Tools for high throughput SNP chip data

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

November 15, 2006

Introduction

SNPchip defines classes and methods useful for organizing high throughput genomic data. The classes defined here extend the `eSet` class in *Biobase*, utilizing the existing Bioconductor infrastructure for organizing high dimensional genomic data. This provides a foundation upon which statistical and visualization tools can be further developed.

1 Simple Usage

We illustrate the structure of the class `AnnotatedSnpSet` with a small dataset provided with the package.

```
> library(SNPchip)

KernSmooth 2.22 installed
Copyright M. P. Wand 1997

> data(annSnpset)
> annSnpset

SnpCallSet (storageMode: lockedEnvironment)
assayData: 5850 features, 5 samples
  element names: calls, callsConfidence, cnConfidence, copyNumber
phenoData
  sampleNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, ..., NA17105_
  varLabels and varMetadata: none
featureData
  rowNames: 501741, 384421, ..., 432871 (5850 total)
  varLabels and varMetadata:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position
experimentData: use 'experimentData(object)'
Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
...
     centromereStart centromereEnd chromosomeSize
chrY        11237315      12237315       57701691
```

annSnpset is an instance of the AnnotatedSnpSet class. Here, the assayData slot in AnnotatedSnpSet contains 5850 SNPs with estimates of copy number and genotype calls, as well as a corresponding confidence score. Typically, such an object would contain 100,000 - 500,000 estimates of genotype calls and copy number. We illustrate in Section 3 how to create an instance of AnnotatedSnpSet from probe-level summaries of SNP chip data. In addition to estimates of genotype call and copy number, the annSnpset contains both chromosome-level annotation, as well as SNP-level annotation. The chromosome-level annotation includes the centromere start and stop sites and chromosome size (in number of base pairs), and could be extended to include location of cytobands, or any other feature of a chromosome.

```
> data(chromosomeAnnotation)
> chromosomeAnnotation[1:5, ]
```

```
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
chr3        90587544      93487544      199505740
chr4        49501045      52501045      191411218
chr5        46441398      49441398      180857866
```

We provide SNP-level annotation that can be downloaded as static tables maintained at our website. The method addFeatureData checks the annotation slot of the AnnotatedSnpSet object, and then downloads the appropriate static table and stores this data in the featureData slot.

```
> annotation(annSnpset)
```

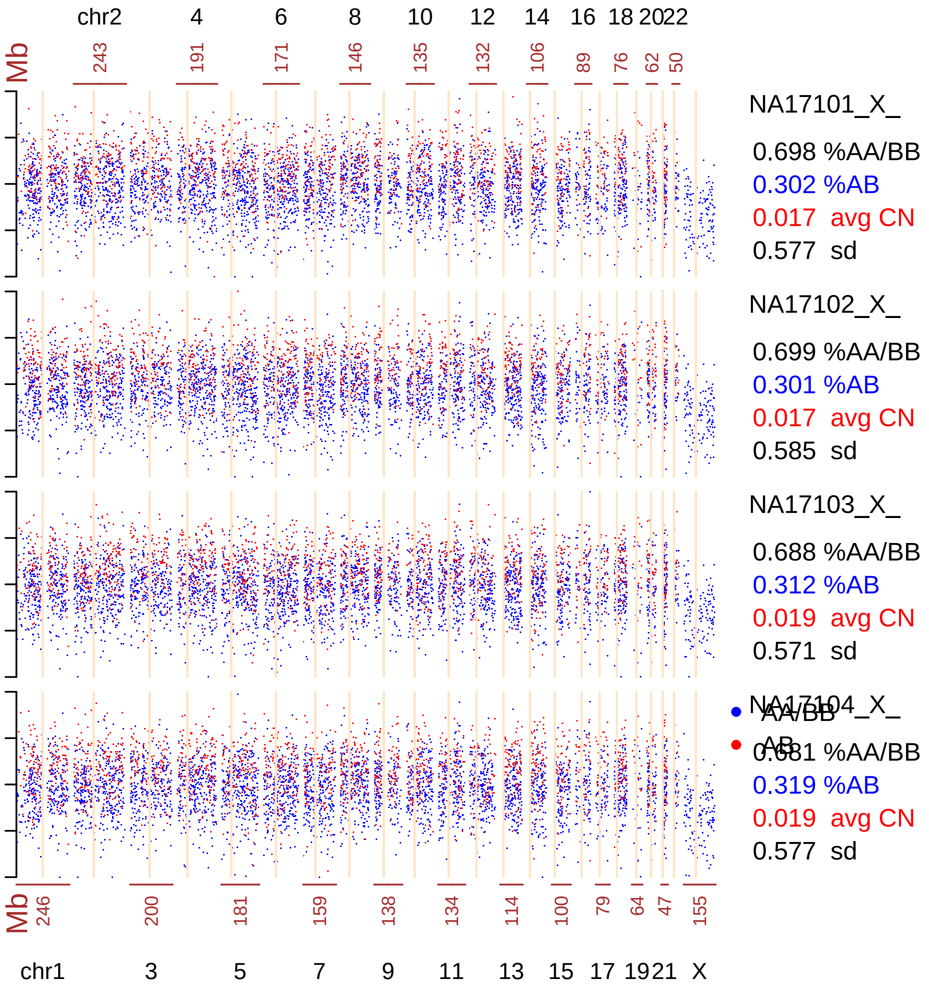
```
[1] "mapping100k"
```

```
> addFeatureData(annSnpset)
```

As annotation packages for SNP chips become more fully developed and supported, the use of static tables for SNP annotation will be deprecated.

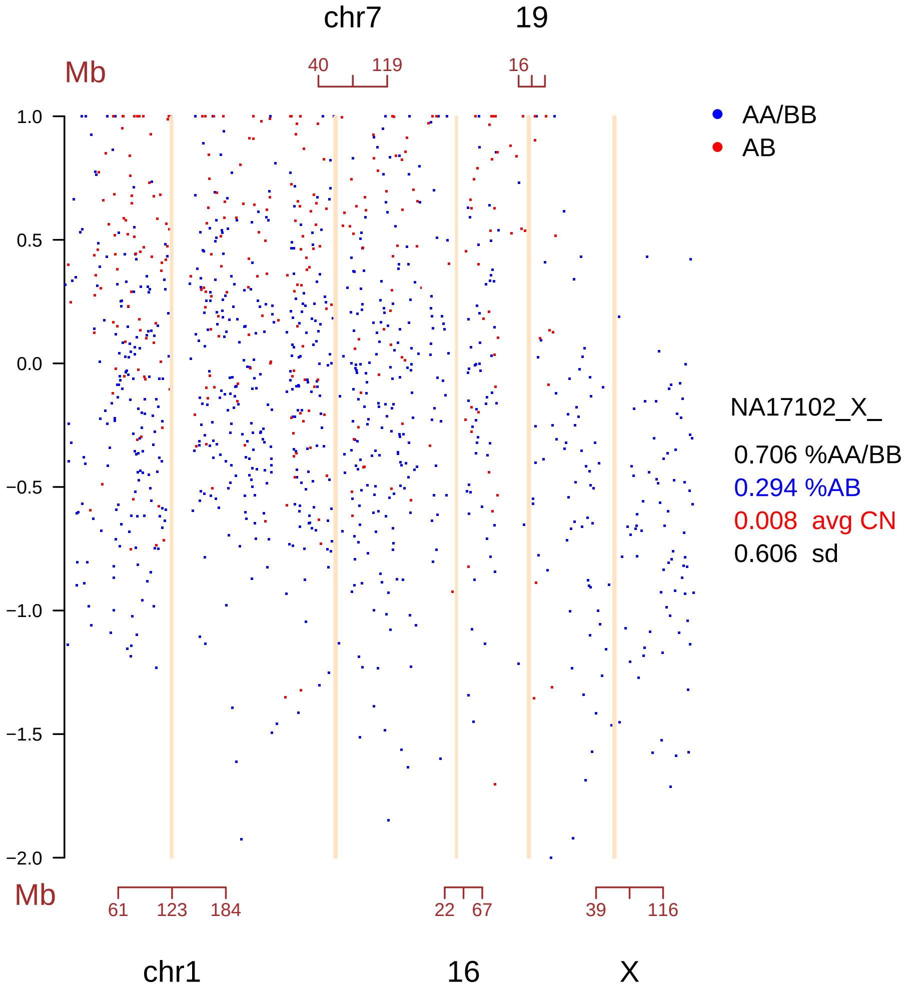
A genome-wide view of copy number and genotype calls versus physical position can be made using `plotSnp`. Here, we plot chromosomes 1-22 and X (the integer 23 is used to represent X) of samples 1 - 4 in the object `annSnpset`:

```
> plotSnp(annSnpset, 1:23, 1:4, summaryPanel = TRUE, width.right = 20,
+     cex.chr = 0.6)
```



The copy number estimates have been centered to have mean zero – a centered copy number of 0 corresponds to a copy number of two. The default plot layout generally works well, but can be adjusted through the arguments mar, oma, and width.right in `plotSnp`. The latter argument specifies how much room to allow for the summary panel relative to the size of the smallest chromosome plotted. For instance, if plotting chromosomes 1-22 and X, width.right set to 15 allows a plotting region for the summary paenl that is 15 times larger than chromosome 21. A more focused view of chromosomes 1, 7, 16, 19, and X of sample 2 could be obtained by

```
> plotSnp(annSnpset, c(1, 7, 16, 19, 23), 2, summaryPanel = TRUE,
+     cexAA = 2, cexAB = 2, width.right = 5)
```



The size of the plotted estimates can be adjusted by the genotype call using the arguments cexAA, cexAB, and cexNC for homozygous AA or BB, heterozygous AB, and no call, respectively.

The R functions `plotChromosome` and `plotCytoband` are useful for visualization of 1 chromosome in sample 5. The following code chunk illustrates how easy it is to subset 1 chromosome and one sample, and then plot the resulting `AnnotatedSnpSet` object. Because an additional plotting region is needed to plot the cytoband, we specify two rows in the `layout`.

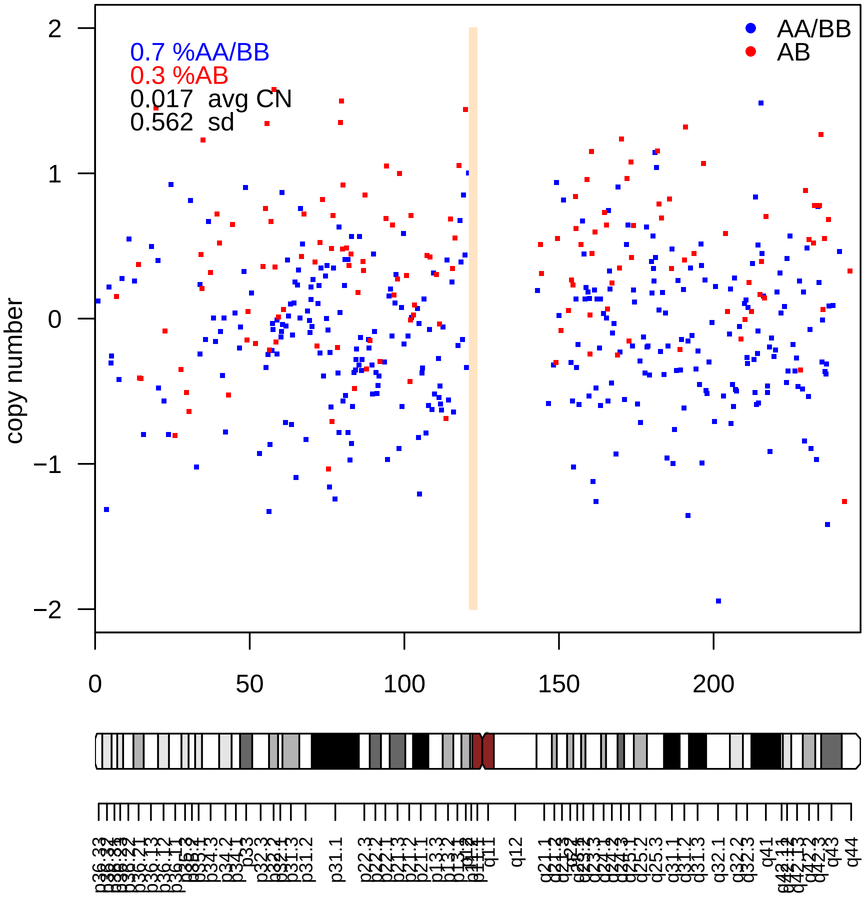
```
> data(cytoband)
> chr1 <- annSnpset[chromosome(annSnpset) == "chr1", 5]
> chr1
```

```
SnpCallSet (storageMode: lockedEnvironment)
assayData: 466 features, 1 samples
```

```
  element names: calls, callsConfidence, cnConfidence, copyNumber
phenoData
  rowNames: NA17105_X_hAF_A5_4000091.CEL
  varLabels and varMetadata: none
featureData
  rowNames: 501741, 384421, ..., 353201 (466 total)
  varLabels and varMetadata:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position
experimentData: use 'experimentData(object)'
Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
...
     centromereStart centromereEnd chromosomeSize
chrY        11237315      12237315       57701691
```

```
> layout(matrix(1:2, nr = 2, nc = 1), heights = c(1, 0.05))
> par(oma = c(6, 4, 0, 3))
> plotChromosome(chr1, mar = c(3, 0, 1, 0), colCentromere = "bisque",
+     cexAA = 3, cexAB = 3, cexNC = 3, panel.xaxis = TRUE, xlab = "Mb",
+     cex.axis = 0.7, cex.legend = 0.7)
> par(las = 3)
> mtext("copy number", 2, line = 2, outer = TRUE, cex = 0.7)
> par(las = 1)
> plotCytoband(chr1, cytoband, cex.axis = 0.5)
```



2 Available annotation

Currently, we provide annotation via static tables to download from our website. Bioconductor is developing annotation packages that will be current and more memory efficient. Here we provide a list of the annotation that is currently available for each SNP.

```
> try(load(url("http://biostat.jhsph.edu/~iruczins/publications/sm/2006.scharpf.bioinf
> colnames(mapping10k$annotation)

 [1] "probeSetId"                     "Affy.SNP.ID"
 [3] "dbSnpId"                        "chrom"
 [5] "Genome.Version"                 "DB.SNP.Version"
 [7] "start"                          "strand"
 [9] "ChrX.pseudo.autosomal.region" "Cytoband"
[11] "Flank"                          "Allele.A"
```

```
[13] "Allele.B"                     "Associated.Gene"
[15] "Genetic.Map"                  "Microsatellite"
[17] "Fragment.Length.Start.Stop"   "Freq.Asian"
[19] "Freq.AfAm"                    "Freq.Cauc"
[21] "Het.Asian"                    "Het.AfAm"
[23] "Het.Cauc"                     "af.asian.called"
[25] "af.afam.called"               "af.cauc.called"
[27] "alleles"                      "enzyme"
```

3 High throughput SNP classes

All that is needed to create an instance of `AnnotatedSnpCallSet` or `AnnotatedSnpCopyNumberSet`is a matrix of genotype calls and copy number, respectively, and their corresponding confidence scores. If estimates of both copy number and genotype calls are available, we can create an `AnnotatedSnpSet` that inherits methods from both `AnnotatedSnpCopyNumberSet` and `AnnotatedSnpCallSet`. In this way, *SNPchip* is completely independent of the pre-processing method used to produce probe-level summaries. To illustrate, the following code chunk loads a list of matrices obtained from normal subjects in the Hapmap project and pre-processed by CRLMM (B. Carvalho *et. al*, *Biostatistics*, in press). Only every 10th SNP from the Xba 50k chip is included in the matrices.

```
> data(hapmap)
> str(hapmap)
```

```
List of 3
 $ calls          : int [1:5850, 1:5] 1 1 2 1 1 2 2 3 2 3 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5850] "SNP_A-1677174" "SNP_A-1705537" "SNP_A-1737197" "SNP_A-1665024"
  .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "NA17
 $ callsConfidence: num [1:5850, 1:5]  308 1292  263  236  671 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5850] "SNP_A-1677174" "SNP_A-1705537" "SNP_A-1737197" "SNP_A-1665024"
  .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "NA17
 $ copyNumber     : num [1:5850, 1:5]  0.190 -1.300  0.155 -0.386 -0.298 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5850] "SNP_A-1677174" "SNP_A-1705537" "SNP_A-1737197" "SNP_A-1665024"
  .. ..$ : chr [1:5] "NA17101_X_hAF_A1_4000091.CEL" "NA17102_X_hAF_A2_4000091.CEL" "NA17
```

Each matrix in the list contains probeset summaries (rows) by column (samples). Currently, we only provide annotation for the Affymetrix SNP chips and so the rownames of the matrices should be Affymetrix probeset id's. For purposes of visualization, an identifier from any technology could be used so long as the SNP-level annotation stored in the featureData slot is a data.frame with columns `Physical.Position` and `chromosome`.

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

[1] "SNP_A-1677174" "SNP_A-1705537" "SNP_A-1737197" "SNP_A-1665024"
[5] "SNP_A-1667950"

> snpset <- new("AnnotatedSnpSet", calls = hapmap$calls, callsConfidence = hapmap$call
+     cnConfidence = hapmap$callsConfidence, copyNumber = hapmap$copyNumber,
+     annotation = "mapping100k", chromosomeAnnotation = chromosomeAnnotation)
```

If you have an internet connection, you can add the SNP-level annotation:

```
> annSnpset <- addFeatureData(snpset)
```

This may take several minutes depending on your internet connection. To do this manually, the annotation files can be downloaded from `http://biostat.jhsph.edu/ iruczins/publications/sm/2` Then, specifying the path to the downloaded files in the `addFeatureData` function:

```
> annSnpset <- addFeatureData(snpset, path = "./")
```

Below, we illustrate how one might convert output from Affymetrix CNAT software to an object of class `AnnotatedSnpSet`. For instance, any one of the .txt files for the CEPH trios provided at the Affymetrix website can be converted as follows

```
> fname <- "100k_trios.Hind.1.txt"
> 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)
> trios <- new("AnnotatedSnpSet", calls = calls, copyNumber = copyNumber,
+     callsConfidence = callsConfidence, cnConfidence = cnConfidence,
+     annotation = "mapping100k", chromosomeAnnotation = chromosomeAnnotation)
```

The SNP-level annotation for the trios data can be retrieved as described previously.

4 Descriptive and statistical summaries

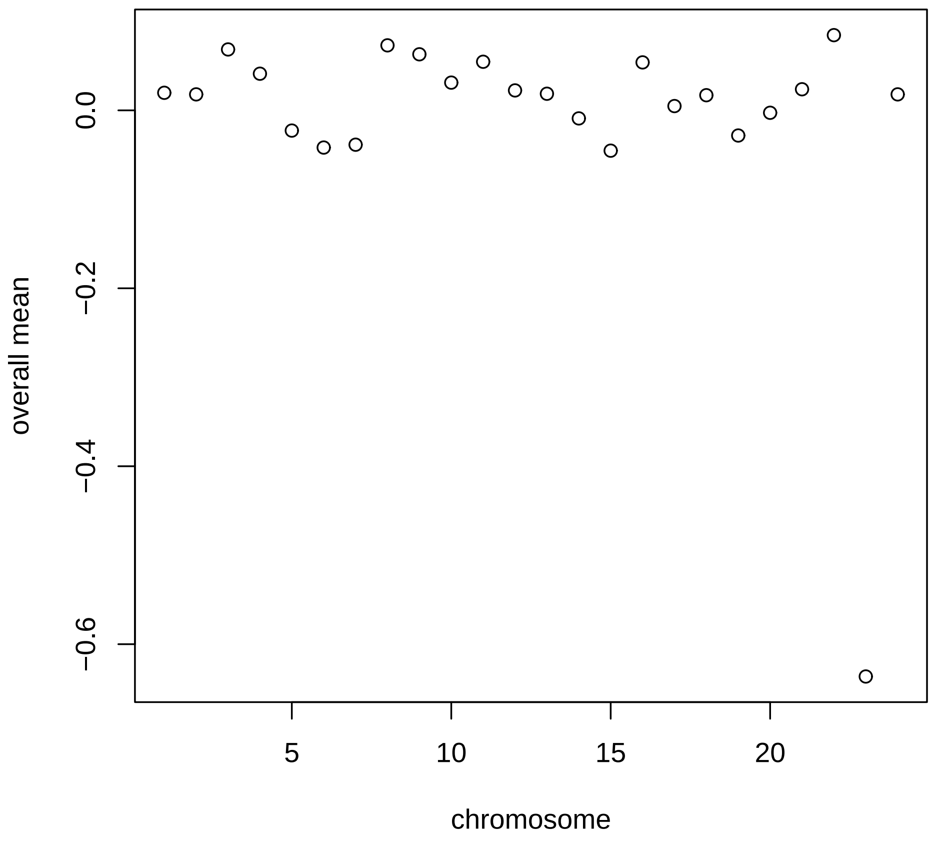
Descriptive statistics for copy number and genotype calls are provided with the `summary` method. For each chromosome in the `AnnotatedSnpSet`, `summary` calculates the average and standard deviation of the copy number estimates, as well as the % homozygous and heterozygous calls. In addition, summary 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] 0.0168 0.0313 0.0165 0.0167 0.0173 ...
  .. ..- 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] 0.01971 0.00906 0.00000 0.69528 0.01793 ...
  ..- 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" ...
```

Plot the overall means:

```
> plot(x[[2]][1, ], xlab = "chromosome", ylab = "overall mean")
```



Smoothing example

The basic unit for all of the above visualization tools and summary methods is an `AnnotatedSnpSet` of a single chromosome. For instance, `plotSnp` converts the `AnnotatedSnpSet` to a list of `AnnotatedSnpSet`, where each element in the list is an `AnnotatedSnpSet` of a single chromosome. In the code below, we are interested in a quick method for smoothing copy number estimates for each chromosome and apply a loess smoother to each element in an object of class `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. We simulated a deletion of 50 consecutive SNPs and then converted the `AnnotatedSnpSet` to an object of class `AnnotatedSnpSetList`.

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

```
SnpCallSet (storageMode: lockedEnvironment)
assayData: 2215 features, 3 samples
  element names: calls, callsConfidence, cnConfidence, copyNumber
phenoData
  rowNames: NA17101_X_hAF_A1_4000091.CEL, NA17102_X_hAF_A2_4000091.CEL, NA17103_X_hAF_A3
  varLabels and varMetadata: none
featureData
  rowNames: 501741, 384421, ..., 271851 (2215 total)
  varLabels and varMetadata:
    Probe.Set.ID: Probe.Set.ID
    dbSNP.RS.ID: dbSNP.RS.ID
    Chromosome: Chromosome
    Physical.Position: Physical.Position
experimentData: use 'experimentData(object)'
Annotation [1] "mapping100k"

chromosomeAnnotation
     centromereStart centromereEnd chromosomeSize
chr1       121147476     123387476      245522847
chr2        91748045      94748045      243018229
...
     centromereStart centromereEnd chromosomeSize
chrY        11237315      12237315       57701691
```

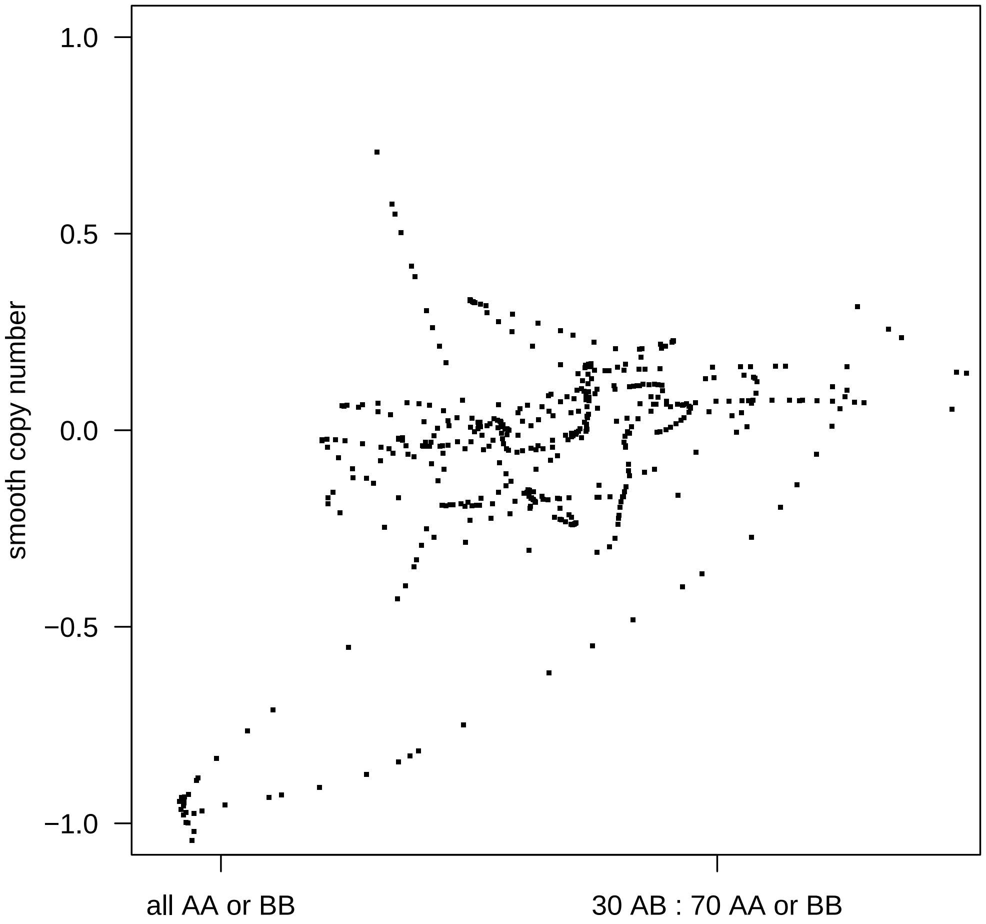
```
> copyNumber(sim)[101:150, 1] <- copyNumber(sim)[101:150, 1] -
+     1
> calls(sim)[101:150, 1] <- 0
```

The `smoothSnp` converts an object of class `AnnotatedSnpSet` to an `AnnotatedSnpSetList` and fits a loess smoother to each element in the list:

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

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, mar = c(4, 4, 0.5, 0.5), oma = rep(0, 4))
> 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(-0.05, 30/70 + 0.2))
> axis(side = 1, at = c(0, 30/70), labels = c("all AA or BB", "30 AB : 70 AA or BB"))
```



5 Integration with other Bioconductor packages

To retreive additional annotation on the known SNP's in the region of this simulated deletion, we could use the *RSNPper*. The installation instructions for *RSNPper* is available at Bioconductor.

```
> 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"]
```