

An Introduction to *Guitar* Package

Xiao Du

Modified: 26 April, 2019. Compiled: October 29, 2019

1 Quick Start with *Guitar*

This is a manual for *Guitar* package. The *Guitar* package is aimed for RNA landmark-guided transcriptomic analysis of RNA-related genomic features.

The *Guitar* package enables the comparison of multiple genomic features, which need to be stored in a name list. Please see the following example, which reads 1000 RNA m6A methylation sites into R for detection. Of course, in actual data analysis, features may come from multiple sets of resources.

```
library(Guitar)

## Loading required package: GenomicFeatures
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall,
##   clusterEvalQ, clusterExport, clusterMap,
##   parApply, parCapply, parLapply, parLapplyLB,
##   parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##   Filter, Find, Map, Position, Reduce,
##   anyDuplicated, append, as.data.frame, basename,
##   cbind, colnames, dirname, do.call, duplicated,
##   eval, evalq, get, grep, grepl, intersect,
##   is.unsorted, lapply, mapply, match, mget, order,
##   paste, pmax, pmax.int, pmin, pmin.int, rank,
##   rbind, rownames, sapply, setdiff, sort, table,
##   tapply, union, unique, unsplit, which, which.max,
##   which.min
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##   expand.grid
```

```

## Loading required package: IRanges
## Loading required package: GenomeInfoDb
## Loading required package: GenomicRanges
## Loading required package: AnnotationDbi
## Loading required package: Biobase
## Welcome to Bioconductor
##
##   Vignettes contain introductory material; view
##   with 'browseVignettes()'. To cite Bioconductor,
##   see 'citation("Biobase)", and for packages
##   'citation("pkgname)".
## Loading required package: rtracklayer
## Loading required package: magrittr
## Loading required package: ggplot2
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:AnnotationDbi':
##
##   select
## The following object is masked from 'package:Biobase':
##
##   combine
## 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 objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
##
## Attaching package: 'Guitar'
## The following object is masked from 'package:BiocGenerics':
##
##   normalize

# genomic features imported into named list
stBedFiles <- list(system.file("extdata", "m6A_mm10_exomePeak_1000peaks_bed12.bed",
                             package="Guitar"))

```

With the following script, we may generate the transcriptomic distribution of genomic features to be tested, and the result will be automatically saved into a PDF file under the working directory with prefix "example". With the `GuitarPlot` function, the gene annotation can be downloaded from internet automatically with a genome assembly number provided; however, this feature requires working internet and might take a longer time. The toy `Guitar` coordinates generated internally should never be re-used in other real data analysis.

```
count <- GuitarPlot(txGenomeVer = "mm10",
                    stBedFiles = stBedFiles,
                    miscOutFilePrefix = NA)
```

In a more efficient protocol, in order to re-use the gene annotation and *Guitar coordinates*, you will have to build `Guitar Coordinates` from a `txdb` object in a separate step. The `transcriptDb` contains the gene annotation information and can be obtained in a number of ways, .e.g, download the complete gene annotation of species from UCSC automatically, which might takes a few minutes. In the following analysis, we load the `Txdb` object from a toy dataset provided with the `Guitar` package. Please note that this is only a very small part of the complete hg19 transcriptome, and the `Txdb` object provided with `Guitar` package should not be used in real data analysis. With a `Txdb` object that contains gene annotation information, we in the next build *Guitar coordinates*, which is essentially a bridge connects the transcriptomic landmarks and genomic coordinates.

```
txdb_file <- system.file("extdata", "mm10_toy.sqlite",
                        package="Guitar")
txdb <- loadDb(txdb_file)
guitarTxdb <- makeGuitarTxdb(txdb = txdb, txPrimaryOnly = FALSE)

## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate components for mRNA"
## [1] "generate components for lncRNA"
## [1] "generate chipped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "generate coverage checking ranges for mrna"
## [1] "generate coverage checking ranges for ncrna"

# Or use gff. file to generate guitarTxdb
# Or use getTxdb() to download TxDb from internet:
# txdb <- getTxdb(txGenomeVer="hg19")
# guitarTxdb <- makeGuitarTxdb(txdb)
```

You may now generate the `Guitar` plot from the named list of genome-based features.

```
GuitarPlot(txTxdb = txdb,
           stBedFiles = stBedFiles,
           miscOutFilePrefix = "example")

## [1] "20191029213645"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
```

```

## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate components for mRNA"
## [1] "generate components for lncRNA"
## [1] "generate chiped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "generate coverage checking ranges for mrna"
## [1] "generate coverage checking ranges for ncrna"
## [1] "20191029213658"
## [1] "import BED file /tmp/RtmpvB2d3J/Rinst7cf748831e94/Guitar/extdata/m6A_mm10_exomePeak_1000peaks_b
## [1] "sample 10 points for Group1"
## [1] "start figure plotting for tx ..."
## [1] "start figure plotting for mrna ..."
## [1] "start figure plotting for ncrna ..."

```

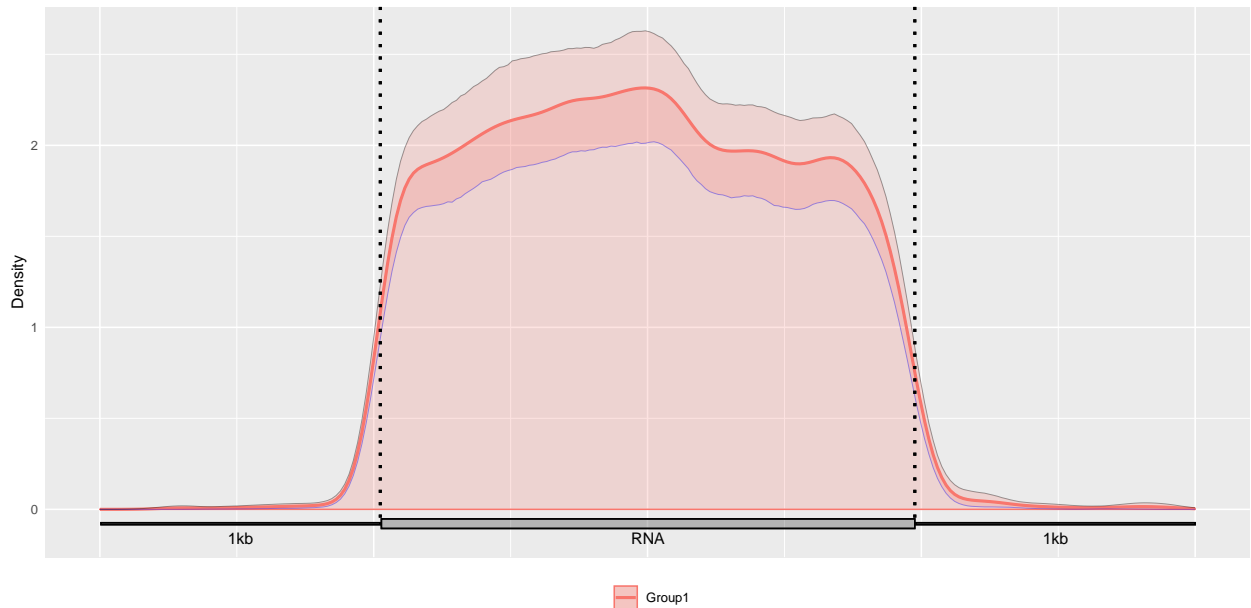
Alternatively, you may also optionally include the promoter DNA region and tail DNA region on the 5' and 3' side of a transcript in the plot with parameter `headOrtail = TRUE`.

```

GuitarPlot(txTxdb = txdb,
           stBedFiles = stBedFiles,
           headOrtail = TRUE)

## [1] "20191029213734"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate components for mRNA"
## [1] "generate components for lncRNA"
## [1] "generate chiped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "generate coverage checking ranges for mrna"
## [1] "generate coverage checking ranges for ncrna"
## [1] "20191029213748"
## [1] "import BED file /tmp/RtmpvB2d3J/Rinst7cf748831e94/Guitar/extdata/m6A_mm10_exomePeak_1000peaks_b
## [1] "sample 10 points for Group1"
## [1] "start figure plotting for tx ..."

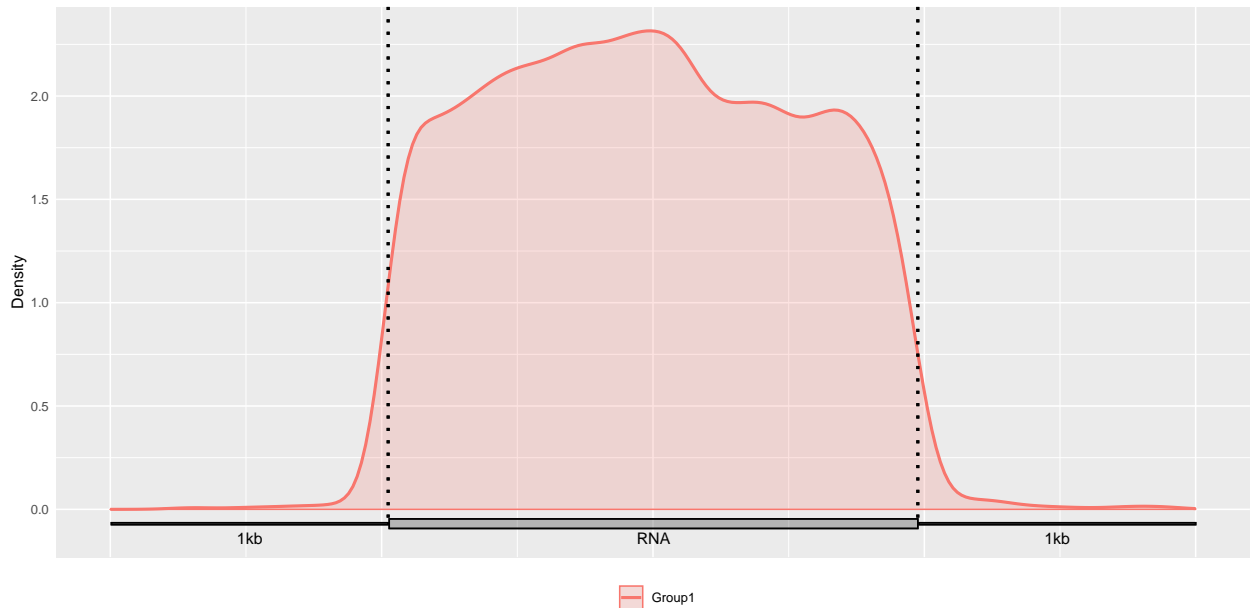
```



Alternatively, you may also optionally include the Confidence Interval for guitar plot with parameter `enableCI = FALSE`.

```
GuitarPlot(txTxdb = txdb,
           stBedFiles = stBedFiles,
           headOrtail = TRUE,
           enableCI = FALSE)

## [1] "20191029213809"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate components for mRNA"
## [1] "generate components for lncRNA"
## [1] "generate chipped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "generate coverage checking ranges for mrna"
## [1] "generate coverage checking ranges for ncrna"
## [1] "20191029213822"
## [1] "import BED file /tmp/RtmpvB2d3J/Rinst7cf748831e94/Guitar/extdata/m6A_mm10_exomePeak_1000peaks_b"
## [1] "sample 10 points for Group1"
## [1] "start figure plotting for tx ..."
```



2 Supported Data Format

Besides BED file, Guitar package also supports GRangesList and GRanges data structures. Please see the following examples.

```
# import different data formats into a named list object.
# These genomic features are using mm10 genome assembly
stBedFiles <- list(system.file("extdata", "m6A_mm10_exomePeak_1000peaks_bed12.bed",
                             package="Guitar"),
                  system.file("extdata", "m6A_mm10_exomePeak_1000peaks_bed6.bed",
                             package="Guitar"))

# Build Guitar Coordinates
txdb_file <- system.file("extdata", "mm10_toy.sqlite",
                        package="Guitar")
txdb <- loadDb(txdb_file)

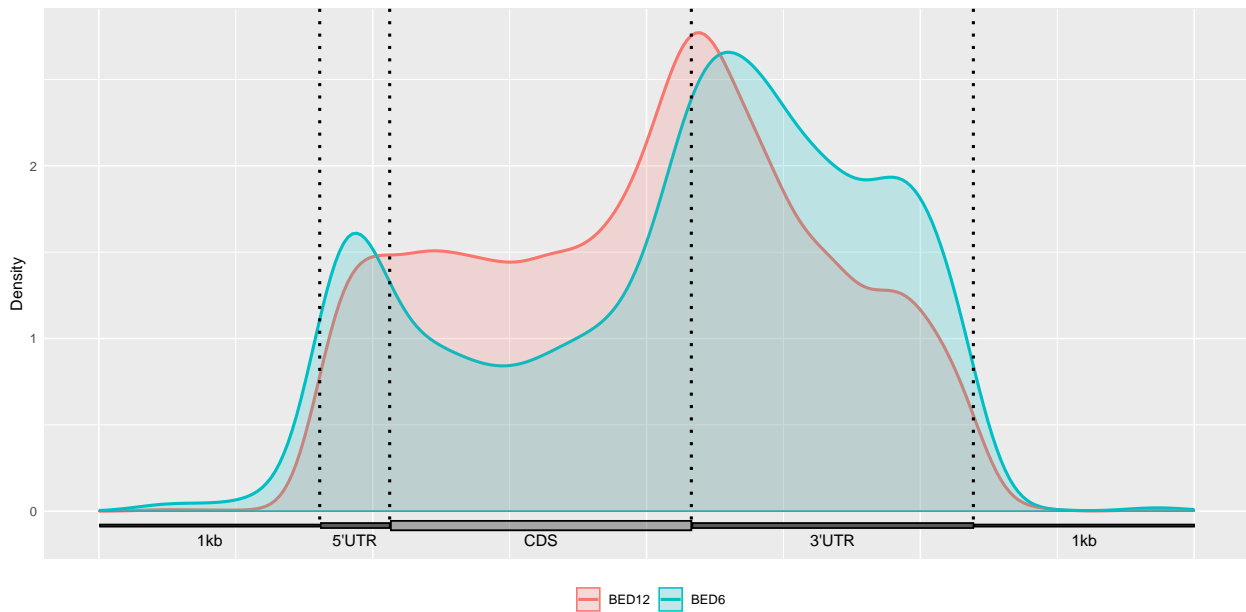
# Guitar Plot
GuitarPlot(txTxdb = txdb,
           stBedFiles = stBedFiles,
           headOrtail = TRUE,
           enableCI = FALSE,
           mapFilterTranscript = TRUE,
           pltTxType = c("mrna"),
           stGroupName = c("BED12", "BED6"))

## [1] "20191029213824"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
```

```

## [1] "generate components for mRNA"
## [1] "generate chipped transcriptome"
## [1] "generate coverage checking ranges for mrna"
## [1] "20191029213837"
## [1] "import BED file /tmp/RtmpvB2d3J/Rinst7cf748831e94/Guitar/extdata/m6A_mm10_exomePeak_1000peaks_b
## [1] "import BED file /tmp/RtmpvB2d3J/Rinst7cf748831e94/Guitar/extdata/m6A_mm10_exomePeak_1000peaks_b
## [1] "sample 10 points for BED12"
## [1] "sample 10 points for BED6"
## [1] "start figure plotting for mrna ..."

```



3 Processing of sampling sites information

We can select parameters for site sampling.

```

stGRangeLists = vector("list", length(stBedFiles))
sitesPoints <- list()
for (i in seq_len(length(stBedFiles))) {
  stGRangeLists[[i]] <- blocks(import(stBedFiles[[i]]))
}
for (i in seq_len(length(stGRangeLists))) {
  sitesPoints[[i]] <- samplePoints(stGRangeLists[i],
    stSampleNum = 10,
    stAmblguity = 5,
    pltTxType = c("mrna"),
    stSampleModle = "Equidistance",
    mapFilterTranscript = FALSE,
    guitarTxdb = guitarTxdb)
}

```

4 Guitar Coordinates - Transcriptomic Landmarks Projected on Genome

The `guitarTxdb` object contains the genome-projected transcriptome coordinates, which can be valuable for evaluating transcriptomic information related applications, such as checking the quality of MeRIP-Seq data. The `Guitar` coordinates are essentially the genomic projection of standardized transcript-based coordinates, making a viable bridge between the landmarks on transcript and genome-based coordinates.

It is based on the `txdb` object input, extracts the transcript information in `txdb`, selects the transcripts that match the parameters according to the component parameters set by the user, and saves according to the transcript type (`tx`, `mrna`, `ncrna`).

```
guitarTxdb <- makeGuitarTxdb(txdb = txdb,
                             txAmbiguity = 5,
                             txMrnaComponentProp = c(0.1,0.15,0.6,0.05,0.1),
                             txLncrnaComponentProp = c(0.2,0.6,0.2),
                             pltTxType = c("tx","mrna","ncrna"),
                             txPrimaryOnly = FALSE)

## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate components for mRNA"
## [1] "generate components for lncRNA"
## [1] "generate chipped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "generate coverage checking ranges for mrna"
## [1] "generate coverage checking ranges for ncrna"
```

5 Check the Overlapping between Different Components

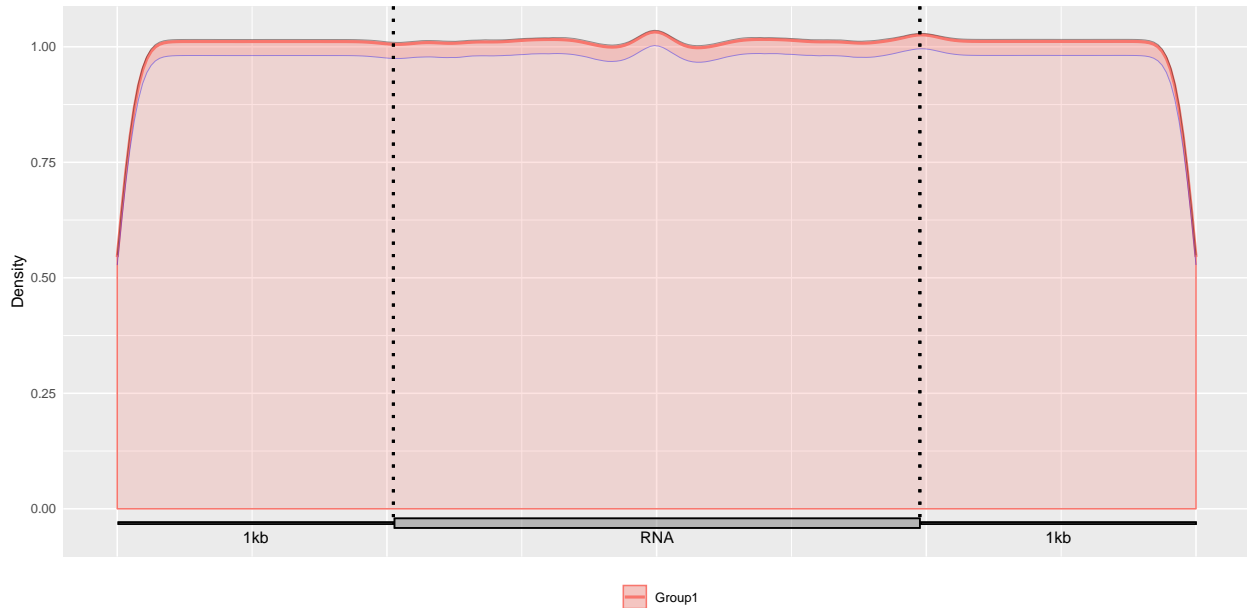
We can also check the distribution of the `Guitar` coordinates built.

```
gcl <- list(guitarTxdb$tx$tx)
GuitarPlot(txTxdb = txdb,
            stGRangeLists = gcl,
            stSampleNum = 200,
            enableCI = TRUE,
            pltTxType = c("tx"),
            txPrimaryOnly = FALSE
            )

## [1] "20191029213854"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate chipped transcriptome"
```



```
## [1] "generate coverage checking ranges for tx"
## [1] "20191029213908"
## [1] "sample 200 points for Group1"
## [1] "start figure plotting for tx ..."
```

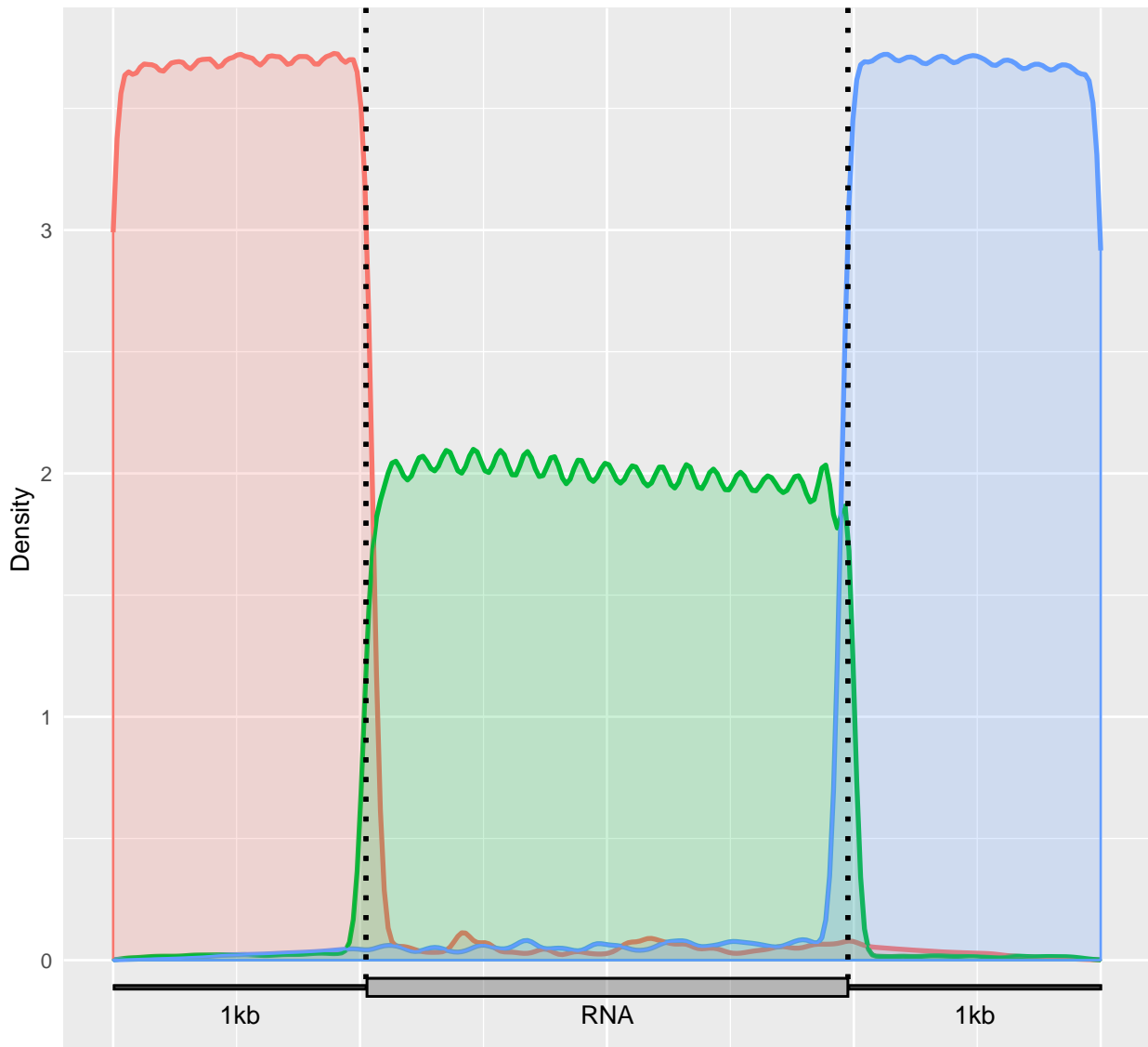


Alternatively, we can extract the RNA components, check the distribution of tx components in the transcriptome

```
GuitarCoords <- guitarTxdb$tx$txComponentGRRange
type <- paste(mcols(GuitarCoords)$componentType,mcols(GuitarCoords)$txType)
key <- unique(type)
landmark <- list(1,2,3,4,5,6,7,8,9,10,11)
names(landmark) <- key
for (i in 1:length(key)) {
  landmark[[i]] <- GuitarCoords[type==key[i]]
}
GuitarPlot(txTxdb = txdb ,
           stGRangeLists = landmark[1:3],
           pltTxType = c("tx"),
           enableCI = FALSE
)
```

```
## [1] "20191029214911"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for all tx"
## [1] "generate chipped transcriptome"
## [1] "generate coverage checking ranges for tx"
## [1] "20191029214928"
## [1] "sample 10 points for Group1"
```

```
## [1] "sample 10 points for Group2"
## [1] "sample 10 points for Group3"
## [1] "start figure plotting for tx ..."
```



■ Group1
 ■ Group2
 ■ Group3

Check the distribution of mRNA components in the transcriptome

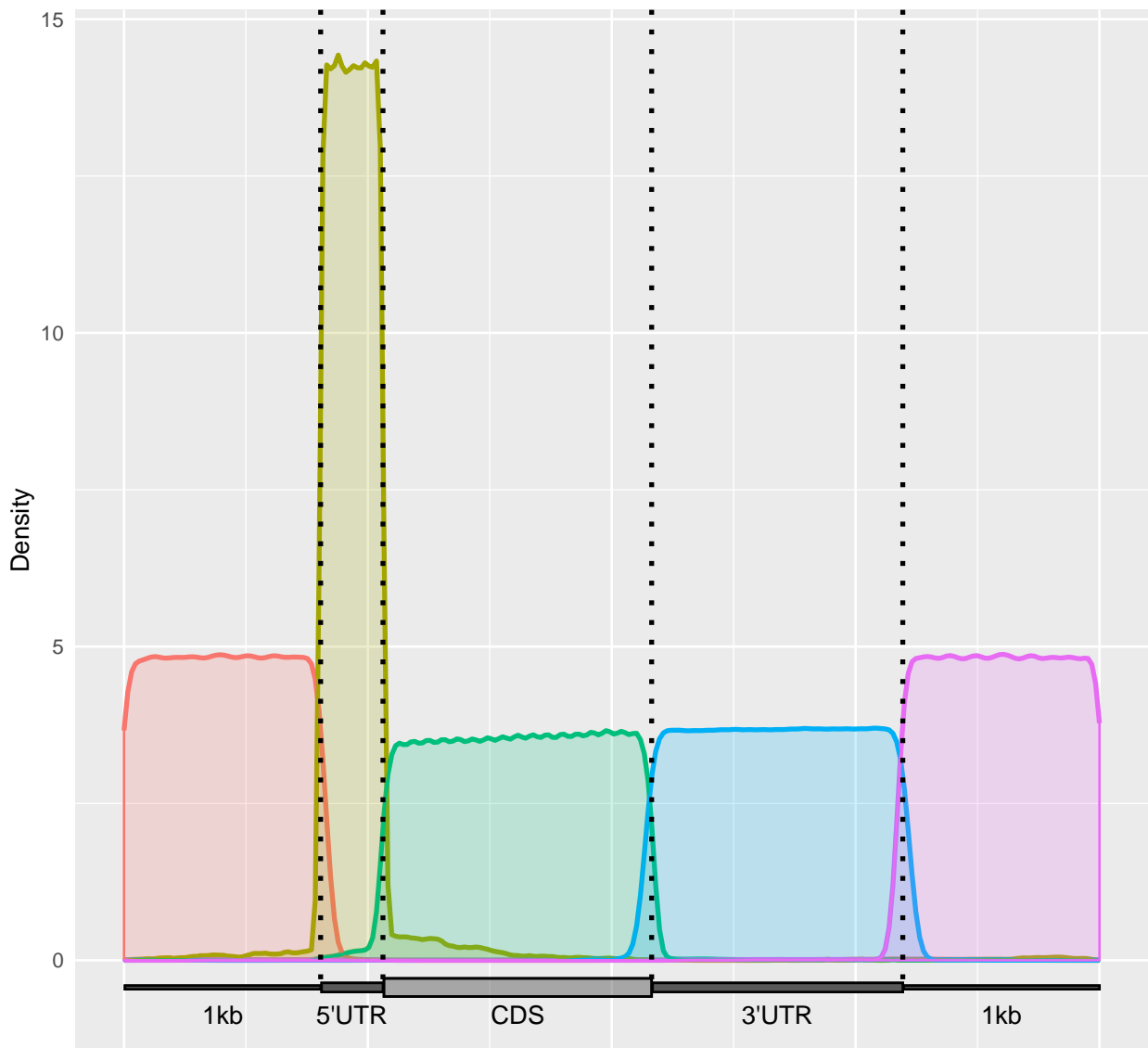
```
GuitarPlot(txTxdb = txdb ,
            stGRangeLists = landmark[4:8],
            pltTxType = c("mrna"),
            enableCI = FALSE
)

## [1] "20191029214956"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
```

```

## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for mRNA"
## [1] "generate chiped transcriptome"
## [1] "generate coverage checking ranges for mrna"
## [1] "20191029215010"
## [1] "sample 10 points for Group1"
## [1] "sample 10 points for Group2"
## [1] "sample 10 points for Group3"
## [1] "sample 10 points for Group4"
## [1] "sample 10 points for Group5"
## [1] "start figure plotting for mrna ..."

```

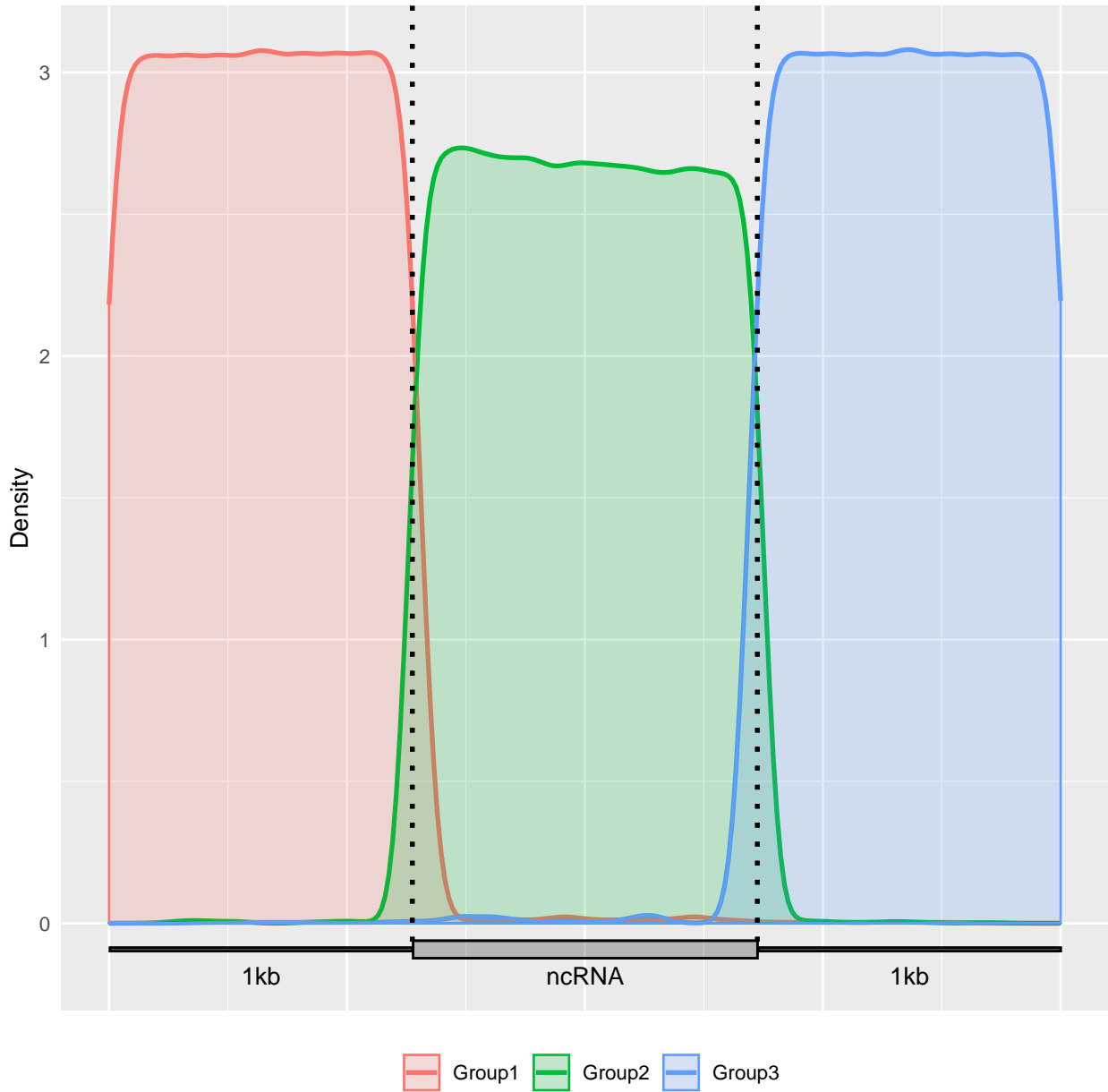


■ Group1
 ■ Group2
 ■ Group3
 ■ Group4
 ■ Group5

Check the distribution of lncRNA components in the transcriptome

```
GuitarPlot(txTxdb = txdb ,
           stGRangeLists = landmark[9:11],
           pltTxType = c("ncrna"),
           enableCI = FALSE
)

## [1] "20191029215023"
## [1] "There are 2946 transcripts of 2946 genes in the genome."
## [1] "total 2946 transcripts extracted ..."
## [1] "total 2719 transcripts left after ambiguity filter ..."
## [1] "total 2719 transcripts left after check chromosome validity ..."
## [1] "total 1342 mRNAs left after component length filter ..."
## [1] "total 307 ncRNAs left after ncRNA length filter ..."
## [1] "generate components for lncRNA"
## [1] "generate chiped transcriptome"
## [1] "generate coverage checking ranges for ncrna"
## [1] "20191029215039"
## [1] "sample 10 points for Group1"
## [1] "sample 10 points for Group2"
## [1] "sample 10 points for Group3"
## [1] "start figure plotting for ncrna ..."
```



6 Session Information

```

sessionInfo()

## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.3 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.10-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.10-bioc/R/lib/libRlapack.so
##

```

```

## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
## [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8    LC_NAME=C
## [9] LC_ADDRESS=C            LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4      parallel  stats      graphics  grDevices
## [6] utils       datasets  methods    base
##
## other attached packages:
## [1] Guitar_2.2.0      dplyr_0.8.3
## [3] ggplot2_3.2.1     magrittr_1.5
## [5] rtracklayer_1.46.0 GenomicFeatures_1.38.0
## [7] AnnotationDbi_1.48.0 Biobase_2.46.0
## [9] GenomicRanges_1.38.0 GenomeInfoDb_1.22.0
## [11] IRanges_2.20.0    S4Vectors_0.24.0
## [13] BiocGenerics_0.32.0 knitr_1.25
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.2          lattice_0.20-38
## [3] prettyunits_1.0.2  Rsamtools_2.2.0
## [5] Biostrings_2.54.0  assertthat_0.2.1
## [7] zeallot_0.1.0      digest_0.6.22
## [9] BiocFileCache_1.10.0 R6_2.4.0
## [11] backports_1.1.5    RSQlite_2.1.2
## [13] evaluate_0.14      httr_1.4.1
## [15] highr_0.8           pillar_1.4.2
## [17] zlibbioc_1.32.0    rlang_0.4.1
## [19] progress_1.2.2     lazyeval_0.2.2
## [21] curl_4.2            blob_1.2.0
## [23] Matrix_1.2-17      labeling_0.3
## [25] BiocParallel_1.20.0 stringr_1.4.0
## [27] RCurl_1.95-4.12    bit_1.1-14
## [29] biomaRt_2.42.0     munsell_0.5.0
## [31] DelayedArray_0.12.0 compiler_3.6.1
## [33] xfun_0.10           pkgconfig_2.0.3
## [35] askpass_1.1         openssl_1.4.1
## [37] tidyselect_0.2.5   SummarizedExperiment_1.16.0
## [39] tibble_2.1.3       GenomeInfoDbData_1.2.2
## [41] matrixStats_0.55.0 XML_3.98-1.20
## [43] withr_2.1.2        crayon_1.3.4
## [45] dbplyr_1.4.2       GenomicAlignments_1.22.0
## [47] bitops_1.0-6       rappdirs_0.3.1
## [49] grid_3.6.1         gtable_0.3.0
## [51] DBI_1.0.0          scales_1.0.0
## [53] stringi_1.4.3      XVector_0.26.0
## [55] vctrs_0.2.0        tools_3.6.1
## [57] bit64_0.9-7        glue_1.3.1
## [59] purrr_0.3.3        hms_0.5.1
## [61] colorspace_1.4-1   memoise_1.1.0

```