

Introduction to BreakPointGenes

Stef van Lieshout, Evert van den Broek

June 29, 2015

1 Running BreakPointGenes

This is a short tutorial on how to use the `BreakPointGenes` package. It covers an example run using included copy number data of one chromosome from 200 samples. The samples are part of the CAIRO study described in REF. Let's start by loading the package.

```
> library(BreakPointGenes)
```

1.1 Loading copy number data

Copy number data can be loaded in two ways. Either from a `cghCall/QDNAseq` object (output of bioconductor packages `CGHcall` or `QDNAseq`) or by providing a `data.frame` with 4 columns: Region-name (usually probe identifier), Chromosome, Start and End. In this tutorial we load a built-in dataset:

```
> data( "copynumber.data.chr20" )
```

Printing this we see that it is an R object of class "cghCall" and that it contains data of 200 samples.

```
> copynumber.data.chr20
cghCall (storageMode: lockedEnvironment)
assayData: 3653 features, 200 samples
  element names: calls, copynumber, probamp, probgain, probloss, probnorm, segmented
protocolData: none
phenoData
  sampleNames: sample_1 sample_2 ... sample_200 (200 total)
  varLabels: Cellularity
  varMetadata: labelDescription
featureData
  featureNames: A_16_P03469195 A_14_P136138 ... A_18_P13856091 (3653 total)
  fvarLabels: Chromosome Start End
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
Annotation:
```

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

Then we need to obtain gene annotations. For hg18 and hg19 reference sequence these are included and can be loaded:

```
> data( gene.annotation.hg19.chr20 )
> head( gene.annotation.hg19.chr20 )
```

	Gene	EnsID	Chromosome	Start	End
2752	RNU7	ENSG00000210768	20	49954915	49954976
8454	RPL7P2	ENSG00000220880	20	2004201	2004468
8455	RPL19P1	ENSG00000217306	20	2709260	2709687
8456	UBE2V1P1	ENSG00000218621	20	3290527	3290971
8458	SF3A3P1	ENSG00000217305	20	3418176	3419652
8463	RPS18P1	ENSG00000217102	20	5461524	5461981

1.4 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.

Next up is the detection of breakpoints

```
> bp <- getBreakpoints( data = copynumber.data.chr20 )
```

Breakpoint detection started...200 samples

Next we filter breakpoints. Different filters can be set with different threshold. Default here is "deltaSeg" filter with a threshold of 0.2. This means that only breakpoints which...

```
> bp <- bpFilter( bp )
```

Applying BP selection...

Next we will add the gene annotation information to the `BreakPointGenes` object. No analysis is done here yet.

```
> bp <- addGeneAnnotation( bp, gene.annotation.hg19.chr20 )
```

```
Adding of gene annotation started on 659 genes by 200 samples
0% ... 25% ... 50% ... 75% ... Adding gene annotation DONE
```

Next we perform the gene analysis. This overlaps the genomic locations of the genes with the copy number data to find breakpoints within genes.

```
> bp <- bpGenes( bp )
```

```
Running bpGenes: 659 genes and 200 samples
0% ... 25% ... 50% ... 75% ... bpGenes DONE
A total of 1029 gene breaks in 241 genes detected
```

Next we determine the significantly recurring breakpoints. This be done at the "gene" or "feature" level and using one of two different methods ("Benjamini Hochberg" or "Gilbert"). The advantage of using...

```
> bp <- bpStats( bp )
```

```
Applying statistical test over 200 samples for: gene breakpoints: BH test...
```

```
> bp
```

This will return an object of class `CopyNumberBreakPoints`.

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_1.7.0
[6] digest_0.6.8
[7] evaluate_0.7
[8] formatR_1.2
[9] highr_0.5
[10] htmltools_0.2.6
[11] impute_1.40.0
[12] knitr_1.10
[13] limma_3.22.7
[14] magrittr_1.5
[15] marray_1.44.0
[16] parallel_3.1.1
```

```
[17] rmarkdown_0.7  
[18] R.methodsS3_1.7.0  
[19] R.oo_1.19.0  
[20] R.utils_2.0.2  
[21] stringi_0.4-1  
[22] stringr_1.0.0  
[23] tools_3.1.1  
[24] yaml_2.1.13
```