DiffBind practical

Dóra Bihary

14 September, 2017

Contents

1 Introduction

The primary aim of the DiffBind package is to identify differentially bound regions between two sample groups. It includes functions to support the processing of peak sets, including overlapping and merging peak sets, counting sequencing reads overlapping intervals in peak sets, and identifying statistically significantly differentially bound sites.

2 Example of running DiffBind

For this tutorial we are using the preprocessed dataset that you can find in the *~/Course_Materials/ChIPSeq/Preprocessed/* folder. This dataset contains samples from a ChIP-seq experiment against two different transcription factors, tp53 and tp73 (TAp73beta isoform) in the human osteosarcoma cell lines Saos-2. Reads were mapped to grch38 genome using BWA and peaks were called by MACS2. The aim of this exercise is to find the differentially bound sites between the two transcription factors.

2.1 Reading in the peaksets

First, let's load the DiffBind library and set the working directory to the folder where you can find the sample sheet describing the data we will process:

```
library("DiffBind")
setwd("~/Course_Materials/ChIPSeq/ChIPQC_DiffBind/")
```

You can read in the sample sheet to have a closer look at it (we use the same file as we did in the ChIPQC practical):

```
samples <- read.csv("sampleSheet.csv")
samples
##       SampleID Tissue    Factor Condition Treatment Replicate
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2
## 3       p53_r1  SAOS2       p53       p53       p53         1
## 4       p53_r2  SAOS2       p53       p53       p53         2
##
## 1 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/TAp73beta_r1.fastq_trimmed.fast
```

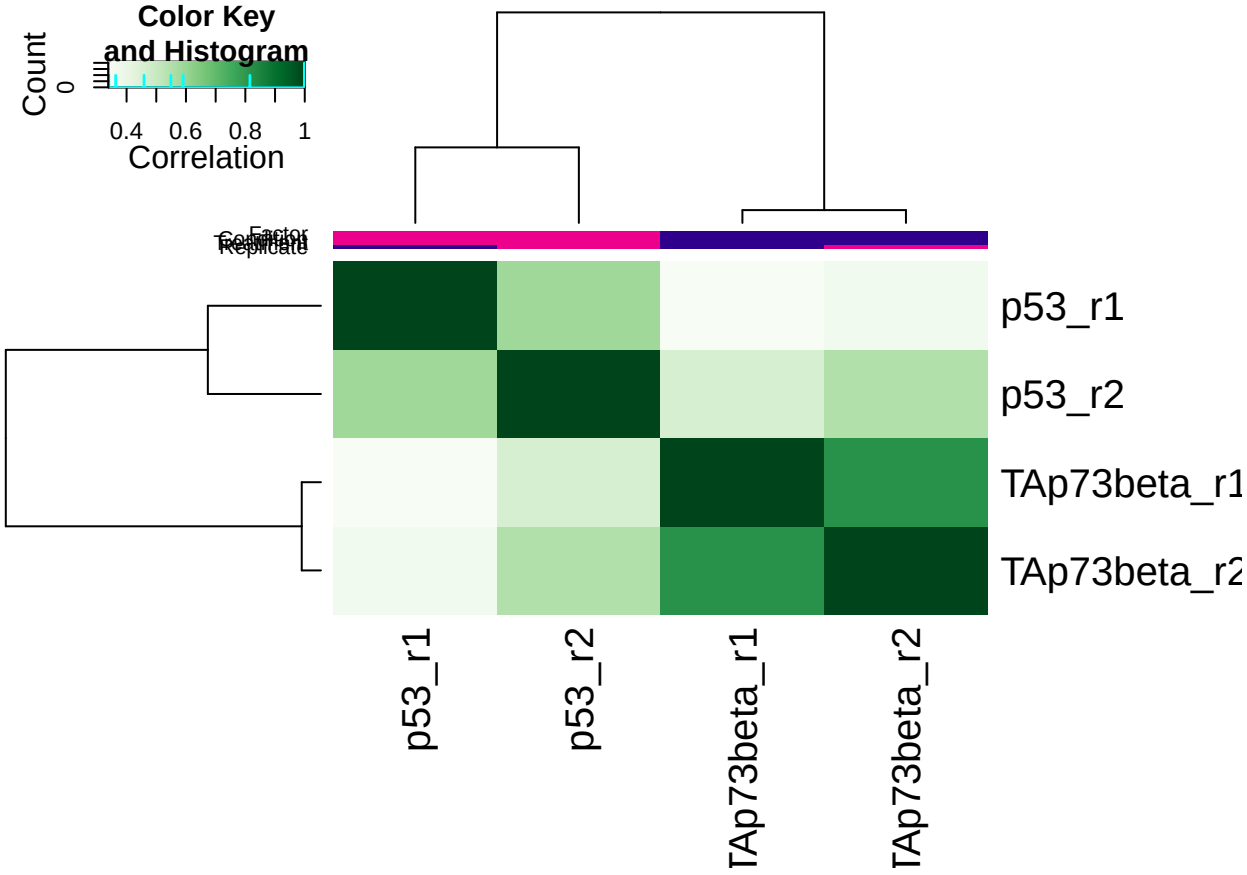
```
## 2 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/TAp73beta_r2.fastq_trimmed.faste
## 3      /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/tp53_r1.fastq_trimmed.faste
## 4      /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/tp53_r2.fastq_trimmed.faste
##   ControlID
## 1     input
## 2     input
## 3     input
## 4     input
##                                                                                                    bamCo
## 1 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorte
## 2 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorte
## 3 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorte
## 4 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorte
##
## 1 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Peaks/TAp73beta_r1.fastq_trimmed.fastq_sorted_
## 2 /home/participant/Course_Materials/ChIPSeq/Preprocessed/Peaks/TAp73beta_r2.fastq_trimmed.fastq_sorted_
## 3      /home/participant/Course_Materials/ChIPSeq/Preprocessed/Peaks/tp53_r1.fastq_trimmed.fastq_sorted_
## 4      /home/participant/Course_Materials/ChIPSeq/Preprocessed/Peaks/tp53_r2.fastq_trimmed.fastq_sorted_
##   PeakCaller
## 1        bed
## 2        bed
## 3        bed
## 4        bed
```

Now you can load and read the sample sheet with the following DiffBind function:

```
DBdata <- dba(sampleSheet=samples)
```

The result is a DBA object; we can display all the metadata and plot a correlation heatmap that gives an initial clustering of the samples using cross-correaltions of the rows in the binding matrix.

```
DBdata
## 4 Samples, 2282 sites in matrix (10050 total):
##             ID Tissue    Factor Condition Treatment Replicate Caller
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1    bed
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2    bed
## 3       p53_r1  SAOS2       p53       p53       p53         1    bed
## 4       p53_r2  SAOS2       p53       p53       p53         2    bed
##   Intervals
## 1      6185
## 2      2962
## 3      1069
## 4      3450
plot(DBdata)
```



Again, each line in DBdata represents a single sample. The table displayed shows some information that was present in the sample sheet and also the number of intervals (peaks) in the different samples. The heatmap shows us that samples cluster together based on condition.

2.2 Counting reads

Our next step is to calculate a binding matrix based on the read counts for every sample rather than only based on the peaks called. Let's generate the counts and have a look at the data:

```
DBdata <- dba.count(DBdata)
DBdata
## 4 Samples, 2282 sites in matrix:
##             ID Tissue    Factor Condition Treatment Replicate Caller
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1 counts
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2 counts
## 3       p53_r1  SAOS2       p53       p53       p53         1 counts
## 4       p53_r2  SAOS2       p53       p53       p53         2 counts
##   Intervals FRiP
## 1      2282 0.12
## 2      2282 0.10
## 3      2282 0.03
## 4      2282 0.05
```

As you can see all the samples are using the same length (2282) consensus peakset. The last column tells us the Fraction of Reads In Peaks (FRiP). This is the proportion of reads that overlap with peaks in the consensus peakset, based on this value we can tell the enrichment of each sample.

2.3 Establishing contrast

Next we have to let DiffBind know how we want to group our samples. In our case we will group based on condition. We also have to set the minMembers parameter to 2 (default is 3) since we only have two samples in each condition.

```
DBdata <- dba.contrast(DBdata, categories=DBA_CONDITION, minMembers = 2)
DBdata
## 4 Samples, 2282 sites in matrix:
##             ID Tissue    Factor Condition Treatment Replicate Caller
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1 counts
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2 counts
## 3       p53_r1  SAOS2       p53       p53       p53         1 counts
## 4       p53_r2  SAOS2       p53       p53       p53         2 counts
##   Intervals FRiP
## 1      2282 0.12
## 2      2282 0.10
## 3      2282 0.03
## 4      2282 0.05
##
## 1 Contrast:
##      Group1 Members1 Group2 Members2
## 1 TAp73beta        2    p53        2
```

You can see almost the same summary table as before except that now the contrast is added at the end.

2.4 Differential binding analysis

We are now ready to run the main function (dba.analyze) of DiffBind that performs differential binding:

```
DBdata <- dba.analyze(DBdata)
DBdata
## 4 Samples, 2282 sites in matrix:
##             ID Tissue    Factor Condition Treatment Replicate Caller
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1 counts
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2 counts
## 3       p53_r1  SAOS2       p53       p53       p53         1 counts
## 4       p53_r2  SAOS2       p53       p53       p53         2 counts
##   Intervals FRiP
## 1      2282 0.12
## 2      2282 0.10
## 3      2282 0.03
## 4      2282 0.05
##
## 1 Contrast:
##      Group1 Members1 Group2 Members2 DB.DESeq2
## 1 TAp73beta        2    p53        2       531
```

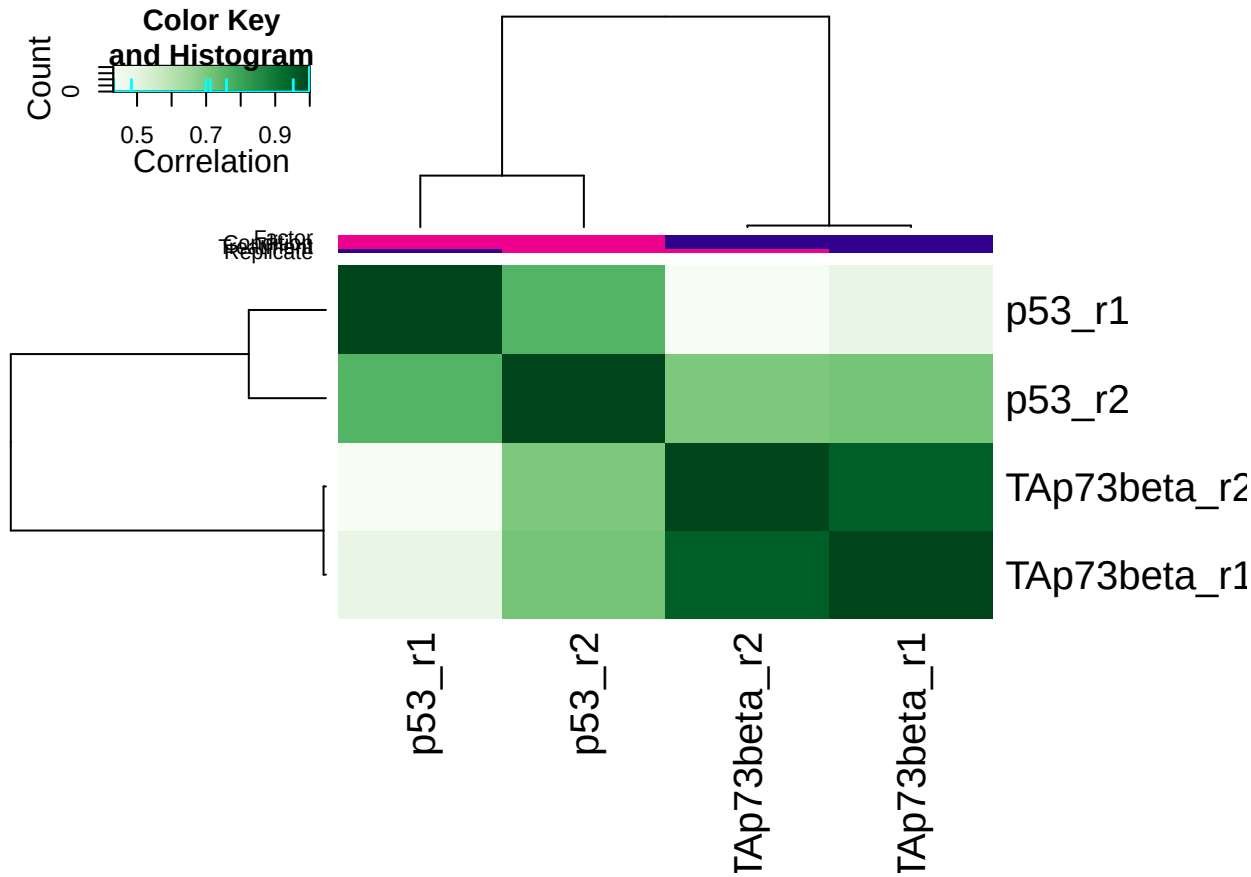
The method uses existing methods created to perform differential expression analysis in RNA-seq datasets. The default method is DESeq2 but you can try edgeR as well. At the end of the summary displayed you can see the amount of differentially bound sites found by each method. This means that out of the 2282 regions DESeq2 identified 531 and edgeR 676 as differentially bound.

```
DBdata <- dba.analyze(DBdata, method=DBA_EDGER)
DBdata
## 4 Samples, 2282 sites in matrix:
```

```
##             ID Tissue    Factor Condition Treatment Replicate Caller
## 1 TAp73beta_r1  SAOS2 TAp73beta TAp73beta TAp73beta         1 counts
## 2 TAp73beta_r2  SAOS2 TAp73beta TAp73beta TAp73beta         2 counts
## 3       p53_r1  SAOS2       p53       p53       p53         1 counts
## 4       p53_r2  SAOS2       p53       p53       p53         2 counts
##   Intervals FRiP
## 1      2282 0.12
## 2      2282 0.10
## 3      2282 0.03
## 4      2282 0.05
##
## 1 Contrast:
##      Group1 Members1 Group2 Members2 DB.edgeR DB.DESeq2
## 1 TAp73beta        2    p53        2      676       531
```

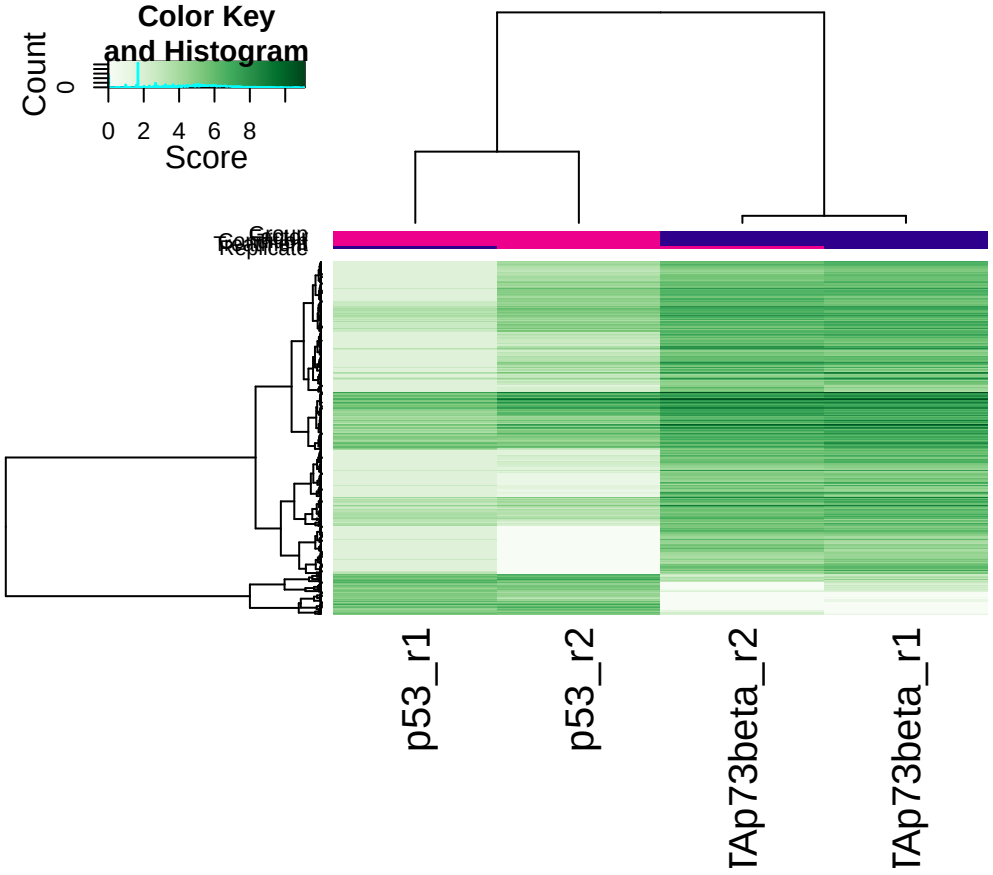
We can now plot the same type of heatmap as at the beginning of our analysis using only the differentially bound sites. This will strengthen a bit the differeneces between the conditions as expected.

```
plot(DBdata)
```



We can also display the binding affinity heatmap to see the binding patterns across the identified regions. You can control which method's result you wish to see by setting method=DBA_EDGER or method=DBA_DESEQ2 (default).

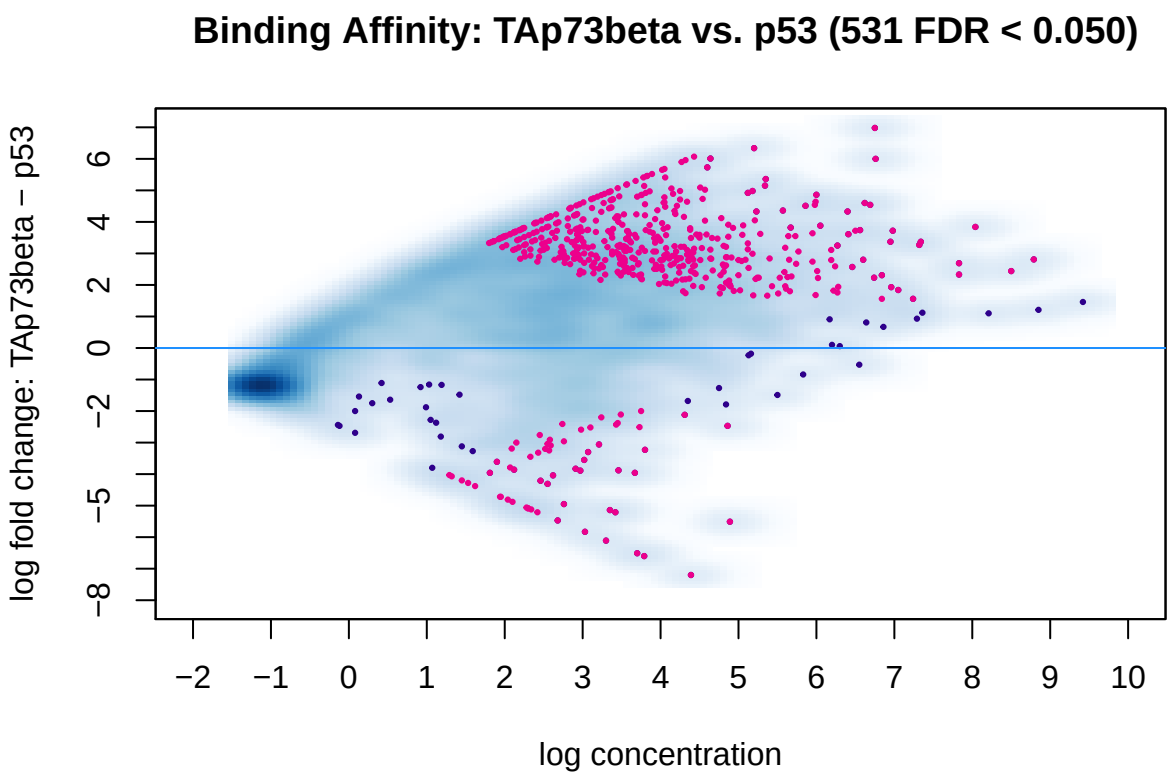
```
dba.plotHeatmap(DBdata, contrast=1, correlations=FALSE)
```



To further analyse the results we can use built in functions to generate MA plots, volcano plots, PCA plots and boxplots. All these functions can be called using either one of the differential binding methods (eg. DESeq2/edgeR).

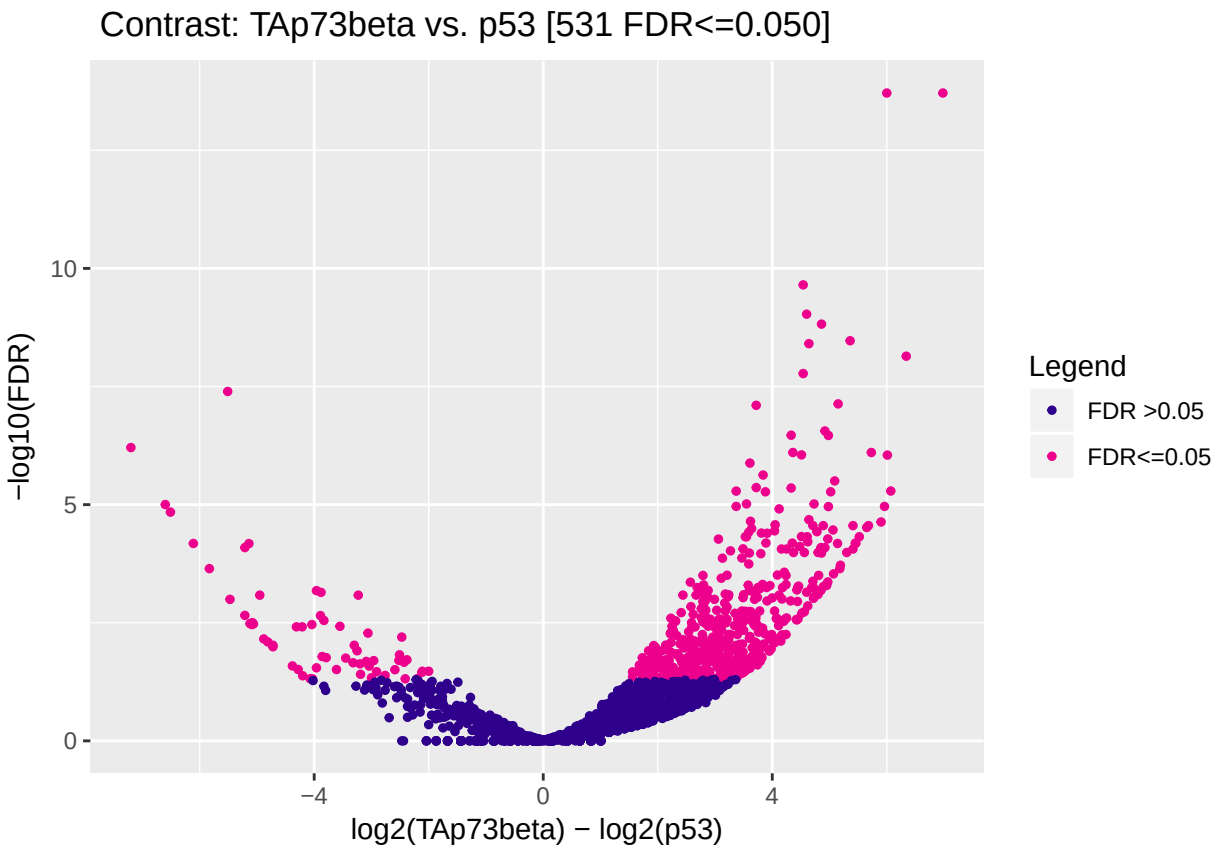
MA plots are useful to visualise which data points are differentially bound. Each of these points represents a binding site and red points indicate differentailly bound ones. These points have a log fold change of at least 2.

```
dba.plotMA(DBdata)
```



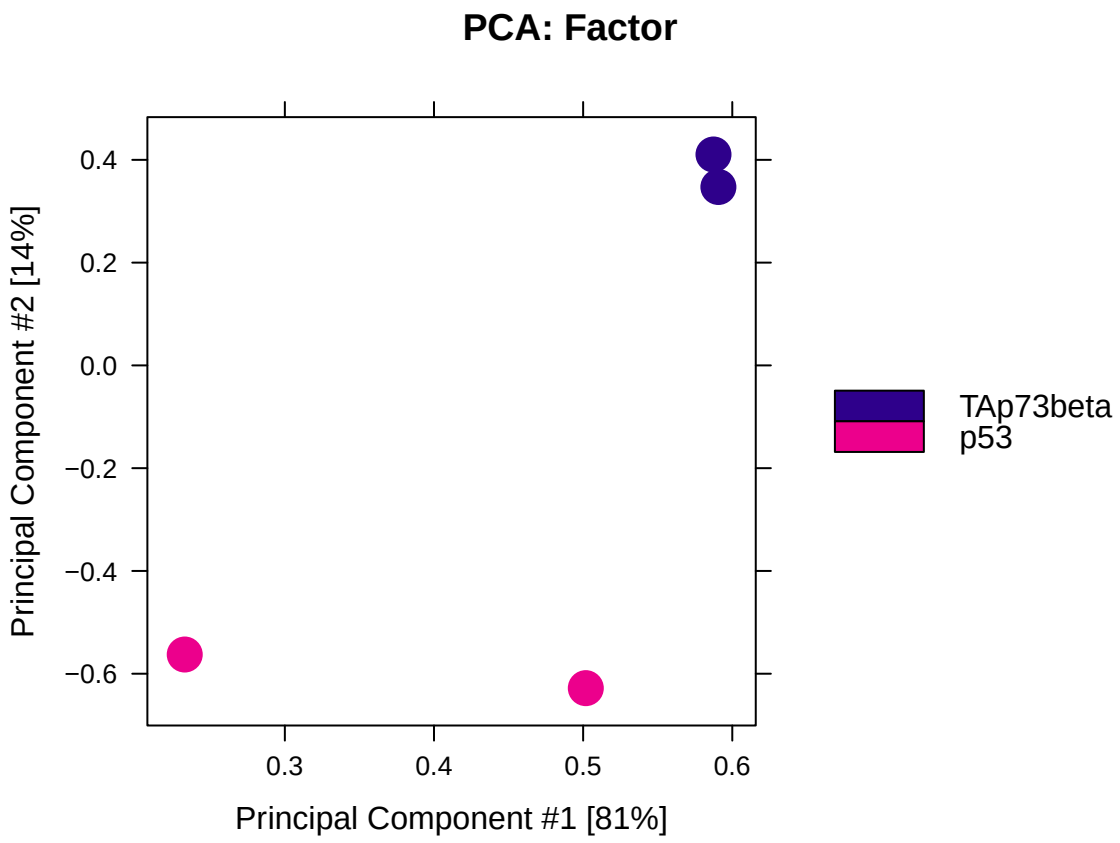
Similarly to MA plots, Volcano plots can show the significantly differentailly bound sites and show their fold enrichment (or p-values).

```
dba.plotVolcano(DBdata)
```

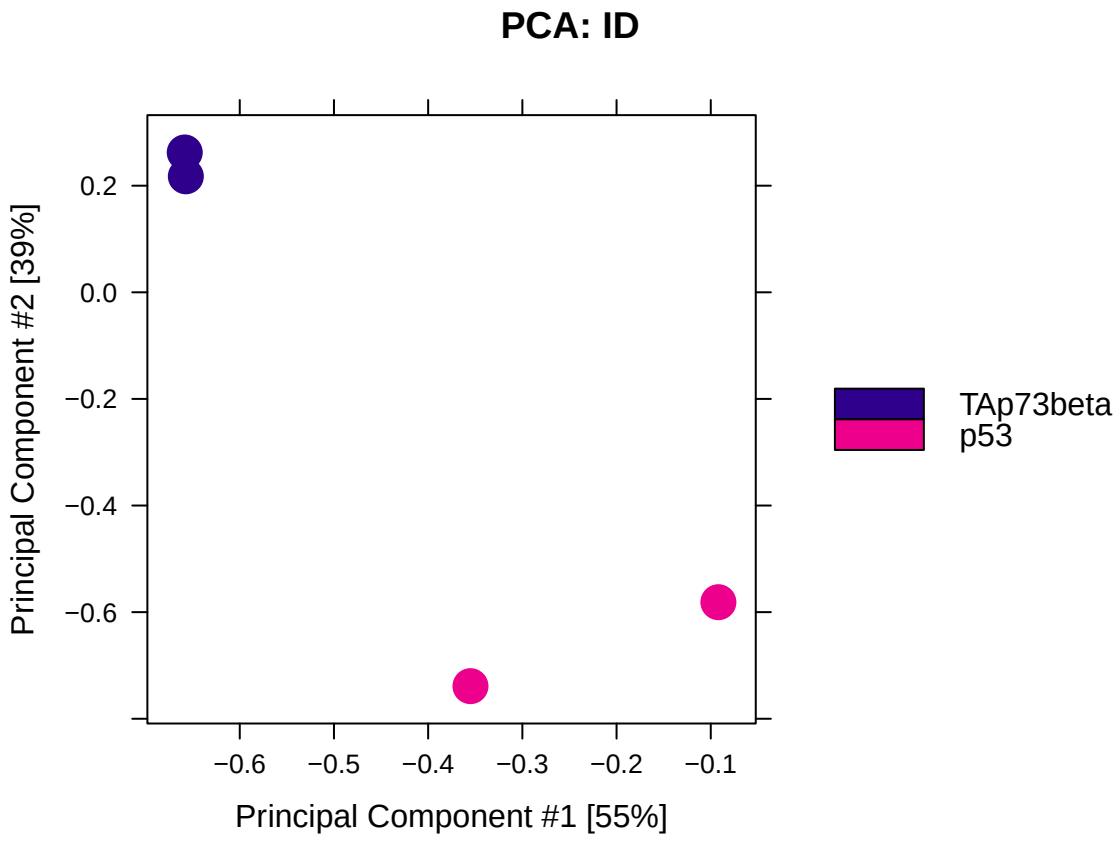


PCA (Principal Component Analysis) can give us different representation of how the samples are associated. We see that samples of the different conditions cluster separately. The first command calculates principal components based on the normalised read counts for all the binding sites; the second one only uses the differentially bound sites.

```
dba.plotPCA(DBdata)
```

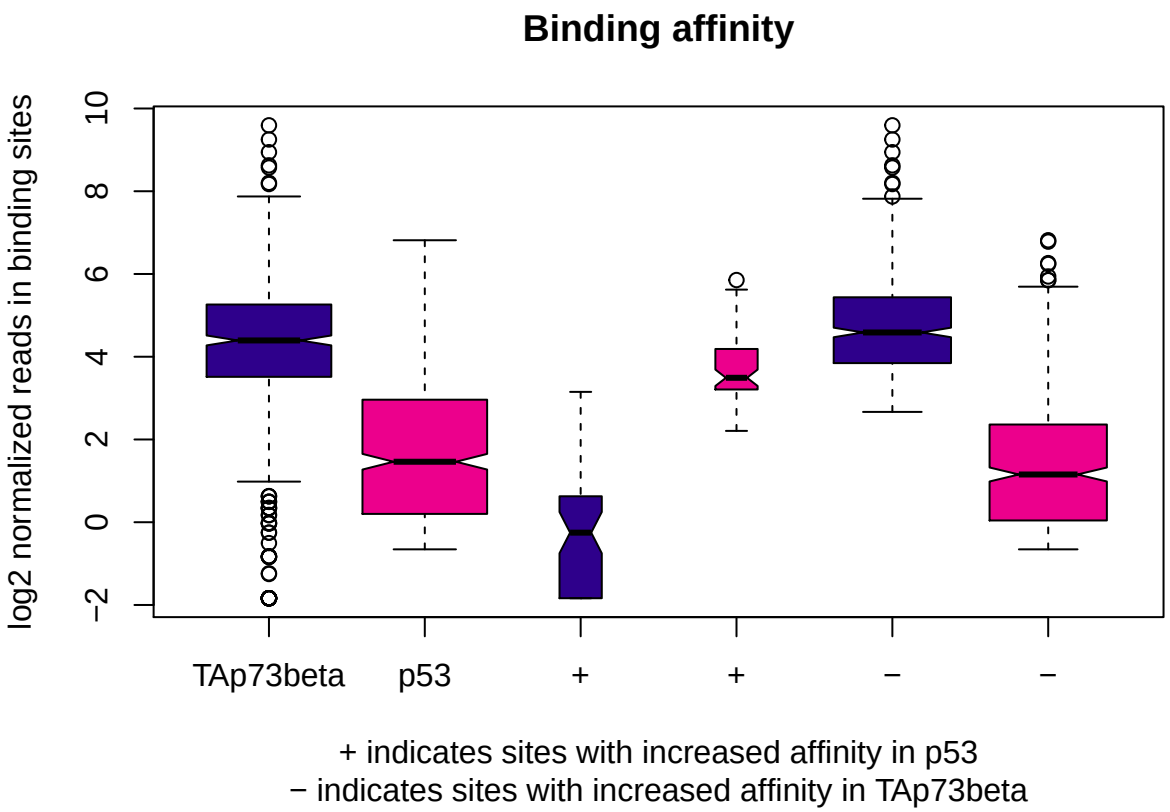



```
dba.plotPCA(DBdata, contrast = 1)
```



Boxplots can give us an idea about the read distribution differences between the classes - in our case the two conditions. The first two boxes show distribution of reads over all differentially bound sites; the middle two show differences on those sites where the affinity increases in p53 and the two boxes on the right show differences where the affinity increases in TAp73beta samples.

```
dba.plotBox(DBdata)
```



And finally we can report the differentially bound peak regions, identified by either method (DESeq2/edgeR).

```
dba.report(DBdata, method=DBA_EDGER)
## GRanges object with 676 ranges and 6 metadata columns:
##        seqnames                ranges strand |      Conc Conc_TAp73beta
##           <Rle>             <IRanges>  <Rle> | <numeric>      <numeric>
##   2054     chr3 [185717812, 185718086]      * |      5.98          -0.65
##   1378     chr3 [134333527, 134334086]      * |      7.94           8.91
##   1678     chr3 [157026110, 157026580]      * |      7.92           8.91
##   1390     chr3 [136142134, 136142383]      * |      5.42          -0.65
##     19     chr3 [  2241629,   2242062]      * |      6.39           7.37
##    ...      ...                   ...    ... .       ...            ...
##    544     chr3 [ 46520543,  46520768]      * |      4.44           3.46
##    279     chr3 [ 22129111,  22129217]      * |      3.05           3.75
##   1415     chr3 [138325984, 138326461]      * |      6.82           7.44
##   1683     chr3 [157268794, 157268843]      * |      2.41           3.13
##   1618     chr3 [152149872, 152150237]      * |      6.18           6.76
##         Conc_p53      Fold   p-value       FDR
##        <numeric> <numeric> <numeric> <numeric>
##   2054      6.97     -7.62  5.94e-19  1.36e-15
##   1378      3.32      5.59  6.38e-17  7.28e-14
##   1678      2.42      6.48  2.01e-16  1.53e-13
##   1390      6.41     -7.05  8.81e-15  5.03e-12
##     19      1.44      5.93     3e-14  1.21e-11
##    ...       ...       ...       ...       ...
##    544      5.02     -1.56    0.0142    0.0482
##    279      1.66      2.09    0.0147    0.0498
```

```
##   1415      5.69      1.75    0.0147    0.0499
##   1683      0.87      2.26    0.0148    0.0499
##   1618       5.2      1.56    0.0148    0.0499
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```