relDistToTSSkmeans

RAC

18/05/2020

```
#refined Function finds relative distance of xlinksite across feature. RNA split into 20 bins in total.
#Also genes are annotated with mature and gene size.

process_distToTSS_1 <- function(dfFilePath, annotationFilePath){


df<-read.csv(dfFilePath, sep = "\t", header =F)
col.names<-c("Sample","chrAnno","startAnno", "stopAnno", "IDAnno", "scoreAnno", "strandAnno","chrRead",
col.numeric<- c("chrAnno","startAnno","stopAnno","totalExons","exonSize","cumSumExons","distToTSS","sta

print("wrangling data")

df1<-df %>%
    setNames(col.names) %>%
      select(-chrRead, -IDRead, -scoreRead, -scoreAnno) %>%
      separate(IDAnno, c("geneName","biotype","exonID", "totalExons", "exonSize","cumSumExons", "distTo
      select(-exonDesc)

df1[col.numeric] <- sapply(df1[col.numeric],as.numeric)

print("processing rel distance from TSS")
df2<-df1 %>%
    mutate(rel.pos = ifelse(strandAnno == "+", ( (startRead-startAnno) + (cumSumExons-exonSize) ),
                     ifelse(strandAnno == "-", ( (stopAnno-stopRead) + (cumSumExons-exonSize) ),
                     "no"))) %>%
    select(Sample, geneName, biotype, geneDesc, rel.pos)


print("loading annotation file")

annoDF<-read.csv(annoFilePath , sep = "\t", header =F)

print("making table with total sizes of RNAs")

totalSizes<-annoDF %>%
    setNames(c("chr","start","stop","ID","score","strand")) %>%
    separate(ID, c("geneName","biotype","exonID", "totalExons", "exonSize","cumSumExons", "distToTSS",
    filter(exonID == totalExons) %>%
    mutate(matureRNA = as.numeric(cumSumExons),
           geneSize = as.numeric(ifelse(totalExons > 1, distToTSS, exonSize))) %>%
    select(geneName, biotype, matureRNA, geneSize)
```

```
df3<-df2 %>%
  left_join(totalSizes)

return(df3)

}
```

```
ScaledNormalise_2_3_RPM_Normalisation <-function(DF, MINSIZE, MAXSIZE, RPM_factor){

annoDF<-read.csv(annoFilePath, sep = "\t", header =F)

totalSizes<-annoDF %>%
  setNames(c("chr","start","stop","ID","score","strand")) %>%
  separate(ID, c("geneName","biotype","exonID", "totalExons", "exonSize","cumSumExons", "distToTSS", "ex
  filter(exonID == totalExons) %>%
  mutate(matureRNA = as.numeric(cumSumExons),
         geneSize = as.numeric(ifelse(totalExons > 1, distToTSS, exonSize))) %>%
  select(geneName, biotype, matureRNA, geneSize)

calculated_test<-relDist_ALYREF %>%
  left_join(totalSizes) %>%
  mutate(rel.pos   = as.numeric(rel.pos),
         matureRNA = as.numeric(matureRNA)) %>%
  filter(matureRNA %in% c(MINSIZE:MAXSIZE)) %>%
  mutate(rel.pos.2 = rel.pos/matureRNA)


#assigning bin numbers to the rel.pos.2

ALL <-calculated_test %>% mutate(rel.pos.2 = ifelse(rel.pos.2 == 0, rel.pos.2 + 0.0000001, rel.pos.2))
max(ALL$rel.pos.2)
min(ALL$rel.pos.2)
ALL$new_distBin <- cut(ALL$rel.pos.2, seq(0, 1, by=0.01), labels=seq("1", "100", by = 1 ) )


combined<-ALL %>%
    separate(Sample, into=c("Sample", "Processing", "readType"), sep="\\.") %>%
    #filter(Timepoint ==  "DMSO" & grepl("coding|histone",biotype) & !grepl("\\*|non", biotype)) %>%
    group_by(geneName, biotype, Sample, matureRNA, geneSize, new_distBin) %>%
    summarise(tally = n()) %>%
  #new steo to normalise to gene expression level
    #left_join(expression_vector) %>%
    #mutate(tally_n = (tally/ctrl_RNAseq_expr)/(matureRNA/100)) #step to normalise count to bin size.
    mutate(tally_n = tally/(matureRNA/100)) #step to normalise count to bin size.

  #  mutate(tally_n = tally/(matureRNA/100)) #step to normalise count to bin size.

totalLibrarySizes<-read.table(RPM_factor) %>%
  setNames(c("Sample","RPMFactor") )%>%
  mutate(RPMFactor= as.numeric(RPMFactor),
         RPMFactor = 1000000/RPMFactor)

totalLibrarySizes$Sample <- gsub("-", "_", totalLibrarySizes$Sample)
```

```
#this step taken from norm profile 3
combined_2<-combined %>%
  left_join(totalLibrarySizes) %>%
  mutate(tally_n = tally_n * RPMFactor) %>%
  select(-RPMFactor) %>%
  group_by(geneName, biotype, Sample, matureRNA) %>%
  mutate(totals_unNorm    = sum(tally),
         pct              = tally_n/sum(tally_n)*100) %>%
  ungroup() %>%
  select(Sample, matureRNA, geneName, biotype, new_distBin, totals_unNorm, pct) %>%
  spread(new_distBin, pct, fill = 0) %>%
  gather("distBin", "value", c('1':'100') )%>%
  mutate(distBin = factor(distBin, levels = c(1:100)))

return(combined_2)
}
```

#Process data in the same way that the tiCLIP count files for ALYREF were.

```
annoDF<-read.csv(annoFilePath, sep = "\t", header =F)

geneStructures<-annoDF %>%
  setNames(c("chr","start","stop","ID","score","strand")) %>%
  separate(ID, c("geneName","biotype","exonID", "totalExons", "exonSize","cumSumExons", "distToTSS", "ex
  separate(geneDesc, into = c("geneStructure"), sep = "-") %>%
  select(geneName, geneStructure) %>%
  unique()
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 117503 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 29, ...].
```

```
totalSizes<-annoDF %>%
    setNames(c("chr","start","stop","ID","score","strand")) %>%
    separate(ID, c("geneName","biotype","exonID", "totalExons", "exonSize","cumSumExons", "distToTSS",
    filter(exonID == totalExons) %>%
    mutate(matureRNA = as.numeric(cumSumExons),
           geneSize = as.numeric(ifelse(totalExons > 1, distToTSS, exonSize))) %>%
    select(geneName, biotype, matureRNA, geneSize)

relDist_ALYREF<-process_distToTSS_1(dfFilePath,annoFilePath )
```

```
## [1] "wrangling data"
## [1] "processing rel distance from TSS"
## [1] "loading annotation file"
## [1] "making table with total sizes of RNAs"
```

```
## Joining, by = c("geneName", "biotype")
```

```
ALYREF_100bins_scaled<-ScaledNormalise_2_3_RPM_Normalisation(relDist_ALYREF,200,100000, "../../data/Che
```

```
## Joining, by = c("geneName", "biotype", "matureRNA", "geneSize")
## `summarise()` has grouped output by 'geneName', 'biotype', 'Sample',
## 'matureRNA', 'geneSize'. You can override using the `.groups` argument.
## Joining, by = "Sample"
```

#Supplemental Figure 4 E

```
clusterID_filepath="../../data/ALYREF_kmeansClustering_bonefide_gene_namelist.tab"

clusterIDs<-read.table(clusterID_filepath, header = T)

ALYREF_100bins_scaled_labelled<-
ALYREF_100bins_scaled %>%
    mutate(experimentalGroups = case_when(
    grepl("CLIP_ALYREF", Sample) ~ "no KD",
    grepl("CLIP_no_antibody", Sample) ~ "no KD",
    grepl("CLIP_in_CBP80_KD_cells_Homo_sapiens", Sample) ~ "CBP80 KD",
    grepl("CLIP_in_PABPN1_KD_cells_Homo_sapiens", Sample) ~ "PABPN1 KD",
    grepl("CLIP_in_CstF64",Sample) ~ "CstF64 KD",
    grepl("CLIP_in_Cntl_cells_rep", Sample) ~ "CstF64 KD"
  )) %>%
  mutate(experimentalSample= case_when(
    grepl("Cntl|no_antibody", Sample) ~ "Control", TRUE ~ "CLIP"
  ))

for_graph<-
ALYREF_100bins_scaled_labelled %>%
filter(experimentalSample == "CLIP") %>%
  filter(grepl("coding",biotype) &
           !grepl("\\*|non", biotype) &
           totals_unNorm >= 20) %>%
  left_join(clusterIDs) %>%
  spread(distBin, value) %>%
  gather("distBin", "value", '1':'100')
```

```
## Joining, by = "geneName"
```

```
df_common_genes<-
for_graph %>%
  filter(!is.na(clusterID) & !grepl("CstF64", Sample)) %>%
  select(Sample, geneName,clusterID) %>%
  unique() %>%
  select(geneName) %>%
  group_by(geneName) %>%
  summarise(n = n()) %>%
  filter(n > 4) %>%
  select(-n)

df_common_genes %>%
  left_join(clusterIDs) %>%
  group_by(clusterID) %>%
  summarise(n = n())
```

```
## Joining, by = "geneName"
```

```
## # A tibble: 2 x 2
##   clusterID     n
##       <int> <int>
## 1         1   273
## 2         2   642
```

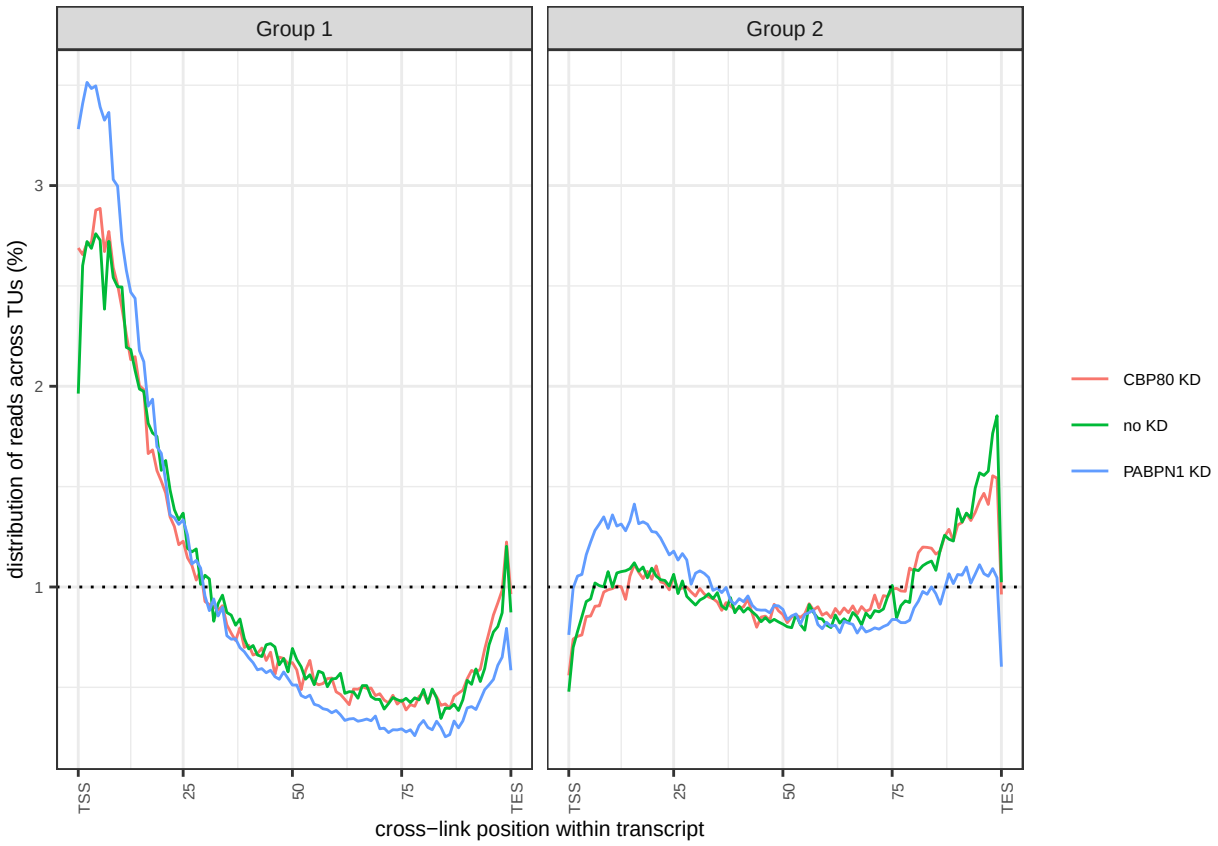
```
for_graph_mean<-
for_graph %>%
```

```
  filter(clusterID %in% c(1:2)) %>%
  right_join(df_common_genes) %>% #selects for genes common to all CLIPs
  group_by(clusterID, Sample, experimentalGroups, distBin, experimentalSample) %>%
  summarise(value = mean(value)) %>%
  mutate(clusterID = gsub("1", "Group 1", gsub("2","Group 2", clusterID )),
         ID = "Shi et al., 2017") %>%
  ungroup()
```

```
## Joining, by = "geneName"
## `summarise()` has grouped output by 'clusterID', 'Sample',
## 'experimentalGroups', 'distBin'. You can override using the `.groups` argument.
```

```
for_graph_mean %>%
  filter(!grepl("CstF64", Sample)) %>%
  ggplot(aes(x=as.numeric(distBin), y=value, col = as.factor(experimentalGroups), group = experimentalG
  stat_summary(fun = mean, geom = "line", size = 0.5) +
  geom_hline(yintercept = 1, linetype = "dotted") +

  scale_x_continuous(breaks = c(1,25,50,75,100),
                     labels = c("TSS",25,50,75,"TES")) +
  facet_grid(. ~ clusterID , scales = "free_y") +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1),
    text = element_text(size = 8),
    legend.position = "right",
    strip.text.x = element_text(size = 8),
    strip.text.y = element_text(size = 8)
  ) +
  ylab("distribution of reads across TUs (%)") +
  labs(col = "") +
  xlab("cross-link position within transcript")
```


Group 1
Group 2
distribution of reads across TUs (%)
3
2
1
TSS
25
50
75
TES
cross−link position within transcript
CBP80 KD
no KD
PABPN1 KD