chipenrich: A package for gene set enrichment testing of genome region data

R.P. Welch, C. Lee, R.A. Smith, P. Imbriano, S. Patil, T. Weymouth, L.J. Scott, M.A. Sartor

October 15, 2013

Contents

1 Introduction

This document describes how to use `chipenrich` to analyze results from ChIP-Seq experiments and other DNA sequencing experiments that result in a set of genomic regions. `chipenrich` includes a method that adjusts for potential confounders of gene set enrichment testing (locus length and mappability of the sequence reads.)

2 Synopsis

After starting R, the package should be loaded using the following:

```
> library(chipenrich)
```

This will load chipenrich, the chipenrich.data package, as well as all other dependency packages. The main function for conducting all gene set enrichment testing is `chipenrich()`.

The defaults for the `chipenrich()` function are

```
chipenrich(peaks, out_name = "chipenrich", out_path = getwd(), genome = "hg19",
genesets =   ('GOBP', 'GOCC', 'GOMF'), locusdef = "nearest_tss ",
method = "chipenrich", fisher_alt = "two.sided", use_mappability = F,
read_length = 24, genome_length = NULL, qc_plots = T)
```

The "peaks" option should be either a data frame or character vector representing the path to a file containing the peaks. The file (or data frame) should have at least 3 columns: "chrom," "start," and "end," denoting the chromosome, starting position, and ending position of the peak. Chromosome should be in UCSC format, e.g. chrX, chrY, chr22, etc. If a file, it must be tab-delimited, and the header must exist. The input file may also be a .bed file. Additional columns can exist, so long as they do not contain tab characters. An example dataset, "peaks_E2F4," is included in the package.

```
> data(peaks_E2F4)
```

```
> data(peaks_E2F4)
> head(peaks_E2F4)
```

```
  chrom     start       end
1  chr1 156186314 156186469
2  chr1  10490456  10490550
3  chr1  46713352  46713436
4  chr1 226496843 226496924
5  chr1 200589825 200589928
6  chr1  47779789  47779907
```

The first task of chipenrich() is to assign the peaks to genes. Currently supported genomes include human (hg19), mouse (mm9), and rat (rn4). Data from older versions of the genome may be converted using UCSC's liftover tool: http://genome.ucsc.edu/cgi-bin/hgLiftOver.

```
> supported_genomes()
```

```
[1] "hg19" "mm10" "mm9"  "rn4"
attr(,"na.action")
[1] 1
attr(,"class")
[1] "omit"
```

Peaks are assigned to genes according to a pre-defined locus definition, i.e. the region where peaks have to occur in order to be assigned to a gene. The following locus definitions are supported in `chipenrich`:

```
> supported_locusdefs()
```

```
[1] "1kb"          "5kb"          "exon"
[4] "nearest_gene" "nearest_tss"
```

Using the options "1kb" or "5kb" will only assign peaks to genes if the peaks are within 1 kilobases (kb) or 5kb of a gene's transcription start site (TSS) respectively. The option "exon" will assign peaks to genes if the peaks occur within a gene's exons. Using "nearest_gene" or "nearest_tss" will assign peaks to genes according to the nearest gene or the nearest TSS. Only the "nearest_gene" and "nearest_tss" locus definitions retain all peaks, others use only a subset of peaks that fall within the defined region. All gene loci are non-overlapping. The command "`help(chipenrich.data)`" may also provide more information on the locus definitions.

Seventeen gene set annotation databases are supported by `chipenrich`:

```
> supported_genesets()
```

```
 [1] "GOBP"                      "GOCC"
 [3] "GOMF"                      "biocarta_pathway"
 [5] "cytoband"                  "drug_bank"
 [7] "ehmn_pathway_gene"         "gene_expression"
 [9] "kegg_pathway"              "mesh"
[11] "metabolite"                "mirbase"
[13] "omim"                      "panther_pathway"
[15] "pfam"                      "protein_interaction_mimi"
[17] "transcription_factors"
```

Two methods for gene set enrichment testing are provided: the main ChIP-Enrich method ("chipenrich"), and Fisher's exact test ("fet"). While "chipenrich" is the main method offered, the "fet" method is also offered for comparison purposes and/or for use in limited situations when its assumptions are met (see examples.)

```
> supported_methods()

[1] "chipenrich" "fet"
```

The default gene set database is Gene Ontology (GO) terms, comprising of all three GO branches (GOBP, GOCC, and GOMF).

Accounting for mappability of reads is optional and can only be accomplished using the ChIP-Enrich method. Mappabilities for the following read lengths are available:

```
> supported_read_lengths()

[1]  24  36  40  50  75 100
```

Read lengths of 24mer are only available for the hg19 genome.

See the section on mappability for more information on how it is calculated.

3 Locus definitions

We define a gene *locus* as the region from which we predict a gene could be regulated. ChIP-seq peaks, or other types of genomic regions, falling within a locus for a gene are assigned to that gene.

We provide a number of different definitions of a gene locus:

nearest_tss The locus is the region spanning the midpoints between the TSSs of adjacent genes.

nearest_gene The locus is the region spanning the midpoints between the boundaries of each gene, where a gene is defined as the region between the furthest upstream TSS and furthest downstream TES for that gene. If two gene loci overlap each other, we take the midpoint of the overlap as the boundary between the two loci. When a gene locus is completely nested within another, we create a disjoint set of 3 intervals, where the outermost gene is separated into 2 intervals broken apart at the endpoints of the nested gene.

1kb The locus is the region within 1 kb of any of the TSSs belonging to a gene. If TSSs from two adjacent genes are within 2 kb of each other, we use the midpoint between the two TSSs as the boundary for the locus for each gene.

5kb The locus is the region within 5 kb of any of the TSSs belonging to a gene. If TSSs from two adjacent genes are within 10 kb of each other, we use the midpoint between the two TSSs as the boundary for the locus for each gene.

exons Each gene has multiple loci corresponding to its exons.

4 Mappability

4.1 Base pair mappability

We define base pair mappability as the average read mappability of all possible reads of size K that encompass a specific base pair location, b. Mappability files from UCSC Genome Browser mappability track were used to calculate base pair mappability. The mappability track provides values for theoretical read mappability, or the number of places in the genome that could be mapped by a read that begins with the base pair location b. For example, a value of 1 indicates a Kmer read beginning at b is mappable to one area in the genome. A value of 0.5 indicates a Kmer read beginning at b is mappable to two areas in the genome. For our purposes, we are only interested in uniquely mappable reads; therefore, all

reads with mappability less than 1 were set to 0 to indicate non-unique mappability. Then, base pair mappability is calculated as:

$$M_i = (\frac{1}{2K-1}) \sum_{j=i-K+1}^{i+(K-1)} M_j \quad (1)$$

where M_i is the mappability of base pair i, and M_j is mappability (from UCSC's mappability track) of read j where j is the start position of the K length read. We calculated base pair mappability for reads of lengths 24, 36, 40, 50, 75, and 100 base pairs for *Homo sapiens* (build hg19) and for reads of lengths 36, 40, 50, 75, and 100 base pairs for *Mus musculus* (build mm9). Mappability is unavailable for *Rattus norvegicus* (build rn4) as there are no tracks available from UCSC.

4.2 Locus mappability

We define locus mappability as the average of all base pair mappability values for a gene's locus. Locus mappability is calculated for each available locus definition.

5 Examples

5.1 Diagnostic plots

If "qc_plots = T" then two pdf files will be output: One with a binomial smoothing spline fitted to the probability of a peak given gene length (this plot is only relevant to when "method = 'chipenrich' ") and one showing the distribution of the distance of peaks to the nearest TSS of a gene. These plots may also be generated using separate functions as illustrated below. Figure 1 shows the distribution of peaks to the nearest TSS. In figure 2, a spline is fitted to the data given gene locus length. The same is shown in figure 3 but here we account for the mappable locus length (*mappability* $\times$ *locuslength*).

There are many combinations of methods, genesets, and mappabiity settings that may be used to do gene set enrichment testing using `chipenrich`. In the following, we include some examples of gene set enrichment testing using the peaks_E2F4 example dataset.

5.2 ChIP-Enrich

Gene set enrichment of Drug Bank using ChIP-Enrich:

```
> results = chipenrich(peaks = peaks_E2F4, genesets = "drug_bank",
+         locusdef = "nearest_tss", qc_plots = F, out_name = NULL)
> results.ce = results$results[order(results$results$P.value),]
> results.ce[1:5,1:5]
```

```
   Geneset.Type Geneset.ID              Description
47    Drug Bank    DB00162                Vitamin A
83    Drug Bank    DB03431 Adenosine-5'-Diphosphate
36    Drug Bank    DB00131  Adenosine monophosphate
66    Drug Bank    DB01016             Glibenclamide
44    Drug Bank    DB00155              L-Citrulline
       P.value       FDR
47 0.008337827 0.5246291
83 0.020880101 0.5246291
36 0.023253198 0.5246291
66 0.031172558 0.5246291
44 0.031366514 0.5246291
```

5.3 ChIP-Enrich with mappability

Gene set enrichment of KEGG pathways using ChIP-Enrich, accounting for mappability:

```
> plot_dist_to_tss(peaks = peaks_E2F4, genome = 'hg19')
```

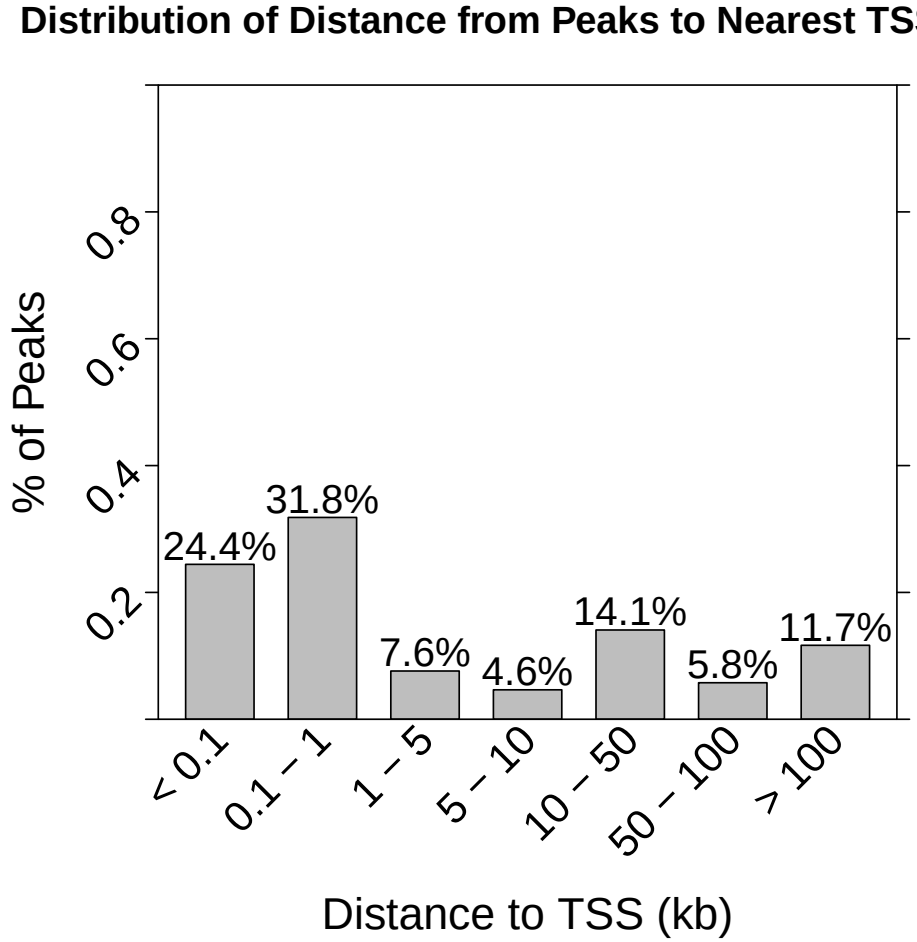

Figure 1: Distribution of distance from each peak in the dataset to the nearest TSS. All peaks are considered in creating this plot, regardless of locus definition.

```
> plot_spline_length(peaks = peaks_E2F4, locusdef = 'nearest_tss', genome = 'hg19')
```

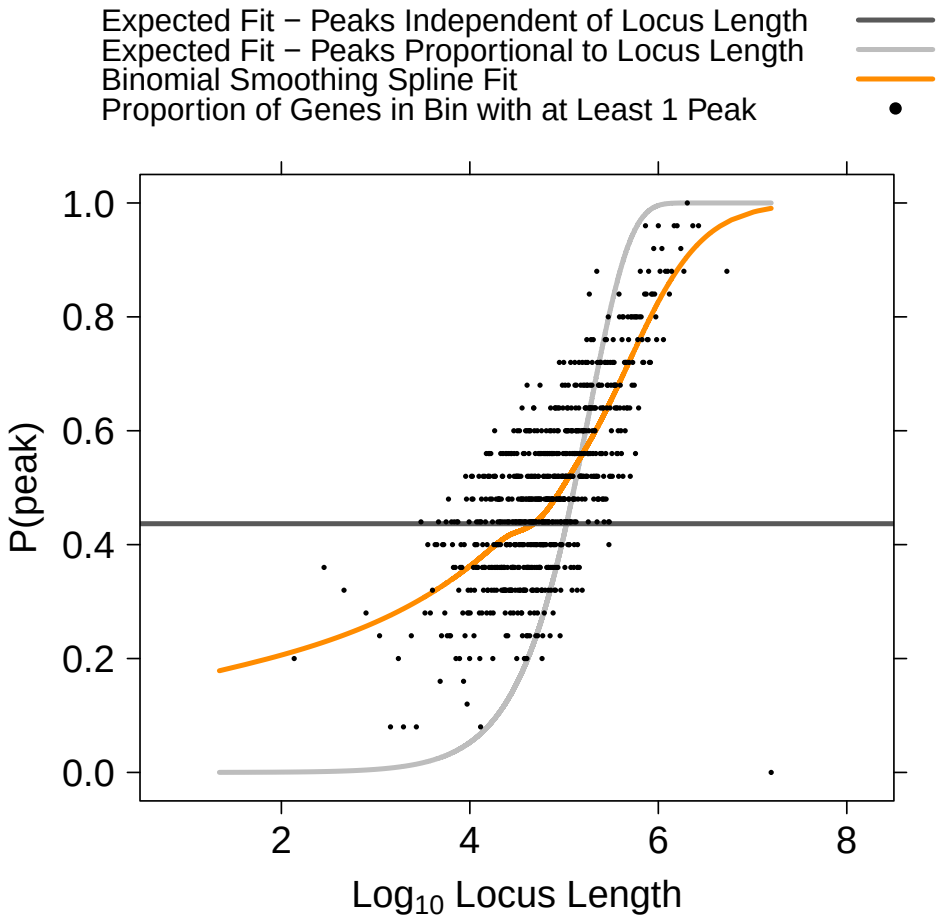

Figure 2: The relationship between the probability of a gene being assigned a peak and locus length. In the E2F4 dataset, we can see such a relationship exists (orange), and does not match either the assumption that each gene has the same probability of a peak (dark grey horizontal line), or that the probability is proportional to locus length (light grey curve.)

```
> plot_spline_length(peaks = peaks_E2F4, locusdef = 'nearest_tss',  genome = 'hg19',
use_mappability = T, read_length = 24)
```

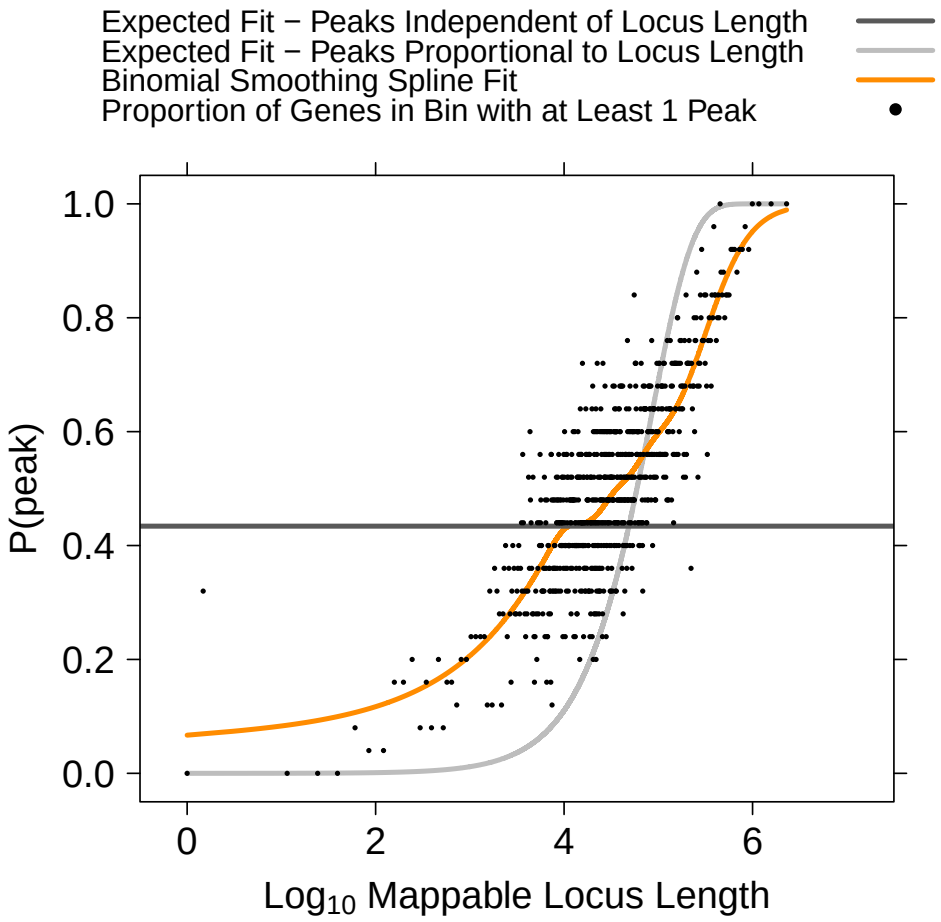

Figure 3: Spline fit using the mappable locus length as opposed to simply the locus length of each gene. In many cases, using mappability improves the spline fit, such as for those long locus length genes with poor mappability (duplicated and/or pseudo-genes.)

```
> results = chipenrich(peaks = peaks_E2F4,genesets = "kegg_pathway",
+         locusdef = "nearest_tss", use_mappability=T,
+         read_length=24, qc_plots = F, out_name = NULL)
> results.cem = results$results[order(results$results$P.value),]
> results.cem[1:5,1:5]

    Geneset.Type    Geneset.ID
122 KEGG Pathway path:hsa04110
121 KEGG Pathway path:hsa04080
119 KEGG Pathway path:hsa04060
182 KEGG Pathway path:hsa05220
47  KEGG Pathway path:hsa00510
                              Description      P.value
122                            Cell cycle 7.048325e-13
121 Neuroactive ligand-receptor interaction 1.354047e-10
119  Cytokine-cytokine receptor interaction 2.059338e-06
182                Chronic myeloid leukemia 1.004192e-05
47                    N-Glycan biosynthesis 6.416873e-05
             FDR
122 1.318037e-10
121 1.266034e-08
119 1.283654e-04
182 4.694597e-04
47  2.346665e-03
```

5.4 Fisher's exact test

Fisher's Exact test assumes that each gene is equally likely to have a peak. We recommend using Fisher's exact test with the "1kb" or "5kb" locus definitions only. This will force all genes to have approximately the same locus length and avoid returning bias results.

Gene set enrichment of KEGG pathways using Fisher's exact test:

```
> results = chipenrich(peaks = peaks_E2F4, genesets = c("kegg_pathway"),
+         locusdef = "5kb", method = "fet", fisher_alt = "two.sided", qc_plots = F, out_name = NULL)
> results.fet = results$results[order(results$results$P.value),]
> results.fet[1:5,1:5]

    Geneset.Type    Geneset.ID
122 KEGG Pathway path:hsa04110
121 KEGG Pathway path:hsa04080
119 KEGG Pathway path:hsa04060
107 KEGG Pathway path:hsa01430
102 KEGG Pathway path:hsa00980
                                     Description
122                                   Cell cycle
121      Neuroactive ligand-receptor interaction
119       Cytokine-cytokine receptor interaction
107                           Cell Communication
102 Metabolism of xenobiotics by cytochrome P450
         P.value          FDR
122 1.826679e-27 3.415891e-25
121 3.821554e-16 3.573153e-14
119 2.546558e-08 1.587354e-06
107 8.529743e-08 3.987655e-06
102 1.111168e-07 4.155767e-06
```

6 Output

The output of `chipenrich()` is an R object containing the results of the test and the peak to gene assignments. Both of these are also written to text files in the working directory (unless specified otherwised) after the test is completed.

6.1 Peak-to-gene assignments

Peak assignments are stored in `$peaks`. The following is an example of how to access the peak to gene assignments in R after a gene set enrichment test has already been performed:

```
> peaks_to_genes = results$peaks
> head(peaks_to_genes)
```

```
  chrom peak_start  peak_end peak_midpoint    geneid
1  chr1    3817484   3817622       3817553 100133612
2  chr1    3817811   3817975       3817893 100133612
3  chr1   77685073  77685175      77685124     10026
4  chr1   29562944  29563102      29563023     10076
5  chr1  183604939 183605051     183604995     10092
6  chr1  180992048 180992155     180992101     10228
   gene_symbol gene_locus_start gene_locus_end nearest_tss
1 LOC100133612          3816912        3821967     3816967
2 LOC100133612          3816912        3821967     3816967
3         PIGK         77680132       77690132    77685132
4        PTPRU         29560250       29568046    29563027
5        ARPC5        183599985      183605095   183605076
6         STX6        180987046      180997046   180992046
  dist_to_tss nearest_tss_gene nearest_tss_gene_strand
1         585        100133612                       +
2         925        100133612                       +
3           7            10026                       -
4          -3            10076                       +
5          80            10092                       -
6         -54            10228                       -
```

6.2 Gene set enrichment test results

The results of a `chipenrich()` R object is stored in `$results`. It contains 9 columns:

```
> colnames(results$results)
```

```
 [1] "Geneset.Type"          "Geneset.ID"
 [3] "Description"           "P.value"
 [5] "FDR"                   "Odds.Ratio"
 [7] "Status"                "N.Geneset.Genes"
 [9] "N.Geneset.Peak.Genes" "Geneset.Peak.Genes"
```

Geneset.ID is the identifier for a given gene set from the selected database. For example, "GO:0000003."

Geneset.Type specifies from which database the "`Geneset.ID`" originates. For example, "Gene Ontology Biological Process."

Description gives a definition of the"`Geneset.ID`." For example,"reproduction."

P.Value is the probability of observing the degree of enrichment (see"`Odds.Ratio`") of the gene set given the null hypothesis that peaks are not associated with any gene sets.

FDR is the false discovery rate proposed by Bejamini & Hochberg for adjusting the p-value to control for family-wise error rate.

Odds.Ratio is the estimated odds that peaks are associated with a given gene set compared to the odds that peaks are associated with other gene sets, after controlling for locus length and/or mappability. An odds ratio greater than 1 indicates enrichment, and less than 1 indicates depletion.

Status denotes whether the gene set was enriched or depleted.

N.Geneset.Genes is the number of genes in the gene set.

N.Geneset.Peak.Genes is the number of genes in the genes set that were assigned at least one peak.

Geneset.Peak.Genes is the list of genes from the gene set that had at least one peak assigned.

7 References

R.P. Welch, C. Lee, R.A. Smith, P. Imbriano, S. Patil, T. Weymouth, L.J. Scott, M.A. Sartor. "ChIP-Enrich: gene set enrichment testing for ChIP-seq data." In preparation.