Introduction to BreakPointGenes

Stef van Lieshout, Evert van den Broek

July 8, 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 (ouput of bioconductor packages CGHcall or QDNAseq) or by providing a data.frame with at least 5 columns: Chromosome, Start, End and FeatureName (usually probe identifier). Note: when using the data.frame input the column names must be exactly as described here and in the same order! In this tutorial we will use a built-in dataset of chromosome 20:

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

By inspecting the dataset, we see that we are dealing with an R object of class "cghCall" with 3653 features (aCGH probes in this case) and 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.

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

1.3 Loading gene annotation data

Then we need to obtain gene annotations. For hg18 (and hg19, hg38) reference sequence these are included and can be loaded:

```
> data( "ens.gene.ann.hg18" )
```

Inspect the annotation.

```
> head( ens.gene.ann.hg18 )
      Gene           EnsID Chromosome  Start    End   band strand
MIRN1302-2 ENSG00000221311          1  20229  20366 p36.33      1
   FAM138E ENSG00000222027          1  24417  25944 p36.33     -1
   FAM138E ENSG00000222003          1  24417  25944 p36.33     -1
   FAM138A ENSG00000222003          1  24417  25944 p36.33     -1
     OR4F5 ENSG00000177693          1  58954  59871 p36.33      1
    OR4F29 ENSG00000177799          1 357522 358460 p36.33      1
```

1.4 Filtering

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...

```
> breakpointsFiltered <- bpFilter( breakpoints )

Applying BP selection...
```

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

```
> breakpointsAnnotated <- addGeneAnnotation( breakpointsFiltered, ens.gene.ann.hg18 )

Adding of gene annotation started on 271 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.

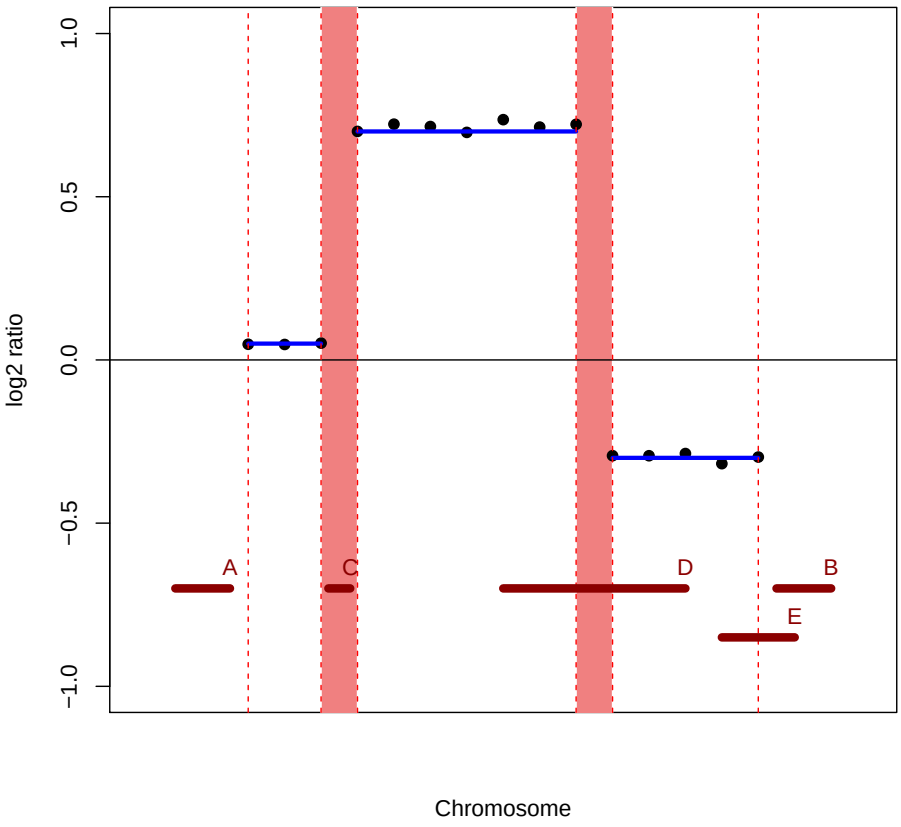

Figure 1: *BreakPointGenes* geneAnnotation

```
> breakpointGenes <- bpGenes( breakpointsAnnotated )

Running bpGenes: 271 genes and 200 samples
0% ... 25% ... 50% ... 75% ... bpGenes DONE
A total of 381 gene breaks in 144 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...

```
> breakpointStats <- bpStats( breakpointGenes )

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

> breakpointStats
```

This will return an object of class `CopyNumberBreakPointGenes`.
By using recurrentGenes() we can observe the recurrent affected genes.

```
> head( recurrentGenes( breakpointStatistics ) )

 A total of 6 recurrent breakpoint genes (at FDR < 0.1)
      Gene sampleCount featureTotal
5274 ALPK2          12           11
2157 ROCK1           9           11
4767 SMAD4           8            4
5081  FECH           7            3
2188 ESCO1           6            6
2300 RBBP8           6            8
           pvalue          FDR
5274 1.914062e-09 5.187107e-07
2157 2.181833e-06 2.956384e-04
4767 5.073395e-06 4.582967e-04
5081 3.810153e-05 2.581379e-03
2188 5.711269e-04 3.095508e-02
2300 7.452691e-04 3.366132e-02
```

2 Gene Annotation creation

The steps below are a description of how the gene annotations in this package were created.

```
> # gene annotations obtained via Biomart.
> # HUGO gene names (HGNC symbol), Ensembl_ID and chromosomal location
>
> # Used (and most) recent releases:
> # HG18: release54
> # HG19: release75
> # HG38: release80 (date: 150629)
>
> library(biomaRt)
> ensembl54 = useMart( host='may2009.archive.ensembl.org', biomart='ENSEMBL_MART_ENSEMBL', d
> ensembl75 = useMart( host='feb2014.archive.ensembl.org', biomart='ENSEMBL_MART_ENSEMBL', d
> ensembl80 = useMart( "ensembl", dataset="hsapiens_gene_ensembl" )
> createAnnotationFile <- function( biomartVersion ) {
+   biomart_result <- getBM(attributes =  c("hgnc_symbol", "ensembl_gene_id", "chromosome_na
+
+   biomart_result[ ,3] <- as.vector( biomart_result[ ,3] )
+   biomart_result$chromosome_name[ biomart_result$chromosome_name=="X" ] <- "23"
+   biomart_result$chromosome_name[ biomart_result$chromosome_name=="Y" ] <- "24"
+
+   biomart_genes <-biomart_result[ which(biomart_result[ ,1]!="" & biomart_result[ ,3] %in%
+   colnames(biomart_genes)[1:5]<-c("Gene","EnsID","Chromosome","Start","End")
+
+   cat( c("Biomart version:", biomartVersion@host, "including:", dim(biomart_genes)[1], "ge
+   return( biomart_genes )
+ }
> ens.gene.ann.hg18 <- createAnnotationFile( ensembl54 )
> ens.gene.ann.hg19 <- createAnnotationFile( ensembl75 )
> ens.gene.ann.hg38 <- createAnnotationFile( ensembl80 )
>
```

3 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
[2] knitr_1.10

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] impute_1.40.0
 [7] limma_3.22.7
 [8] marray_1.44.0
 [9] parallel_3.1.1
[10] R.methodsS3_1.7.0
[11] R.oo_1.19.0
[12] R.utils_2.0.2
[13] tools_3.1.1
```