

# Introduction to BreakPointGenes

Stef van Lieshout, Evert van den Broek

January 30, 2015

## 1 Running BreakPointGenes

This is a short tutorial on how to use the `BreakPointGenes` package. It covers an example run using the included data set of chromosomes 7–10 of a low grade glioma (LGG) sample. First step is naturally to load the package.

```
> library(BreakPointGenes)
```

### 1.1 Obtaining data from a `cghCall` object (ouput from packages `CGHcall` or `QDNAseq`)

Then we need to obtain gene annotations (mainly for the location) For hg19 reference they can be downloaded with (NOT AVAILABLE YET):

```
> geneAnnotations <- getGeneAnnotations()  
Downloading gene annotations for genome hg19
```

After downloading, the gene annotations can be saved locally with `saveRDS`, and in the future be read from the local file with `loadRDS` instead of relying on downloading.

### 1.2 Obtaining data from a `cghCall` object (ouput from packages `CGHcall` or `QDNAseq`)

Then we need to obtain copy number data. Because the `BreakPointGenes` package is an extension of the work done in packages `CGHcall` (for array data) and `QDNAseq` (for sequencing data), an object of class "`cghCall`" can be used as input.

```
> bp <- getBreakpoints( data = cghCallObject )
```

### 1.3 Processing bam files

Next step is to load the copy number data from a `cghCall` file or from a text file. This can be done for example with one of the commands below.

```

> bp <- getBreakpoints( data = cghCallObject )
> # using the output of CGHcall or QDNAseq
>
> # or
>
> bp <- getBreakpoints( data = matrix(), data2 = matrix() )
> # providing segments and call values directly (NOT YET POSSIBLE)
>

```

This will return an object of class CopyNumberBreakPoints.

```

> library( "BreakPointGenes" )
> # load built-in dataset (CGHcall)
> data( "LGG150.data" )
> # load built-in gene annotation dataset
> data( gene.annotation.hg19 )
> # setup the breakpoint data
> bp <- getBreakpoints( data = LGG150.data )

```

Breakpoint detection started...1 samples

```

> # optionally filter the data
> bp <- bpFilter( bp, filter = "deltaSeg", threshold = 0.2 )

```

Applying BP selection...

```

> # setup the gene data
> bp <- addGeneAnnotation( bp, gene.annotation.hg19 )

```

Adding of gene annotation started on 1000 genes by 1 samples  
0% ... 25% ... 50% ... 75% ... Adding gene annotation DONE

```

> # perform gene analysis
> bp <- bpGenes( bp )

```

Running bpGenes: 1000 genes and 1 samples  
bpGenes DONE  
A total of 0 gene breaks in 0 genes detected

## 2 Session Information

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

R version 3.1.1 (2014-07-10)

Platform: x86\_64-pc-linux-gnu (64-bit)

locale:

```
[1] LC_CTYPE=en_US.UTF-8
[2] LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8
[4] LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8
[6] LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8
[8] LC_NAME=C
[9] LC_ADDRESS=C
[10] LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8
[12] LC_IDENTIFICATION=C
```

attached base packages:

```
[1] stats      graphics  grDevices
[4] utils      datasets  methods
[7] base
```

other attached packages:

```
[1] BreakPointGenes_0.0.1
```

loaded via a namespace (and not attached):

```
[1] Biobase_2.26.0
[2] BiocGenerics_0.12.1
[3] CGHbase_1.26.0
[4] CGHcall_2.28.0
[5] devtools_1.6.1
[6] digest_0.6.4
[7] DNACopy_1.40.0
[8] impute_1.40.0
[9] limma_3.22.1
[10] marray_1.44.0
[11] matrixStats_0.12.2
[12] parallel_3.1.1
[13] Rcpp_0.11.3
[14] R.methodsS3_1.6.1
[15] R.oo_1.18.0
[16] roxygen2_4.1.0
```

```
[17] R.utils_1.34.0  
[18] stringr_0.6.2  
[19] tools_3.1.1
```