A Modification Motif Analysis Demo

Pat Marks

April 29, 2012

1 Getting started

We demonstrate a typical analysis of kinetic modification detection data on a bacterial genome. The main steps in the analysis are: load and quality check the modification calls, find commonly methylated sequence motifs, and find the motifs in the genome sequence.

We install some packages from bioconductor:

We start by loading some utilities:

```
> 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.g`
This can be downloaded from the job results page in SMRTPortal, or accessed directly from the SMRTportal job folder on the file server. 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 – we pull out the center base (position 21). Look at the summary of the detected hits. Confident m6A detection requires coverage >20x (per strand). See

```
> gffFile <- "/mnt/secondary/Smrtanalysis/userdata/jobs/040/040041/data/modifications.gff.
> hits <- readModificationsGff(gffFile)

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

```
> 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
1 TCAGGTGCGGGCTTTTTTCTGTGTTTCCTGTACGCGTCAGC     3.43
2 ACGGTGGCCACCTGCCCCTGCCTGGCATTGCTTTCCAGAAT     2.04
3 TTTATTTGGGCAAATTCCTGATCGACGAAAGTTTTCAATTG     5.18
4 GCCCCAACAAACTAATGCCATGCAGGACATGTTTTATTTGG     1.83
5 ATACGCCGGCCATAATGGCGATCGACATTTTCTCGCCACGG     2.83
6 CGCGCTTCTAATACGCCGGCCATAATGGCGATCGACATTTT     1.95

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

   seqname             source            feature              start
 Length:107758      Length:107758      Length:107758      Min.   :    132
 Class :character   Class :character   Class :character   1st Qu.:1158852
 Mode  :character   Mode  :character   Mode  :character   Median :2348764
                                                          Mean   :2332150
                                                          3rd Qu.:3497812
                                                          Max.   :4639410
      end              score           strand             frame
 Min.   :    132   Min.   : 20.00   Length:107758      Length:107758
 1st Qu.:1158852   1st Qu.: 23.00   Class :character   Class :character
 Median :2348764   Median : 31.00   Mode  :character   Mode  :character
 Mean   :2332150   Mean   : 50.89
 3rd Qu.:3497812   3rd Qu.: 81.00
 Max.   :4639410   Max.   :363.00
    coverage        context             IPDRatio      CognateBase
 Min.   :  5.00   Length:107758      Min.   : 1.350   Length:107758
 1st Qu.: 41.00   Class :character   1st Qu.: 1.740   Class :character
 Median : 47.00   Mode  :character   Median : 2.090   Mode  :character
 Mean   : 47.86                      Mean   : 2.848
 3rd Qu.: 54.00                      3rd Qu.: 3.920
 Max.   :223.00                      Max.   :10.910

> hitsSubset <- hits[sample(nrow(hits), min(nrow(hits), 50000)),
+     ]
```

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

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 IPD of observed IPD 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

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
```

3 Motif Finding

We use the motif finding package `cosmo` (http://www.bioconductor.org/packages/release/bioc/html/cosm to search for conserved motifs in the DNA sequences surrounding our modification detections. First, we convert the context snippets into a cosmo compatible format. Next, we select a small number of hits to pass to `cosmo` (it will not return if you provide too many at once). We feed the contexts to cosmo, then look at the resulting logo plot. We found 'GATC', starting at position 21-2 = 19

```
> sequenceList <- llply(workHits$context, function(x) list(seq = as.character(x),
+     desc = ""))
> smallHitList <- sequenceList[1:200]
> cc2 <- cosmo(seqs = smallHitList, minW = 4, maxW = 10, revComp = F,
+     silent = T)
> motif <- "GATC"
> position <- 20
```

View the logo plot (Figure 3)

We found 'GATC', starting at position 21-1 = 20. Pull out the -2 to +1 context window from the hits contexts. How many hits do we get with this motif. Roughly how many would we expect

```
> p <- qplot(score, colour = CognateBase, geom = "freqpoly", data = hits)
> show(p)
```

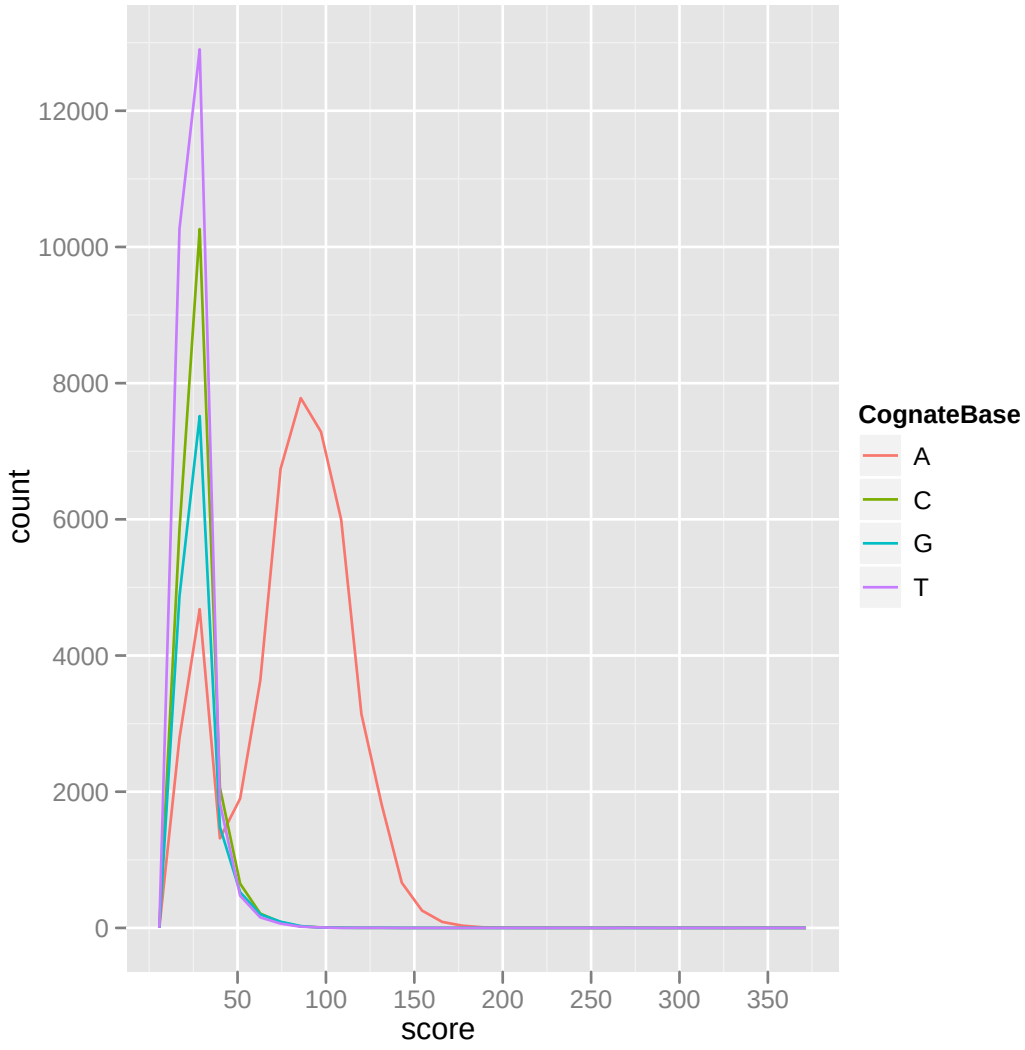

Figure 1: Modification Scores by cognate base

```
> p <- qplot(coverage, score, colour = CognateBase, alpha = I(0.3),
+     data = hitsSubset)
> show(p)
```

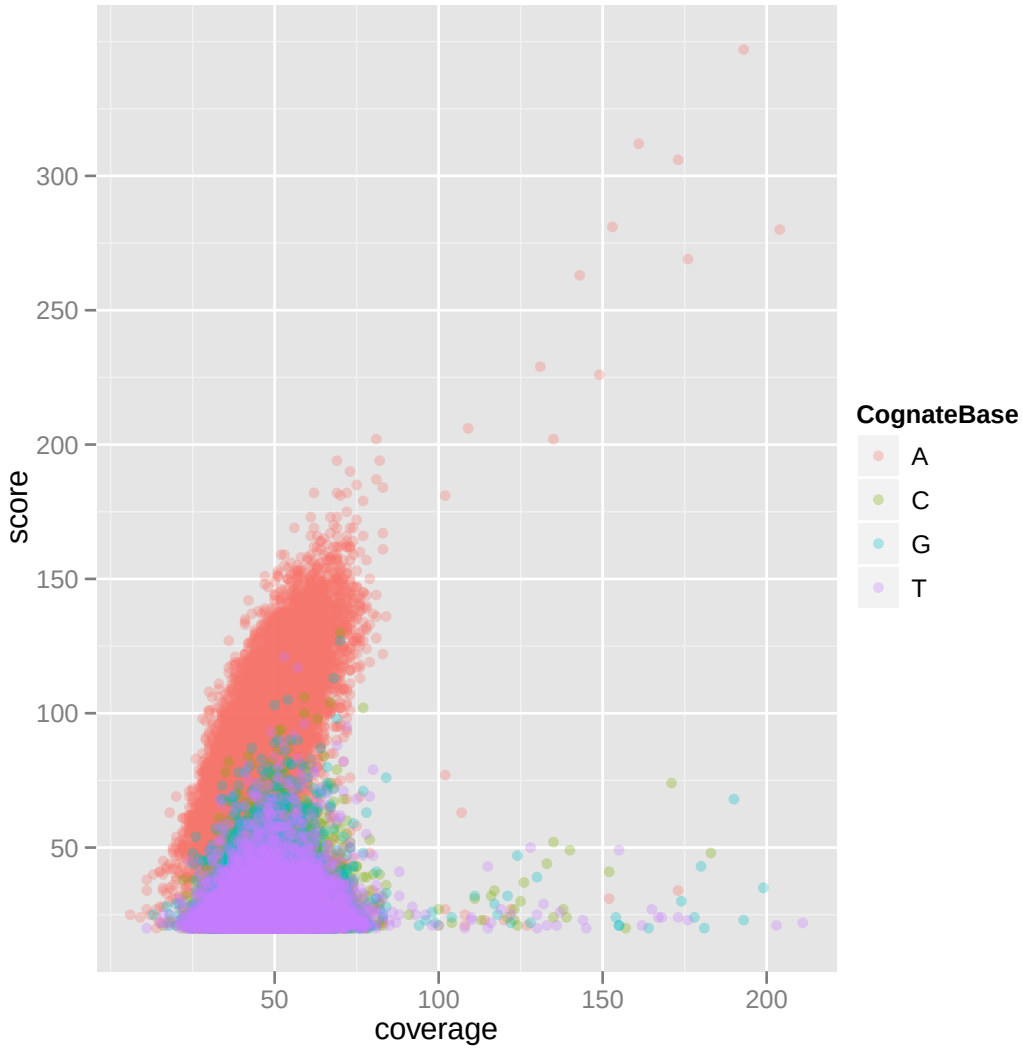

Figure 2: Score vs. Coverage

```
> plot(cc2)
```

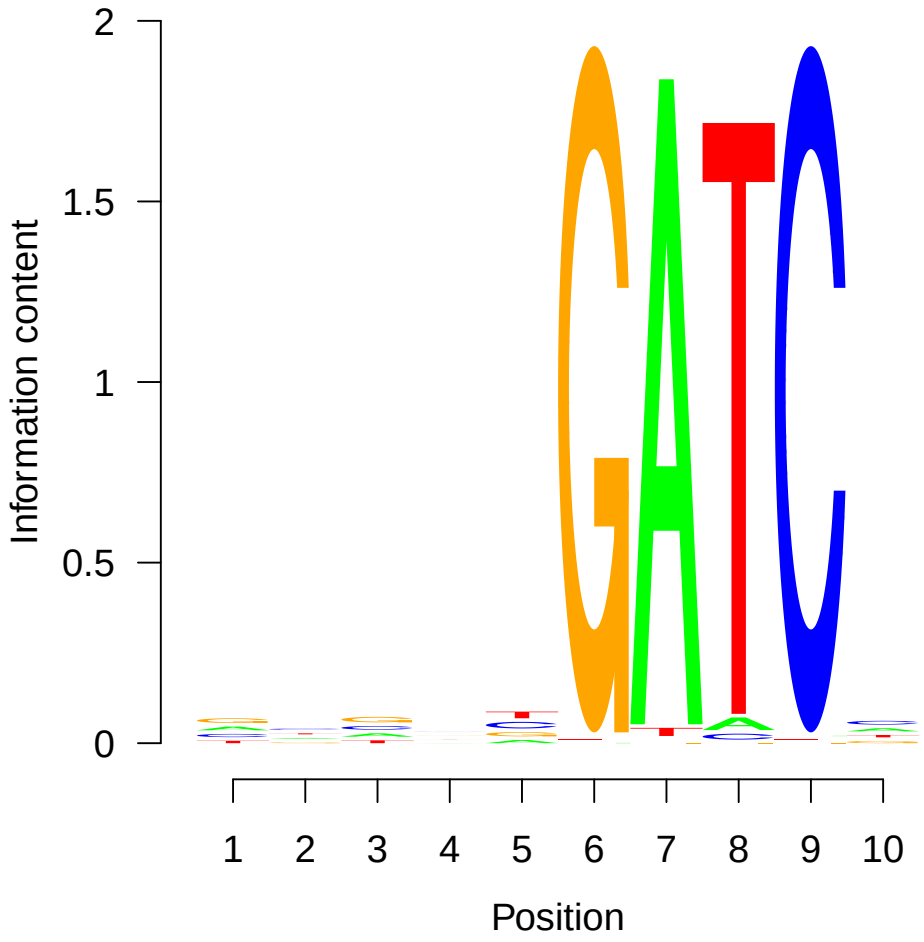

Figure 3: Logo plot

if everything was methylated in a random genome. We see the correct number of hits! We are now done with 'CTATG'. Remove the 'CTATG' hits from working list and try again. We still haev lots of unidentified motifs

```
> motif <- "GATC"
> position <- 20
> snip <- substr(workHits$context, position, position + nchar(motif) -
+     1)
> observedHits <- sum(motif == snip)
> observedHits
```

```
[1] 37728
```

```
> genomeSize <- max(hits$end)
> expectedHits <- genomeSize/(4^nchar(motif))
> expectedHits
```

```
[1] 18122.7
```

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

```
[1] 4192
```

Not clear what's happening now – we probably have a mixture of motifs in similar proportion which will confuse cosmo. Continue on by hand, or wait for the fancy new motif finder! Now imagine we have identified our motifs and we want to label

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

```
> 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 4.

5 Appendix

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 (available on CRAN), cosmo, Biostrings (available on Bioconductor)

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

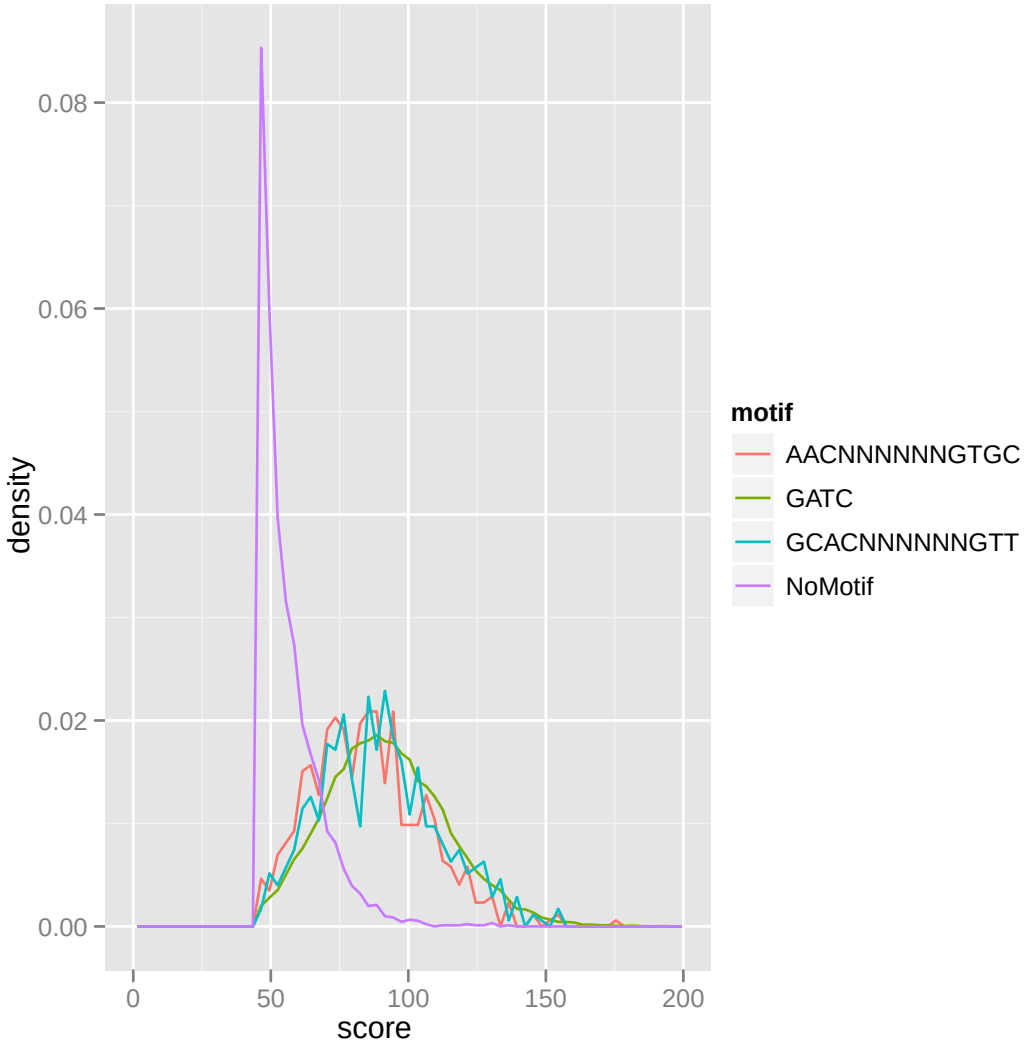

Figure 4: Score distribution by motif annotation