promoter_comparison

Promoters are upstream regions of all protein-coding genes

```
library(biomaRt)
mart = getParamart()
```

```
## Database connected
## biomart      ...      parasite_mart
## host         ...      https://parasite.wormbase.org:443/biomart/martservice
## dataset      ...      wbps_gene
```

```
UPSTREAM=1000
DOWNSTREAM=200
promoters = getCElegansPromoters(mart, upstream = UPSTREAM, downstream = DOWNSTREAM)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
```

```
## The following object is masked from 'package:biomaRt':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## getBM(filter = c("biotype", "species_id_1010"), value = list(
##     biotype = "protein_coding", species_id_1010 = "caelegprjna13758"),
```

```
##     attributes = c("wbps_gene_id", "external_gene_id", "chromosome_name",
##     "start_position", "end_position", "strand"))
```

```
## Warning in valid.GenomicRanges.seqinfo(x, suggest.trim = TRUE): GRanges object contains 1 out-of-bou
##   Note that ranges located on a sequence whose length is unknown (NA) or
##   on a circular sequence are not considered out-of-bound (use
##   seqlengths() and isCircular() to get the lengths and circularity flags
##   of the underlying sequences). You can use trim() to trim these ranges.
##   See ?`trim,GenomicRanges-method` for more information.
```

```
promoters = trim(sort(promoters, ignore.strand=T)) # trim because one interval is chrIV:-359-840 at -10
head(promoters)
```

```
## GRanges object with 6 ranges and 2 metadata columns:
##       seqnames      ranges strand |   wbps_gene_id external_gene_id
##          <Rle>   <IRanges>  <Rle> |    <character>      <character>
##   [1]     chrI 10031-11230      - | WBGene00022277           homt-1
##   [2]     chrI 10495-11694      + | WBGene00022276           nlp-40
##   [3]     chrI 26582-27781      - | WBGene00022278           rcor-1
##   [4]     chrI 32951-34150      - | WBGene00022279           sesn-1
##   [5]     chrI 42733-43932      + | WBGene00022275            txt-7
##   [6]     chrI 46461-47660      + | WBGene00044345        Y48G1C.12
##   -------
##   seqinfo: 7 sequences (1 circular) from ce11 genome
```

```
selfOverlaps = findOverlaps(promoters, ignore.strand=T)
#head(selfOverlaps)

# selfOverlaps includes everything against itself + overlaps between promoters
# Filter out the self hits, and retain the "between" hits as "collisions".
collisions = selfOverlaps[!isSelfHit(selfOverlaps)]

overlappingPromoterRows = unique(c( from(collisions), to(collisions)))
length(overlappingPromoterRows)
```

```
## [1] 6749
```

```
sprintf("There are %d overlaps between %d promoters.", length(collisions), length(overlappingPromoterRo
```

```
## [1] "There are 8008 overlaps between 6749 promoters."
```

```
filtered.promoters = promoters[-overlappingPromoterRows]
filtered.promoters = filtered.promoters[-which(seqnames(filtered.promoters) == 'chrM')]
sprintf("There are %d unambiguous promoters.", length(filtered.promoters))
```

```
## [1] "There are 13246 unambiguous promoters."
```

```
# -500,+200
# "There are 4256 overlaps between 4067 promoters."
# "There are 15922 unambiguous promoters."

# -1000,+200
#"There are 8008 overlaps between 6749 promoters."
#"There are 13246 unambiguous promoters."
```

```
OUTPUT_03 = normalizePath("../03_output")
PROMOTOR_BED_PATH = sprintf("%s/filtered.promoters.minus%d_plus%d.bed", OUTPUT_03, UPSTREAM, DOWNSTREAM
write.table(filtered.promoters, PROMOTOR_BED_PATH, sep="\t", quote=F, row.names=F, col.names=F)
```

Setup a conda environment in your shell

I had to call my local setup script .zshrc, where I have initialized conda, to have access to the "base" environment, where I have installed wiggletools and ucsc user apps.

```
$ wiggletools apply_paste filtered.promoters.minus1000_plus200.df meanI maxI filtered.promoters.minus10
ELT2_LE_combined_subtracted.bw
```

The same can be done for the IDR peaks.

```
$ wiggletools apply_paste LE_IDR_peaks.df meanI maxI ELT2_LE_combined.IDR.bed ELT2_LE_combined_subtracte
```

```
PROMOTOR_DF_PATH = sprintf("%s/filtered.promoters.minus%d_plus%d.df", OUTPUT_03, UPSTREAM, DOWNSTREAM)
promoters.agg = read.table(PROMOTOR_DF_PATH)
colnames(promoters.agg) <- c("chrom", "start","end","len", "strand", "wbps_gene_id", "gene_name", "chip

IDR_peaks.agg = read.table(file.path(OUTPUT_03,"LE_IDR_peaks.df"))
IDR_peaks.agg$V4 = NULL
IDR_peaks.agg$V5 = NULL
IDR_peaks.agg$V6 = NULL
IDR_peaks.agg$V8 = NULL
colnames(IDR_peaks.agg) <- c("chrom", "start","end","intensity","nlogq","offset","signal.mean","signal.

gr.IDR = makeGRangesFromDataFrame(IDR_peaks.agg,keep.extra.columns = T)
seqinfo(gr.IDR) <- Seqinfo(genome="ce11")

gr.promoters = makeGRangesFromDataFrame(promoters.agg,keep.extra.columns = T)
seqinfo(gr.promoters) <- Seqinfo(genome="ce11")
```

```
chipmean.minval = min(gr.promoters$chip_signal_mean,na.rm=T)
chipmean.minval
```

```
## [1] -100.4667
```

```
chipmax.minval = min(gr.promoters$chip_signal_max,na.rm=T)
chipmax.minval
```

```
## [1] -80.85739
```

```
chipmean.log = log(-chipmean.minval + 1 + gr.promoters$chip_signal_mean,base=2)
chipmax.log = log(-chipmax.minval + 1 + gr.promoters$chip_signal_max,base=2)

gr.promoters$log_chip_signal_mean = chipmean.log
gr.promoters$log_chip_signal_max = chipmax.log
head(gr.promoters)
```

```
## GRanges object with 6 ranges and 7 metadata columns:
##       seqnames      ranges strand |       len   wbps_gene_id   gene_name
##          <Rle>   <IRanges>  <Rle> | <integer>    <character> <character>
##   [1]     chrI 26582-27781      - |      1200 WBGene00022278      rcor-1
##   [2]     chrI 32951-34150      - |      1200 WBGene00022279      sesn-1
##   [3]     chrI 42733-43932      + |      1200 WBGene00022275       txt-7
##   [4]     chrI 46461-47660      + |      1200 WBGene00044345   Y48G1C.12
##   [5]     chrI 48921-50120      + |      1200 WBGene00021677       pgs-1
##   [6]     chrI 63867-65066      - |      1200 WBGene00021678    Y48G1C.5
##       chip_signal_mean chip_signal_max log_chip_signal_mean log_chip_signal_max
##              <numeric>       <numeric>            <numeric>           <numeric>
```

```
##   [1]        116.59365       220.93678              7.76858              8.24219
##   [2]         23.56896        38.75358              6.96620              6.91422
##   [3]          7.16118        18.78316              6.76325              6.65307
##   [4]         26.93845        43.20576              7.00456              6.96651
##   [5]         11.93393        34.69149              6.82529              6.86479
##   [6]         -5.76947         9.25825              6.58041              6.50963
##   -------
##   seqinfo: 7 sequences (1 circular) from ce11 genome
```

```
LOG_PROMOTOR_DF_PATH = sprintf("%s/log_filtered.promoters.minus%d_plus%d.df", OUTPUT_03, UPSTREAM, DOWNS
write.table(as.data.frame(gr.promoters), file = LOG_PROMOTOR_DF_PATH,quote=F, row.names=F,sep="\t")
```

```
laps = findOverlaps(gr.promoters,gr.IDR)
length(laps)
```

```
## [1] 1424
```

```
head(laps)
```

```
## Hits object with 6 hits and 0 metadata columns:
##       queryHits subjectHits
##       <integer>   <integer>
##   [1]         1           4
##   [2]        29           7
##   [3]        31           8
##   [4]        32           9
##   [5]        38          14
##   [6]        42          16
##   -------
##   queryLength: 13246 / subjectLength: 4098
```

```
gr.promoters$IDR_mean = NaN
gr.promoters$IDR_max = NaN
gr.promoters[from(laps)]$IDR_max = gr.IDR[to(laps)]$signal.max
gr.promoters[from(laps)]$IDR_mean = gr.IDR[to(laps)]$signal.mean
```

(base) Cumbernauld:ELT-2-ChIP-promoters david$ bigWigInfo ELT2_LE_combined_subtracted.bw version: 4 isCompressed: yes isSwapped: 0 primaryDataSize: 7,724,199 primaryIndexSize: 38,660 zoomLevels: 9 chromCount: 7 basesCovered: 35,608,464 mean: 0.350340 min: -3276.705811 max: 5699.461426 std: 46.132783