DiffBind : differential binding analysis of ChIP-Seq peak data

Rory Stark
rory.stark@cancer.org.uk

Gordon Brown
gordon.brown@cancer.org.uk

Cancer Research UK
Cambridge Research Institute

14 December 2011

1 Introduction

This document offers an introduction and overview of the R Bioconductor package DiffBind , which provides functions for processing ChIP-Seq data enriched for genomic loci where specific protein/DNA binding occurs, including peak sets identified by ChIP-Seq peak callers and aligned sequence read datasets. It is designed to work with multiple peak sets simultaneously, representing different ChIP experiments (antibodies, transcription factor and/or histone marks, experimental conditions, replicates) as well as managing the results of multiple peak callers.

The primary emphasis of the package is on identifying sites that are differentially bound 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 based on evidence of binding affinity (measured by differences in read densities). To this end it uses statistical routines developed in an RNA-Seq context (primarily the Bioconductor packages edgeR and DESeq). Additionally, the package builds on R graphics routines to provide a set of standardized plots to aid in binding analysis.

This guide includes a brief overview of the processing flow, followed by three sections of examples: the first focusing on the core task of obtaining differentially bound sites based on affinity data, the second working through the main plotting routines, and the third revisiting occupancy data (peak calls) in more detail, as well as comparing the results of an occupancy-based analysis with an affinity-based one. Finally, some technical aspects of the how these analyses are accomplished are detailed.

2 Processing overview

DiffBind works primarily with peaksets, which are sets of genomic intervals representing candidate protein binding sites. Each interval consists of a chromosome, a start and end position, and usually a score of some type indicating confidence in, or strength of, the peak. Associated with each peakset are metadata relating to the experiment from which the peakset was derived. Additionally, files containing mapped sequencing reads (BAM/SAM/BED) can be associated with each peakset (one for the ChIP data, and optionally another representing a control dataset).

Generally, processing data with DiffBind involves five phases:

1. `Reading in peaksets`: The first step is to read in a set of peaksets and associated metadata. Peaksets are derived either from ChIP-Seq peak callers, such as MACS (Zhang et al. [2008]), or using some other criterion (e.g. all the promoter regions in a genome). The easiest way to read in peaksets is using a comma-separated value (csv) sample sheet with one line for each peakset. A single experiment can have more than one associated peakset, e.g. if multiple peak callers are used for comparison purposes, and hence have more than one line in the sample sheet. Once the peaksets are read in, a merging function finds all overlapping peaks and derives a single set of unique genomic intervals covering all the supplied peaks.

2. `Occupancy analysis`: Peaksets, especially those generated by peak callers, provide an insight into the potential *occupancy* of the protein being ChIPed for at specific genomic loci. After the peaksets have been loaded, it can be useful to perform some exploratory plotting to determine how these occupancy maps agree with each other, e.g. between experimental replicates (re-doing the ChIP under the same conditions), between different peak callers on the same experiment, and within groups of samples representing a common experimental condition. DiffBind provides functions to enable overlaps to be examined, as well as functions to determine how well similar samples cluster together. Beyond quality control, the product of an occupancy analysis may be a *consensus peakset*, representing an overall set of candidate binding sites to be used in further analysis.

3. `Counting reads`: Once a consensus peakset has been derived, DiffBind can use the supplied sequence read files to count how many reads overlap each interval for each unique sample. The result of this is a *binding affinity matrix* containing a (normalized) read count for each sample at every potential binding site. With this matrix, the samples can be re-clustered using affinity, rather than occupancy, data. The binding affinity matrix is used for QC plotting as well as for subsequent differential analysis.

4. `Differential binding affinity analysis`: The core functionality of DiffBind is the differential binding affinity analysis, which enables binding sites to be identified that are statistically significantly differentially bound between sample groups. To accomplish this, first a contrast (or contrasts) is established, dividing the samples into groups to be compared. Next the core analysis routines are executed, by default using `edgeR` . This will assign a p-value and FDR to each candidate binding site indicating the significance of their being differentially bound.

5. `Plotting and reporting`: Once one or more contrasts have been run, DiffBind provides a number of functions for reporting and plotting the results. MA plots give an overview of the results of the analysis, while correlation heatmaps and PCA plots show how the groups cluster based on differentially bound sites. Boxplots show the distribution of reads within differentially bound sites corresponding to whether they gain or lose affinity between the two sample groups. A reporting mechanism enables differentially bound sites to be extracted for further processing, such as annotation and/or pathway analysis.

3 Example: obtaining differentially bound sites

This section offers a quick example of how to use DiffBind to identify significantly differentially bound sites using affinity (read count) data.

The dataset for this example consists of ChIPs against the transcription factor ERa using five breast cancer cell lines (Ross-Innes et al. [2012]). Three of these cell lines are responsive to tamoxifen, while two others are resistant to tamoxifen treatment. There are at least two replicates for each of the cell lines, with one cell line having three replicates, for a total of eleven sequenced libraries. For each sample, we have one peakset originally derived using the MACS peak caller (Zhang et al. [2008]), for a total of eleven peaksets. Note that to save space in the package, only data for chromosome 18 is used. The metadata and peak data are available in the `extra` subdirectory of the DiffBind package directory; you can make this your working directory by entering:

```
> library(DiffBind)
> setwd(system.file("extra", package="DiffBind"))
```

Obtaining the sites significantly differentially bound (DB) between the samples that respond to tamoxifen and those that are resistant can be done in a five-step script:

```
> tamoxifen = dba(sampleSheet="tamoxifen.csv")
> tamoxifen = dba.count(tamoxifen)
> tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION)
> tamoxifen = dba.analyze(tamoxifen)
> tamoxifen.DB = dba.report(tamoxifen)
```

The following subsections describe these steps in more detail.

3.1 Reading in the peaksets

Table 1 shows the sample sheet, saved in a file called `tamoxifen.csv`. The peaksets are read in using the following DiffBind function:

```
> tamoxifen = dba(sampleSheet="tamoxifen.csv")
```

Table 1: Tamoxifen dataset sample sheet (tamoxifen.csv).

SampleID	Tissue	Factor	Condition	Replicate	bamReads	bamControl	Peaks
BT474.1-	BT474	ER	Resistant	1	BT474_ER_1.bed.gz	BT474_Input.bed.gz	BT474_ER_1.bed.gz
BT474.2-	BT474	ER	Resistant	2	BT474_ER_2.bed.gz	BT474_Input.bed.gz	BT474_ER_2.bed.gz
MCF7.1+	MCF7	ER	Responsive	1	MCF7_ER_1.bed.gz	MCF7_Input.bed.gz	MCF7_ER_1.bed.gz
MCF7.2+	MCF7	ER	Responsive	2	MCF7_ER_2.bed.gz	MCF7_Input.bed.gz	MCF7_ER_2.bed.gz
MCF7.3+	MCF7	ER	Responsive	3	MCF7_ER_3.bed.gz	MCF7_Input.bed.gz	MCF7_ER_3.bed.gz
T47D.1+	T47D	ER	Responsive	1	T47D_ER_1.bed.gz	T47D_Input.bed.gz	T47D_ER_1.bed.gz
T47D.2+	T47D	ER	Responsive	2	T47D_ER_2.bed.gz	T47D_Input.bed.gz	T47D_ER_2.bed.gz
TAMR.1-	TAMR	ER	Resistant	1	TAMR_ER_1.bed.gz	TAMR_Input.bed.gz	TAMR_ER_1.bed.gz
TAMR.2-	TAMR	ER	Resistant	2	TAMR_ER_2.bed.gz	TAMR_Input.bed.gz	TAMR_ER_2.bed.gz
ZR75.1+	ZR75	ER	Responsive	1	ZR75_ER_1.bed.gz	ZR75_Input.bed.gz	ZR75_ER_1.bed.gz
ZR75.2+	ZR75	ER	Responsive	2	ZR75_ER_2.bed.gz	ZR75_Input.bed.gz	ZR75_ER_2.bed.gz

The result is a `DBA object`; the metadata associated with this object can be displayed simply as follows:

```
> tamoxifen
```

```
11 Samples, 2602 sites in matrix (3557 total):
         ID Tissue Factor  Condition Peak.caller Replicate Intervals
1  BT474.1-  BT474     ER  Resistant         raw         1      1084
2  BT474.2-  BT474     ER  Resistant         raw         2      1115
3   MCF7.1+   MCF7     ER Responsive         raw         1      1513
4   MCF7.2+   MCF7     ER Responsive         raw         2      1037
5   MCF7.3+   MCF7     ER Responsive         raw         3      1372
6   T47D.1+   T47D     ER Responsive         raw         1       509
7   T47D.2+   T47D     ER Responsive         raw         2       347
8   TAMR.1-   TAMR     ER  Resistant         raw         1      1148
9   TAMR.2-   TAMR     ER  Resistant         raw         2       933
10  ZR75.1+   ZR75     ER Responsive         raw         1      2111
11  ZR75.2+   ZR75     ER Responsive         raw         2      1975
```

This shows how many peaks are in each peakset, as well as (in the first line) total number of unique peaks after merging overlapping ones (3,557) and the default binding matrix of 11 samples by the 2,602 sites that overlap in at least two of the samples. This object is available for loading using `data(tamoxifen_peaks)`.

Using only this peak caller data, a correlation heatmap can be generated which gives an initial clustering of the samples using the cross-correlations of each row of the binding matrix:

```
> plot(tamoxifen)
```

The resulting plot (Figure 1) shows that while the replicates for each cell line cluster together appropriately, the cell lines do not cluster into groups corresponding to those that are responsive (MCF7, T47D, and ZR75) vs. those resistant (BT474 and TAMR) to tamoxifen treatment. It also shows that the two most highly correlated cell lines are TAMR and MCF7. This is probably due to the fact that the TAMR cell line is derived directly from the MCF7 cell line by exposing it to tamoxifen until a resistant strain emerges.

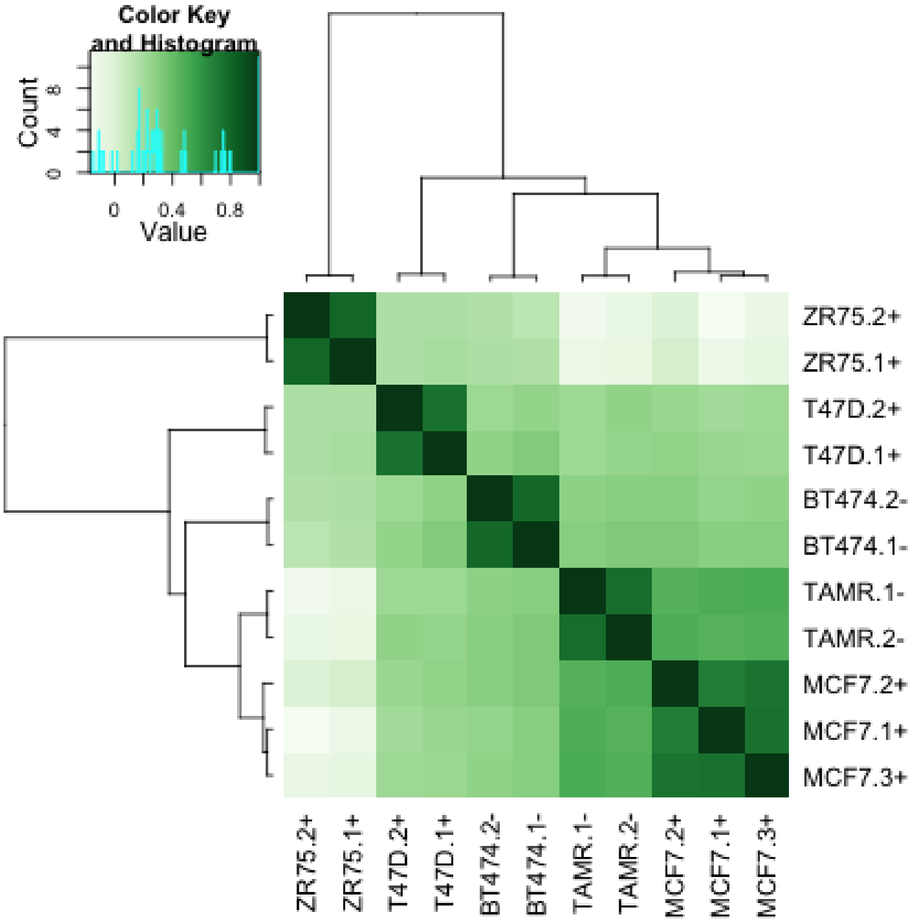


Figure 1: Correlation heatmap, using occupancy (peak caller score) data. Generated by: `plot(tamoxifen)`; can also be generated by: `dba.plotHeatmap(tamoxifen)`.

3.2 Counting reads

The next step is to calculate a binding matrix with scores based on read counts for every sample (affinity scores), rather than confidence scores for only those peaks called in a specific sample (occupancy scores). These reads are obtained using the dba.count function:[1]

```
> tamoxifen = dba.count(tamoxifen, minOverlap=3)
```

This object is available loading using data(tamoxifen_counts). The dba.count call plots a new correlation heatmap based on the affinity scores, seen in Figure 2a. This figure shows two main clusters, with the MCF7-derived samples (MCF7 and TAMR) in one cluster and the other cell line samples in the other cluster. Responsiveness to tamoxifen treatment does not appear to form a basis for clustering when using all of the affinity scores.

3.3 Establishing a contrast

Before running the differential analysis, we need to tell DiffBind which cell lines fall in which groups. This is done using the dba.contrast function, as follows:

```
> tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION)
```

The uses the *condition* metadata (Responsive vs. Resistant) to set up a a contrast with 4 samples in the Resistant group and 7 samples in the Responsive group.[2]

3.4 Performing the differential analysis

The main differential analysis function is invoked as follows:

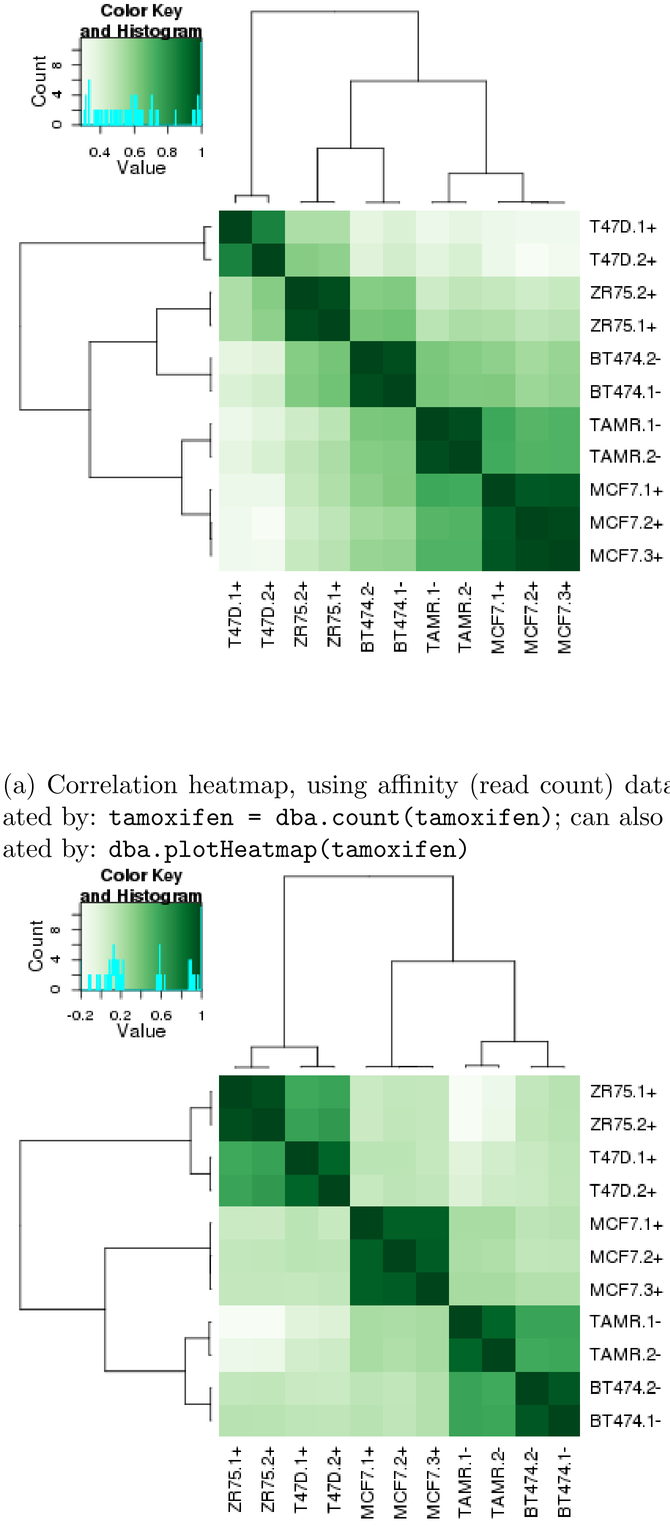
```
> tamoxifen = dba.analyze(tamoxifen)
```

This will run an edgeR analysis (see subsequent section discussing the technical details of the edgeR analysis) on the binding affinity matrix. Displaying the resultant DBA object shows that 176 of the 1,654 sites are identified as being significantly differentially bound (DB) using the default threshold of FDR <= 0.1:

```
> tamoxifen
```

[1] Note that due to space limitations the reads are not shipped with the package. We hope to make them easily available in the future. Alternatively, you can get the result of the dba.count call by loading the supplied R object by invoking data(tamoxifen_counts)

[2] This step is actually optional: if the main analysis function dba.analyze is invoked with no contrasts established, DiffBind will set up a default set of contrasts automatically, which will include the one we are interested in.



(a) Correlation heatmap, using affinity (read count) data. Generated by: `tamoxifen = dba.count(tamoxifen)`; can also be generated by: `dba.plotHeatmap(tamoxifen)`

(b) Correlation heatmap, using only significantly differentially bound sites. Generated by: `tamoxifen = dba.analyze(tamoxifen)`; can also be generated by: `dba.plotHeatmap(tamoxifen, contrast=1)`

Figure 2: Correlation heatmap plots, generated using `dba.plotHeatmap`.

```
11 Samples, 1654 sites in matrix:
         ID Tissue Factor  Condition Peak.caller Replicate Intervals
1  BT474.1-  BT474     ER  Resistant      counts         1      1654
2  BT474.2-  BT474     ER  Resistant      counts         2      1654
3   MCF7.1+   MCF7     ER Responsive      counts         1      1654
4   MCF7.2+   MCF7     ER Responsive      counts         2      1654
5   MCF7.3+   MCF7     ER Responsive      counts         3      1654
6   T47D.1+   T47D     ER Responsive      counts         1      1654
7   T47D.2+   T47D     ER Responsive      counts         2      1654
8   TAMR.1-   TAMR     ER  Resistant      counts         1      1654
9   TAMR.2-   TAMR     ER  Resistant      counts         2      1654
10  ZR75.1+   ZR75     ER Responsive      counts         1      1654
11  ZR75.2+   ZR75     ER Responsive      counts         2      1654

1 Contrast:
     Group1 Members1     Group2 Members2 DB.edgeR
1 Resistant        4 Responsive        7      176
```

By default, `dba.analyze` plots a correlation heatmap if it finds any significantly differentially bound sites, shown in Figure 2b. Using only the differentially bound sites, we now see that the four tamoxifen resistant samples (representing two cell lines) cluster together, although the MCF7 replicates cluster closer to them than to the other tamoxifen responsive samples. Comparing Figure 2a, which uses all 1,654 consensus binding sites, with Figure 2b, which uses only the 176 differentially bound sites, demonstrates how the differential binding analysis isolates sites that help distinguish between the Resistant and Responsive sample groups.

3.5 Retrieving the differentially bound sites

The final step is to retrieve the differentially bound sites as follows:

```
> tamoxifen.DB = dba.report(tamoxifen)
```

These are returned as a `RangedData` object, appropriate for downstream processing (such as annotation using `ChIPpeakAnno`):

```
> tamoxifen.DB
```

```
RangedData with 176 rows and 6 value columns across 1 space
       space               ranges   |      Conc Conc_Resistant Conc_Responsive
    <factor>            <IRanges>   | <numeric>      <numeric>       <numeric>
1      chr18 [  384853,   386517]   |  6.823605       8.096612        4.431367
2      chr18 [62640747, 62642453]   |  6.934470       2.817637        7.555987
3      chr18 [67583814, 67584869]   |  5.342209       1.003085        5.968130
```

```
4      chr18 [72497301, 72498984]   |  7.855845      4.252805        8.464089
5      chr18 [ 8719898,  8721018]   |  6.666497      7.927462        4.358680
6      chr18 [10013642, 10014854]   |  6.331802      7.540164        4.337558
7      chr18 [59042847, 59045816]   |  7.941078      4.041763        8.557560
8      chr18 [32851316, 32852623]   |  6.081242      2.544904        6.687379
9      chr18 [ 7635601,  7636863]   |  6.221846      7.392009        4.413421
...      ...                  ... ...       ...           ...             ...
168    chr18 [45377496, 45378558]   |  4.554629      5.513895        3.435553
169    chr18 [ 3374456,  3375654]   |  8.350504      9.232769        7.401898
170    chr18 [32345729, 32346998]   |  5.147547      6.059360        4.136942
171    chr18 [61454097, 61456301]   |  6.182678      4.476940        6.664234
172    chr18 [ 1607943,  1608842]   |  5.162252      6.065850        4.169283
173    chr18 [54831153, 54832907]   |  7.562856      5.428491        8.090203
174    chr18 [30556844, 30557672]   |  3.792881      1.893885        4.296954
175    chr18 [ 3696381,  3697136]   |  3.679674      1.667921        4.195417
176    chr18 [69611010, 69612526]   |  5.301286      6.179120        4.361647
          Fold      p.value         FDR
     <numeric>    <numeric>   <numeric>
1     3.665246 1.204475e-06 0.001184865
2    -4.738351 1.432727e-06 0.001184865
3    -4.965046 6.535923e-06 0.003254281
4    -4.211284 7.870086e-06 0.003254281
5     3.568782 1.499719e-05 0.004961069
6     3.202605 2.611534e-05 0.006226854
7    -4.515798 3.413955e-05 0.006226854
8    -4.142474 3.436111e-05 0.006226854
9     2.978587 3.556226e-05 0.006226854
...        ...          ...         ...
168   2.078342  0.008895375  0.08757709
169   1.830871  0.008998139  0.08806463
170   1.922418  0.009288795  0.09037451
171  -2.187294  0.009544180  0.09231622
172   1.896566  0.009717737  0.09301905
173  -2.661712  0.009729320  0.09301905
174  -2.403069  0.009992494  0.09498612
175  -2.527496  0.010156676  0.09599510
176   1.817473  0.010367907  0.09743476
```

The value columns show the mean read concentration over all the samples (the default calculation uses log2 normalized ChIP read counts with control read counts subtracted) and the mean concentration over the first (Resistant) group and second (Responsive) group. The Fold column shows the difference in mean concentrations between the two groups (Conc_Resistant - Conc_Responsive), with a positive value indicating increased binding affinity in the Resistant group and a negative value indicating increased binding affinity in the Responsive group. The final two columns give confidence measures for identifying these sites as differentially bound, with a raw p-value and a multiple testing corrected FDR in the final column.

4 Example: plotting

Besides the correlation heatmaps automatically generated by the core functions, a number of other plots are available using the affinity data. This sections covers MA plots, PCA plots, Boxplots, and Heatmaps.

4.1 MA plots

MA plots are a useful way to visualize the effect of normalization on data, as well as seeing which of the datapoints are being identified as differentially bound. An MA plot can be obtained for the resistant-responsive contrast as follows:

```
> dba.plotMA(tamoxifen)
```

The plot is shown in Figure 3. Each point represents a binding site, with point in red representing sites identified as differentially bound. The plot shows how the differentially bound sites have an absolute log fold difference of at least 2. This same data can also be shown with the concentrations of each sample groups plotted against each other plot using `dba.plotMA(tamoxifen, bXY=T)`.

4.2 PCA plots

While the correlation heatmaps already seen are good for showing clustering, plots based on principal components analysis can be used to give a deeper insight into how samples are associated. A PCA plot corresponding to Figure 2a, which includes normalized read counts for all the binding sites, can be obtained as follows:

```
> dba.plotPCA(tamoxifen, contrast=1, th=1)
```

```
           Legend
Resistant  "black"
Responsive "red"
```

This draws the plot and returns a color legend. The resulting plot (Figure 4a) shows the four Resistant samples (black) not separable from the Responsive samples (red) in either the first (horizontal) or the second (vertical) components when looking at all the binding sites.

A PCA plot using only the differentially bound sites (corresponding to Figure 2b), using an FDR threshold of 0.1, can be drawn as follows:

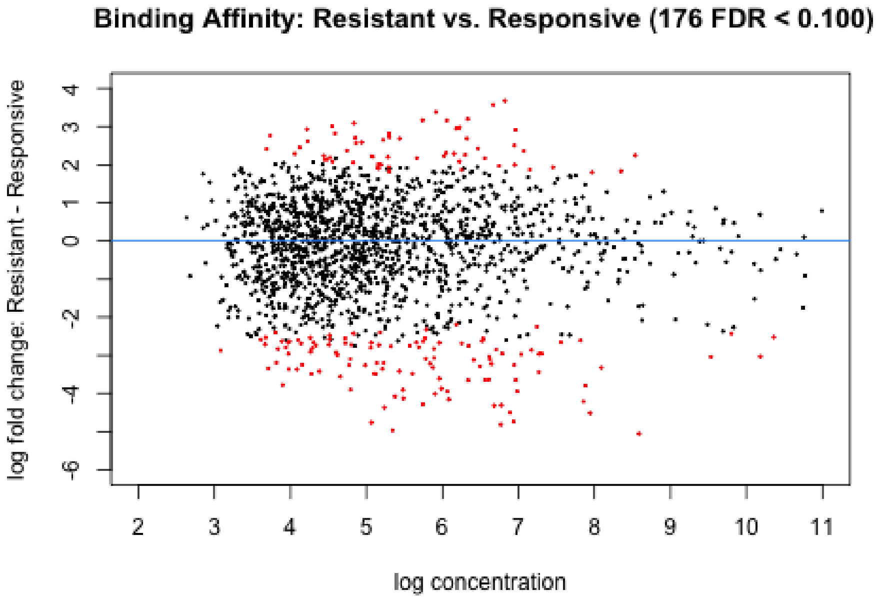


Figure 3: MA plot of Resistant-Responsive contrast, with sites identified as significantly differentially bound shown in red. Generated by: `dba.plotMA(tamoxifen)`

```
> dba.plotPCA(tamoxifen, contrast=1,th=.1)
```

This plot (Figure 4b) shows that the differential analysis identifies sites than can be used to separate the sample groups along both the first and second components.

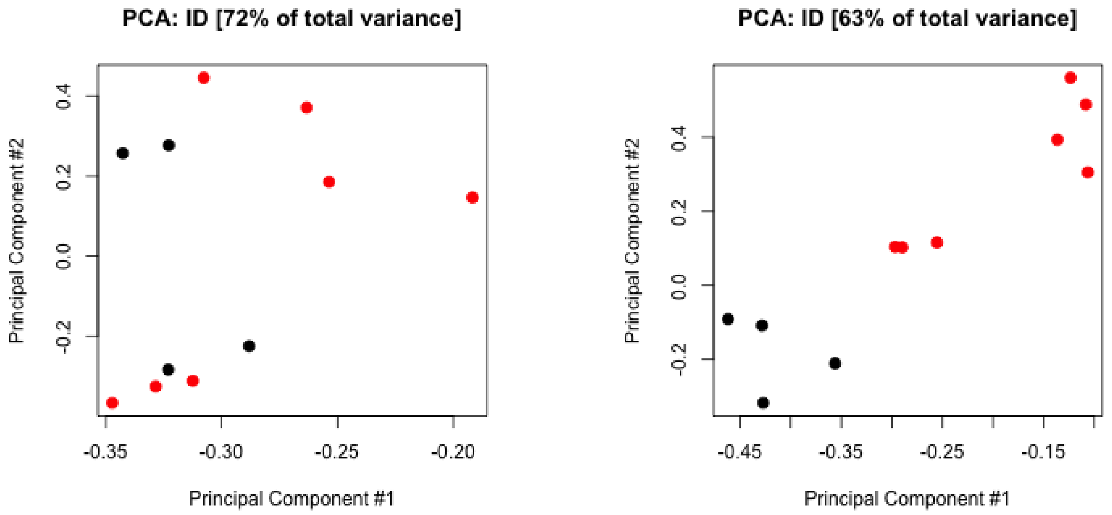
The `dba.plotPCA` function is customizable. For example, if you want to see where each of the cell lines lies, type `dba.plotPCA(tamoxifen, DBA_TISSUE)`. If your installation of R supports 3D graphics using the `rgl` package, try `dba.plotPCA(tamoxifen, b3D=T)`. You may have to rotate the resultant plot top-to-bottom to align it with the standard two-dimension plot, but seeing the first three principal components can be a useful exploratory exercise.

4.3 Boxplots

Boxplots provide a way to view how read distributions differ between classes of binding sites. Consider the example, where the 176 differentially bound sites are identified. The MA plot (Figure 3) shows that these are not distributed evenly between those that increase binding affinity in the Responsive group vs. those that increase binding affinity in the Resistant groups. This can be seen quantitatively using the sites returned in the report:

```
> sum(tamoxifen.DB$Fold<0)

[1] 116
```



(a) PCA plot using affinity data for all sites. Generated by: `dba.plotPCA(tamoxifen,contrast=1,th=1)`

(b) PCA plot using affinity data for only differentially bound sites. Generated by: `dba.plotPCA(tamoxifen, contrast=1)`

Figure 4: Principal Component Analysis (PCA) plots, generated using `dba.plotPCA`.

```
> sum(tamoxifen.DB$Fold>0)
```

```
[1] 60
```

But how are reads distributed amongst the different classes of differentially bound sites and sample groups? These data can be more clearly seen using a boxplot:

```
> pvals = dba.plotBox(tamoxifen)
```

The default plot (Figure 5) shows in the first two boxes that amongst differentially bound sites overall, the Responsive samples have a somewhat higher mean read concentration. The next two boxes show the distribution of reads in differentially bound sites that exhibit increased affinity in the Responsive samples, while the final two boxes show the distribution of reads in differentially bound sites that exhibit increased affinity in the Resistant samples.

`dba.plotBox` returns a matrix of p-values (computed using a two-sided Wilcoxon 'Mann-Whitney' test) indicating which of these distributions are significantly different from another distribution.

```
> pvals
```

```
               Resistant.DB Responsive.DB Resistant.DB+ Responsive.DB+
Resistant.DB   1.000000e+00  9.336872e-09  4.089980e-06   1.534225e-13
Responsive.DB  9.336872e-09  1.000000e+00  2.033085e-28   2.951266e-04
```

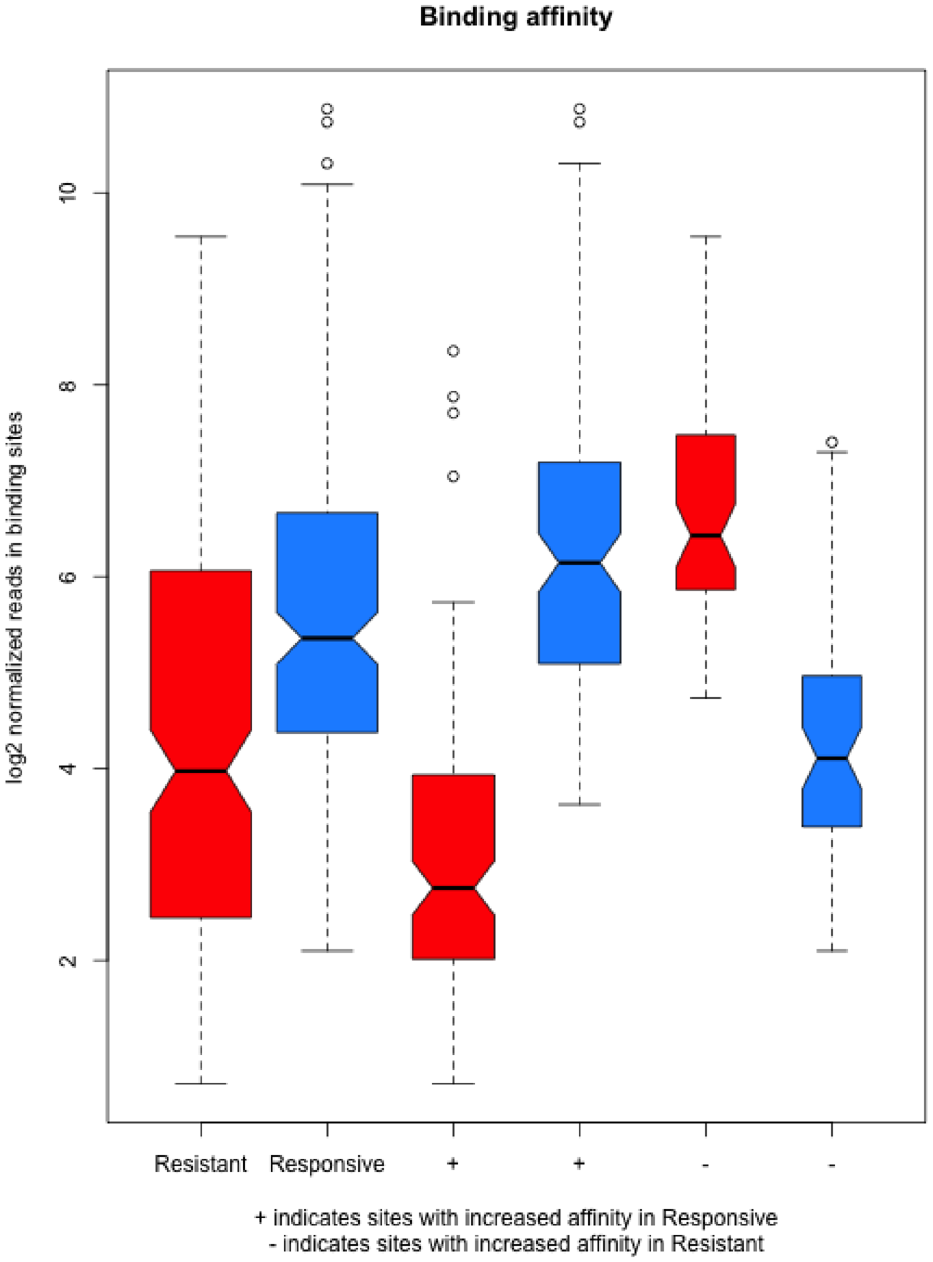


Figure 5: Box plots of read distributions for significantly differentially bound (DB) sites. Tamoxifen resistant samples are shown in red, and responsive samples are shown in blue. Left two boxes show distribution of reads over all DB sites in the Resistant and Responsive groups; middle two boxes show distributions of reads in DB sites that increase in affinity in the Responsive group; last two boxes show distributions of reads in DB sites that increase in affinity in the Resistant group. Generated by: `dba.plotBox(tamoxifen)`

```
Resistant.DB+  4.089980e-06  2.033085e-28  1.000000e+00   1.057582e-31
Responsive.DB+ 1.534225e-13  2.951266e-04  1.057582e-31   1.000000e+00
Resistant.DB-  1.062977e-12  3.581463e-07  3.273505e-24   1.563846e-02
Responsive.DB- 5.580471e-01  2.195392e-08  1.407336e-08   1.513744e-15
               Resistant.DB- Responsive.DB-
Resistant.DB    1.062977e-12   5.580471e-01
Responsive.DB   3.581463e-07   2.195392e-08
Resistant.DB+   3.273505e-24   1.407336e-08
Responsive.DB+  1.563846e-02   1.513744e-15
Resistant.DB-   1.000000e+00   4.196047e-16
Responsive.DB-  4.196047e-16   1.000000e+00
```

The significance of the overall difference in distribution of concentrations amongst the differentially bound sites in the two groups is shown to be p-value=9.336872e-09, while those between the Resistant and Responsive groups in the individual cases (increased in Responsive or Resistant) have p-values computed as 1.107548e-31 and 4.018128e-16.

4.4 Heatmaps

DiffBind provides two types of heatmaps. This first, correlation heatmaps, we have already seen. For example, the heatmap shown in Figure 2b can be generated as followws:

```
> corvals = dba.plotHeatmap(tamoxifen,contrast=1)
```

Another way to view the patterns of binding affinity directly in the differentially bound sites is via a binding affinty heatmap. This can be plotted for the example case as follows:

```
> corvals = dba.plotHeatmap(tamoxifen, contrast=1, correlations=FALSE)
```

Figure 6 shows the affinities and clustering of the differentially bound sites (rows), as well as the sample clustering (columns). This plot can be tweaked to get more contrast, for example by using row-scaling `dba.plotHeatmap(tamoxifen, contrast=1, correlations=FALSE, scale=row)`.

5 Example: occupancy analysis and overlaps

In this section, we look at the tamoxifen resistance ER-binding dataset in some more detail, showing what a pure occupancy-based analysis would look like, and comparing it to the results obtained using the affinity data. For this we will start by re-loading the peaksets:

```
> data(tamoxifen_peaks)
```

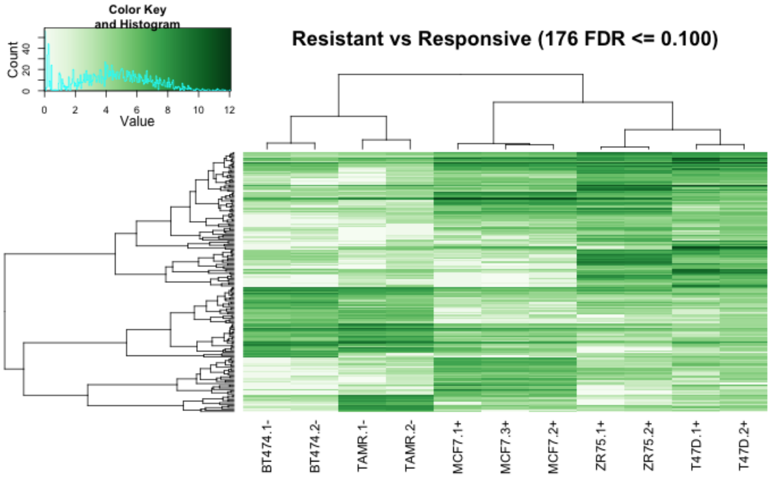


Figure 6: Binding affinity heatmap showing affinities for differentially bound sites. Samples cluster first by whether they are responsive to tamoxifen treatment, then by cell line. Clusters of binding sites show distinct patterns of affinity levels. Generated by: `dba.plotHeatmap(tamoxifen, contrast=1, correlations=FALSE)`

5.1 Overlap rates

One reason to do an occupancy-based analysis is to determine what candidate sites should be used in a subsequent affinity-based analysis. In the example so far, we took all sites that were identified in peaks in at least three of the eleven peaksets, reducing the number of sites from 3,557 overall to the 1,654 sites used in the differential analysis. We could have used a more stringent criterion, such as only taking sites identified in five or six of the peaksets, or a less stringent one, such as including all 3,557 sites. In making the decision of what criteria to use many factors come into play, but it helps to get an idea of the rates at which the peaksets overlap (for more details on how overlaps are determined, see Section 6.1 on peak merging). A global overview can be obtained using the RATE mode of the `dba.overlap` function as follows:

```
> olap.rate = dba.overlap(tamoxifen,mode=DBA_OLAP_RATE)
> olap.rate
```

```
 [1] 3557 2602 1654 1299 1002  764  620  455  352  187  118
```

`olap.rate` is a vector containing the number of peaks that appear in at least one, two, three, and so on up to all eleven peaksets.

These values can be plotted to show the overlap rate drop-off curve:

```
> plot(olap.rate,type='b')
```

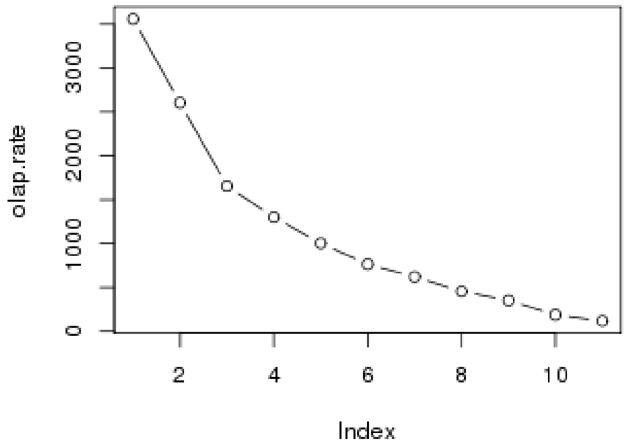


Figure 7: Overlap rate plot, showing how the number of overlapping peaks decreases as the overlap criteria becomes more stringent. X axis shows the number of peaksets in which the site is identified, while the Y axis shows the number of overlapping sites. Generated by plotting the result of: `dba.overlap(tamoxifen,mode=DBA_OLAP_RATE)`

The rate plot is shows in Figure 7. These curves typically exhibit a roughly geometric drop-off, with the number of overlapping sites halving as the overlap criterion become stricter by one site. When the drop-off is extremely steep, this is an indication that the peaksets do not agree very well. For example, if there are replicates you expect to agree, there may be a problem with the experiment. In the current example, peak agreement is high and the curve exhibits a better than geometric drop-off.

5.2 Deriving consensus peaksets

When performing an overlap analysis, it is often the case that the overlap criteria are set stringently in order to lower noise and drive down false positives.[3] The presence of a peak in multiple peaksets is an indication that it is a ”real” binding site, in the sense of being identifiable in a repeatable manner. The use of biological replicates (performing the ChIP multiple times), as in the tamoxifen dataset, can be used to guide derivation of a consensus peakset. Alternatively, an inexpensive but less powerful way to help accomplish this is to use multiple peak callers for each ChIP dataset and look for agreement between peak callers (Li et al. [2011]).

Consider for example the MCF7 cell line, represented by three replicates in this dataset. How well do the replicates agree on their peak calls? The overlap rate for just the MCF7 samples can

[3] It is less clear that limiting the potential binding sites in this way is appropriate when focusing on affinity data, as the differential binding analysis method will identify only sites that are significantly differentially bound, even if operating on peaksets that include incorrectly identified sites.

be isolated using a *sample mask*. A set of sample masks are automatically associated with a `DBA object` in the `$masks` field:

```
> names(tamoxifen$masks)
```

```
 [1] "BT474"       "MCF7"        "T47D"        "TAMR"        "ZR75"       
 [6] "ER"          "Resistant"   "Responsive"  "raw"         "Replicate.1"
[11] "Replicate.2" "Replicate.3"
```

Arbitrary masks can be generated using the `dba.mask` function, or simply by specifying a vector of peakset numbers. In this case, a mask that isolates the MCF7 samples can be passed into the `dba.overlap` function:

```
> dba.overlap(tamoxifen,tamoxifen$masks$MCF7,mode=DBA_OLAP_RATE)
```

```
[1] 1767 1222  874
```

There are 874 peaks (out of 1,767) identified in all three replicates. A finer grained view of the overlaps can be obtained with the `dba.plotVenn` function:

```
> dba.plotVenn(tamoxifen,tamoxifen$masks$MCF7)
```

The resultant plot is shown as Figure 8. This plot shows the 874 consensus peaks identified as common to all replicates, but further breaks down how the replicates relate to each other. The same can be done for each of the replicated cell line experiments, and rather than applying a global cutoff (3 of 11), each cell line could be dealt with individually in deriving a final peakset. For example we could replace the three MCF7 peaksets with one including the 1,222 peaks identified in at least two replicates by masking the non-consensus MCF7 peaksets (the $Consensus mask filters peaksets formed from existing peaksets using `dba.peakset`) as follows:

```
> tamoxifen = dba.peakset(tamoxifen,tamoxifen$masks$MCF7,sampID="MCF7+")
> tamoxifen = dba(tamoxifen,!(!tamoxifen$masks$Consensus&tamoxifen$masks$MCF7))
> tamoxifen
```

```
9 Samples, 2377 sites in matrix (3322 total):
        ID Tissue Factor  Condition Peak.caller Replicate Intervals
1 BT474.1-  BT474     ER  Resistant         raw         1      1084
2 BT474.2-  BT474     ER  Resistant         raw         2      1115
3  T47D.1+   T47D     ER Responsive         raw         1       509
4  T47D.2+   T47D     ER Responsive         raw         2       347
5  TAMR.1-   TAMR     ER  Resistant         raw         1      1148
6  TAMR.2-   TAMR     ER  Resistant         raw         2       933
7  ZR75.1+   ZR75     ER Responsive         raw         1      2111
8  ZR75.2+   ZR75     ER Responsive         raw         2      1975
9    MCF7+   MCF7     ER Responsive         raw     1-2-3      1222
```

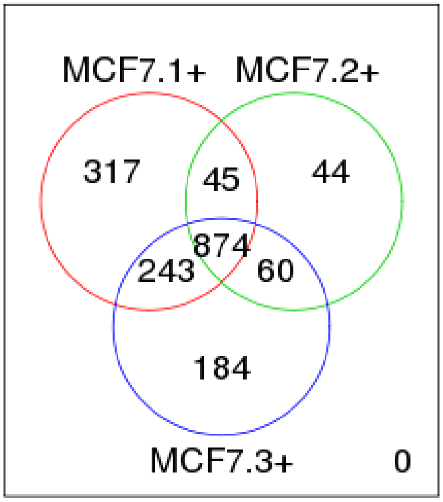


Figure 8: Venn diagram showing how the ER peak calls for three replicates of MCF7 cell line overlap. Generated by plotting the result of: `dba.venn(tamoxifen,tamoxifen$masks$MCF7)`

5.3 A complete occupancy analysis: identifying sites unique to a sample group

Occupancy-based analysis, in addition to offering many ways of deriving consensus peaksets, can also be used to identify sites unique to a group of samples. This is analogous to, but not the same as, finding differentially bound sites. In these subsections, the two approaches are directly compared.

Returning to the original tamoxifen dataset:

```
> data(tamoxifen_peaks)
```

We can derive consensus peaksets for the Resistant and Responsive groups. First we examine the overlap rates:

```
> dba.overlap(tamoxifen,tamoxifen$masks$Resistant,mode=DBA_OLAP_RATE)

[1] 1875 1298  597  436

> dba.overlap(tamoxifen,tamoxifen$masks$Responsive,mode=DBA_OLAP_RATE)

[1] 3208 2293 1217  807  584  265  161
```

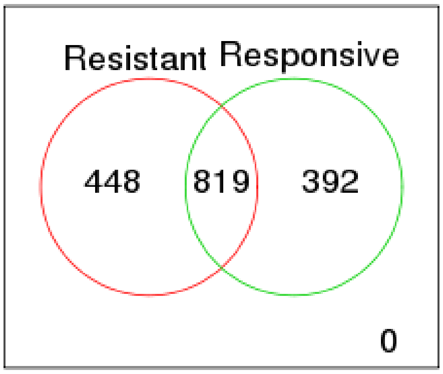


Figure 9: Venn diagram showing how the ER peak calls for two response groups overlap. Generated by plotting the result of: `dba.plotVenn(tamoxifen, tamoxifen$masks$Consensus)`

For the Resistant group, we choose an overlap rate of two, leaving 1,298 sites, while for the Responsive group, we use an overlap rate of 3, leaving 1,217 sites, and look at the overlap between the two groups:

```
> tamoxifen = dba.peakset(tamoxifen,tamoxifen$masks$Resistant,
+                         sampID="Resistant",minOverlap=2)
> tamoxifen = dba.peakset(tamoxifen,tamoxifen$mask$Responsive,
+                         sampID="Responsive",minOverlap=3)
> dba.plotVenn(tamoxifen,tamoxifen$masks$Consensus)
```

Figure 9 shows that 448 sites are unique to the Resistant group, and 392 sites are unique to the Responsive group, with 819 sites being identified in both groups (meaning in at least half the Resistant samples and at least three of the seven Responsive samples). If our primary interest is in finding binding sites that are different between the two groups, it may seem reasonable to consider the 819 common sites to be uninteresting, and focus on the 840 sites that are unique to a specific group. The sites can be obtained using `dba.overlap`:

```
> tamoxifen.OL = dba.overlap(tamoxifen, tamoxifen$masks$Consensus)
```

The sites unique to the Resistant group are accessible in `tamoxifen.OL$onlyA`, with the Responsive-unique sites in `tamoxifen.OL$onlyB`:

```
> tamoxifen.OL$onlyA

RangedData with 448 rows and 1 value column across 1 space
       space               ranges   |  Resistant
    <factor>            <IRanges>   |  <numeric>
1      chr18     [301531, 302172]   | 0.12897827
2      chr18     [346557, 347362]   | 0.02545366
3      chr18     [361121, 362106]   | 0.01780650
4      chr18     [384853, 386517]   | 0.08871906
5      chr18     [479200, 480032]   | 0.08687365
6      chr18     [562348, 563067]   | 0.03202105
7      chr18     [639178, 639957]   | 0.05527764
8      chr18     [647089, 647869]   | 0.05521693
9      chr18     [914869, 915486]   | 0.06593881
...      ...                  ... ...        ...
440    chr18 [74267807, 74268931]   | 0.08594548
441    chr18 [74313069, 74313720]   | 0.05233002
442    chr18 [74629529, 74630608]   | 0.01670948
443    chr18 [74675317, 74676231]   | 0.06404468
444    chr18 [75157775, 75158504]   | 0.03463882
445    chr18 [75163026, 75163816]   | 0.03066147
446    chr18 [75401417, 75402162]   | 0.04940022
447    chr18 [75525519, 75526188]   | 0.06458959
448    chr18 [75826088, 75826939]   | 0.02528083
```

```
> tamoxifen.OL$onlyB

RangedData with 392 rows and 1 value column across 1 space
       space               ranges   | Responsive
    <factor>            <IRanges>   |  <numeric>
1      chr18   [ 336592,  337347]   | 0.04664408
2      chr18   [ 439109,  440079]   | 0.03310468
3      chr18   [ 988767,  989698]   | 0.04701041
4      chr18   [1065304, 1066051]   | 0.06951515
5      chr18   [1231653, 1232311]   | 0.05523114
6      chr18   [1369773, 1370746]   | 0.08923925
7      chr18   [1470236, 1470902]   | 0.06573040
8      chr18   [2161751, 2162455]   | 0.03927969
9      chr18   [2197080, 2197908]   | 0.04607203
...      ...                  ... ...        ...
384    chr18 [72790966, 72791819]   | 0.08770689
385    chr18 [72914352, 72914983]   | 0.03892986
386    chr18 [73517930, 73518921]   | 0.02559103
```

```
387     chr18 [74835590, 74836200]    | 0.12889933
388     chr18 [74850775, 74851581]    | 0.07360388
389     chr18 [74860617, 74861878]    | 0.05664056
390     chr18 [74906349, 74907306]    | 0.03568147
391     chr18 [75641971, 75642647]    | 0.06198451
392     chr18 [76087923, 76089259]    | 0.12598012
```

The scores associated with each site are derived from the peak caller confidence score, and are a measure of confidence in the peak call (occupancy), not a measure of how strong or distinct the peak is.

5.4 Comparison of occupancy and affinity based analyses

So how does this occupancy-based analysis compare to the previous affinity-based analysis?

First, different criteria were used to select the overall consensus peakset. We can compare them to see how well they agree:

```
> tamoxifen = dba.peakset(tamoxifen,tamoxifen$masks$Consensus,
+                         minOverlap=1,sampID="OL Consensus")
> tamoxifen = dba.peakset(tamoxifen,!tamoxifen$masks$Consensus,
+                         minOverlap=3,sampID="Consensus_3")
> dba.plotVenn(tamoxifen,14:15)
```

Figure 10 shows that the two sets agree on about 85% of their sites, so the results should be directly comparable.[4]

Next re-load the affinity analysis:

```
> data(tamoxifen_analysis)
```

To compare the sites unique to each sample group identified from the occupancy analysis with those sites identified as differentially bound based on affinity (read count) data, we use a feature of `dba.report` that facilitates evaluating the occupancy status of sites. Here we obtain a report of all the sites (`th=1`) with occupancy statistics (`bCalled=T`):

```
> tamoxifen.rep = dba.report(tamoxifen,bCalled=T,th=1)
```

The `bCalled` option adds two columns to the report (`Called1` and `Called2`), one for each group, giving the number of samples within the group in which the site was identified as a peak in the original peaksets generated by the peak caller. We can use these to recreate the overlap criteria used in the occupancy analysis:

[4] Alternatively, we could re-run the analysis using the newly derived consensus peakset by passing it into the counting function: `> tamoxifen = dba.count(tamoxifen, peaks = dba.peakset(tamoxifen, peaks=14, bRetrieve=T))`

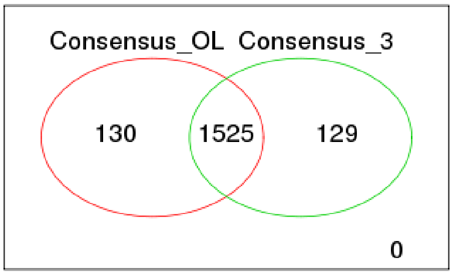


Figure 10: Venn diagram showing how the ER peak calls for two different ways of deriving consensus peaksets. Generated by plotting the result of: `dba.plotVenn(tamoxifen,14:15)`

```
> onlyResistant = tamoxifen.rep$Called1>=2 & tamoxifen.rep$Called2<3
> sum(onlyResistant )

[1] 313

> onlyResponsive = tamoxifen.rep$Called2>=3 & tamoxifen.rep$Called1<2
> sum(onlyResponsive)

[1] 391

> bothGroups = tamoxifen.rep$Called1>=2 & tamoxifen.rep$Called2>=3
> sum(bothGroups)

[1] 821
```

Comparing these numbers verifies the similarity with those seen in Figure 9.

Focusing on only those sites identified as significantly differentially bound (FDR <= 0.1), however, shows a different story than that obtainable using only occupancy data:

```
> tamoxifen.DB = dba.report(tamoxifen,bCalled=T,th=.1)
> onlyResistant = tamoxifen.DB$Called1>=2 & tamoxifen.DB$Called2<3
> sum(onlyResistant)

[1] 32

> onlyResponsive = tamoxifen.DB$Called2>=3 & tamoxifen.DB$Called1<2
> sum(onlyResponsive)
```

```
[1] 87
```

```
> bothGroups = tamoxifen.DB$Called1>=2 & tamoxifen.DB$Called2>=3
> sum(bothGroups)
```

```
[1] 54
```

There are a number of notable differences in the results. First there are many fewer sites identified as differentially bound. Indeed, most of the sites identified in the occupancy analysis as unique to a sample group are not found to be significantly differentially bound using the affinity data. While partly this is a result of the stringency of the statistical tests, it shows how the affinity analysis can discriminate between sites where peak callers are making occupancy decisions that do not reflect significant differences in read densities at these sites. Secondly, while the sites unique to a sample group are more likely to be identified as differentially bound, differentially bound sites are as likely to be called in the consensus of both response groups as they are to be unique to one group, as *nearly a third of the sites identified as significantly differentially bound are called as peaks in both response groups.* Finally, the differentially bound peaks identified using the affinity analysis are associated with significance statistics (p-value and FDR) that can be used to rank them for further examination, while the occupancy analysis yields a relatively unordered list of peaks, as the peak caller statistics refer only to the significance of occupancy, and not of differential binding.

6 Technical notes

This section includes some technical notes explaining some of the internal technical details of DiffBind processing.

6.1 Merging peaks

When forming the global binding matrix consensus peaksets, DiffBind first identifies all unique peaks amongst the relevant peaksets. As part of this process, it merges overlapping peaks, replacing them with a single peak representing the narrowest region that covers all peaks that overlap by at least one base. There are at least two consequences of this that are worth noting.

First, as more peaksets are included in analysis, the average peak width tends to become longer as more overlapping peaks are detected and the start/end points are adjusted outward to account for them. Secondly, peak counts may not appear to add up as you may expect due to merging. For example, if one peakset contains two small peaks near to each other, while a second peakset includes a single peak that overlaps both of these by at least one base, these will all be replaced in the merged matrix with a single peak. As more peaksets are added, multiple peaks from multiple peaksets may be merged together to form a single, wider peak.

6.2 edgeR analysis

When dba.analyze is invoked using the default method=DBA_EDGER, a standardized differential analysis is performed using the edgeR package (Robinson et al. [2010]). This section details the precise steps in that analysis.

For each contrast, a separate analysis is performed. First, a matrix of counts is constructed for the contrast, with columns for all the samples in the first group, followed by columns for all the samples in the second group. The raw read count is used for this matrix; if the bSubControl parameter is set to TRUE (as it is by default), the raw number of reads in the control sample (if available) will be subtracted (with a minimum final read count of 1). Next the library size is computed for each sample for use in subsequent normalization. By default, this is the total number of reads in peaks (the sum of each column). Alternatively, if the bFullLibrarySize parameter is set to TRUE, the total number of reads in the library (calculated from the source BAM/SAM/BED file) is used. The default setting is appropriate for situations when the overall signal is expected to be directly comparable between the samples; using the full library size may be preferable if samples are expected to have dramatically different signals (e.g., if some are expected to have very low binding rates compared to others). The first step concludes with a call to edgeR 's DGEList function.

The DGEList object that results is next passed to calcNormFactors with all other parameters retained as defaults, returning an updated DGEList object. This is passed to estimateCommonDisp with default parameters. If bTagwise is TRUE (most useful when there are at least three members in each group of a contrast), the resulting DGEList object is then passed to estimateTagwiseDisp, with the prior set to 50 divided by two less than the total number of samples in the contrast. This final DGEList for contrast n is stored in the DBA object as

DBA$contrasts[[n]]$edgeR

and may be examined and manipulated directly for further customization.

The final steps are to perform testing to determine the significance measure of the differences between the sample groups by calling exactTest using the DGEList with the dispersion set based on the bTagwise parameter. The resulting DGEExact object is stored in the DiffBind object as:

DBA$contrasts[[n]]$edgeR$db.

When data are retrieved for plotting or reporting, topTags is called on the DGEExact object, using the default FDR method BH.

6.3 edgeR analysis with blocking factor

When dba.analyze is invoked using method=DBA_EDGER, and a blocking factor has been added to the contrast, and additional edgeR analysis is carried out using a generalized linear model. The analysis mostly follows the matched tumor-normal example in the edgeR User's Guide. If samples are not perfectly matched (every sample in the first group has exactly one matching sample in the second group) a warning is generated (from dba.contrast) but the analysis should still complete successfully. A design matrix is constructed via a call to model.matrix; the first columns is the Intercept column, the last column represents the second group, and there will be one fewer middle columns than the number of unique replicates.

For each contrast, a separate analysis is performed. First, a matrix of counts is constructed for the contrast, with columns for all the samples in the first group, followed by columns for all the samples in the second group. The raw read count is used for this matrix; if the `bSubControl` parameter is set to `TRUE` (as it is by default), the raw number of reads in the control sample (if available) will be subtracted (with a minimum final read count of 1). Next the library size is computed for each sample for use in subsequent normalization. By default, this is the total number of reads in peaks (the sum of each column). Alternatively, if the `bFullLibrarySize` parameter is set to TRUE, the total number of reads in the library (calculated from the source BAM/SAM/BED file) is used. The default setting is appropriate for situations when the overall signal is expected to be directly comparable between the samples; using the full library size may be preferable if samples are expected to have dramatically different signals (e.g., if some are expected to have very low binding rates compared to others). The first step concludes with a call to `edgeR` 's `DGEList` function.

The `DGEList` object that results is next passed to `calcNormFactors` with all other parameters retained as defaults, returning an updated `DGEList` object. This is passed to `estimateGLMCommonDisp` along with the design matrix. If bTagwise is TRUE (most useful when there are at least three members in each group of a contrast), the resulting `DGEList` object is then passed to `estimateGLMTagwiseDisp`, along with the design matrix. This final `DGEList` for contrast n is stored in the `DBA object` as

`DBA$contrasts[[n]]$edgeR$block`

and may be examined and manipulated directly for further customization.

Next `glmFit` is called with the design matrix, and the result stored. Finally, `glmLRT` is called based on the highest number coefficient (number of replicates plus one), returning a `DGEGLM` object, stored in

`DBA$contrasts[[n]]$edgeR$block$LRT`

When data are retrieved for plotting or reporting, `topTags` is called on the `DGEGLM` object, using the default FDR method `BH`.

6.4 DESeq analysis

When `dba.analyze` is invoked using `method=DBA_DESEQ`[5], a standardized differential analysis is performed using the `DESeq` package (Anders and Huber [2010]). This section details the precise steps in that analysis.

For each contrast, a separate analysis is performed. First, a matrix of counts is constructed for the contrast, with columns for all the samples in the first group, followed by columns for all the samples in the second group. The raw read count is used for this matrix; if the `bSubControl` parameter is set to `TRUE` (as it is by default), the raw number of reads in the control sample (if available) will be subtracted. Next the library size is computed for each sample for use in subsequent normalization. By default, this is the total number of reads in peaks (the sum of each

[5] Note that `DESeq` can be made the default analysis method for a DBA object by setting `DBA$config$AnalysisMethod=DBA_DESEQ`.

column). Alternatively, if the bFullLibrarySize parameter is set to TRUE, the total number of reads in the library (calculated from the source BAM/SAM/BED file) is used. The first step concludes with a call to DESeq's newCountDataSet function, which returns a CountDataSet object. If bFullLibrarySize is set to TRUE, then sizeFactors is called with the number of reads in the BAM/SAM/BED files for each ChIP sample, divided by the minimum of these; otherwise, estimateSizeFactors is invoked. Either way, the resulting CountDataSet object is accessible as

DBA$contrasts[[n]]$DESeq$DEdata

Next, estimateDispersions is called with the CountDataSet object and fitType set to local. If there are no replicates, (only one sample in each group), method is set to blind. Otherwise, if bTagwise is TRUE, method is set to per-condition; if it is FALSE, method is set to pooled. Finally, nbinomTest is called, and the result (reordered by adjusted p-value) saved for reporting.

7 Acknowledgements

This package was developed at Cancer Research UK's Cambridge Research Institute with the help and support of many people there. We wish to acknowledge everyone the Bioinformatics Core under the leadership of Matthew Eldridge, as well as the Nuclear Receptor Transcription Laboratory under the leadership of Jason Carroll. Researchers who contributed ideas and/or pushed us in the right direction include Caryn-Ross Innes, Vasiliki Theodorou, and Tamir Chandra among many others. We also thank members of the Gordon Smyth laboratory at the WEHI, Melbourne, particularly Mark Robinson and Davis McCarthy, for helpful discussions.

8 Setup

This vignette was built on:

```
> sessionInfo()

R version 2.14.2 (2012-02-29)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=C                 LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
```

```
[8] base

other attached packages:
[1] DiffBind_1.0.9 Biobase_2.14.0

loaded via a namespace (and not attached):
 [1] IRanges_1.12.6     RColorBrewer_1.0-5 amap_0.8-7         edgeR_2.4.6
 [5] gdata_2.8.2        gplots_2.10.1      gtools_2.6.2       limma_3.10.3
 [9] tools_2.14.2       zlibbioc_1.0.1
```

References

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, Oct 2010. doi: 10.1186/gb-2010-11-10-r106.

Q. Li, J.B. Brown, H. Huang, and P. Bickel. Measuring reproducibility of high-throughput experiments. *Annals of Applied Statistics*, 2011.

Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–40, Jan 2010. doi: 10.1093/bioinformatics/btp616. URL `http://bioinformatics.oxfordjournals.org/cgi/content/full/26/1/139`.

C.S. Ross-Innes, R. Stark, A.E. Teschendorff, K.A. Holmes, H.R. Ali, M.J. Dunning, G.D. Brown, O. Gojis, I.O. Ellis, A.R. Green, et al. Differential oestrogen receptor binding is associated with clinical outcome in breast cancer. *Nature*, 481(7381):389–393, 2012.

Y. Zhang, T. Liu, C.A. Meyer, J. Eeckhoute, D.S. Johnson, B.E. Bernstein, C. Nussbaum, R.M. Myers, M. Brown, W. Li, et al. Model-based analysis of chip-seq (macs). *Genome Biol*, 9(9): R137, 2008.