Human Bi-Directional Promoters

Bob Horton

Bidirectional promoters are common in the human genome, and we would like to see what we can learn about them by exploring an annotation table listing human genes and their transcription start sites. It is very easy to download this type of data from the UCSC database, since they have made their MySql host available over the Internet. Here is the command I used to grab the entire 'refGene' table from the UCSC hg19 assembly using the stand-alone mysql client:

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e "SELECT * FROM refGene" hg19 >
hg19_refgene.txt
```

First we load the table into R, and list the column names to get an idea of what this table can tell us:

```
library(sqldf)
```

```
## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite
## Loading required package: DBI
## Loading required package: RSQLite.extfuns
```

```
hg19_refgene <- read.delim("hg19_refgene.txt", comment.char="#", stringsAsFactors=F)
colnames(hg19_refgene)
```

```
##  [1] "bin"          "name"         "chrom"        "strand"
##  [5] "txStart"      "txEnd"        "cdsStart"     "cdsEnd"
##  [9] "exonCount"    "exonStarts"   "exonEnds"     "score"
## [13] "name2"        "cdsStartStat" "cdsEndStat"   "exonFrames"
```

This table contains a lot of information about exons and coding sequences, but we are really only interested in transcription start sites, which is the same as txStart on the '+' strand, and txEnd on the '-' strand. We can therefore simplify the table a bit. Here we do three simplification steps: first, we use SQL to select only certain columns from the big table. In this step, we also change the name of the 'name2' column to 'symbol'. The resulting table is saved as hg19. Second, we add a new column to represent the transcription start site; on the positive strand, that is the left end of the range (txStart), and on the negative strand (any gene not on the top strand) it is the right end (txEnd). After that we can simplify again, since we don't need the txStart and txEnd columns any more. For variety, we use R code this time to specify the columns we want to keep instead of SQL.

```
hg19 <- sqldf("SELECT name, chrom, strand, txStart, txEnd, name2 as symbol FROM hg19_refgene")
```

```
## Loading required package: tcltk
```

```
hg19$tss <- ifelse(hg19$strand == '+', hg19$txStart, hg19$txEnd)
hg19 <- hg19[, c("name", "chrom", "strand", "symbol", "tss")]

head(hg19, n=10)
```

```
##            name chrom strand  symbol       tss
## 1     NM_032291  chr1      +   SGIP1  66999824
## 2     NM_052998  chr1      +     ADC  33546713
## 3  NM_001080397  chr1      + SLC45A1   8384389
## 4     NM_013943  chr1      +   CLIC4  25071759
## 5     NM_032785  chr1      -   AGBL4  50489626
## 6     NM_018090  chr1      +  NECAP2  16767166
## 7  NM_001145278  chr1      +  NECAP2  16767166
## 8  NM_001145277  chr1      +  NECAP2  16767166
## 9     NM_001918  chr1      -     DBT 100715409
## 10    NM_003243  chr1      -  TGFBR3  92351836
```

Note that some of the gene symbols are repeated, with multiple transcript names (isoforms) coming from a single start site. We can use SQL to group them by chromosome, strand, and start site, then rename them by concatenating a group of symbols together into a comma-separated list. The *DISTINCT* keyword keeps any given symbol from being repeated more than once. This new consolidated table replaces the old hg19 table. To see some examples of TSSs with multiple gene symbols, we can search for symbols with commas (here collected into a new table called *multiSymbols*).

```
hg19 <- sqldf("SELECT group_concat(DISTINCT name) as name,
                group_concat(DISTINCT symbol) as symbol,
                chrom, strand, tss
              FROM hg19 GROUP BY chrom, strand, tss")

multiSymbols <- hg19[grep(',',hg19$symbol),]
print(head(multiSymbols), row.names=F)
```

```
##                                      name
##               NR_028327,NR_028325,NR_028322
##      NM_001005221,NM_001005277,NM_001005224
##  NM_199006,NM_198544,NM_199294,NM_001270517
##            NR_037187,NR_036462,NM_001243768
##                         NR_003022,NR_003025
##                   NM_001010890,NM_001098376
##                                    symbol chrom strand      tss
##  LOC100133331,LOC100132062,LOC100132287  chr1      +   323891
##                      OR4F29,OR4F16,OR4F3  chr1      +   367658
##                       APITD1-CORT,APITD1  chr1      + 10490158
##                       APITD1-CORT,APITD1  chr1      + 10490803
##                        SNORA59B,SNORA59A  chr1      + 12567299
##                          PRAMEF9,PRAMEF15  chr1      + 13421175
```

Only the first few lines are shown above; the *multiSymbols* table shows a total of 517 transcription start sites that are annotated with multiple symbols.

Now we use SQL to find pairs of start sites on opposite strands, within a certain distance of one another. This query uses sub-select statements to pull out the start sites on the positive and negative strands, then looks for sites on one strand within a certain distance of a start site on the other strand. Here we use 1000 bp, which is the distance set in [Trinklein 2004].

```
bdp <- sqldf(
    "SELECT pos.chrom as chrom, pos.tss as pos_tss, neg.tss as neg_tss,
        pos.symbol as pos_gene, neg.symbol as neg_gene,
```

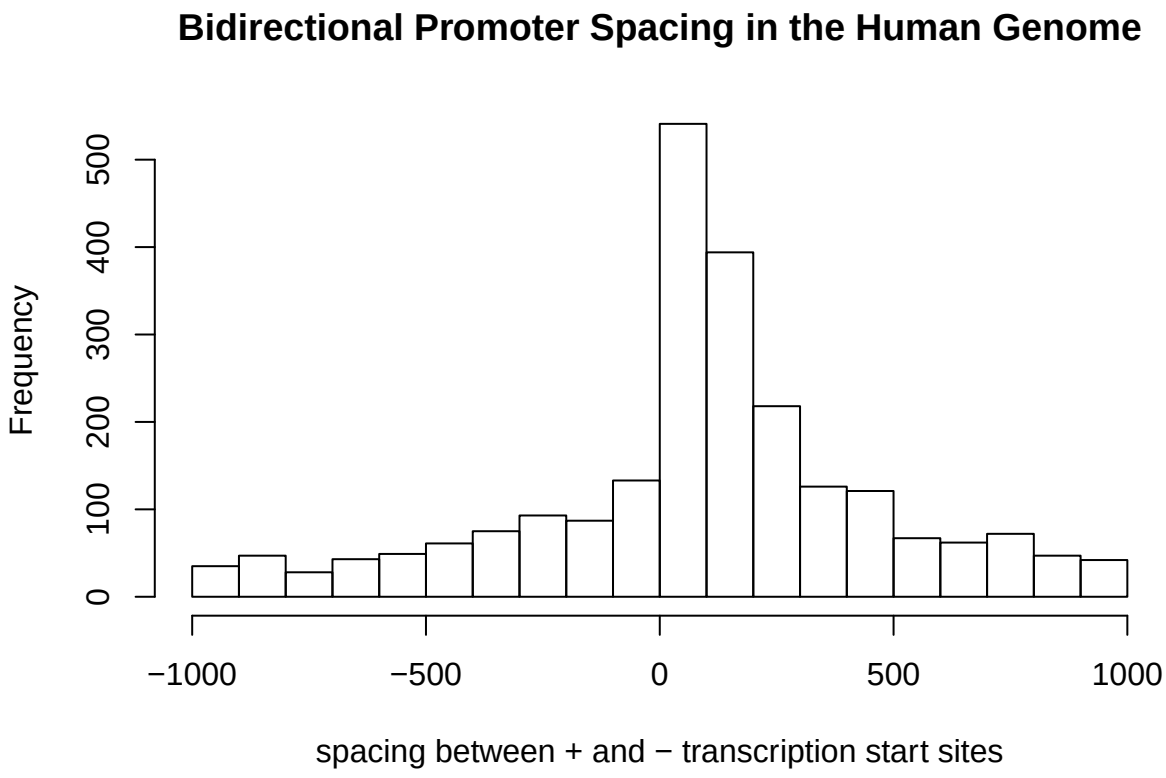
```
        (pos.tss - neg.tss) as spacing
    FROM  (SELECT * from hg19 WHERE strand = '+') AS pos,
          (SELECT * from hg19 WHERE strand = '-') AS neg
        WHERE pos.chrom = neg.chrom
        AND   abs(neg.tss - pos.tss) < 1000")

head(bdp)
```

```
##   chrom pos_tss neg_tss  pos_gene  neg_gene spacing
## 1  chr1  762970  762902 LOC643837 LINC00115      68
## 2  chr1  763177  762902 LOC643837 LINC00115     275
## 3  chr1 1167628 1167447   B3GALT6      SDF4     181
## 4  chr1 1243993 1243269     PUSL1     ACAP3     724
## 5  chr1 1260142 1260067    GLTPD1    CPSF3L      75
## 6  chr1 1334909 1334718 LOC148413     CCNL2     191
```

With this table we can get a general overview of how the start sites are spaced in bidirectional promoters:

```
hist(bdp$spacing, breaks=21, main="Bidirectional Promoter Spacing in the Human Genome",
     xlab="spacing between + and - transcription start sites")
```



Note that if a bidirectional promoter is defined as the region between the oppositely oriented promoters, then we can't include pairs with negative spacing.

Percentage of human promoters that are bidirectional

```
num_bidir <- nrow(bdp[bdp$spacing > 0,])
total_tss <- nrow(hg19)
```

According to our criteria, there are 1690 bidirectional gene pairs in the human genome, accounting for 11.2655% of all the promoters in the human genome. The paper by [Trinklein 2004] reported 1352 pairs of bidirectional promoters (11% of all genes annotated at that time).

Exercises:

- Given a list of gene symbols, use an asterisk to mark the ones with bidirectional promoters.
- Search the "bdp" table to see if you can find the pairs of bidirectional reporter mentioned in the introduction to [Adachi 2002]. Have any of the gene symbols changed?
- Use SQL queries on these tables to reproduce some of the promotor-spacing histograms from [Trinklein 2004].
- Generate a new bdp table, allowing up to 10 kb between start sites. Generate a histogram of promoter spacing. Filter this new table to show only those promoters with spacing of +/-1kb; is this table the same as the original? Why might pairs with negative spacing be underrepresented?
- Repeat this analysis with the mouse using the refGene table from the mm9 assembly: `mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e "SELECT * FROM refGene" mm9 > mm9_refgene.txt`

References

- Trinklein ND, Aldred SF, Hartman SJ, Schroeder DI, Otillar RP, Myers RM. An abundance of bidirectional promoters in the human genome. Genome Res. 2004 14(1):62-6. PubMed
- Adachi N, Lieber MR. Bidirectional gene organization: a common architectural feature of the human genome. Cell. 2002 Jun 28;109(7):807-9. PubMed.
- Bidirectional promoters in Wikipedia.