1 Introduction

The emergence of ChIP-seq technology for genome-wide profiling of transcription factor binding sites (TFBS) has made it possible to categorize very precisely the TFBS motifs. How to harness the power of huge volume of data generated by this new technology presents many computational challenges. We propose a novel motif discovery algorithm that is scalable to large databases, and performs discriminative motif discovers by searching the most differential motifs between a foreground and background sequence dataset. This tool can be used in a traditional setting in which the foreground sequence dataset is derived from a ChIP-seq binding profile, and background sequence dataset is either sampled from the genome or generated from a null model. It can also be used for comparative study involving two TFBS binding profiles.

In a nutshell, the method works as the following: we enumerate all fixed-length n-mers exhaustively, and measure their discriminative power by a logistic regression model. The top ranking seed motif is then iteratively refined by allowing IUPAC degenerate letters and extended to a longer motif automatically. We introduce a bootstrapping robustness test to avoid over-fitting in the optimization process. The logistic regression framework offers direct measurement of statistical significance, and we demonstrate by permutation tests that the z-value statistics do reflect the probability of occurrence by chance. Compared to traditional motif finding tool, use of proper control sequences for comparison avoids the difficulty of modeling true genomic background, which usually presents complicated high order structure such as dinucleotide sequence preference, repeats, nucleosome positions signals, etc. When used to compare two similar ChIP-Seq samples, the discriminative motifs usually leads to insights on sample specificity.

2 Quick Start

The method requires foreground and background sequence datasets. The users can use fasta files as input.

```
> library(motifRG)
> MD.motifs <- findMotifFasta(system.file("extdata", "MD.peak.fa",package="motifRG"),
+                             system.file("extdata", "MD.control.fa", package="motifR
+                             max.motif=3,enriched=T)
```

The output motifs are:

```
> motifLatexTable(main="MyoD motifs", MD.motifs, prefix="myoD")
```

The foreground sequences correspond to subset of MyoD ChIP-Seq peaks in mouse fibroblast transfected with MyoD. MyoD binds to CANNTG ebox. The motif prediction results suggest that MyoD binds to CAGCTG and CAGGTG eboxes.

Alternatively, the users can fetch sequence given the sequence coordinates.

Table 1: MyoD motifs

Consensus	scores	ratio	fg.frac	bg.frac	logo
NNDCAGCTGN	21	4.6	0.78	0.31	
NNCAHCTGNYNN	13	3.2	0.46	0.18	
NNCAGCAGNN	11	3.0	0.41	0.17	

```
> data(YY1.peak)
> data(YY1.control)
> library(BSgenome.Hsapiens.UCSC.hg19)
> YY1.peak.seq <- getSequence(YY1.peak, genome=Hsapiens)
> YY1.control.seq <- getSequence(YY1.control, genome=Hsapiens)
> YY1.motif.1 <- findMotifFgBg(YY1.peak.seq, YY1.control.seq, enriched=T)
```

3 Fine tuning results

Let's examine the motif prediction results for the YY1 dataset.

```
> motifLatexTable(main="YY1 motifs", YY1.motif.1, prefix="YY1-1")
```

All motifs are GC rich motifs, and do not include known YY1 motif with consensus CCAT. We can check the GC content of the foreground and background sequences:

```
> summary(letterFrequency(YY1.peak.seq, "CG", as.prob=T))
```

```
      C|G
 Min.   :0.3905
 1st Qu.:0.5741
 Median :0.6285
 Mean   :0.6279
```

Table 2: YY1 motifs

Consensus	scores	ratio	fg.frac	bg.frac	logo
NNDNSCCGCCNN	16	4.5	0.75	0.36	
NNGCDGCCNN	16	2.7	0.91	0.60	
NNGGCTCCNN	15	2.8	0.79	0.48	
NNCGGCSCNN	15	4.1	0.67	0.31	
NNCGBGGCNN	15	3.6	0.65	0.29	

```
 3rd Qu.:0.6841
 Max.   :0.8304
```

```
> summary(letterFrequency(YY1.control.seq, "CG", as.prob=T))
```

```
      C|G
 Min.   :0.2197
 1st Qu.:0.4378
 Median :0.4949
 Mean   :0.5053
 3rd Qu.:0.5624
 Max.   :0.8351
```

It is clear that foreground sequences have significant GC bias.

We also examine width of the foreground sequences:

```
> summary(width(YY1.peak.seq))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  158.0   434.0   623.0   718.5   927.8  2771.0
```

Considering that YY1 has a very degenerate motif, it is likely to occur by chance in such wide peaks.

Assuming that YY1 peak summits occur within the center of the peaks, we can narrow the peaks to increase signal to noise ratio. We can also fit GC content as covariants for the regression model to balance this bias. In addition, in many ChIP-Seq datasets, the stronger peaks are more likely to be direct targets than the weaker peaks, and more likely to contain the transcription factor motif. But it is hard to make the cutoff without knowing the motif in priori. One can weight the foreround sequences based on peak intensity, and use the weights in motif prediction:

To narrow the peak:

```
> YY1.narrow.seq <- subseq(YY1.peak.seq,
+                          pmax(round((width(YY1.peak.seq) - 200)/2), 1),
+                          width=pmin(200, width(YY1.peak.seq)))
> YY1.control.narrow.seq <- subseq(YY1.control.seq,
+                                  pmax(round((width(YY1.control.seq) - 200)/2),1),
+                                  width=pmin(200, width(YY1.control.seq)))
> category=c(rep(1, length(YY1.narrow.seq)), rep(0, length(YY1.control.narrow.seq)))
```

To compute GC bias:

```
> all.seq <- append(YY1.narrow.seq, YY1.control.narrow.seq)
> gc <- as.integer(cut(letterFrequency(all.seq, "CG", as.prob=T),
+                      c(-1, 0.4, 0.45, 0.5, 0.55, 0.6, 2)))
```

To weight sequences:

```
> all.weights = c(YY1.peak$weight, rep(1, length(YY1.control.seq)))
```

Use all of above for motif prediction:

```
> YY1.motif.2 <- findMotif(all.seq,category, other.data=gc,
+                          max.motif=5,enriched=T, weights=all.weights)

> motifLatexTable(main="Refined YY1 motifs", YY1.motif.2,prefix="YY1-2")
```

Table 3: Refined YY1 motifs

Consensus	scores	ratio	fg.frac	bg.frac	logo
NNSGCCGCNN	8.7	6.8	0.62	0.127	
NNRATGGCNN	6.7	1.5	0.25	0.182	
NNCGGAAVNN	6.7	3.1	0.27	0.094	
NNCGGRGCNN	5.1	4.6	0.56	0.151	
NNCCKCGCNN	4.6	3.8	0.32	0.092	

The predicted ATGGC motif for YY1 matches the reverse complement of the known motif. The results also incude ETS motif CGGAA, and other GC rich motifs. It is difficult to completely balance the effects of GC content, because it is unclear what should be the proper transformation so it is very easy to over-correct or under-correct the bais, and GC bias usually reflects other biases, such as enrichment of promoters, CpG islands, etc. The best approach to adjust for such a bias is to select a control dataset with matching distribution of GC content, promoters etc, if one has the freedom to choose arbitrary control.

4 Refine PWM model

Motifs found by findMotif tend to be relatively short, as longer and more specific motif models do not necessarily provide better discrimination of foreground background vs background if they are already well separated. However, one can refine and extend a PWM model given the motif matches by findMotif as seed for more specific model. The method below exploits a MEME-like EM algorithm to refine the basic motif pattern to more informative PWM model.

```
> data(ctcf.motifs)
> ctcf.seq <- readDNAStringSet(system.file("extdata", "ctcf.fa",package="motifRG"))
> pwm.match <- refinePWMMotif(ctcf.motifs$motifs[[1]]@match$pattern, ctcf.seq)
> library(seqLogo)
```

```
> seqLogo(pwm.match$model$prob)
```

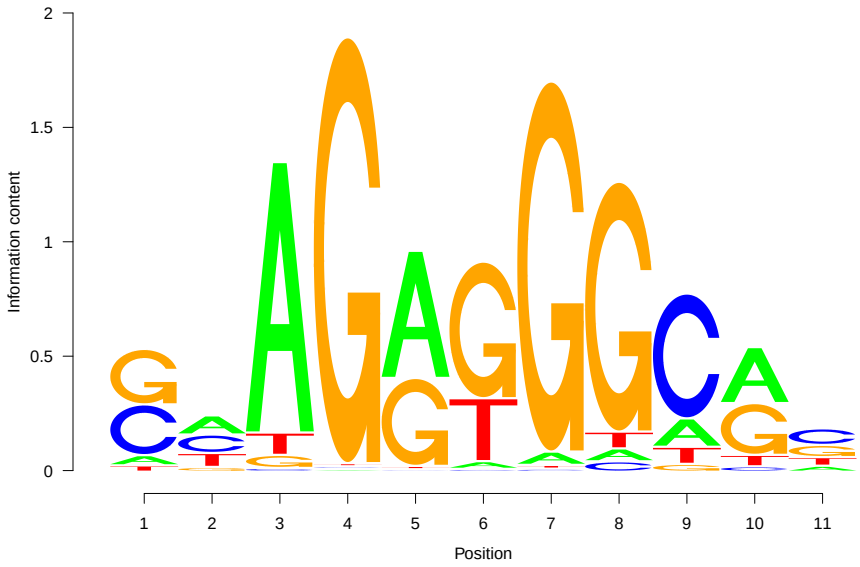

Figure 1: PWM logo of CTCF PWM matches

We use refienPWMMotifExtend function to automatically extend the PWM motif if the flanking region is also informative.

```
> pwm.match.extend <-
+     refinePWMMotifExtend(ctcf.motifs$motifs[[1]]@match$pattern, ctcf.seq)
```

```
> seqLogo(pwm.match.extend$model$prob)
```

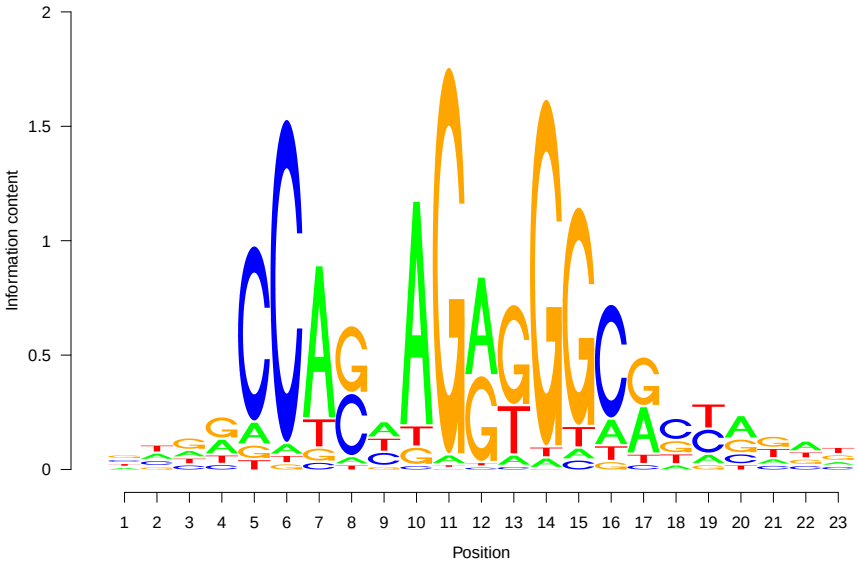

Figure 2: PWM logo of CTCF PWM matches

```
> plotMotif(pwm.match.extend$match$pattern)
```