The CopyNumber450k User's Guide: Calling CNVs from Illumina 450k Methylation Arrays

Simon Papillon-Cavanagh[1,3], Jean-Philippe Fortin[2], and Nicolas De Jay[3]

[1]McGill University and Genome Quebec Innovation Centre, Montreal, QC, Canada
[2]Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, USA
[3]Department of Human Genetics, McGill University, Montreal, QC, Canada

October 13, 2015

Contents

1 Introduction

The CopyNumber450k package provides tools for calling copy number variantions (CNVs) from Illumina 450k methylation arrays. CopyNumber450k uses the sum of methylated and unmethylated alleles as a proxy of DNA molecule abundance. Aberrant intensities is used as an indication of putative CNVs. CopyNumber450k provides a set of user-friendly methods to infer, plot and assess statistical significance of CNVs, as described below.

1.1 Loading libraries

```
> library(CopyNumber450k)
> library(CopyNumber450kData)
> library(minfiData)
```

2 Example Dataset

The starting point of CopyNumber450k is an `RGChannelSet`: an object that must be created with the minfi package directly from the raw IDAT files. To learn how to create such an object, please refer to the detailed vignette of the minfi package. The `RGChannelSet` object is read into a `CNV450kSet` object, which internally extracts the needed data and annotation. We strongly suggest that the reader consults minfi's vignette on how to load data and to use quality control plots to ensure that the data is of acceptable quality [3].

2.1 Data requirements

In order to construct a `CNV450kSet` object, the input sample set must contain at least three samples whose associated `Sample_Group` phenotypic data value is `control`. A sample whose `Sample_Group` value is not `control` is considered to be a case sample and will be analyzed and segmented.

In order to facillitate the use of this package, we provide 52 control samples in the CopyNumber450kData R/Bioconductor package. We strongly suggest that you use those controls if you do not have sufficient control samples in your data set.

```
> # Load control data (n=52) from CopyNumber450kData
> data(RGcontrolSetEx)
> # Load example data (n=6) from minfiData
> data(RGsetEx)
> # In order to reduce example time, let's use only one sample
> # and 30 controls (instead of 52). In real life situations, it is advised to
> # use all the available controls.
> RGsetEx <- RGsetEx[, 5]
> RGcontrolSetEx <- RGcontrolSetEx[, sample(1:ncol(RGcontrolSetEx), 30)]
```

2.2 Creating the object

Most users will directly load their `RGChannelSet` using the `CNV450kSet(RGset)` method. However, the minfi package provides the `combine` function, a helper method which merges two `RGChannelSets` into a single object. This method may be used when using the CopyNumber450kData control set in order to combine it with the user's data set.

It is important to ensure that samples have been assigned to relevant groups (i.e. control samples have `control` as `Sample_Group`). Note that the provided control samples in CopyNumber450kData are already properly annotated.

Please note that in the following code example we will use only a subset of probes to proceed. This is to be avoided in real life cases as it will significantly reduce the resolution and will output unrealistic results by artificially smoothing the segmentation.

```
> # Ensure that Sample_Group values are valid
> head(pData(RGcontrolSetEx)$Sample_Group)
```

```
[1] "control" "control" "control" "control" "control" "control"
> head(pData(RGsetEx)$Sample_Group)
[1] "GroupA"
> # Combine both RGsets in a single RGset
> RGset <- combine(RGcontrolSetEx, RGsetEx)
> # Create the object
> mcds <- CNV450kSet(RGset)
> # In order to speed up example computation, we will randomly subset the
> # probes used by CopyNumber450k. THIS SHOULD NEVER BE DONE AS IT SERVES
> # ONLY FOR SPEEDING UP THE EXAMPLE.
> mcds <- mcds[sample(1:nrow(mcds), 10000), ]
```

2.3 SNP probes

Single nucleotide polymorphisms (SNPs) affect probe hydridization affinity and may therefore artificially skew intensity levels. Hence, it is useful to drop the probes that contain or target SNPs. Although minfi offers a function to drop SNP probes, it does so only for probes that directly target SNP CpGs. We recommend that the CopyNumber450k `dropSNPprobes` method be used with our package. Since `dropSNPprobes` drops probes that both target and contain known SNPs, it leads to a more representative normalization and segmentation output.

```
> mcds <- dropSNPprobes(mcds, maf_threshold=0.01)
```

3 Data Visualization and Quality Assessment

CopyNumber450k offers quality assessment plots to identify potential batch and normalization effects.

3.1 Density Plot

The density plot allows the user to observe differences in density distributions between samples. Certain effects, such as array position on the chip, can be clearly observed using the density plot (Figure 1). Additionnaly, we offer coloring features which can be based on different sample phenotypic features. Please refer to the method documentation for all possible groupings.

```
> plotDensity(mcds, main="Pre-normalization density plot",
+          color.by="array.row")
```

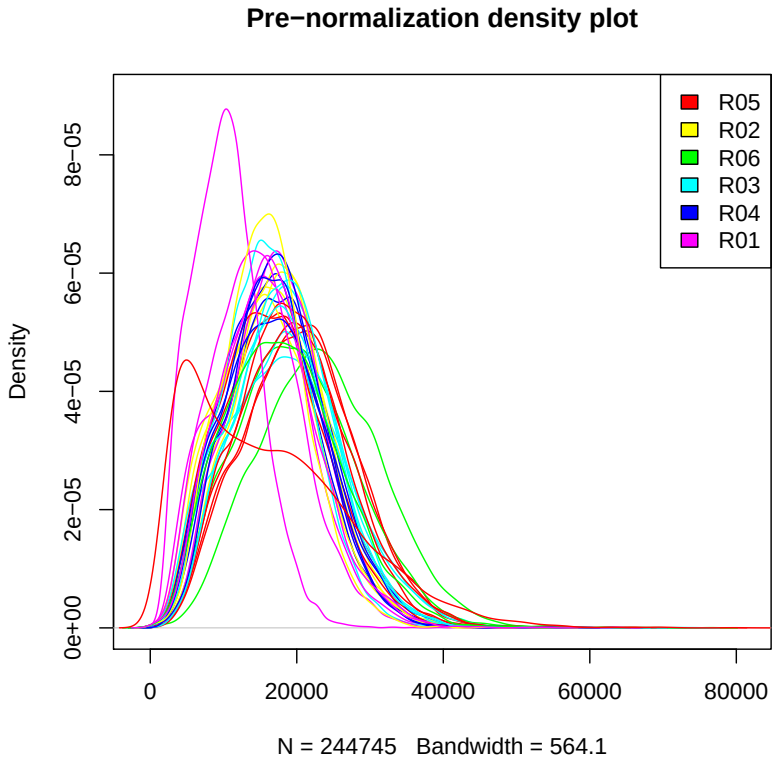


Figure 1: Global sample intensity by array position

3.2 Principal Component Analysis (PCA) Plot

A PCA plotting method, with the same coloring features as `plotDensity` is provided (Figure 2). This method may allow users to identify confounding factors influencing data variability, such a chip bias or batch effects.

4 Normalization

Currently, CopyNumber450k offers two normalization procedures: quantile normalization [1] and functional normalization [2]. Underlying quantile normalization is the assumption that the distributions of signal intensities are similar across subjects[1]. Therefore, we suggest that this method be used for datasets in which no global differences in CNVs across samples are expected.

4.1 Quantile normalization

The quantile normalization will match all the samples distribution and mostly remove any batch effects [1]. However, in samples with gross chromosomal aberrations, the signal may be lost. We suggest that you use the quantile normalization in samples where you expect small aberrations.

```
> mcds.q <- normalize(mcds, "quantile")
```

4.2 Functional normalization

In situations where large-scale differences are expected (e.g. in rare cases of polyploidy), we recommend the functional normalization, a new method developed specifically for the 450k array in which the similarity of distributions between samples is not assumed [2]. In this sense functional normalization is more conservative than quantile normalization.

Compared to the previous `densityPlot` (Figure 1), post-normalization density distributions are much more similar to one another and the array position bias is removed (Figure 3).

5 Segmentation

After normalization, it is necessary to bin probes together in order to identify possibly amplified or deleted genomic segments. We use a binary circular

```
> plotPCA(mcds, main="Pre-normalization PCA plot",
+         color.by="sample.group")
```

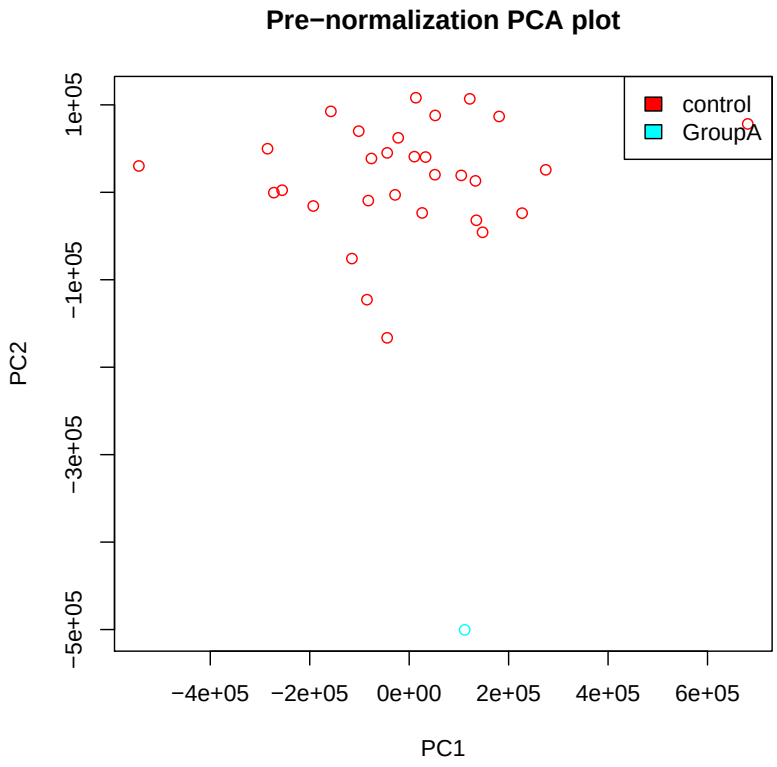


Figure 2: PCA plot of sample intensities

```
> mcds.f <- normalize(mcds, "functional")
> plotDensity(mcds.f, main="Density plot of functional normalized data")
```

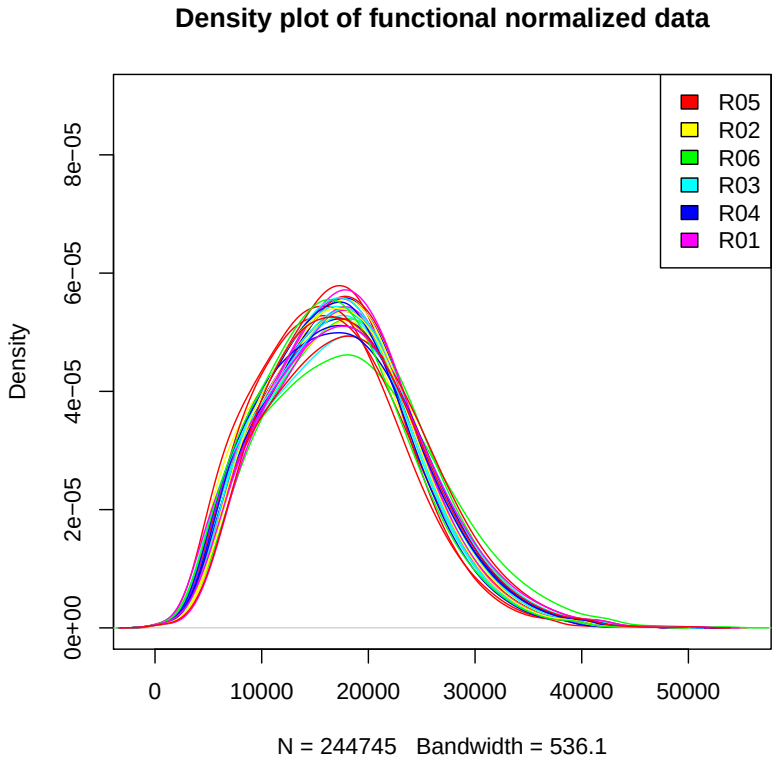


Figure 3: Density plot of sample intensities after functional normalization

segmentation algorithm provided by the DNAcopy package [4]. After segmentation, each segment is further processed by comparing its median intensity to a distribution inferred from the control samples. This allows us to assess whether the segment intensity value is attributable to random variability or if it is significantly different from what would be expected by chance. Finally, each segment is annotated with the genes it contains.

For simplicity, let us assume that the assayed sample does not contain gross aberrations and proceed with the object that has been normalized with the quantile normalization (`mcds.q`).

```
> mcds.q <- segmentize(mcds.q)
```

```
Analyzing: GroupA_1
```

6 Results

Provided that the segmentation has been performed, calling `getSegments` on the object returns a list containing the genomic segments in which each row represents a genomic segment found in a sample.

6.1 Text format

The segments can be saved in a CSV format, by using the common `write.csv` function that was redefined to `CNV450kSet` objects.

```
> write.csv(mcds.q, file="segments.csv")
```

6.2 A plot of the sample's genome

In order to obtain a visual representation of the sample's genome, CopyNumber450k offers the `plotSample` function, which takes the segmentized `CNV450kSet` object and the index of the sample to plot as input. In this example, since we randomly subsetted the probes, the resolution will be too low to find any interesting results. (Figure 4).

The user may specify a region of interest and plot only this region to observe chromosomal breakpoints, as illustrated by the focal amplification on chr1 in the assayed sample (Figure 6.2).

```
> plotSample(mcds.q, 1, main="Genomic view of Sample 1")
```

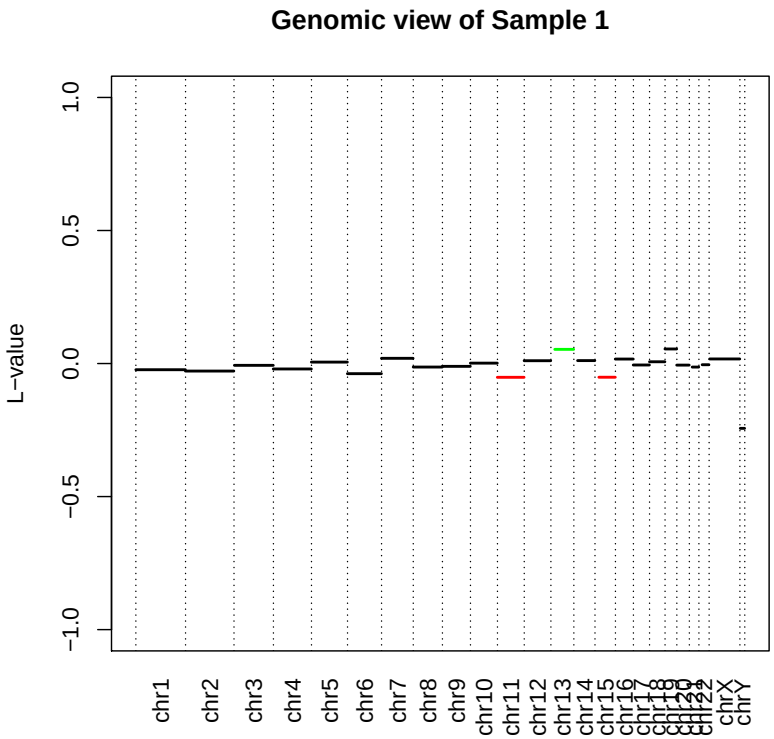
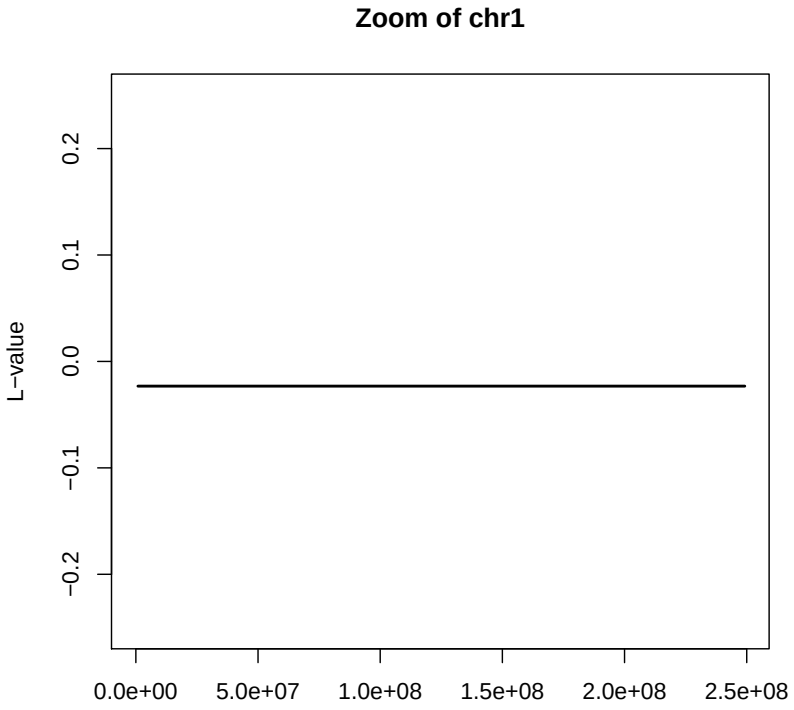


Figure 4: Genomic view of a sample

```
> plotSample(mcds.q, 1, chr="chr1",
+         main="Zoom of chr1", ylim=c(-.25,.25))
```



7 SessionInfo

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

- R version 3.2.2 (2015-08-14), `x86_64-apple-darwin13.4.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.30.0, BiocGenerics 0.16.0, Biostrings 2.38.0, CopyNumber450k 1.6.0, CopyNumber450kData 1.5.0, DNAcopy 1.44.0, GenomeInfoDb 1.6.0, GenomicRanges 1.22.0, IRanges 2.4.0, IlluminaHumanMethylation450kanno.ilmn12.hg19 0.2.1, IlluminaHumanMethylation450kmanifest 0.4.0, S4Vectors 0.8.0, SummarizedExperiment 1.0.0, XVector 0.10.0, bumphunter 1.10.0, foreach 1.4.3, iterators 1.0.8, lattice 0.20-33, locfit 1.5-9.1, minfi 1.16.0, minfiData 0.11.0, preprocessCore 1.32.0
- Loaded via a namespace (and not attached): AnnotationDbi 1.32.0, BiocParallel 1.4.0, DBI 0.3.1, GEOquery 2.36.0, GenomicAlignments 1.6.0, GenomicFeatures 1.22.0, MASS 7.3-44, RColorBrewer 1.1-2, RCurl 1.95-4.7, RSQLite 1.0.0, Rcpp 0.12.1, Rsamtools 1.22.0, XML 3.98-1.3, annotate 1.48.0, base64 1.1, beanplot 1.2, biomaRt 2.26.0, bitops 1.0-6, codetools 0.2-14, colorspace 1.2-6, digest 0.6.8, doRNG 1.6, ellipse 0.3-8, futile.logger 1.4.1, futile.options 1.0.0, genefilter 1.52.0, ggplot2 1.0.1, grid 3.2.2, gtable 0.1.2, igraph 1.0.1, illuminaio 0.12.0, lambda.r 1.1.7, limma 3.26.0, magrittr 1.5, matrixStats 0.14.2, mclust 5.0.2, mixOmics 5.1.2, multtest 2.26.0, munsell 0.4.2, nlme 3.1-122, nor1mix 1.2-1, pheatmap 1.0.7, pkgmaker 0.22, plyr 1.8.3, proto 0.3-10, quadprog 1.5-5, registry 0.3, reshape 0.8.5, reshape2 1.4.1, rgl 0.95.1367, rngtools 1.2.4, rtracklayer 1.30.0, scales 0.3.0, siggenes 1.44.0, splines 3.2.2, stringi 0.5-5, stringr 1.0.0, survival 2.38-3, tools 3.2.2, xtable 1.7-4, zlibbioc 1.16.0

References

[1] B M Bolstad, R A Irizarry, M Astrand, and T P Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19:185, 2003.

[2] Jean-Philippe Fortin, Aurélie Labbe, Elana J Fertig, Mathieu Lemire, Brent W Zanke, Thomas J Hudson, Celia M T Greenwood, and Kasper H Hansen. Functional normalization: a better alternative to quantile normalization for methylation data. *(in preparation)*, 2013.

[3] Kasper Daniel Hansen and Martin Aryee. *minfi: Analyze Illumina's 450k methylation arrays*. R package version 1.8.3.

[4] Venkatraman E. Seshan and Adam Olshen. *DNAcopy: DNA copy number data analysis*. R package version 1.36.0.