A Modification Motif Analysis Demo

Pat Marks

April 30, 2012

1 Introduction

This document demonstrates a typical analysis of kinetic modification detection data from bacterial genome. The raw data used here is generated by sequencing native E.Coli K12 on the PacBio RS, and performing secondary analysis using the RS.Modification.Detection workflow. Some words about RM dogma – decoding the RM system is the goal of this analysis.

- Load and explore the modification calls from the RS.Modification.Detection workflow
- Find the sequence motifs that are common to most of the modified genome hits
- Find instances of those motifs in the genome sequence
- Determine the what fraction of the motifs instances were detected as methylated

1.1 R Setup

The analysis workflow demonstrated here on R 2.14 and R 2.15 on Linux and Windows. From `CRAN` we require ggplot2 and plyr. From `Bioconductor` we require Biostrings and cosmo. See the appendix for details of download and installation detatils.

Please prepare your R environment as follows. `scripts.R` is included with this tutorial.

```
> library(ggplot2)
> library(plyr)
> library(Biostrings)
> library(cosmo)
```

```
Welcome to cosmo version 1.18.0

cosmo is free for research purposes only. For more details, type
license.cosmo(). Type citation('cosmo') for details on how to cite
cosmo in publications.
```

```
> source('scripts.R')
```

2 Loading modification data

We start by loading the raw modifications calls from that are produced by SMRTPortal in `modifications.gff.gz`. This can be downloaded from the job results page in SMRTPortal, or accessed directly from the SMRTportal job folder on the file server. `modifications.gff.gz` is compliant with `GFFv3` specification ([www.sequenceontology.org]). See Table ?? for a description of the columns in the `GFF` file.

The included `R` script contains a `GFF` file reader that extracts some extra atrribute columns used by the modification detection tool. Take a look at the data contained in the `GFF` file. The context field contains a 41 base context centered around the detected modification – pull out the center base (position 21). Summarize the `coverage` column of the `GFF` to see how many reads contribute to each modification call. Confident m6A detection generally requires coverage $>$ 20 per strand.

Column	Description
seqid	Reference tag (e.g. ref00001)
source	Name of tool – 'kinModCall'
type	Modification type – currently we use a generic tag "modified_base"
start	Location of modification
end	Location of modification plus one
score	Phred transformed p-value of detection
strand	Sample strand containing modification
phase	Not applicable
attributes	Fields below are packed in the GFF attributes column
IPDRatio	Ratio between mean IPD of observed data to IPD of unmodified DNA
context	Reference sequence -20bp to +20bp around start, converted to current strand
coverage	Number of valid IPD observations at this site

Table 1: Contents of modifications.gff.gz file

```
> # Load GFF file
> gff <- '/mnt/secondary/Smrtanalysis/userdata/jobs/040/040041/data/modifications.gff.gz'
> hits <- readModificationsGff(gff)

Reading /mnt/secondary/Smrtanalysis/userdata/jobs/040/040041/data/modifications.gff.gz

> hits$CognateBase <- substr(hits$context, 21, 21)
> head(hits)

    seqname     source       feature start end score strand frame coverage
1 ref000001 kinModCall modified_base   271 271    26      -     .       34
2 ref000001 kinModCall modified_base   423 423    21      -     .       36
3 ref000001 kinModCall modified_base   621 621    81      -     .       37
4 ref000001 kinModCall modified_base   653 653    21      -     .       35
5 ref000001 kinModCall modified_base   728 728    65      -     .       40
6 ref000001 kinModCall modified_base   738 738    30      -     .       39
                                    context IPDRatio CognateBase
1 TCAGGTGCGGGCTTTTTTCTGTGTTTCCTGTACGCGTCAGC     3.43           G
2 ACGGTGGCCACCTGCCCCTGCCTGGCATTGCTTTCCAGAAT     2.04           C
3 TTTATTTGGGCAAATTCCTGATCGACGAAAGTTTTCAATTG     5.18           A
4 GCCCCAACAAACTAATGCCATGCAGGACATGTTTTATTTGG     1.83           T
5 ATACGCCGGCCATAATGGCGATCGACATTTTCTCGCCACGG     2.83           A
6 CGCGCTTCTAATACGCCGGCCATAATGGCGATCGACATTTT     1.95           C

> summary(hits$coverage)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   41.00   47.00   47.86   54.00  223.00
```

We now make some plots from the GFF data to assess the quality and type of the modification calls. Figure 1 shows a histogram of the scores of the GFF entries, coloured by the cognate base. This will give you a sense of how strong your signals are and whether the strongest signals are enriched on any base. For our E.Coli test genome the predominant modification is 6-methyl adenosine, so most of the significant modification detections are at A positions.

The histogram in Figure 1 indicates that the interesting A bases have a score cutoff of roughly 45. We select these hits, then sort in decreasing order of score, so we consider the strongest signal first.

```
> goodHits <- subset(hits, score > 45)
> goodHits <- goodHits[order(goodHits$score, decreasing=T),]
> workHits <- goodHits
> p <- qplot(score, colour=CognateBase, geom='freqpoly', data=hits, binwidth=5) +
+   geom_abline(slope=0.9) + geom_vline(x=25)
> show(p)
```

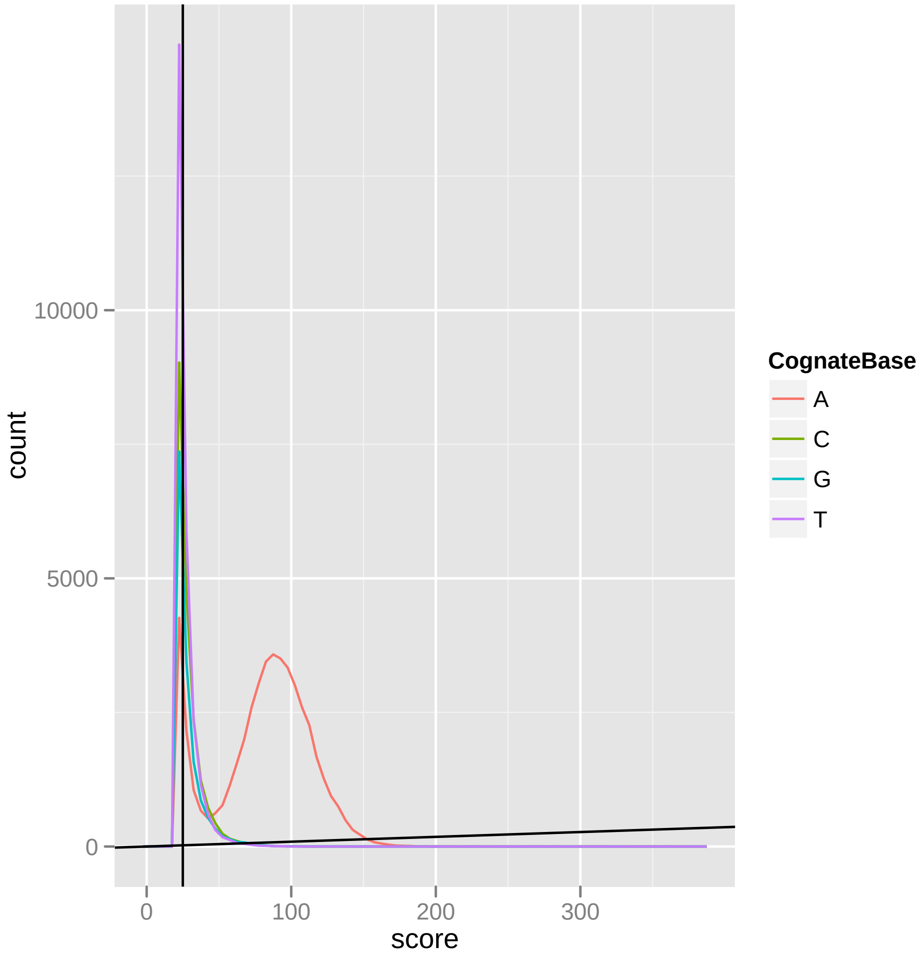

Figure 1: Modification Scores by cognate base

```
> p <- qplot(coverage, score, colour=CognateBase, alpha=I(0.3), data=hits[1:10000,])
> show(p)
```

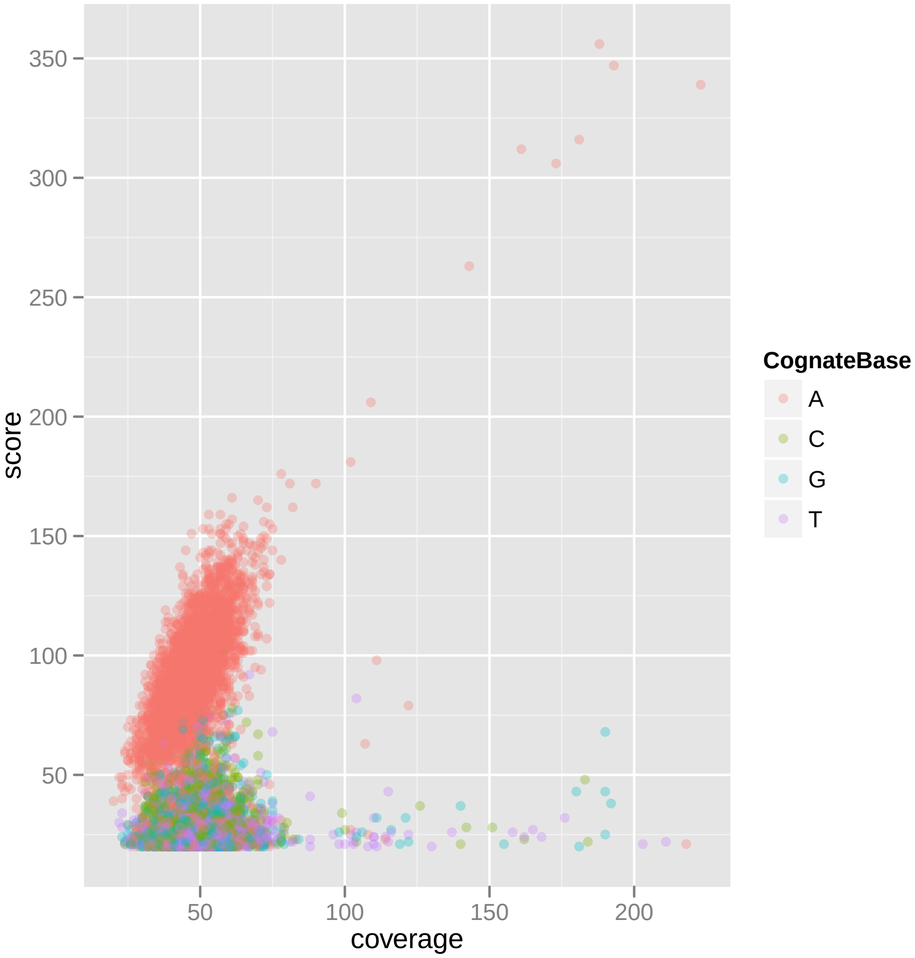

Figure 2: Score vs. Coverage

3 MEMEChip Motif Finding

We use the online motif finding server 'MEMEChip' (http://meme.sdsc.edu/meme/cgi-bin/meme-chip.cgi). The MEMEChip server requires the source sequences to uploaded in FASTA format. We write the context string of the top 1000 hits to 'contexts.fasta'. Fill in blurb about how to use the web server.

```
> writeContextFasta(workHits$context[1:1000], 'contexts.fasta')
```

As we get motif results from MEMEChip, we can check which hits match each motif using the labelContexts function. The labelContexts takes a charactervector of context strings, and a vector of motif strings and their associated methlyated positions. We make a rough approximation of the number of GATC sites we would expect on both stands of a random genome of length $|G|$ as $2|G|/4^n$ where n is the number of bases in the motif. We see that most GATC are methylated. We will perform a more accurate analysis in the next section. We can now remove the 'GATC' hits from working list. We still have many of unassigned GFF hits, so we can run MEMEChip on the reduced list of hits.

```
> motif <- 'GATC'
> position <- 2
> motifLabels <- labelContexts(workHits$context, motif, position)
> table(motifLabels)

motifLabels
 GATC  None
37728  4192

> genomeSize <- max(hits$end)
> expectedHits <- 2 * genomeSize / (4^4)
> expectedHits

[1] 36245.39

> workHits <- workHits[motifLabels != motif,]
> nrow(workHits)

[1] 4192
```

In fact our E.Coli sample contains a pair of methyltransferases that methylate reverse-complementary motifs that share a common 4 base sub-motif. These two motifs occur in equal numbers. cosmo was not designed to handle situation like this, so we must continue on by hand, inspecting the hits manually. Hand inspection of the hits shows that 'GATC', 'GCACNNNNNNGTT', 'AACNNNNNNGTGC' are the fully methylated motifs in our sample.

4 Genome Annotation

We can load the genome sequence and annotate it instances of each motif, to determine the genome-wide methylation fraction of our motifs. The genomeAnnotation function returns a data.frame containing one row for each match in the supplied genome of each motif supplied. Genome positions can match multiple motifs, which gives multiples row at the same genome position

```
> # Load the genome sequence
> seq_path <- '/mnt/secondary/Smrtanalysis/userdata/references/ecoli/sequence/ecoli.fasta'
> dna_seq <- read.DNAStringSet(seq_path)
> motifs = c('GATC', 'GCACNNNNNNGTT', 'AACNNNNNNGTGC')
> positions = c(2, 3, 2)
> genomeAnnotations <- genomeAnnotation(dna_seq, motifs, positions)
> head(genomeAnnotations)
```

```
  strand start motif onTarget seqid
1      +   620  GATC       On     1
2      +   727  GATC       On     1
3      +   782  GATC       On     1
4      +   881  GATC       On     1
5      +  1168  GATC       On     1
6      +  1570  GATC       On     1
```

```
> table(genomeAnnotations$motif)

AACNNNNNNGTGC          GATC GCACNNNNNNGTT
          595         38240           595
```

We merge the genomeAnnotation output with our GFF data.frame to count the number of motif instances that exist in the genome and the number that were detected as methylated. We merge the genomeAnnotation and goodHits tables by genome position and strand, with all=T to include GFF hits that are not annotated with a motif, and genome motif instances that do not have a GFF hit. We adjust the merged data.frame to indicate these cases.

```
> goodHits$seqid <- as.integer(substr(goodHits$seqname, 4,11))
> mm <- merge(goodHits, genomeAnnotations, all=T)
> mm$motif[is.na(mm$motif)] <- 'NoMotif'
> mm$feature[is.na(mm$feature)] <- 'not_detected'
> table(mm$feature, mm$motif)

                AACNNNNNNGTGC  GATC GCACNNNNNNGTT NoMotif
  modified_base           576 37728           584    3032
  not_detected             19   512            11       0
```

In our goodHits table we have 3032 'modified_base' calls that do not occur at an annotated genome position. For genome positions matching the GATC motif, 37728 were present detected ('modified_base') and 512 were not ('not_detected'). We can assess whether we are likely to have missed any methylated motifs by comparing the score distributions for the positions matching a motif to the 'NoMotif' set, as in Figure 3.

5 Unmethylated Sites

Our comparison of the genome annotation and the detected methylations reveals that there are a small number of GATC sites in the genome that were not detected as methylated at our score cutoff of 45. We may be interested in looking in more detail at the kinetic evidence of the undetected GATC sites to determine whether these sites are false negatives caused by our analysis, or are truly unmethylated in the organism. The GFF file only contains genome posititions with a score > 20. The modifications.csv.gz file contains the complete statistics for all sites in the genome.

Here we load the CSV file, and merge it with the genomeAnnotations table, keeping only rows containing an identified motif. We can recreate our detection table by cutting on *score* > 45. Figure 4 shows the low end of the score distributions of the matching motif sites. It appears that there is a small population of GATC sites with very low score. We look at the table of hits with score < 10. These sites appear to have good coverage, and small ipd ratio. Interestingly, 12 of 16 of these sites are paired, showing no methylation on both strands, which gives us more confidence in their unmethylated status.

```
> # Load CSV file
> if(F)
+   {
+ csv <- '/mnt/secondary/Smrtanalysis/userdata/jobs/040/040041/data/modifications.csv.gz'
+ rawKin <- read.csv(csv)
+
+ rawKin$seqid <- as.integer(substr(rawKin$refId,4,11))
+ rawKin$strand[rawKin$strand == 1] <- '-'
```

```
> p <- qplot(score, ..density..,
+            colour=motif, geom='freqpoly',
+            data=subset(mm, feature=='modified_base'),
+            binwidth=3, xlim=c(0,200))
> show(p)
```

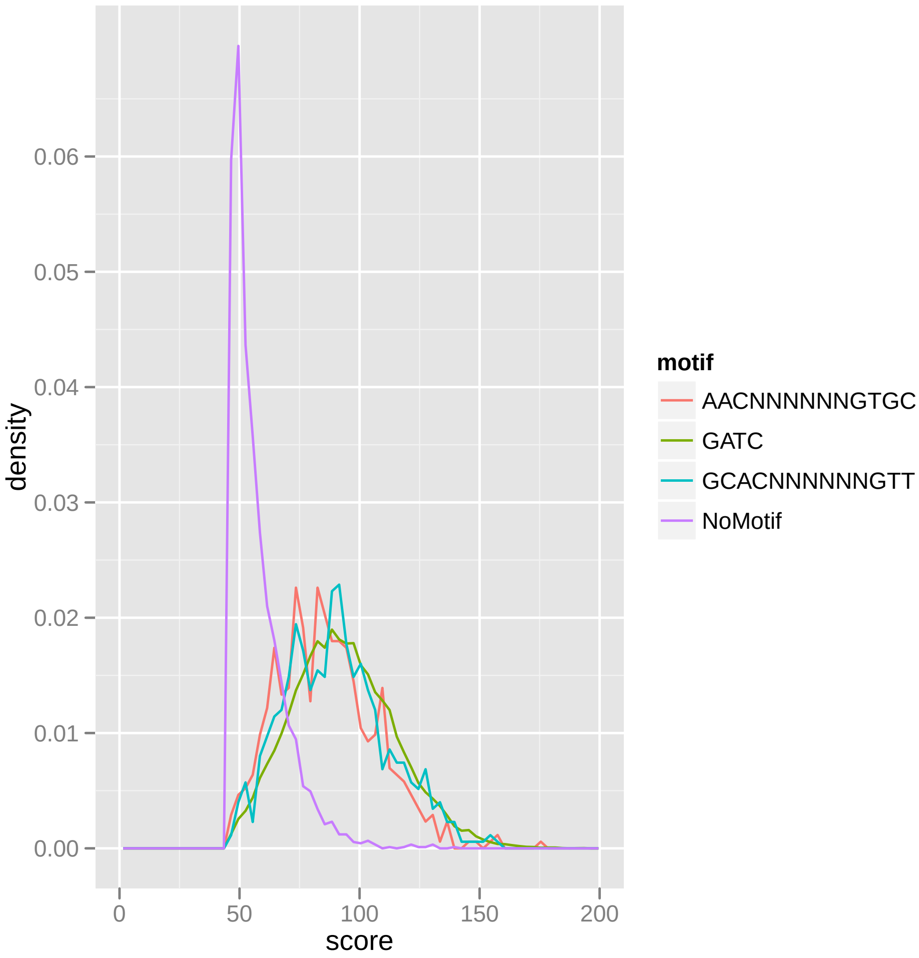

Figure 3: Score distribution by motif annotation

```
+ rawKin$strand[rawKin$strand == 0] <- '+'
+ mga <- merge(genomeAnnotations, rawKin, by.x=c('start', 'strand'), by.y=c('tpl', 'strand'))
+ table(mga$motif, mga$score > 45)
+
+ colsToShow <- c('start', 'strand', 'motif', 'score', 'coverage', 'ipdRatio')
+ subset(mga, score < 10 & motif=='GATC')[,colsToShow]
+   }
```

A R Package Installation

R can be downloaded for Linux, Mac and Windows from http://r-project.org We also recommend the graphical front-end RStudio, available from rstudio.org. The following R packages need to be installed to run through this demo. ggplot2, plyr are available through the built-in R pacakge manager. Instructions for installing Bioconductor packages is available here: www.bioconductor.org/install. cosmo and Biostrings are required from Bioconductor

A.1 Session Info

Here we give information about the version of R and installed packages used to generate this document.

```
> sessionInfo()

R version 2.14.2 Patched (2012-02-29 r59005)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
[1] C

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

other attached packages:
[1] cosmo_1.18.0      seqLogo_1.18.0    Biostrings_2.22.0 IRanges_1.12.6   
[5] plyr_1.7.1        ggplot2_0.9.0    

loaded via a namespace (and not attached):
 [1] MASS_7.3-17        RColorBrewer_1.0-5 colorspace_1.1-1   dichromat_1.2-4   
 [5] digest_0.5.2       memoise_0.1        munsell_0.3        proto_0.3-9.2     
 [9] reshape2_1.2.1     scales_0.2.0       stringr_0.6        tools_2.14.2      
```

For Research Use Only. Not for use in diagnostic procedures. Copyright 2011, Pacific Biosciences of California, Inc. All rights reserved. Information in this document is subject to change without notice. Pacific Biosciences assumes no responsibility for any errors or omissions in this document. Certain notices, terms, conditions and/or use restrictions may pertain to your use of Pacific Biosciences products and/or third party products. Please refer to the applicable Pacific Biosciences Terms and Conditions of Sale and to the applicable license terms at http://www.pacificbiosciences.com/licenses.html.

Pacific Biosciences, the Pacific Biosciences logo, PacBio, SMRT and SMRTbell are trademarks of Pacific Biosciences in the United States and/or certain other countries. All other trademarks are the sole property of their respective owners.

```
> #p <- qplot(score, ..density.., colour=motif,
> #           log='y', data=mga,
> #           geom='freqpoly', xlim=c(0,100), binwidth=1)
> #show(p)
> p<-qplot(1,1)
> show(p)
```

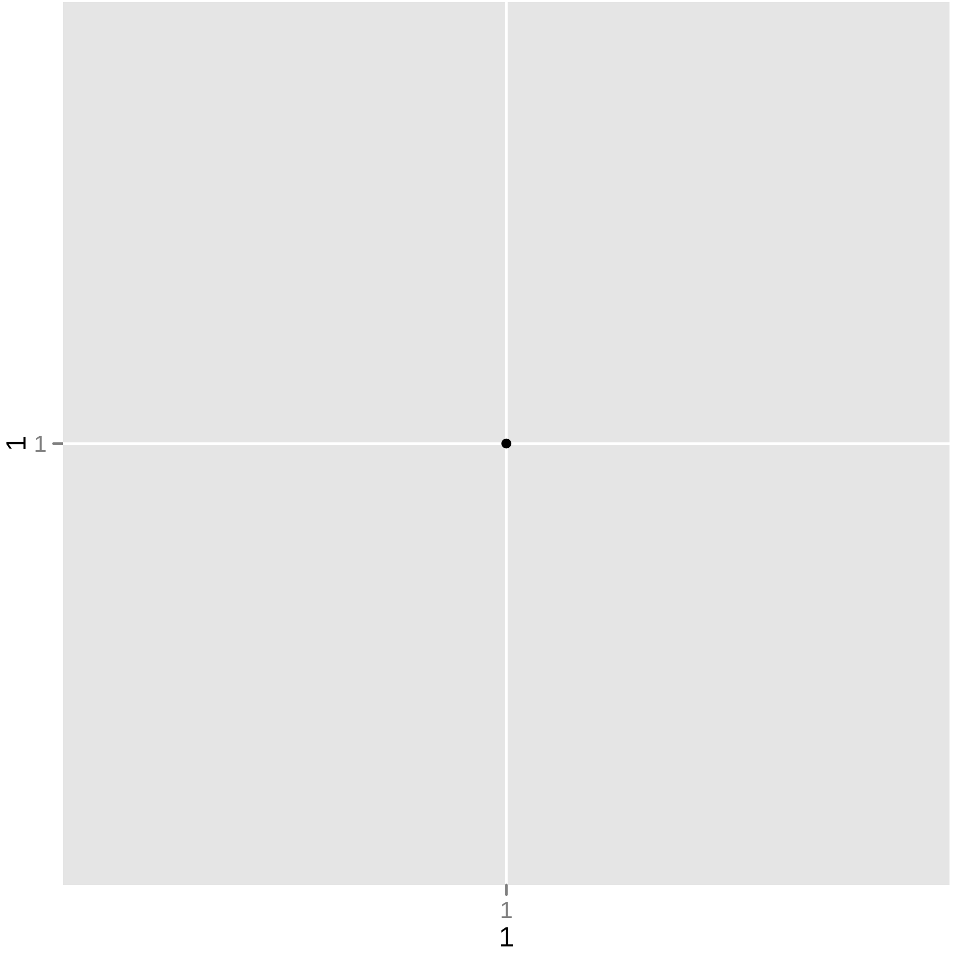

Figure 4: Score distribution of motif sites