promoter_comparison

```
RNASEQ=params$rob.data

UPSTREAM=1000
DOWNSTREAM=200
IDR_BED = sprintf("../01_input/ELT2_%s_combined_IDR.bed", params$stage) # peaks input file
OUTPUT_03 = normalizePath("../03_output")

# output files from genomic ranges
PROMOTOR_BED_PATH = sprintf("%s/filtered.promoters.minus%d_plus%d.bed",
                            OUTPUT_03,
                            UPSTREAM,
                            DOWNSTREAM)
# colliding promoters removed
NR_PROMOTOR_BED_PATH = sprintf("%s/nr.promoters.minus%d_plus%d.bed",
                               OUTPUT_03,
                               UPSTREAM,
                               DOWNSTREAM)

# input signal file for wiggle tool step
SIGNAL_BW = sprintf("../01_input/ELT2_%s_combined_subtracted.bw", params$stage)
# input signal file for deeptools (e.g. ELT2_L1_combined_subtracted.interp.bw)
INTERP_SIGNAL_BW = sprintf("../01_input/ELT2_%s_combined_subtracted.interp.bw", params$stage)

# output files from wiggle tool step
PROMOTOR_DF_PATH = sprintf("%s/filtered.promoters.minus%d_plus%d.df",
                           OUTPUT_03,
                           UPSTREAM,
                           DOWNSTREAM)

NR_PROMOTOR_DF_PATH = sprintf("%s/nr.promoters.minus%d_plus%d.df",
                           OUTPUT_03,
                           UPSTREAM,
                           DOWNSTREAM)

IDR_DF = sprintf("../01_input/ELT2_%s_combined_IDR.df", params$stage) # peaks with signal agg

#################
```

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
```

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

```
## 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"))
```

```
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[-which(seqnames(promoters) == 'chrM')]

# to remove overlapping promoters, uncomment below
nr.promoters = filtered.promoters[-overlappingPromoterRows]
sprintf("There are %d unambiguous promoters.", length(nr.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."

write.table(filtered.promoters, PROMOTOR_BED_PATH, sep="\t", quote=F, row.names=F, col.names=F)
write.table(nr.promoters, NR_PROMOTOR_BED_PATH, sep="\t", quote=F, row.names=F, col.names=F)
```

Setup a conda environment in your shell

Install a conda environment containing wiggletools and ucsc user apps via `root/David/01_promoters/02_scripts/conda_env`

To pass variable names to the *bash* chunk by setting them in the environment with `Sys.setenv`.

```
Sys.setenv(PROMOTOR_BED_PATH=PROMOTOR_BED_PATH, # all promoters
           NR_PROMOTOR_BED_PATH = NR_PROMOTOR_BED_PATH, # overlapping removed
           IDR_BED = IDR_BED,
           IDR_DF = IDR_DF,
           SIGNAL_BW = SIGNAL_BW,
           PROMOTOR_DF_PATH = PROMOTOR_DF_PATH,
           NR_PROMOTOR_DF_PATH = NR_PROMOTOR_DF_PATH,
           STAGE=params$stage
           )
```

Run wiggletools in a bash session.

```
source $HOME/.bash_profile
conda activate elt-2-rev

set -ue # exit 1 if any vars are not set (using Sys.setenv above)
echo PROMOTOR_BED_PATH $PROMOTOR_BED_PATH
echo NR_PROMOTOR_BED_PATH $NR_PROMOTOR_BED_PATH
echo NR_PROMOTOR_DF_PATH $NR_PROMOTOR_DF_PATH
echo IDR_BED $IDR_BED
echo IDR_DF $IDR_DF
echo SIGNAL_BW $SIGNAL_BW
echo STAGE $STAGE

#wiggletools
wiggletools apply_paste - meanI maxI $PROMOTOR_BED_PATH $SIGNAL_BW > $PROMOTOR_DF_PATH
echo $PROMOTOR_DF_PATH
head -5 $PROMOTOR_DF_PATH

wiggletools apply_paste - meanI maxI $NR_PROMOTOR_BED_PATH $SIGNAL_BW > $NR_PROMOTOR_DF_PATH
echo $NR_PROMOTOR_DF_PATH
head -5 $NR_PROMOTOR_DF_PATH

wiggletools apply_paste - meanI maxI $IDR_BED $SIGNAL_BW > $IDR_DF
echo $IDR_DF
head -5 $IDR_DF
```

```
## PROMOTOR_BED_PATH /Users/david/work/ELT-2-ChIP-revision/David/01_promoters/03_output/filtered.promot
## NR_PROMOTOR_BED_PATH /Users/david/work/ELT-2-ChIP-revision/David/01_promoters/03_output/nr.promoters
## NR_PROMOTOR_DF_PATH /Users/david/work/ELT-2-ChIP-revision/David/01_promoters/03_output/nr.promoters.
## IDR_BED ../01_input/ELT2_LE_combined_IDR.bed
## IDR_DF ../01_input/ELT2_LE_combined_IDR.df
## SIGNAL_BW ../01_input/ELT2_LE_combined_subtracted.bw
## STAGE LE
## /Users/david/work/ELT-2-ChIP-revision/David/01_promoters/03_output/filtered.promoters.minus1000_plus
## chrI 10031   11230   1200    -   WBGene00022277  homt-1  17.987559   94.528107
## chrI 10495   11694   1200    +   WBGene00022276  nlp-40  47.095758   101.579247
## chrI 26582   27781   1200    -   WBGene00022278  rcor-1  116.593648  220.936783
## chrI 32951   34150   1200    -   WBGene00022279  sesn-1  23.568960   38.753582
## chrI 42733   43932   1200    +   WBGene00022275  txt-7   7.161179    18.783163
## /Users/david/work/ELT-2-ChIP-revision/David/01_promoters/03_output/nr.promoters.minus1000_plus200.df
## chrI 26582   27781   1200    -   WBGene00022278  rcor-1  116.593648  220.936783
## chrI 32951   34150   1200    -   WBGene00022279  sesn-1  23.568960   38.753582
## chrI 42733   43932   1200    +   WBGene00022275  txt-7   7.161179    18.783163
## chrI 46461   47660   1200    +   WBGene00044345  Y48G1C.12   26.938451   43.205757
## chrI 48921   50120   1200    +   WBGene00021677  pgs-1   11.933928   34.691494
## ../01_input/ELT2_LE_combined_IDR.df
## chrI 3661    4117    .   0   .   79.0848644469403    -1  2.86484056630441    228 99.779078   107.7549
## chrI 11112   11568   .   0   .   83.9050179045692    -1  2.94001815500769    228 85.471255   101.5792
## chrI 16762   17218   .   0   .   99.4021006146189    -1  2.97266559226614    228 98.322681   108.1028
## chrI 26839   27295   .   0   .   199.906215809772    -1  3.57760667736254    228 194.236994  220.9367
## chrI 110411  110867  .   0   .   81.0040191671889    -1  2.95965456213624    228 99.372743   118.4187
```

Read in the results of the wiggletools commands.

```
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(IDR_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.m

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")
```

Attach log scale promoter signal values.

```
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 10031-11230      - |      1200 WBGene00022277      homt-1
##   [2]     chrI 10495-11694      + |      1200 WBGene00022276      nlp-40
##   [3]     chrI 26582-27781      - |      1200 WBGene00022278      rcor-1
##   [4]     chrI 32951-34150      - |      1200 WBGene00022279      sesn-1
##   [5]     chrI 42733-43932      + |      1200 WBGene00022275       txt-7
##   [6]     chrI 46461-47660      + |      1200 WBGene00044345   Y48G1C.12
##       chip_signal_mean chip_signal_max log_chip_signal_mean log_chip_signal_max
##              <numeric>       <numeric>            <numeric>           <numeric>
##   [1]         17.98756         94.5281              6.90031             7.46259
##   [2]         47.09576        101.5792              7.21493             7.51914
##   [3]        116.59365        220.9368              7.76858             8.24219
##   [4]         23.56896         38.7536              6.96620             6.91422
##   [5]          7.16118         18.7832              6.76325             6.65307
##   [6]         26.93845         43.2058              7.00456             6.96651
##   -------
##   seqinfo: 7 sequences (1 circular) from ce11 genome
```

```
# output file
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")
```

Find overlaps between promoters and IDR peaks. Populate IDR signal fields when a peak exists, leave NaN otherwise.

```
laps = findOverlaps(gr.promoters,gr.IDR, ignore.strand=T,minoverlap = 100)

head(laps)
```

```
## Hits object with 6 hits and 0 metadata columns:
##       queryHits subjectHits
##       <integer>   <integer>
##   [1]         1           2
##   [2]         2           2
##   [3]         3           4
##   [4]        16           5
##   [5]        17           5
##   [6]        37           7
##   -------
##   queryLength: 19985 / subjectLength: 4098
```

```
gr.promoters$IDR_mean = NaN
gr.promoters$IDR_max = NaN
gr.promoters$IDR_value = NaN
gr.promoters$nlogq = 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
gr.promoters[from(laps)]$IDR_value = gr.IDR[to(laps)]$intensity
gr.promoters[from(laps)]$nlogq = gr.IDR[to(laps)]$nlogq
print("Number of promoters overlapping an IDR peak:")
```

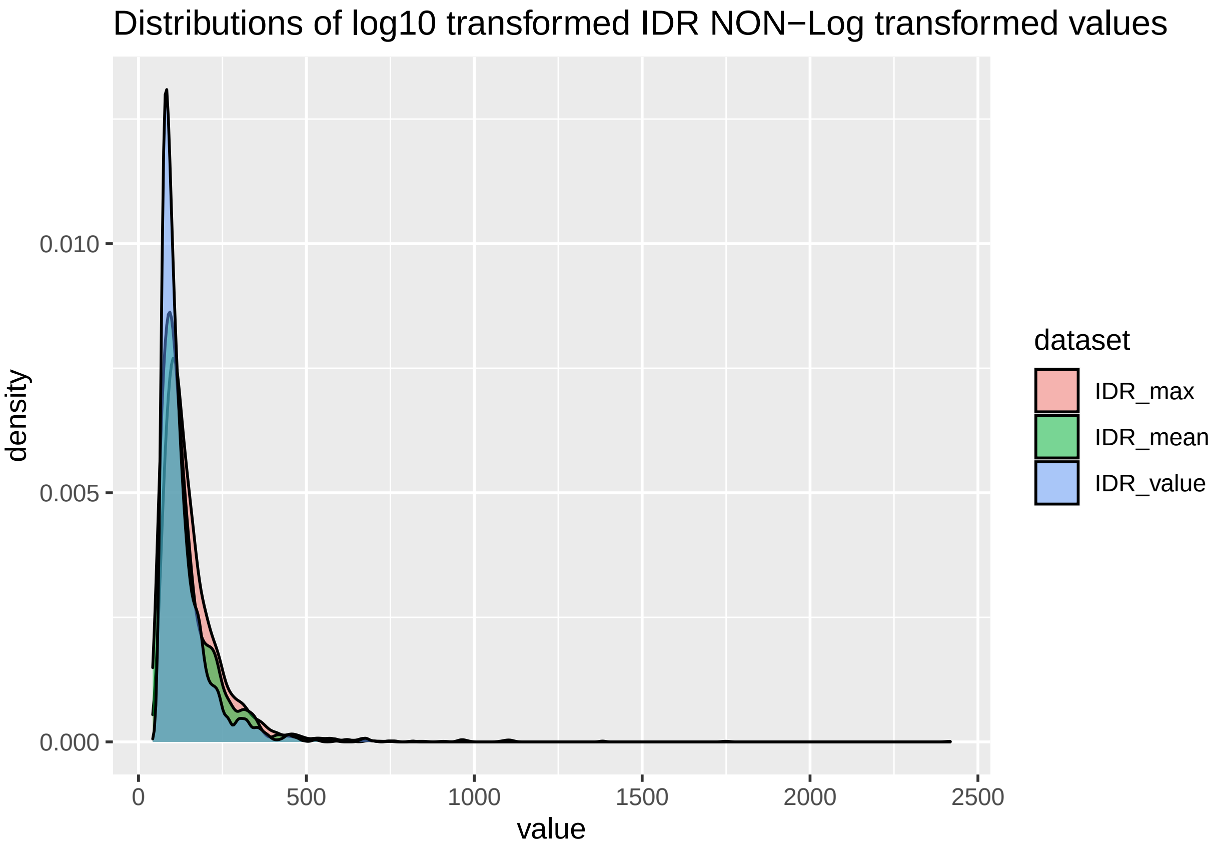
```
## [1] "Number of promoters overlapping an IDR peak:"
```

```
sum(!is.nan(gr.promoters$IDR_max))
```

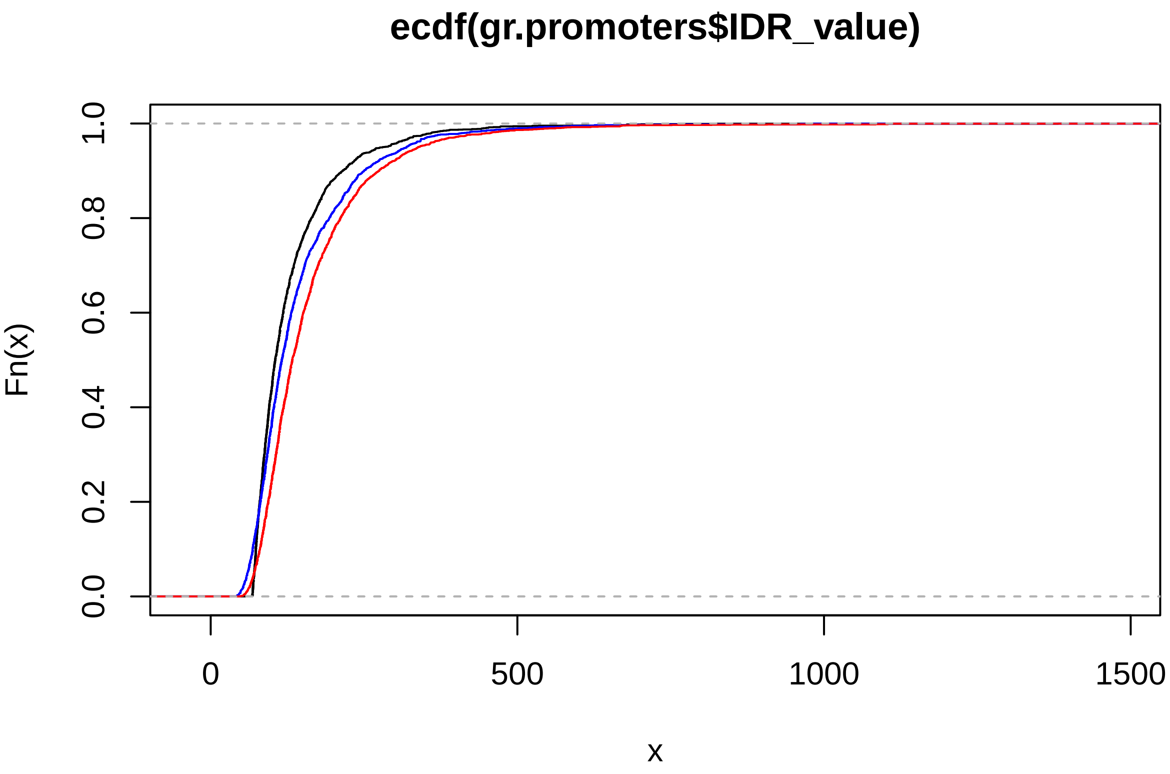
```
## [1] 2629
```

```
idr.nonlog = gather(as.data.frame(gr.promoters)[,c('IDR_value','IDR_mean','IDR_max')], key="dataset")
ggplot(idr.nonlog, aes(x=value, fill=dataset))  + geom_density(alpha=.5) + labs(title="Distributions of
```

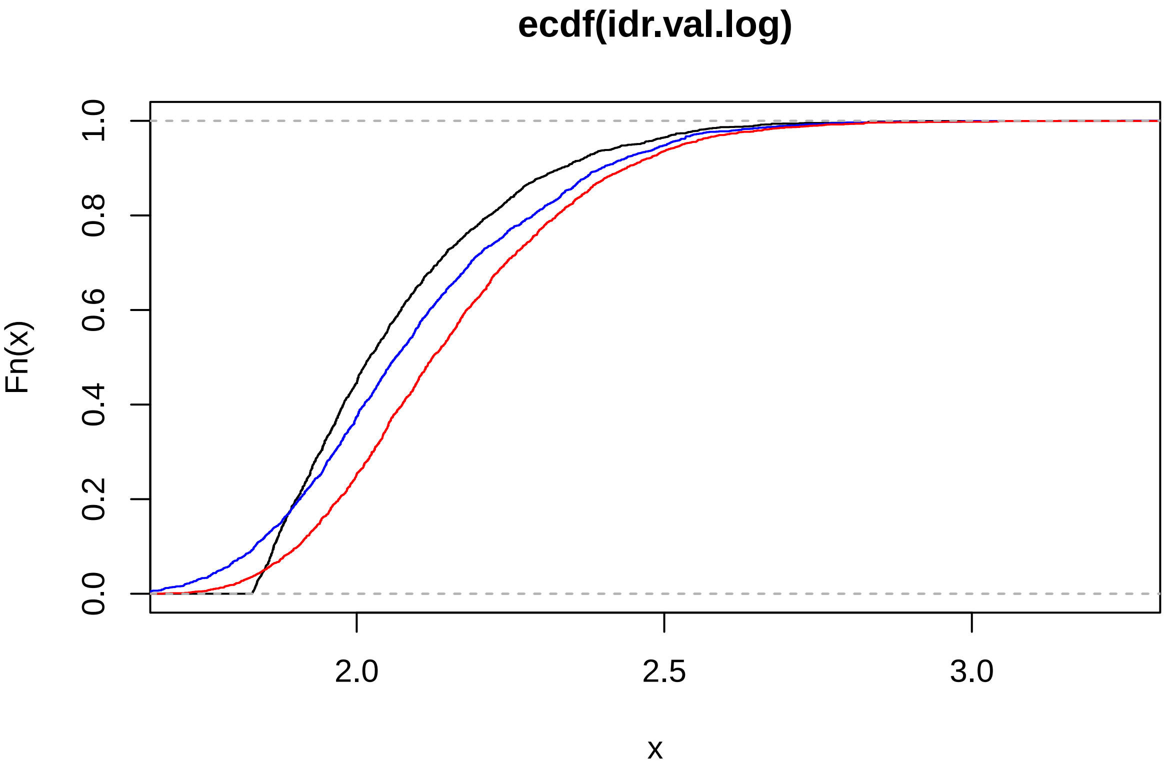
```
## Warning: Removed 52068 rows containing non-finite values (stat_density).
```



```
idr.val.ecdf = ecdf(gr.promoters$IDR_value)
idr.mean.ecdf = ecdf(gr.promoters$IDR_mean)
idr.max.ecdf = ecdf(gr.promoters$IDR_max)

plot(idr.val.ecdf)
lines(idr.mean.ecdf,col="blue")
lines(idr.max.ecdf,col="red")
```



```
# the data currently have all positive values, so no adjustment made for log
idr.val.log = log10(gr.promoters$IDR_value)
idr.mean.log = log10(gr.promoters$IDR_mean)
idr.max.log = log10(gr.promoters$IDR_max)

idr.val.log.ecdf = ecdf(idr.val.log)
idr.mean.log.ecdf = ecdf(idr.mean.log)
idr.max.log.ecdf = ecdf(idr.max.log)

plot(idr.val.log.ecdf)
lines(idr.mean.log.ecdf,col="blue")
lines(idr.max.log.ecdf,col="red")
```

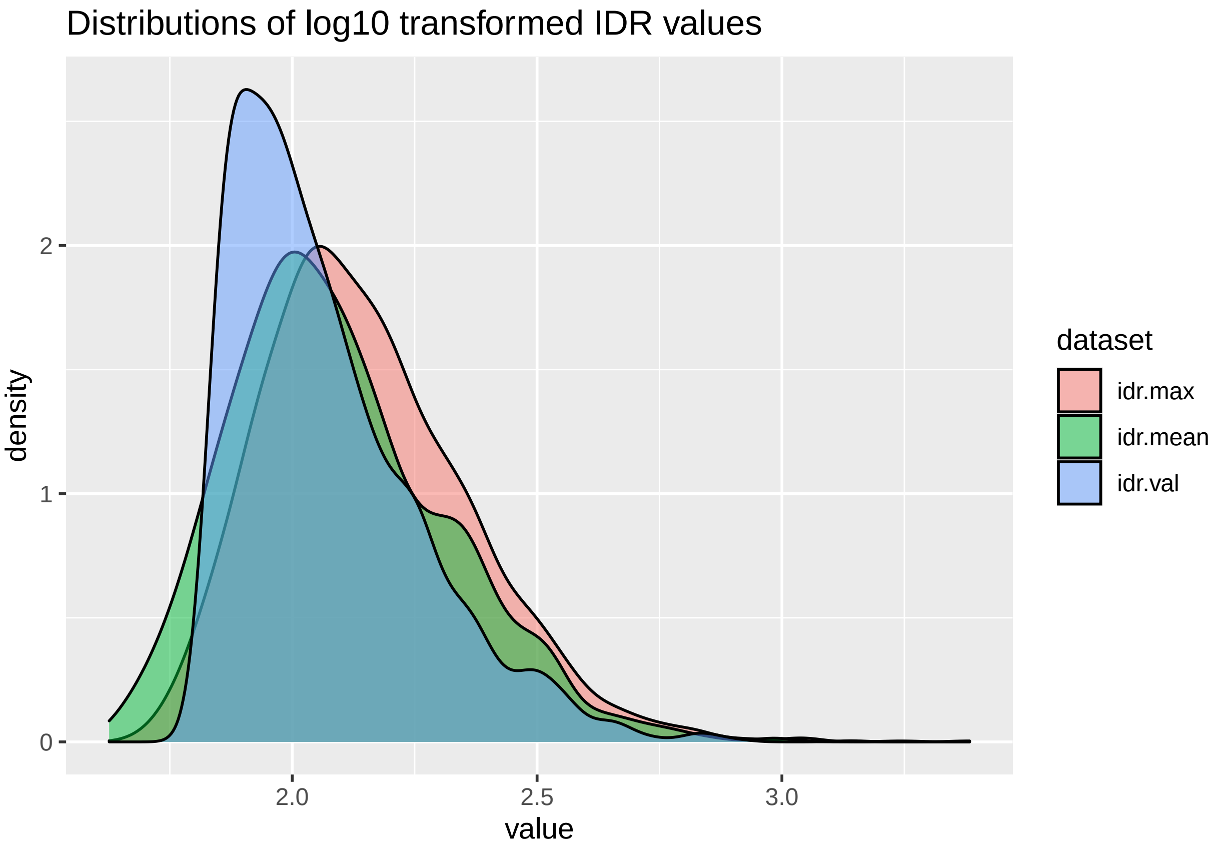


```
log.vals = data.frame(idr.mean = idr.mean.log, idr.val = idr.val.log, idr.max = idr.max.log)

long.log.vals = gather(log.vals, key="dataset")
head(long.log.vals)
```

```
##    dataset    value
## 1 idr.mean 1.931820
## 2 idr.mean 1.931820
## 3 idr.mean 2.288332
## 4 idr.mean      NaN
## 5 idr.mean      NaN
## 6 idr.mean      NaN
```

```
ggplot(long.log.vals, aes(x=value, fill=dataset))  + geom_density(alpha=.5) + labs(title="Distributions
```

```
## Warning: Removed 52068 rows containing non-finite values (stat_density).
```



```
gr.promoters$IDR_logTEN_max = idr.max.log
gr.promoters$IDR_logTEN_mean = idr.mean.log
gr.promoters$IDR_logTEN_value = idr.val.log
sum(idr.mean.log > 2.5, na.rm=T)
```

```
## [1] 136
```

Read in RNA-seq data, join promoters by wbps geneid, and then sort logFoldChange high to low.

```
# input file
rnaseq = read.csv(RNASEQ)
rownames(rnaseq) <- rnaseq$WBGeneID

mcols(gr.promoters) <- mcols(gr.promoters) %>%
  cbind(rnaseq[gr.promoters$wbps_gene_id,2:6])  %>%
  as.data.frame() %>%
  dplyr::rename(IDR_nlogq = nlogq)

names(gr.promoters) <- gr.promoters$wbps_gene_id

# sort promoters high to low by log2FC
gr.promoters = gr.promoters[order(gr.promoters$log2FoldChange,decreasing=T)]

head(gr.promoters)
```

```
## GRanges object with 6 ranges and 19 metadata columns:
##                   seqnames            ranges strand |       len   wbps_gene_id
##                      <Rle>         <IRanges>  <Rle> | <integer>    <character>
##   WBGene00007725      chrV 19410658-19411857      - |      1200 WBGene00007725
##   WBGene00044723    chrIV     670356-671555      - |      1200 WBGene00044723
##   WBGene00008044   chrIII   9318941-9320140      + |      1200 WBGene00008044
##   WBGene00001932    chrIV 11339907-11341106      - |      1200 WBGene00001932
##   WBGene00044291     chrV 19404729-19405928      + |      1200 WBGene00044291
##   WBGene00010745    chrIV 12975303-12976502      + |      1200 WBGene00010745
##                    gene_name chip_signal_mean chip_signal_max
##                  <character>        <numeric>       <numeric>
##   WBGene00007725     C25F9.5         28.17897        51.42597
##   WBGene00044723   K11H12.11          1.78768         9.95639
##   WBGene00008044     C40H1.9         10.46317        32.82366
##   WBGene00001932      his-58          7.83876        23.42412
##   WBGene00044291    C25F9.10          8.84522        21.00955
##   WBGene00010745      dod-17         -7.73933        19.86670
##                  log_chip_signal_mean log_chip_signal_max  IDR_mean   IDR_max
##                             <numeric>           <numeric> <numeric> <numeric>
##   WBGene00007725              7.01843             7.05835       NaN       NaN
##   WBGene00044723              6.69006             6.52064       NaN       NaN
##   WBGene00008044              6.80645             6.84148       NaN       NaN
##   WBGene00001932              6.77222             6.71811       NaN       NaN
##   WBGene00044291              6.78545             6.68464       NaN       NaN
##   WBGene00010745              6.55040             6.66852       NaN       NaN
##                  IDR_value IDR_nlogq IDR_logTEN_max IDR_logTEN_mean
##                  <numeric> <numeric>      <numeric>       <numeric>
##   WBGene00007725       NaN       NaN            NaN             NaN
##   WBGene00044723       NaN       NaN            NaN             NaN
##   WBGene00008044       NaN       NaN            NaN             NaN
##   WBGene00001932       NaN       NaN            NaN             NaN
##   WBGene00044291       NaN       NaN            NaN             NaN
##   WBGene00010745       NaN       NaN            NaN             NaN
##                  IDR_logTEN_value  baseMean log2FoldChange     lfcSE
##                         <numeric> <numeric>      <numeric> <numeric>
##   WBGene00007725              NaN   314.398        13.2990  2.823009
##   WBGene00044723              NaN   212.125        12.4555  2.728411
##   WBGene00008044              NaN   123.845        11.9332  2.826257
##   WBGene00001932              NaN 14889.029        11.6219  0.725403
##   WBGene00044291              NaN   100.927        11.3942  2.684083
##   WBGene00010745              NaN   112.920        11.2570  2.679958
##                       pvalue        padj
##                    <numeric>   <numeric>
##   WBGene00007725 9.65618e-21 2.64520e-19
##   WBGene00044723 5.82794e-20 1.52209e-18
##   WBGene00008044 7.80849e-16 1.55963e-14
##   WBGene00001932 4.75752e-58 7.06924e-56
##   WBGene00044291 2.64330e-17 5.79281e-16
##   WBGene00010745 4.60768e-16 9.35560e-15
##   -------
##   seqinfo: 7 sequences (1 circular) from ce11 genome
```

```
# look at the number filtered by DESeq2
# as described by https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#
baseMean_is_zero = rnaseq$baseMean == 0
pval_na = is.na(rnaseq$pvalue)
padj_na = is.na(rnaseq$padj)
# case one
```

```
sum(baseMean_is_zero & pval_na & padj_na)
```

```
## [1] 0
```

```
# case two
sum(!baseMean_is_zero & pval_na & padj_na)
```

```
## [1] 52
```

```
# case three
sum(!pval_na & padj_na)
```

```
## [1] 3088
```

```
# divide groups by peak and padj
enriched_intestine = gr.promoters$padj<.05 & !is.na(gr.promoters$padj) & gr.promoters$log2FoldChange >
has_peak = !is.nan(gr.promoters$IDR_max)
classA = enriched_intestine & has_peak
classB = !enriched_intestine & has_peak
classC = enriched_intestine & !has_peak
classD = !enriched_intestine & !has_peak

m = matrix( c(sum(classA),
            sum(classB),
            sum(classC),
            sum(classD)), ncol = 2)
m.chisq = chisq.test(m)

gr.promoters$class = "classA"
gr.promoters$class[classB] <- "classB"
gr.promoters$class[classC] <- "classC"
gr.promoters$class[classD] <- "classD"
```

```
promoters.hilo = as.data.frame(gr.promoters)

# BED format
write.table(promoters.hilo, file.path(OUTPUT_03, "promoters.hilo.bed"), quote=F, sep="\t", row.names=F,

# Matrix format readable into R
write.table(promoters.hilo, file.path(OUTPUT_03, "promoters.hilo.tsv"), quote=F, sep="\t", row.names=T,


PROMOTERS_HILO_BED_PATH = file.path(OUTPUT_03, "promoters.hilo.bed")
PROMOTERS_HILO_BED_PATH_A = file.path(OUTPUT_03, "promoters.hilo.classA.bed")
PROMOTERS_HILO_BED_PATH_B = file.path(OUTPUT_03, "promoters.hilo.classB.bed")
PROMOTERS_HILO_BED_PATH_C = file.path(OUTPUT_03, "promoters.hilo.classC.bed")
PROMOTERS_HILO_BED_PATH_D = file.path(OUTPUT_03, "promoters.hilo.classD.bed")


write.table(promoters.hilo[classA,],
            PROMOTERS_HILO_BED_PATH_A, quote=F, sep="\t", row.names=F, col.names=F)
write.table(promoters.hilo[classB,],
            PROMOTERS_HILO_BED_PATH_B, quote=F, sep="\t",
row.names=F, col.names=F)
write.table(promoters.hilo[classC,],
            PROMOTERS_HILO_BED_PATH_C, quote=F, sep="\t",
row.names=F, col.names=F)
write.table(promoters.hilo[classD,],
            PROMOTERS_HILO_BED_PATH_D, quote=F, sep="\t",
row.names=F, col.names=F)

#### deeptooling up versus down only, no other filters
promoters.hilo.up = promoters.hilo %>% filter(log2FoldChange > 0)
promoters.hilo.down = promoters.hilo %>% filter(log2FoldChange < 0)

PROMOTERS_HILO_BED_PATH_UP = file.path(OUTPUT_03, "promoters.hilo.up.bed")
PROMOTERS_HILO_BED_PATH_DOWN = file.path(OUTPUT_03, "promoters.hilo.down.bed")


write.table(promoters.hilo.up,
            PROMOTERS_HILO_BED_PATH_UP,
            quote=F,
            sep="\t",
row.names=F, col.names=F)

write.table(promoters.hilo.down,
            PROMOTERS_HILO_BED_PATH_DOWN,
            quote=F,
            sep="\t",
row.names=F, col.names=F)
```

To produce the deeptools output, execute DEEPTOOLS.bash.

It will compute promoters.hilo.mx and promoters.hilo.pdf.

Deeptools PDFs indicate a font called dejavu, if you're tired of replacing it in Illustrator, install it from: https://sourceforge.net/projects/dejavu/

```
Sys.setenv(UPSTREAM=UPSTREAM,
           DOWNSTREAM=DOWNSTREAM,
           INTERP_SIGNAL_BW=INTERP_SIGNAL_BW,
            PROMOTERS_HILO_BED_PATH=PROMOTERS_HILO_BED_PATH,
            PROMOTERS_HILO_BED_PATH_A=PROMOTERS_HILO_BED_PATH_A,
            PROMOTERS_HILO_BED_PATH_B=PROMOTERS_HILO_BED_PATH_B,
            PROMOTERS_HILO_BED_PATH_C=PROMOTERS_HILO_BED_PATH_C,
            PROMOTERS_HILO_BED_PATH_D=PROMOTERS_HILO_BED_PATH_D,
            PROMOTERS_HILO_BED_PATH_UP=PROMOTERS_HILO_BED_PATH_UP,
            PROMOTERS_HILO_BED_PATH_DOWN=PROMOTERS_HILO_BED_PATH_DOWN)
```

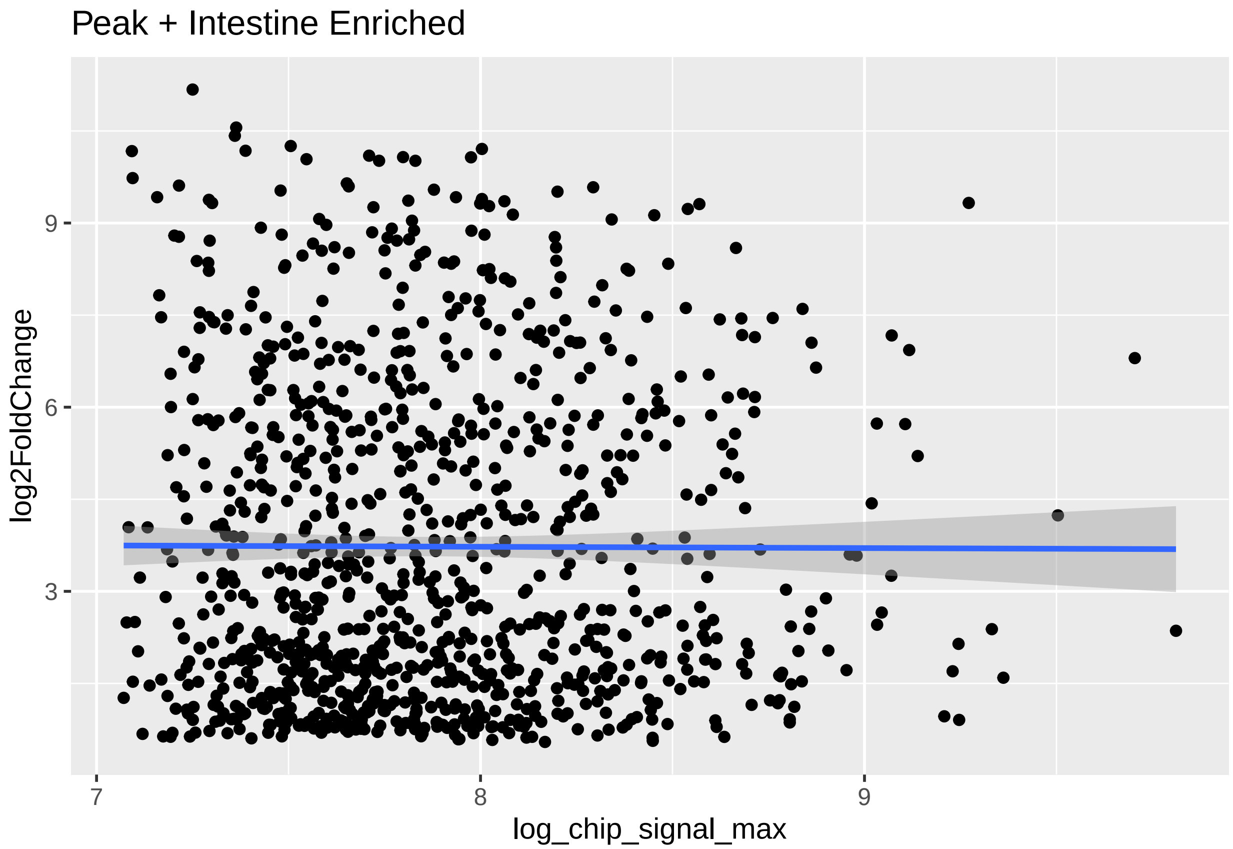
```
source $HOME/.bash_profile
conda activate derptools # yaml environ in 02_scripts/conda_envs

set -ue # exit 1 if any vars are not set (using Sys.setenv in prev chunks)
BODYLENGTH=$(($UPSTREAM + $DOWNSTREAM))

# real  1m59.354s
# user  3m47.980s
# sys   0m2.663s
time computeMatrix scale-regions --regionBodyLength $BODYLENGTH \
                                --startLabel 'up-1Kb' \
                                --endLabel down+200 \
                                --beforeRegionStartLength $UPSTREAM\
                                --afterRegionStartLength $DOWNSTREAM\
                                -R $PROMOTERS_HILO_BED_PATH_A $PROMOTERS_HILO_BED_PATH_B $PROMOTERS_HIL
                                -S $INTERP_SIGNAL_BW\
                                -p 4 -o promoters.olap100.hilo.mx

plotHeatmap  --matrixFile promoters.olap100.hilo.mx\
             -out promoters.olap100.hilo.pdf\
             --sortRegions no\
             --colorMap RdYlBu_r\
             --startLabel '' --endLabel ''\
             --regionsLabel 'peak+int. enrich.' 'peak+ NOT int. enrich.' 'NO peak + int. enrich.' 'NO p
             --samplesLabel 'ELT-2 signal (reps. combined subtracted)'
```

```
##
## real 2m13.865s
## user 4m10.511s
## sys  0m3.680s
```

```
source $HOME/.bash_profile
conda activate derptools # yaml environ in 02_scripts/conda_envs
BODYLENGTH=$(($UPSTREAM + $DOWNSTREAM))
set -ue # exit 1 if any vars are not set (using Sys.setenv in prev chunks)
time computeMatrix scale-regions --regionBodyLength $BODYLENGTH \
                            --startLabel 'up-1Kb' \
                            --endLabel down+200 \
                            --beforeRegionStartLength $UPSTREAM\
                            --afterRegionStartLength $DOWNSTREAM\
                            -R $PROMOTERS_HILO_BED_PATH_UP $PROMOTERS_HILO_BED_PATH_DOWN\
                            -S $INTERP_SIGNAL_BW\
                            -p 4 -o promoters.hilo.updown.mx

plotHeatmap  --matrixFile promoters.hilo.updown.mx\
             -out promoters.updown.pdf\
             --sortRegions no\
             --colorMap RdYlBu_r\
             --startLabel '' --endLabel ''\
             --regionsLabel 'log2FC > 0' 'log2FC < 0'\
             --samplesLabel 'ELT-2 signal (reps. combined subtracted)'
```

```
##
## real 1m43.253s
## user 3m25.571s
## sys  0m3.078s
```

```
gr.promoters.classA = gr.promoters[classA]

# scatter plot with linear mods on logFC up and down separately
gr.promoters.classA %>% as.data.frame() %>%
  ggplot(
    aes(x=log_chip_signal_max,
        y=log2FoldChange,
        group=log2FoldChange>0)) + geom_point() +
        geom_smooth(method='lm', formula= y~x) +
        ggtitle("Peak + Intestine Enriched")
```



```
classA.up = promoters.hilo %>% as.data.frame() %>% filter(classA & log2FoldChange > 0)
up.table = classA.up[,c('log2FoldChange',
                      'chip_signal_mean',
                      'chip_signal_max',
                      'log_chip_signal_mean',
                      'log_chip_signal_max',
                      'IDR_mean',   'IDR_max', 'IDR_value')]

cor.up.table = cor(up.table)
options(digits=3)
knitr::kable(cor.up.table, caption="Pairwise correlations")
```

Table 1: Pairwise correlations

	log2FoldChange	chip_signal_mean	chip_signal_max	log_chip_signal_mean	log_chip_signal_max	IDR_mean	IDR_max	IDR_value
log2FoldChange	1.000	-0.035	-0.002	-0.047	-0.004	0.002	- 0.003	0.041
chip_signal_mean	-0.035	1.000	0.918	0.983	0.903	0.892	0.896	0.705
chip_signal_max	-0.002	0.918	1.000	0.893	0.971	0.974	0.984	0.838
log_chip_signal_mean	-0.047	0.983	0.893	1.000	0.915	0.871	0.872	0.675
log_chip_signal_max	-0.004	0.903	0.971	0.915	1.000	0.951	0.957	0.794
IDR_mean	0.002	0.892	0.974	0.871	0.951	1.000	0.990	0.883
IDR_max	-0.003	0.896	0.984	0.872	0.957	0.990	1.000	0.854
IDR_value	0.041	0.705	0.838	0.675	0.794	0.883	0.854	1.000

```
cor.test(classA.up[,'log2FoldChange'],classA.up[,'IDR_mean'])
```

```
##
##  Pearson's product-moment correlation
##
## data:  classA.up[, "log2FoldChange"] and classA.up[, "IDR_mean"]
## t = 0.06, df = 1027, p-value = 1
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0593  0.0629
## sample estimates:
##     cor
## 0.00183
```

```
cor.test(classA.up[,'log2FoldChange'],classA.up[,'log_chip_signal_mean'])
```

```
##
##  Pearson's product-moment correlation
##
## data:  classA.up[, "log2FoldChange"] and classA.up[, "log_chip_signal_mean"]
## t = -1, df = 1027, p-value = 0.1
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1074  0.0146
## sample estimates:
##     cor
## -0.0466
```

```
Sys.setenv(PROMOTOR_BED_PATH=PROMOTOR_BED_PATH,
           NR_PROMOTOR_BED_PATH=NR_PROMOTOR_BED_PATH)
```

```
source $HOME/.bash_profile
conda activate elt-2-rev
wiggletools
```

```
## WiggleTools
##
## Copyright [1999-2017] EMBL-European Bioinformatics Institute
## Development contact: Daniel Zerbino zerbino@ebi.ac.uk
##
## Citation: Zerbino DR, Johnson N, Juettemann T, Wilder SP and Flicek PR: WiggleTools: parallel process
##
## This library parses wiggle files and executes various operations on them streaming through lazy evalu
##
## Inputs:
##  The program takes in Wig, BigWig, BedGraph, Bed, BigBed, Bam, VCF, and BCF files, which are disting
##  Note that wiggletools assumes that every bam file has an index .bai file next to it.
##
## Outputs:
##  The program outputs a wiggle file in stdout unless the output is squashed
##
## Command line:
##  wiggletools --help
##  wiggletools program
##
```

```
## Program grammar:
##  program = (iterator) | do (iterator) | (extraction) | (statistic) | run (file)
##  iterator = (in_filename) | (unary_operator) (iterator) | (binary_operator) (iterator) (iterator) |
##  unary_operator = unit | coverage | write (output) | write_bg (ouput) | smooth (int) | abs | exp | ln
##  output = (out_filename) | -
##  in_filename = *.wig | *.bw | *.bed | *.bb | *.bg | *.sam | *.bam | *.cram | read_count *.sam | read_
##  statistic = (statistic_function) (iterator) | ndpearson (multiplex) (multiplex)
##  statistic_function = AUC | meanI | varI | minI | maxI | stddevI | CVI | energy (wavelength) | pears
##  binary_operator = diff | ratio | overlaps | trim | noverlaps | nearest | apply (statistic) | fillIn
##  reducer = cat | sum | product | mean | var | stddev | entropy | CV | median | min | max
##  setComparison = ttest | ftest | wilcoxon
##  multiplex_list = (multiplex) | (multiplex) : (multiplex_list)
##  multiplex = (iterator_list) | map (unary_operator) (multiplex) | strict (multiplex)
##  iterator_list = (iterator) | (iterator) : (iterator_list)
##  extraction = profile (output) (int) (iterator) (iterator) | profiles (output) (int) (iterator) (ite
##      | apply_paste (out_filename) (statistic) (bed_file) (iterator)
```