or BB", "30 AB : 70 AA or BB"))
```



To retreive additional annotation on the known SNP's in the region of this simulated deletion, we could use the *RSNPper*. For instance,

```
> library(RSNPper)
> x <- as.character(c(position(smooth.obj)[101], position(smooth.obj)[110]))
> itemsInRange(item = "countsnps", chr = "chr1", start = x[1],
+     end = x[2])
```

To find all the genes in the region of the deletion, and then find additional annotation on the SNPs that these genes carry:

```
> gir <- itemsInRange(item = "genes", chr = "chr1", start = x[1],
+     end = x[2])
> f <- function(x) {
+     allGeneMeta(geneInfo(x["NAME"]))["GENEID"]
+ }
> id <- lapply(gir[1:5], f)
> str(id)
> snpinfo <- geneSNPs("817")
> names(snpinfo[[1]])
> snpinfo[[1]]["ROLE"]
```