Performing Some Basic Quality Checking and Analysis on Sequencing Data with Repitools

Mark Robinson Aaron Statham Dario Strbenac

November 12, 2010

1 Introduction

Repitools is a package that allows statistics of differential epigenetic marking to be calculated, as well as summaries of genome - wide trends to be visualised in a variety of formats. Some basic quality checking utilities are also available for sequencing data. The utility of Repitools comes from that most of the functionality available is implemented for both microarrays and next generation sequencing, with very similar function calls for both types of data.

In this vignette, quality checking of the sequencing data, followed by analysis and visualisation will be demonstrated. A more detailed description of the package can be found in the associated Bioinformatics Applications Note [1]

To start with, load the Repitools package.

```
> library(Repitools)
```

2 Data

A GRangesList of short reads from an Illumina Genome Analyser of four samples is included with the package. Only reads on chromosome 21 were kept, to have fast - running examples. The details of the samples are :

```
> dataPath <- system.file("data", package = "Repitools")
> load(paste(dataPath, "samplesList.Rdata", sep = .Platform$file.sep))
> names(samplesList)

[1] "Cancer Input"  "Cancer MBD2IP" "Normal Input"  "Normal MBD2IP"
```

Also, an annotation of genes located on chromosome 21 is included.

```
> geneAnno <- read.csv(paste(dataPath, "chr21genes.csv", sep = .Platform$file.sep),
+     stringsAsFactors = FALSE)
> head(geneAnno)
```

[1] Repitools: an R package for the analysis of enrichment-based epigenomic data

```
       name   chr strand    start      end   symbol
1 NM_199260 chr21      -  9928613 10012791     TPTE
2 NM_182482 chr21      - 10042712 10120796    BAGE2
3 NM_001187 chr21      - 10079666 10120808     BAGE
4 NR_026916 chr21      + 13332357 13412440 C21orf99
5 NM_174981 chr21      + 13904368 13935777    POTED
6 NR_026755 chr21      - 14137325 14142556 C21orf15
```

Lastly, there is matrix of gene expression difference data, with each element related to the corresponding row in the gene annotation table. These values were artificially generated. The expression differences matrix will be used when illustrating some of the visualisation functionality later in the vignette.

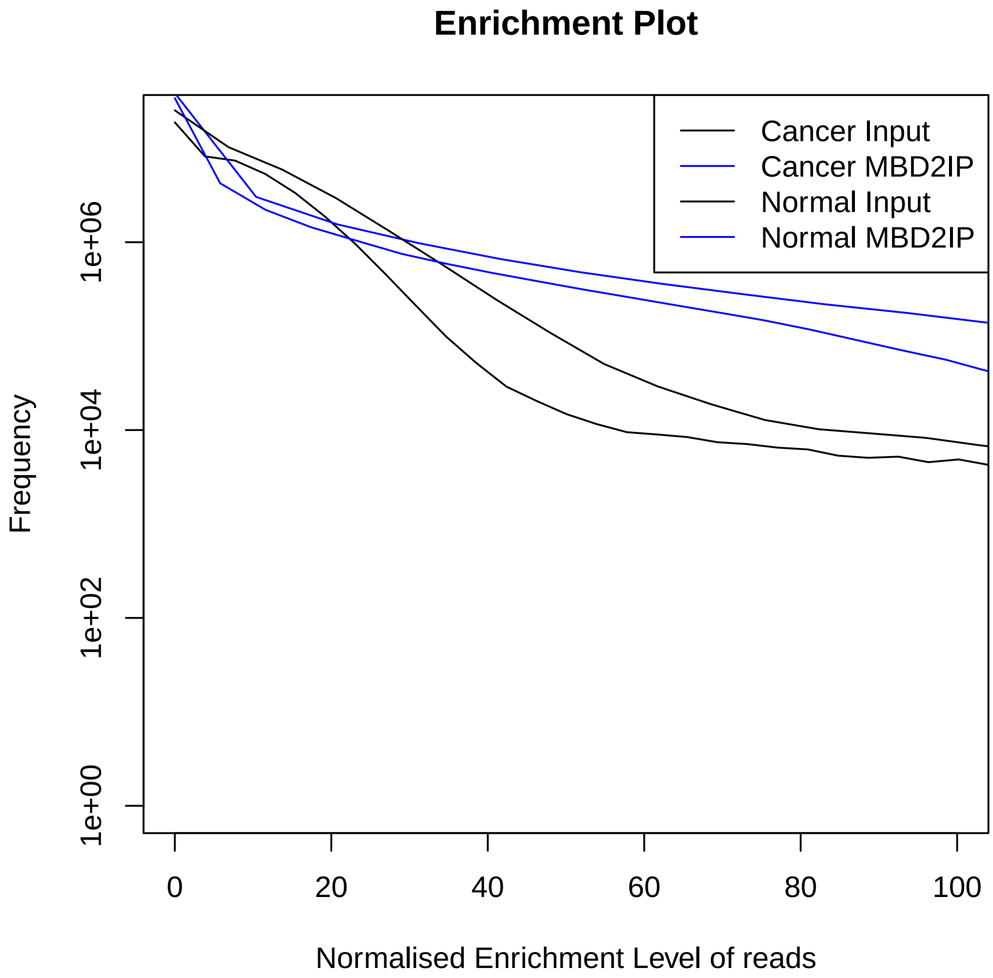
```
> load(paste(dataPath, "expr.Rdata", sep = .Platform$file.sep))
> head(expr)
```

```
          Expression Difference
NM_199260             1.0900000
NM_182482             8.5100000
NM_001187            -0.1758591
NR_026916             3.3350484
NM_174981            -1.1676130
NR_026755            -1.8425325
```

3 Quality Checking

Notice that two of the samples are MBD2 IPs, and two are inputs. Therefore, the IP samples should differ to the inputs in two ways. Firstly, they should be more CpG rich, since DNA methylation rarely ever occurs outside of this sequence context. Also, since DNA methylation tends to occur in peaks, rather than spread out regions, a higher frequency of bases should have high coverage of reads in the IP samples than in input samples. The `enrichmentPlot` and `cpgDensityPlot` functions allow examination of this.

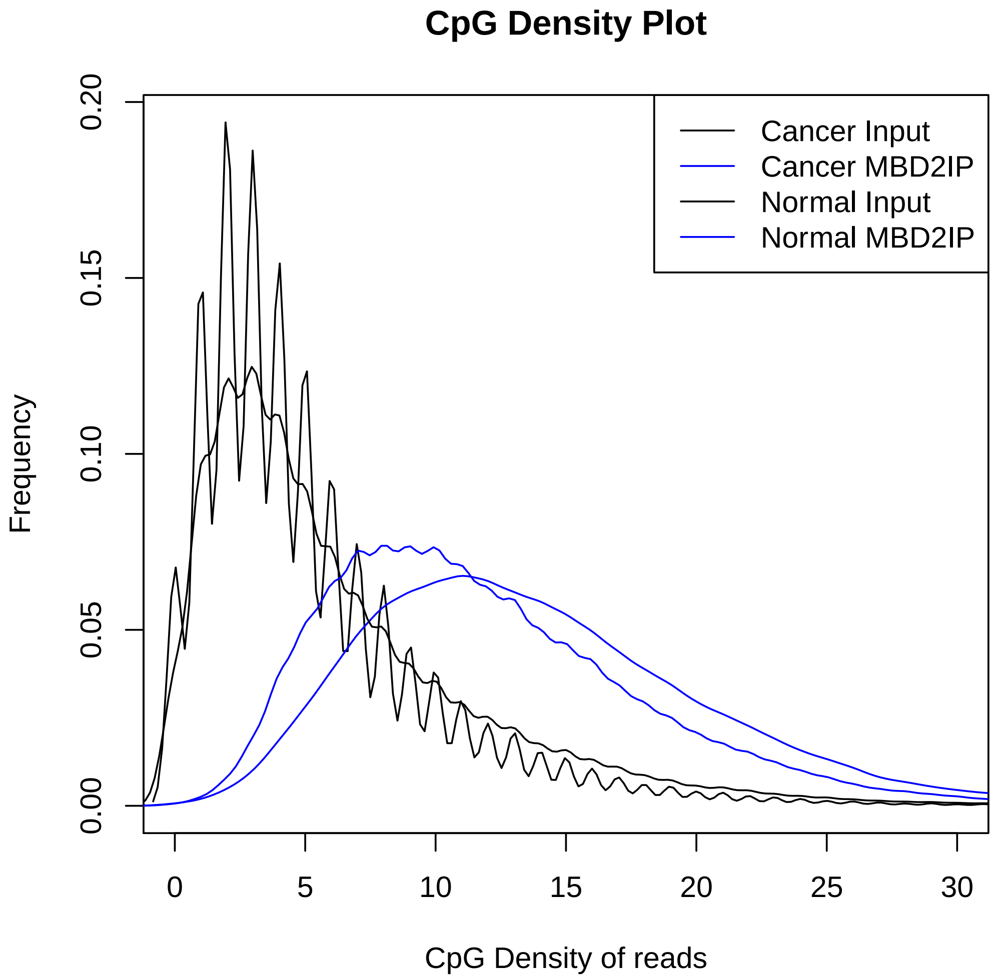
```
> library(BSgenome.Hsapiens.UCSC.hg18)
> enrichmentPlot(samplesList, Hsapiens, 300, cols = c("black",
+     "blue", "black", "blue"), xlim = c(0, 100))
```



The above code uses the Hsapiens object to get the maximum base of chromosomes. The normalisation of the coverage used is to scale every coverage value by 10 $million/number\_of\_reads\_in\_sample$. 300 is passed in as the seqLen parameter, because that is approximately the real length of the fragments sequenced. As expected, many more bases in the IP samples have high read coverages.

Next, the CpG density of reads is examined.

```
> cpgDensityPlot(samplesList, cols = c("black", "blue", "black",
+     "blue"), xlim = c(0, 30), wFunction = "none", organism = Hsapiens,
+     seqLen = 300)
```



This time the `Hsapiens` annotation is required so that the 300 base DNA sequence (tags are only 36 bp long) may be fetched. The `wFunction` parameter allows the count of CpGs to be weighted. In this example, raw counts are used.

Notice that at lower CpG densities, the two input samples have a higher frequency of reads than the two IP samples. At higher CpG densities, this trend is reversed. This suggests that the DNA methylation IP has worked.

4 Analyses and Visualisations

The `doSeqStats` function is a convenient way to do a statistical test of differential enrichment between two groups or treatments, either for counts in windows genome wide, or for counts in windows surrounding some genomic landmarks, like TSSs. The function leverages the package `edgeR`'s modelling of counts as negative binomial distributed and its adaptation of Fisher's exact test to overdispersed data. The `doSeqStats` function is distinct in that it uses copy number segmentations from the input sequencing in the statistical testing procedure by using pseudo library sizes, therefore controlling for the effect of copy number changes in the difference in counts between two conditions.

```
> stats <- doSeqStats(samplesList, seqLen = 300, whichInputs = c(3,
+     1), whichControl = 4, whichTreat = 2, blockSize = 20000,
+     minCount = c(20, 10), CNlevels = 5, blocksTable = geneAnno,
+     bpUp = 1000, bpDown = 1000)
```

```
Analyzing: Cancer.Input...Normal.Input.Fold.Change
Comparison of groups:  T - C
Comparison of groups:  T - C
Comparison of groups:  T - C
Comparison of groups:  T - C
Comparison of groups:  T - C
```

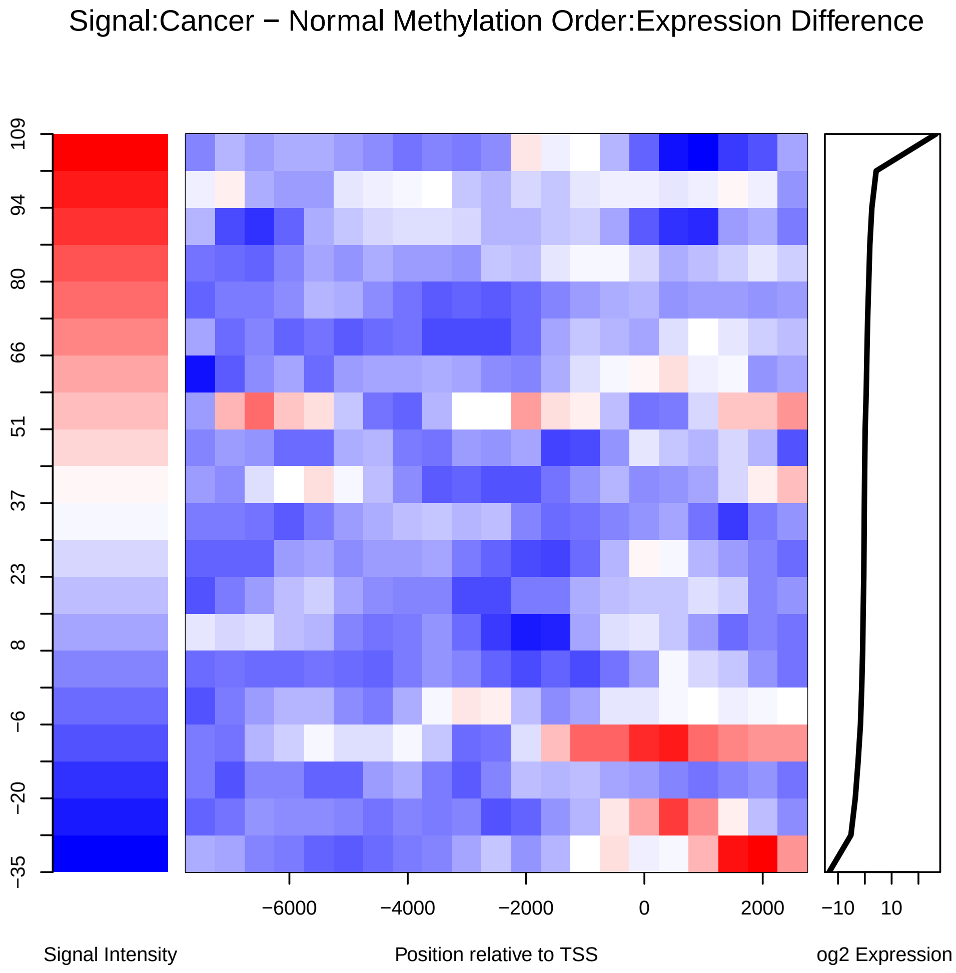
```
> head(stats)
```

```
       name   chr strand   symbol    start      end featureStart featureEnd
1 NM_199260 chr21      -     TPTE 10011791 10013791      9928613   10012791
2 NM_182482 chr21      -    BAGE2 10119796 10121796     10042712   10120796
3 NM_001187 chr21      -     BAGE 10119808 10121808     10079666   10120808
4 NR_026916 chr21      + C21orf99 13331357 13333357     13332357   13412440
5 NM_174981 chr21      +    POTED 13903368 13905368     13904368   13935777
7 NR_027270 chr21      - C21orf81 14273636 14275636     14237966   14274636
    logConc       logFC      p.value TreatmentCN
1 -6.795713  0.08113602 9.062943e-01    0.853581
2 -5.220464 -1.31197489 1.828683e-10    0.853581
3 -5.224117 -1.30467009 2.584061e-10    0.853581
4 -8.269183 -2.56960673 5.417769e-07    1.341270
5 -8.309269 -2.48943638 1.636124e-06    1.341270
7 -9.144194 -0.81958498 3.592834e-01    1.341270
  Normal MBD2IP Per 10 Million Reads Cancer MBD2IP Per 10 Million Reads
1                               2377                               3220
2                              11479                               5922
3                              11421                               5922
4                               2145                                727
5                               2029                                727
7                                638                                727
     adj.p.val     zScore zeroReads totalReads
1 9.793826e-01  0.1177139         0       5597
2 1.225218e-08 -6.3750793         0      17401
3 1.298491e-08 -6.3218764         0      17343
4 7.259810e-06 -5.0108942         0       2872
5 1.730847e-05 -4.7938478         0       2756
7 4.845368e-01 -0.9167313         0       1365
```

The `whichInputs` parameter specifies which two samples are the inputs, and in order of control input then treatment input. `whichControl` and `whichTreat` are vectors that give the indices in `samplesList` of the control IP samples and treatment IP samples, respectively. In this example, there is only one of each. `blocksSize` is a vector of length two, usually. It specifies the width of tiled windows genome wide that counts are made in the inputs, and in the IPs. Since windows surrounding gene annotations are being used as the regions of interest, the above example only needs a single number - the width of read windows to be used for counting in the input samples. The `CNlevels` parameter gives the number of distinct levels to break the copy number segmentations into. The `minCount` parameter is a vector of length 2. The first number sets a limit on each window's sum of counts of the input samples to be above, or else they are discarded from the segmentation. The second number is a such a cutoff for the IP samples. If the sum of counts for a gene is below the limit, no statistics are reported for that gene.

Epigenomic data is often gathered with other data, such as gene expression. It may be of interest to see the profile of epigenetic mark enrichment at a variety of distances from TSSs, and stratify this into groups by the expression of genes. The `binPlots` function is a convenient way to look at these interactions.

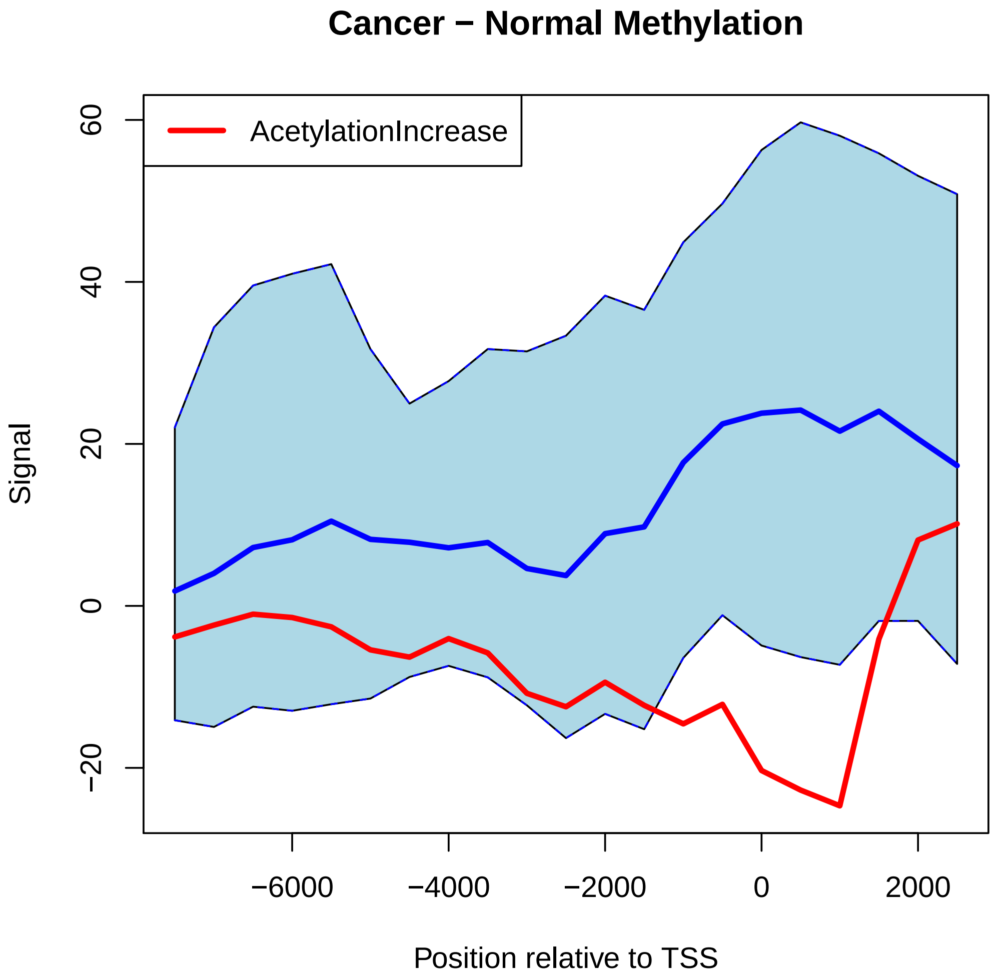
```
> differenceMatrix <- matrix(c(0, 1, 0, -1), dimnames = list(names(samplesList),
+     "Cancer - Normal Methylation"))
> differenceMatrix

              Cancer - Normal Methylation
Cancer Input                            0
Cancer MBD2IP                           1
Normal Input                            0
Normal MBD2IP                          -1

> binPlots(samplesList, geneAnno, design = differenceMatrix, by = 500,
+     bw = 500, seqLen = 300, ordering = expr, plotType = "heatmap",
+     nbins = 20)

gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```



This example made counts in 500 base non - overlapping windows between -7500 bases upstream and 2500 bases downstream for each gene, then split them into categories based on the expression difference value, and averaged over all counts for each particular window and expression category. Apart from the heatmap visualisation, there are a number of other styles. Details can be found in the documentation of the function. Trends are much less noisy when the full dataset with real expression difference values is used.

Some genes may be of interest to the researcher for some reason. This subset of genes may be known to be strongly marked with another epigenetic mark, or change in expression in the same direction strongly, or many other reasons. No matter what the reason for selecting the subset is, the profile of intensities or counts can be plotted versus the profile of randomly selected gene lists and compared with the `significancePlots` function. In the following example, it will be assumed that the first 25 genes of the annotation have been previously found to have a significant gain of histone acetylation.

```
> significancePlots(samplesList, geneAnno, geneList = list(AcetylationIncrease = 1:25),
+     design = differenceMatrix, by = 500, bw = 500, seqLen = 300)
```



The blue region forms the null distribution that was created by sampling random gene lists of the same size as the user - specified gene list a number of times, as set by the `nSamples` parameter. By default, the null region is a between the 0.025 and 0.975 quantiles of the null distribution. In this example, it appears that the high acetylation gene set has a significant loss of methylation around gene TSSs.

5 Summary

Repitools has a number of useful functions for quality checking, analysis, and comparison of trends. Many of the functions work seamlessly on array data, as well as sequencing data. Consult the package documentation for instructions on how to use functions that were not demonstrated by this vignette.

6 Environment

This vignette was created in:

```
> sessionInfo()
```

```
R version 2.12.0 (2010-10-15)
Platform: x86_64-unknown-linux-gnu (64-bit)

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

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] gplots_2.8.0                        caTools_1.10
 [3] bitops_1.0-4.1                      gdata_2.8.0
 [5] gtools_2.6.2                        DNAcopy_1.24.0
 [7] edgeR_1.8.1                         BSgenome.Hsapiens.UCSC.hg18_1.3.16
 [9] Repitools_1.46                      BSgenome_1.18.0
[11] Biostrings_2.18.0                   GenomicRanges_1.2.0
[13] IRanges_1.8.0                       R.methodsS3_1.2.1

loaded via a namespace (and not attached):
[1] Biobase_2.10.0 limma_3.6.5    tools_2.12.0
```