Working with nucleosome-derived data (MNase-Seq)

Tobias Straub

2020-05-05

Alignment of MNaseSeq-Data

Paired-end data is aligned as follows using bowtie2.

- only concordant reads are reported
- maximum insert size is kept to 250 (a bit larger than nucleosomes)
- multiple matches of to the genome are suppressed, orphaned mates are eliminated
- BAM file is then converted to BED in which chr, start, end of the sequenced fragement is reported

Cleaning up of orphaned reads is performed with this script taken from (https://www.biostars.org/p/95929/)

```
#!/usr/bin/env python
import csv
import sys

f = csv.reader(sys.stdin, dialect="excel-tab")
of = csv.writer(sys.stdout, dialect="excel-tab")
last_read = None
for line in f :
    #take care of the header
    if(line[0][0] == "@") :
        of.writerow(line)
        continue

    if(last_read == None) :
        last_read = line
    else :
        if(last_read[0] == line[0]) :
            of.writerow(last_read)
            of.writerow(line)
            last_read = None
        else :
            last_read = line
```

```
BOWTIE_INDEX= <dir>/Bowtie2Index/genome
BOWTIE_OPTS="-p 10 -X 250 --no-discordant --no-mixed --no-unal"
bowtie2 $BOWTIE_OPTS -x $BOWTIE_INDEX -1 mate_1.fastq.gz -2 mate_2.fastq.gz > aligned.sam
samtools view -hf 0x2 aligned.sam | grep -v "XS:i:" | filter_orphans.py | samtools view -b -o al
samtools sort -n -m 1G -@ 8 -o aligned.s.bam aligned.bam
bamToBed -i aligned.s.bam -bedpe > aligned.bed 2>/dev/null
cut -f 1,2,6 aligned.bed > aligned.s.bed
```

Single-read data is aligned as follows using bowtie2.

- multiple matches of to the genome are suppressed
- BAM file is then converted to BED in which chr, start, end and strand of the read is reported

```
BOWTIE_INDEX= <dir>/Bowtie2Index/genome
BOWTIE_OPTS="-p 10 --no-unal"
bowtie2 $bowtie_opts -x $BOWTIE_INDEX ${SRUN}.fastq.gz > aligned.sam
samtools view -h aligned.sam | grep -v "XS:i:" | samtools view -b -o aligned.bam
samtools sort -m 1G -@ 8 -o aligned.s.bam aligned.bam
bamToBed -i aligned.s.bam > aligned.bed 2>/dev/null
cut -f 1,2,3,6 aligned.bed > aligned.s.bed
```

Conversion to coverage object

```
library(tsTools)
library(IRanges)
```

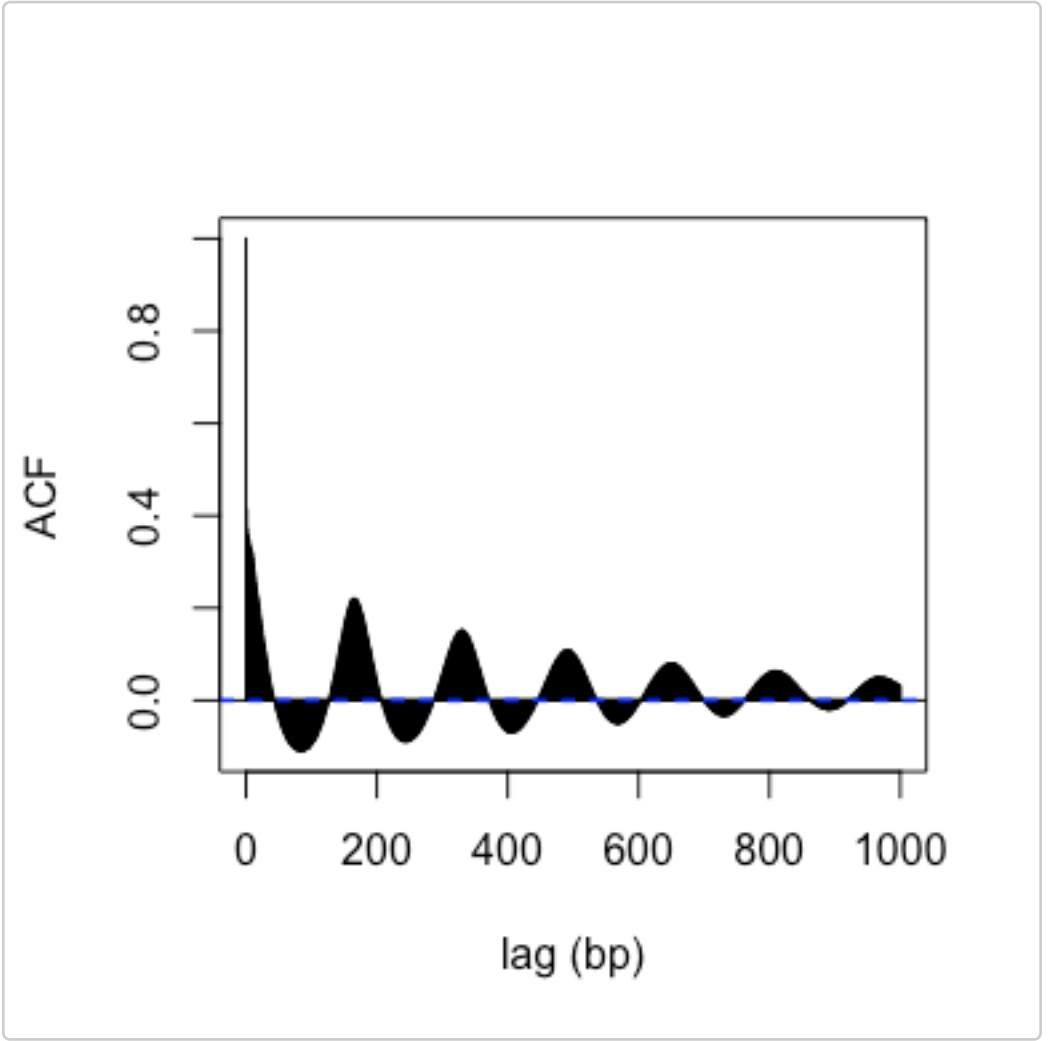
As an example we load the data included in the package (S. cerevisiae MNase-Seq, subset to chromosome IV)

```
fpath <- system.file("extdata", "SRR2154281_IV.s.bed", package="tsTools")
cov <- bed2dyad(fpath,"PAIRED",1)
```

Autocorrelation Function

The autocorrelation function on the signal vector can provide a hint as to whether we have something nucleosomal in the data. Given the beads on a string arrangement of nucleosomes we expect a periodic autocorrelation pattern.

```
acf(as.vector(cov[['IV']]), lag.max = 1000, main="", xlab="lag (bp)")
```

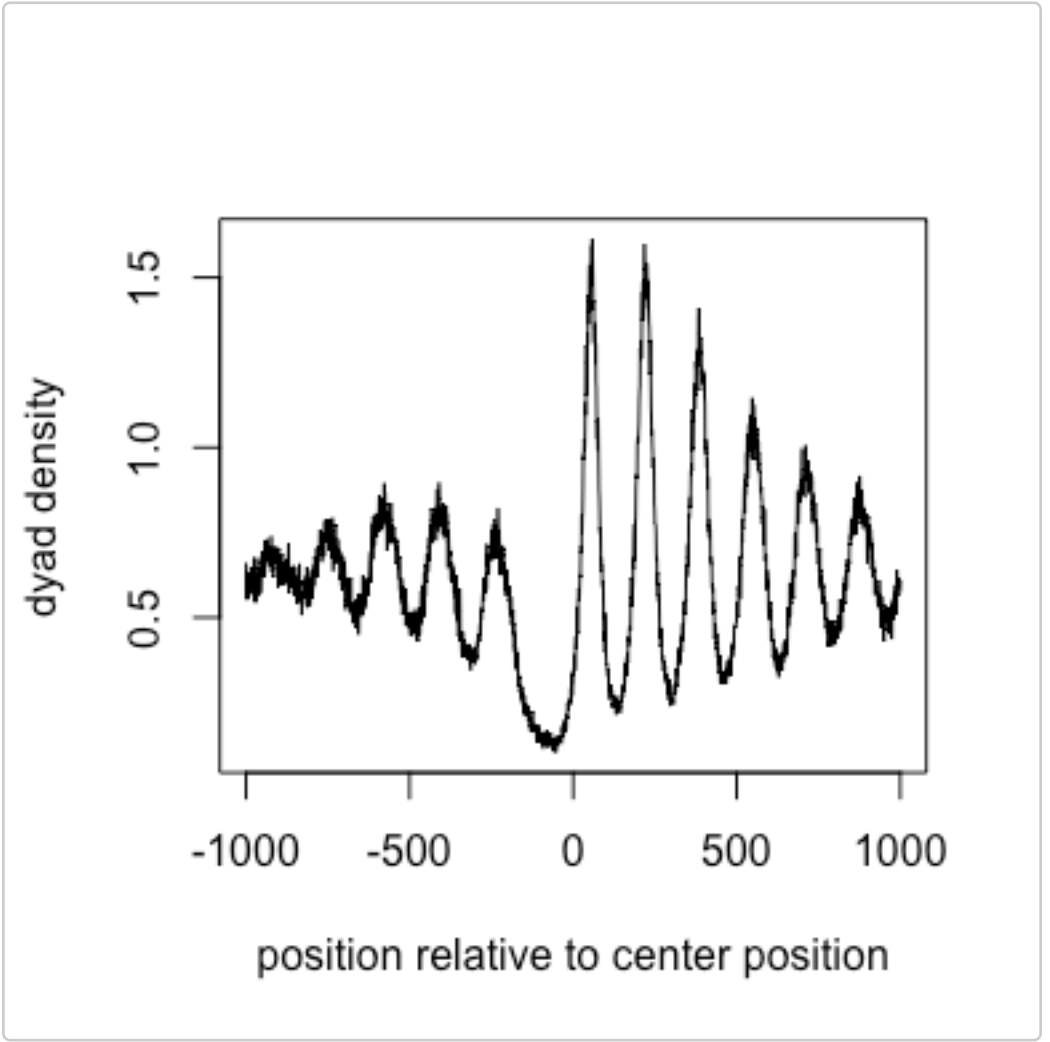


Cumulative Plots

```
data(ann)
head(ann)
```

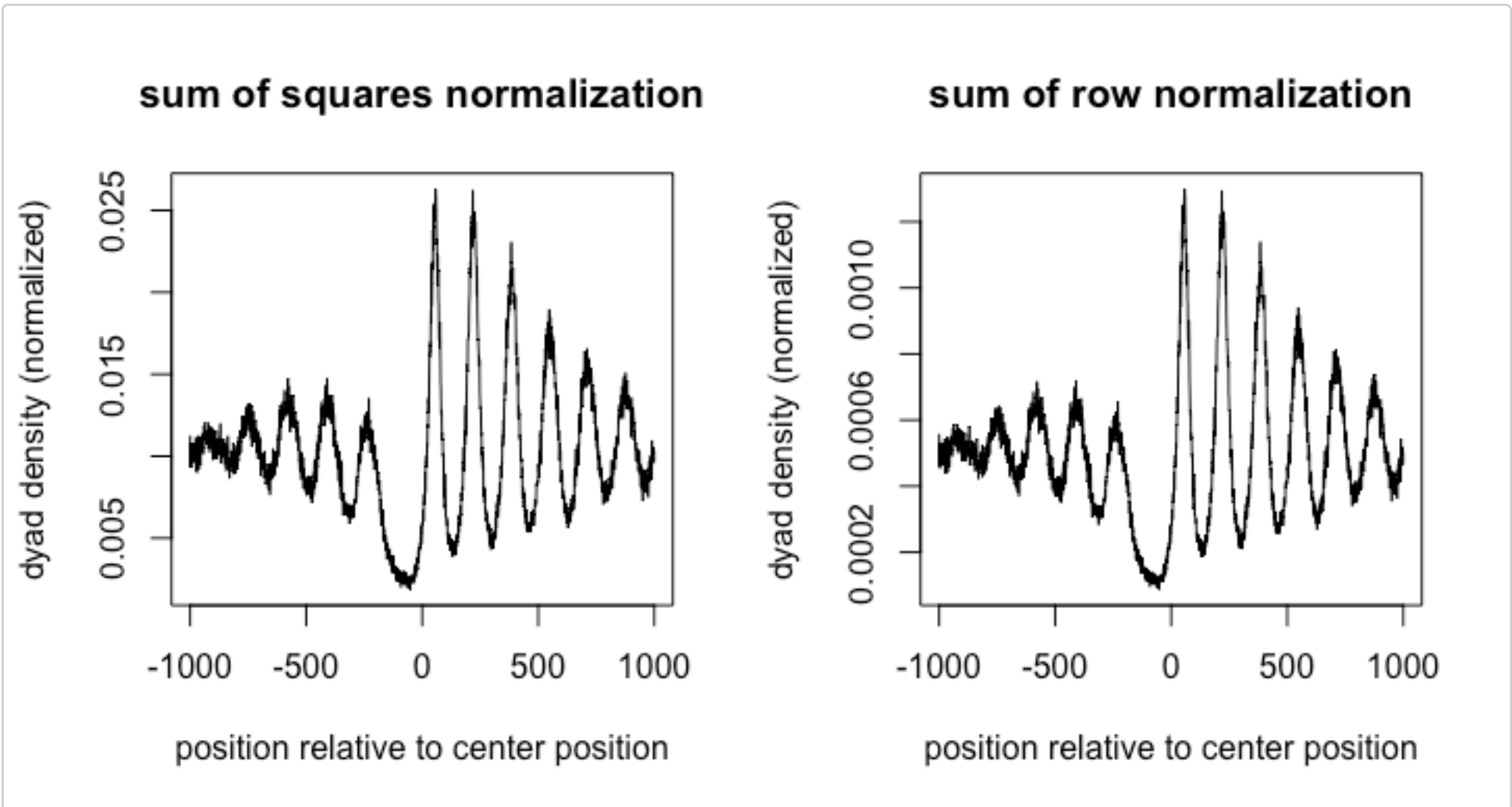
```
##        chr start   end strand class    name commonName  endConfidence    source
## ST3634   I  5074  6237      -   SUT  SUT432     SUT432 bothEndsMapped    Manual
## ST3635   I  7275  9260      -   ORF YAL067C       SEO1 bothEndsMapped    Manual
## ST0001   I  9367  9600      +   SUT  SUT001     SUT001 bothEndsMapped    Manual
## ST3636   I 10731 11140      -   CUT  CUT436     CUT436 bothEndsMapped Automatic
## ST3637   I 28082 29772      -   SUT  SUT433     SUT433 bothEndsMapped    Manual
## ST0002   I 30071 30904      +   CUT  CUT001     CUT001 bothEndsMapped Automatic
```

```
ann <- ann[ann$chr=="IV",]
centers <- data.frame(chr=ann$chr,
                      center=ifelse(ann$strand=="+", ann$start, ann$end),
                      strand=ann$strand)
rownames(centers) <- rownames(ann)
mat <- coverageWindowsCenteredStranded(centers, window.size=2000, cov)
x <- seq(-1000,1000)
plot(x, apply(mat, 2, mean), type="l",
     xlab="position relative to center position", ylab="dyad density")
```



row-wise normalization

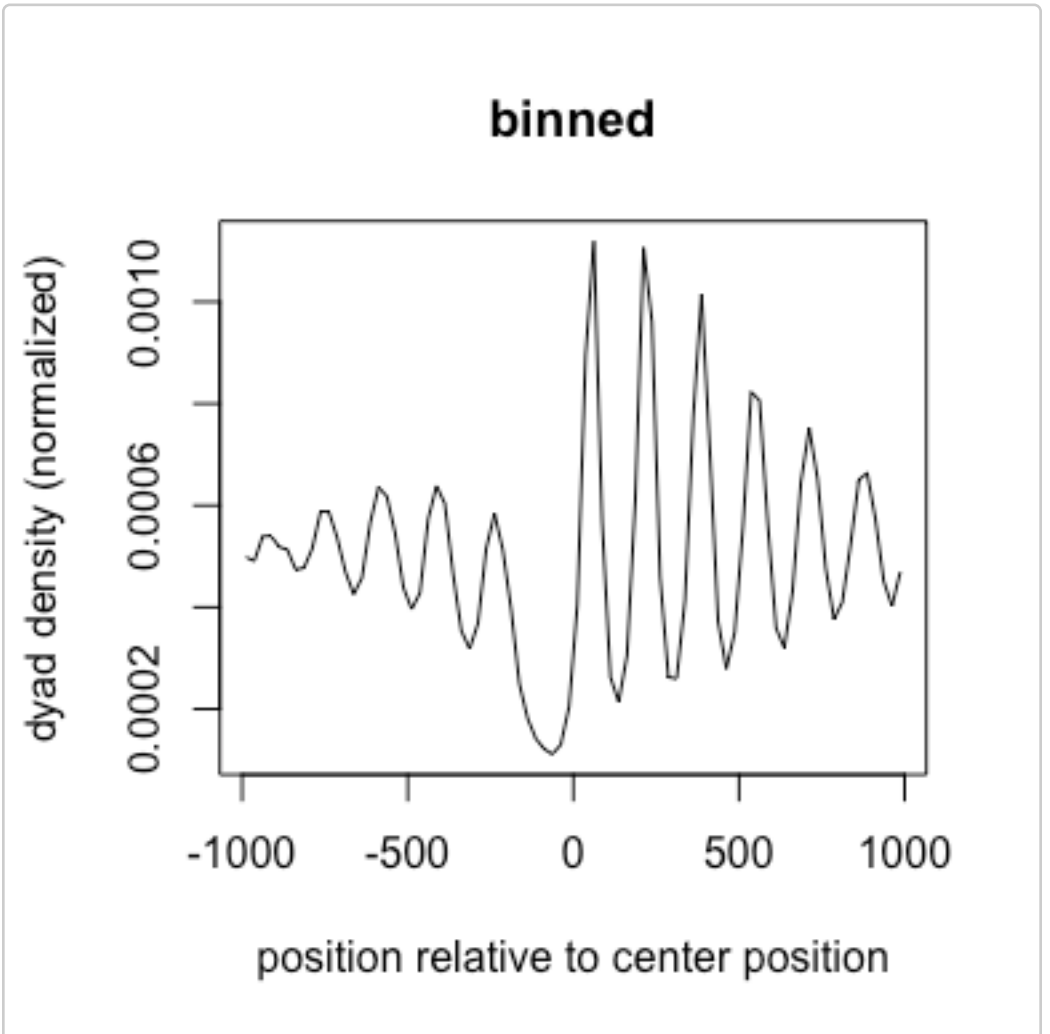
```
mat.sq <- norm.square(mat)
mat.sum <- norm.sum(mat)

par(mfrow=c(1,2))
plot(x, apply(mat.sq, 2, mean), type="l",
     xlab="position relative to center position", ylab="dyad density (normalized)", main="sum of
plot(x, apply(mat.sum, 2, mean), type="l",
     xlab="position relative to center position", ylab="dyad density (normalized)", main="sum of
```



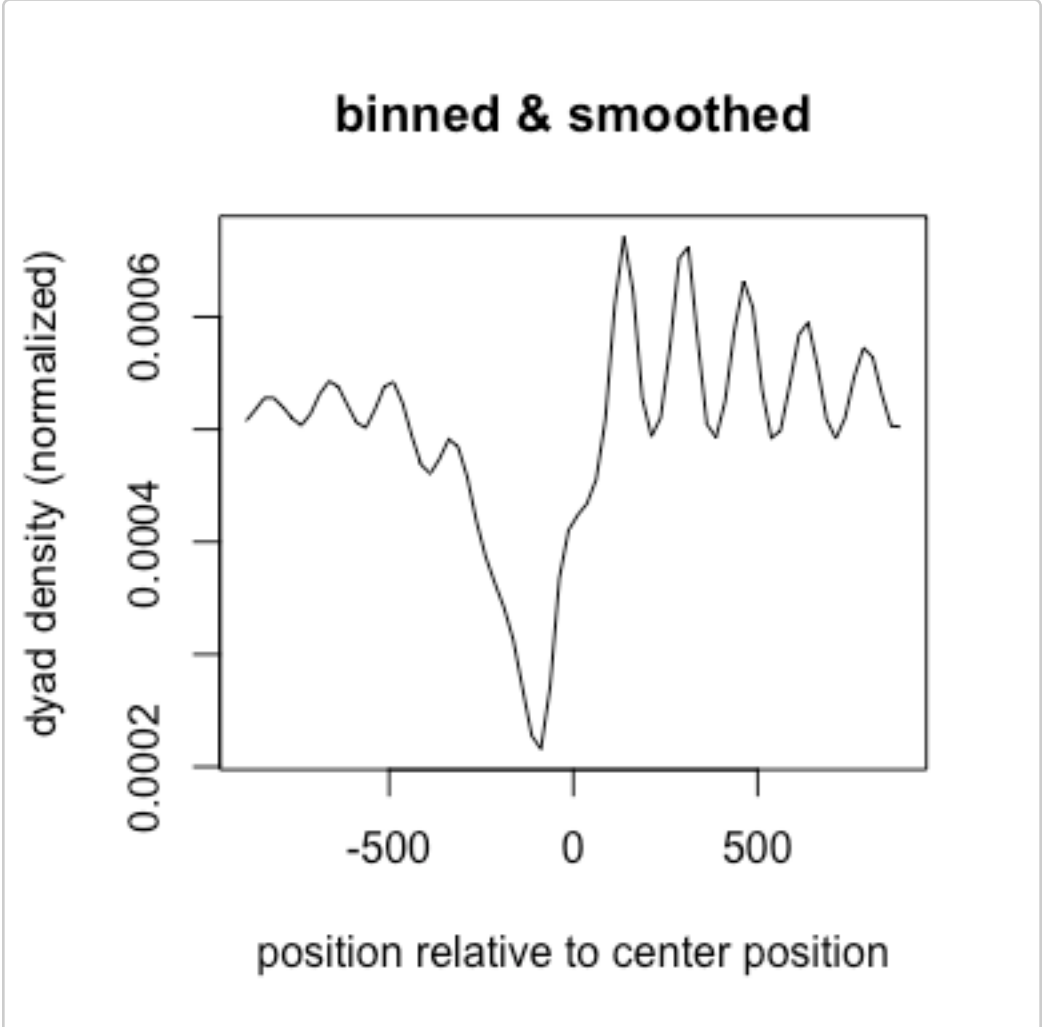

```
mat.bin <- bin.matrix.rows(mat.sum,25)
x <- as.integer(colnames(mat.bin))
plot(x, apply(mat.bin, 2, mean), type="l",
     xlab="position relative to center position", ylab="dyad density (normalized)", main="binned
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

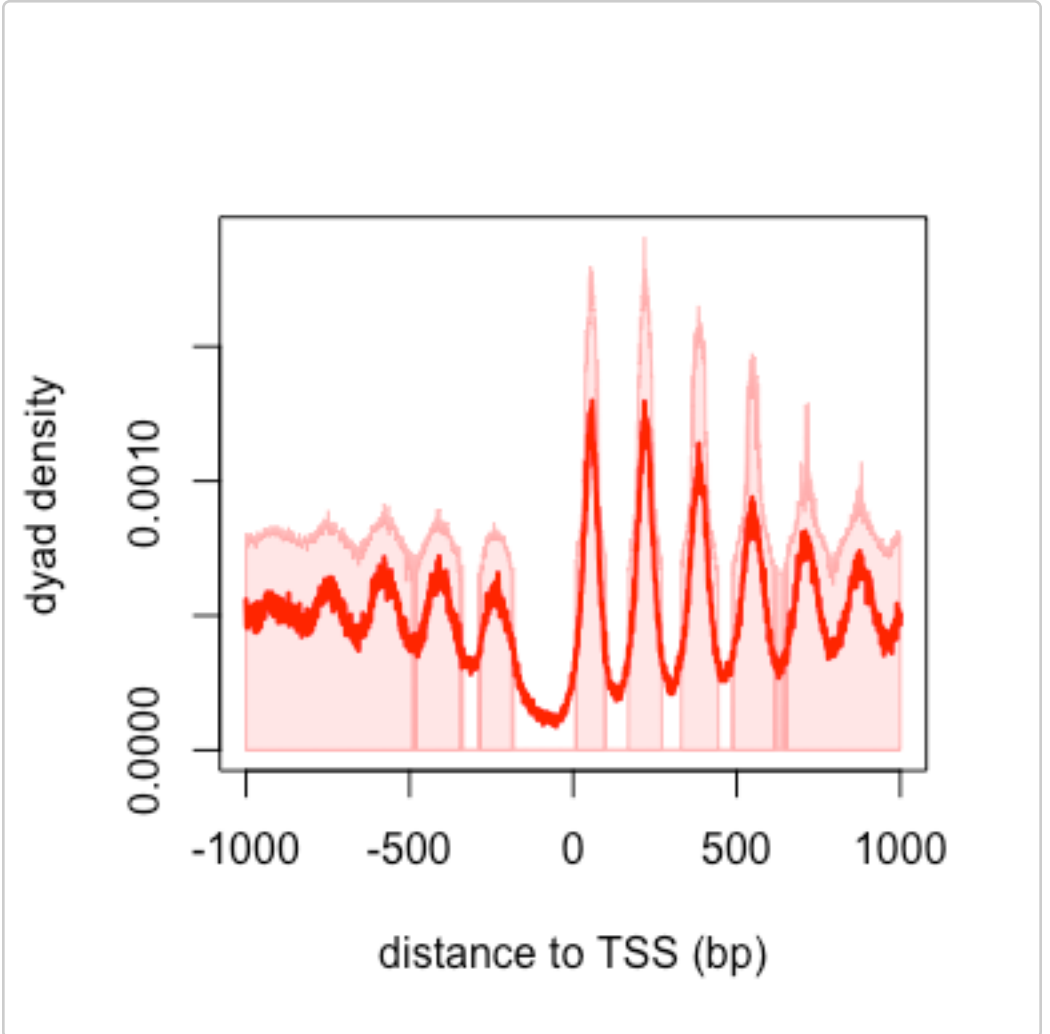



```
mat.bin.smoothed <- t(apply(mat.bin, 1, function(x) {zoo::rollmean(x, 9)}))
x <- as.integer(colnames(mat.bin.smoothed))
plot(x, apply(mat.bin.smoothed, 2, mean), type="l",
     xlab="position relative to center position", ylab="dyad density (normalized)", main="binned
```



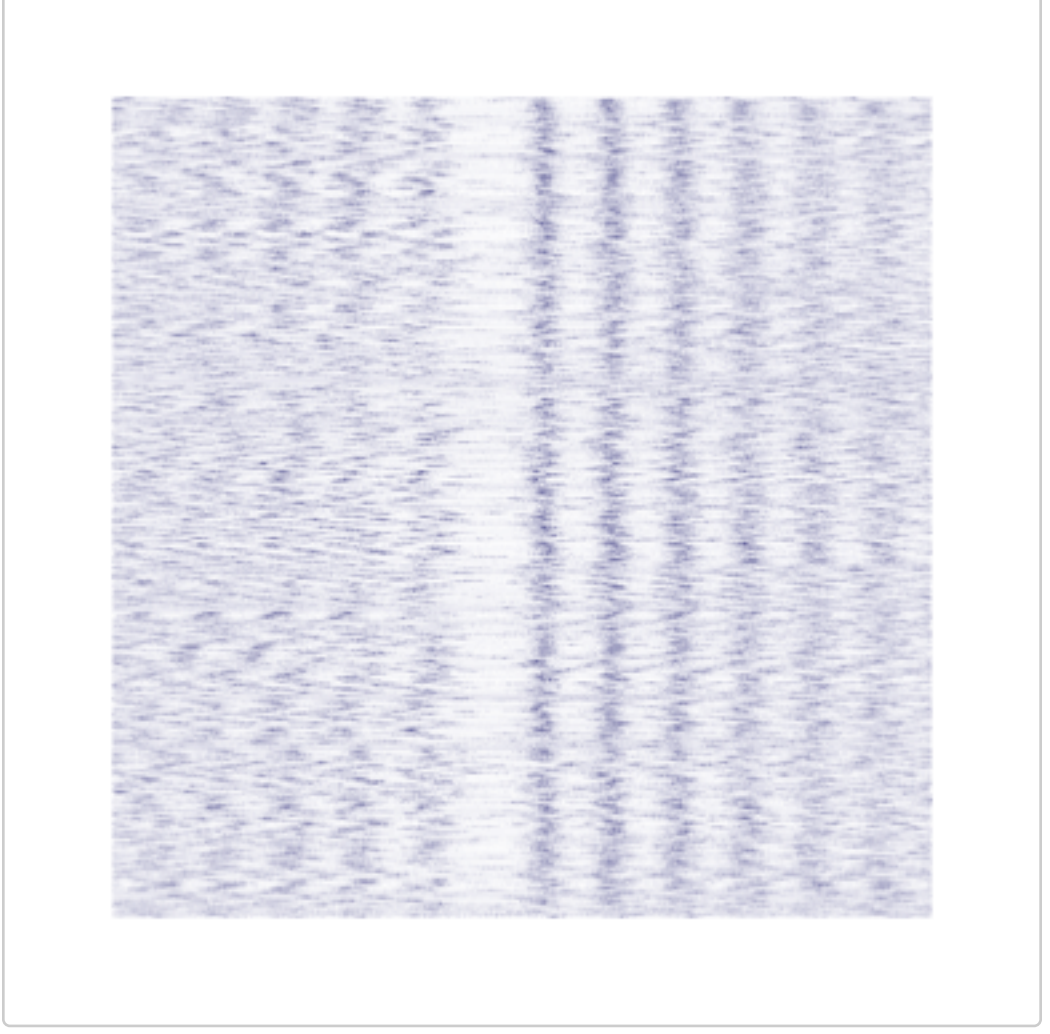
Cumulative plot with scatter

```
cumPlot(mat.sum, base.col = "#FF0000")
```



Heatmaps

```
library(grid)
hmat <- meanScale(mat)
plotRasterHeatmap(hmat)
```



Getting Spacing and Phasing along Reference points

```
ann[1:10,]
```

```
##        chr start   end strand class    name commonName  endConfidence    source
## ST0356  IV 20192 21761      +   ORF YDL241W    YDL241W bothEndsMapped    Manual
## ST4012  IV 21868 22453      -   CUT  CUT468     CUT468 bothEndsMapped Automatic
## ST0357  IV 22640 26185      +   ORF YDL240W       LRG1 bothEndsMapped    Manual
## ST4013  IV 26276 28957      -   ORF YDL239C       ADY3        mapped3    Manual
## ST4014  IV 28956 30485      -   ORF YDL238C       GUD1        mapped5    Manual
## ST0358  IV 30624 32065      +   ORF YDL237W    YDL237W bothEndsMapped    Manual
## ST4015  IV 31716 32045      -   CUT  CUT469     CUT469 bothEndsMapped Automatic
## ST0359  IV 32248 33409      +   ORF YDL236W      PHO13 bothEndsMapped    Manual
## ST4016  IV 33300 34036      -   ORF YDL235C       YPD1 bothEndsMapped    Manual
## ST4017  IV 34139 36596      -   ORF YDL234C       GYP7 bothEndsMapped    Manual
```

```
result <- ocampo(cov, ann[1:10,])
result
```

```
##           r space shift
## ST0356 0.59   168   -15
## ST4012 0.88   191    38
## ST0357 0.83   165    43
## ST4013 0.43   149   -70
## ST4014 0.74   157    59
## ST0358 0.87   166    53
## ST4015 0.84   167    36
## ST0359 0.89   169    60
## ST4016 0.77   169    36
## ST4017 0.91   161    35
```

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
##  [1] grid      stats4    parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
## [1] zoo_1.8-8           IRanges_2.20.2      S4Vectors_0.24.4
## [4] BiocGenerics_0.32.0 tsTools_0.1.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6           knitr_1.28             XVector_0.26.0
##  [4] magrittr_1.5           GenomicRanges_1.38.0   zlibbioc_1.32.0
##  [7] lattice_0.20-41        rlang_0.4.6            stringr_1.4.0
## [10] GenomeInfoDb_1.22.1    tools_3.6.3            data.table_1.12.8
## [13] xfun_0.13              htmltools_0.4.0        yaml_2.2.1
## [16] digest_0.6.25          GenomeInfoDbData_1.2.2 bitops_1.0-6
## [19] RCurl_1.98-1.2         evaluate_0.14          rmarkdown_2.1
## [22] stringi_1.4.6          compiler_3.6.3
```