An Introduction to *Guitar* Package

Jia Meng, PhD

Modified: 2 April, 2016. Compiled: May 3, 2016

1 Quick Start with Guitar

This is a manual for Guitar package. The Guitar package is aimed for RNA landmark-**gui**ded **t**ranscriptomic **a**nalysis of RNA-reated genomic featu**r**es.

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 will read 1000 RNA m6A methylation sites into R, and then splitted them into 3 groups to be examined. Of course, in real data analysis, the 3 groups of features are likely to be from different resources.

```
> library(Guitar)
> narrowPeak <- system.file(
+   "extdata", "m6A_hg19_1000peaks_macs2.narrowPeak",
+   package="Guitar")
> # genomic features imported into named list
> m6A_Bcell <- narrowPeaktoGRanges(narrowPeak)
> m6A_Bcell_1 <- m6A_Bcell[1:300]
> m6A_Bcell_2 <- m6A_Bcell[301:600]
> m6A_Bcell_3 <- m6A_Bcell[601:900]
> feature_hg19 <- list(m6A_Bcell_1, m6A_Bcell_2, m6A_Bcell_3)
> names(feature_hg19) <- c("Group_1","Group_2","Group_3")
```

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(feature_hg19,
+            genome="hg19",
+            saveToPDFprefix = "example")
```

In a more efficent protocol, in order to re-use the gene annotation and *Guitar coordinates*, you will have to build Guitar Coordiantes 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, with command `makeTxDbFromUCSC` from *GenomicFeatures* package to 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.

```
> txdb_file <- system.file("extdata", "hg19_toy.sqlite",
+                          package="Guitar")
> txdb <- loadDb(txdb_file)
> # Or use makeTxDbFromUCSC() to download TxDb from internet
> # txdb <- makeTxDbFromUCSC(genome="hg19")
```

With a *TxDb* object that contains gene annotation information, we in the next build *Guitar coordiantes*, which is essentially a bridge connects the transcriptomic landmarks and genomic coordinates. The parameter `noBins=20` defines the resolution of your analysis. Higher resolution will lead to finer details revealed and increased computation time and memory usage.

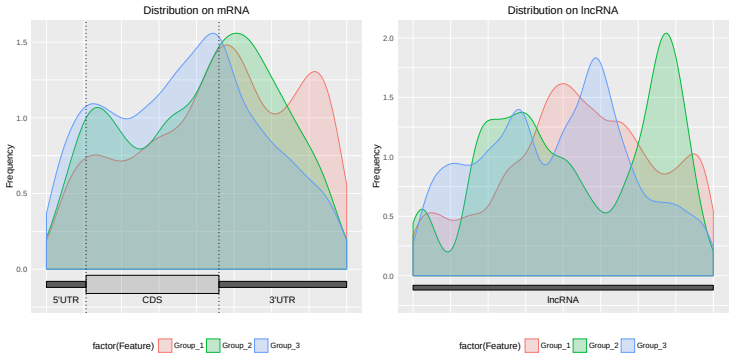
```
> gc_txdb <- makeGuitarCoordsFromTxDb(txdb, noBins=100)
```

```
[1] "total 1932 transcripts extracted ..."
[1] "total 1043 transcripts left after ambiguity filter ..."
[1] "total 471 mRNAs left after component length filter ..."
[1] "total 148 ncRNAs left after ncRNA length filter ..."
[1] "Building Guitar Coordinates. It may take a few minutes ..."
[1] "Guitar Coordinates Built ..."
```

You may now generate the Guitar plot from the named list of genome-based features and the prebuilt *Guitar coordinates*.

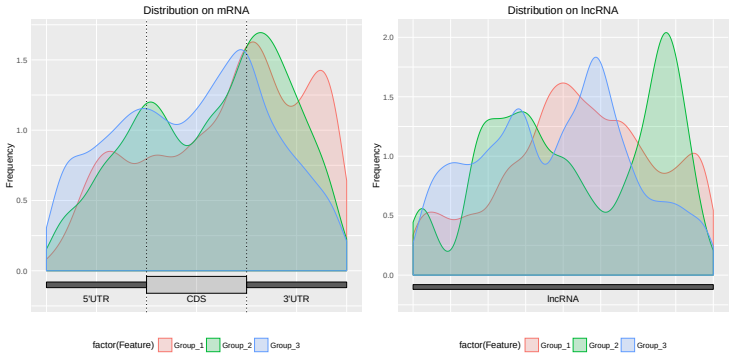
```
> GuitarPlot(gfeatures = feature_hg19,
+            GuitarCoordsFromTxDb = gc_txdb,
+            saveToPDFprefix = "example2")
```

```
[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "Figures saved into example2_Guitar.pdf ..."
```



The figure generated reflects the true size of RNA components (5'UTR, CDS and 3'UTR); alternatively, you may cancel the component rescaling, and treat each component equally with option `rescaleComponent=FALSE`. The figure generated in this way looks clearer, but the relative size of the components will be missing. In this way, 5'UTR, although is mucher smaller, shows up to be of the same size as CDS and 3'UTR.

```
> GuitarPlot(gfeatures = feature_hg19,
+            GuitarCoordsFromTxDb = gc_txdb,
+            rescaleComponent=FALSE)

[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```



Alternatively, you may also stack everthing together to see their relative location-specific density compared with other features tested, or you may also optionally include the promoter DNA region and its complementary region on the 3' side of a transcript in the plot with parameter `includeNeighborDNA =TRUE` and `fill=TRUE`.

```
> GuitarPlot(gfeatures = feature_hg19,
+            GuitarCoordsFromTxDb = gc_txdb,
+            includeNeighborDNA =TRUE,
```

```
+             rescaleComponent=FALSE,
+             fill=TRUE)
```

```
[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```

2 Supported Data Format

Besides BED12 format, *Guitar* package also supports BAM/SAM and bed3 via GRangesList, GRanges and GAlignments data structures. Please see the following examples.

```
> # import different data formats into a named list object.
> # These genomic features are using mm10 genome assembly
> bed3=system.file("extdata", "H3K4me3_mm10_1000peaks.bed", package="Guitar")
> bed12=system.file("extdata", "m6A_mm10_exomePeak_1000peaks_bed12.bed", package="Guitar")
> bam1=system.file("extdata", "866991_part.bam", package="Guitar")
> bam2=system.file("extdata", "866992_part.bam", package="Guitar")
> H3K4me3 <- import.bed(bed3) # bed3 imported as GRanges
> m6A_HepG2 <- BED12toGRangesList(bed12) # bed12 imported as GRangesList
```

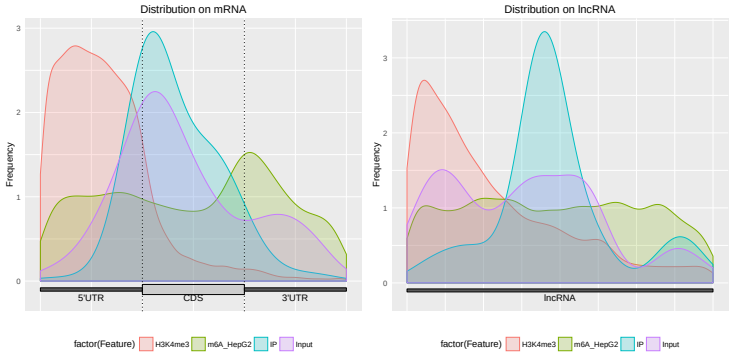
```
[1] "Converting BED12 to GRangesList"
[1] "It may take a few minutes"
```

```
> MeRIP_IP <- readGAlignments(bam1) # bam imported as GAlignments
> MeRIP_Input <- readGAlignments(bam2) # bam imported as GAlignments
> feature_mm10 <- list(H3K4me3,m6A_HepG2,MeRIP_IP,MeRIP_Input)
> names(feature_mm10) <- c("H3K4me3", "m6A_HepG2", "IP","Input")
> # Build Guitar Coordinates
> txdb_file <- system.file("extdata", "mm10_toy.sqlite",
+                          package="Guitar")
> mm10_toy_txdb <- loadDb(txdb_file)
> gc_mm10_txdb <- makeGuitarCoordsFromTxDb(mm10_toy_txdb, noBins=30)
```

```
[1] "total 2946 transcripts extracted ..."
[1] "total 2272 transcripts left after ambiguity filter ..."
[1] "total 1038 mRNAs left after component length filter ..."
[1] "total 353 ncRNAs left after ncRNA length filter ..."
[1] "Building Guitar Coordinates. It may take a few minutes ..."
[1] "Guitar Coordinates Built ..."
```

```
> # Guitar Plot
> GuitarPlot(gfeatures = feature_mm10,
+            GuitarCoordsFromTxDb = gc_mm10_txdb,
+            rescaleComponent=FALSE)
```

```
[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```



3 Analysis of BS-Seq data report from Bismark

Here is one more example for analysis of DNA m5C methylation from BS-Seq data.

```
> f <-system.file("extdata", "DNAm5C_mm10_10000sites_bismark.cov", package="Guitar")
> a <- read.table(f,header=FALSE,sep="\t")
> q <- a[[5]]
> size <- a[[5]]+a[[6]]
> mean_methy <- sum(q)/sum(size)
> d0 <- (pbinom(q, size, mean_methy, lower.tail = TRUE, log.p = FALSE) < 0.05)
> d1 <- (pbinom(q, size, mean_methy, lower.tail = FALSE, log.p = FALSE) < 0.05)
> GR <- GRanges(seqnames = a[,1], ranges = IRanges(start=a[,2], width=1))
> peak <- list(GR[d1],GR[d1+d0 == 0],GR[d0])
> names(peak) <- c( "higher", "unsure", "lower")
> TxDb.Mmusculus.UCSC.mm10.knownGene <- makeTxDbFromUCSC(genome="mm10")
> gc_mm10_txdb <- makeGuitarCoordsFromTxDb(TxDb.Mmusculus.UCSC.mm10.knownGene)
```

```
> GuitarPlot(gfeatures = peak,
+            GuitarCoordsFromTxDb = gc_mm10_txdb,
+            includeNeighborDNA =TRUE,
+            fill=TRUE)
```

4 Guitar Coordinates - Transcriptomic Landmarks Projected on Genome

The `GuitarCoordsFromTxDb` 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 coordiantes, making a viable bridge beween the landmarks on transcript and genome-based coordinates. The referred "transcript id", standardized position, the interval between two adjacent check points on that components, component name and type are assessible with `mcols` function.

```
> head(gc_txdb)
```

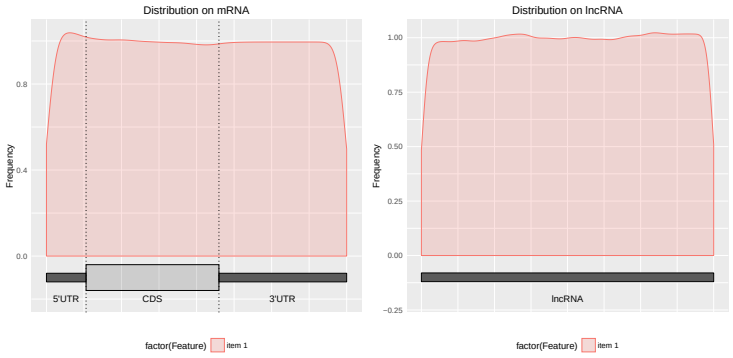
```
GRanges object with 6 ranges and 5 metadata columns:
             seqnames             ranges strand |
                <Rle>          <IRanges>  <Rle> |
  uc010nxq.1     chr1 [  10854,   10904]      + |
  uc009vjk.2     chr1 [ 321017,  321067]      + |
  uc001aau.3     chr1 [ 322872,  322922]      + |
  uc021oeh.1     chr1 [ 323268,  323318]      + |
  uc021oei.1     chr1 [ 326526,  326576]      + |
  uc001acv.3     chr1 [1071377, 1071427]      + |
                 txid       pos  interval     comp
               <factor> <numeric> <numeric> <factor>
  uc010nxq.1 uc010nxq.1     0.005        10    Front
  uc009vjk.2 uc009vjk.2     0.005        10    Front
  uc001aau.3 uc001aau.3     0.005        10    Front
  uc021oeh.1 uc021oeh.1     0.005        10    Front
  uc021oei.1 uc021oei.1     0.005        10    Front
  uc001acv.3 uc001acv.3     0.005        10    Front
             category
             <factor>
  uc010nxq.1     mRNA
  uc009vjk.2     mRNA
  uc001aau.3     mRNA
  uc021oeh.1     mRNA
  uc021oei.1     mRNA
  uc001acv.3     mRNA
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
> mcols(gc_txdb)

DataFrame with 279900 rows and 5 columns
             txid       pos  interval     comp category
         <factor> <numeric> <numeric> <factor> <factor>
1      uc010nxq.1     0.005        10    Front     mRNA
2      uc009vjk.2     0.005        10    Front     mRNA
3      uc001aau.3     0.005        10    Front     mRNA
4      uc021oeh.1     0.005        10    Front     mRNA
5      uc021oei.1     0.005        10    Front     mRNA
...           ...       ...       ...      ...      ...
279896 uc001byp.3     0.995        10     Back   lncRNA
279897 uc001byq.3     0.995        10     Back   lncRNA
279898 uc010ohw.2     0.995        10     Back   lncRNA
279899 uc021olm.1     0.995        10     Back   lncRNA
279900 uc010oid.3     0.995        10     Back   lncRNA
```

5 Check the Overlapping between Different Components

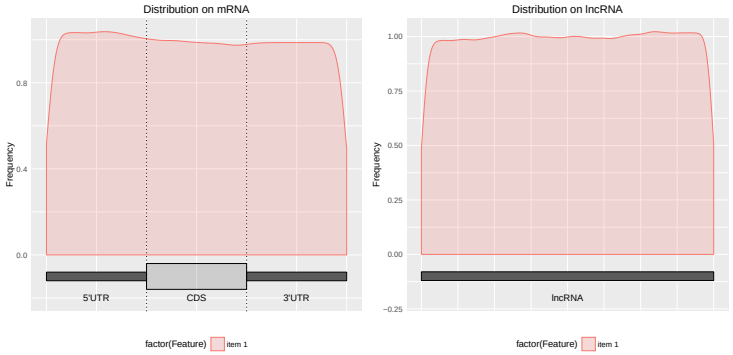
We can also check the distribution of the Guitar coordinates built.

```
> GuitarCoords <- reduce(gc_txdb)  # extract the coordinates
> gcl <- list(GuitarCoords) # put into a list
> GuitarPlot(gfeatures = gcl,
+            GuitarCoordsFromTxDb = gc_txdb,
+            rescaleComponent=TRUE)

[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```



or, check without rescaling

```
> GuitarCoords <- reduce(gc_txdb)  # extract the coordinates
> gcl <- list(GuitarCoords) # put into a list
> GuitarPlot(gfeatures = gcl,
+            GuitarCoordsFromTxDb = gc_txdb,
+            rescaleComponent=FALSE)

[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```

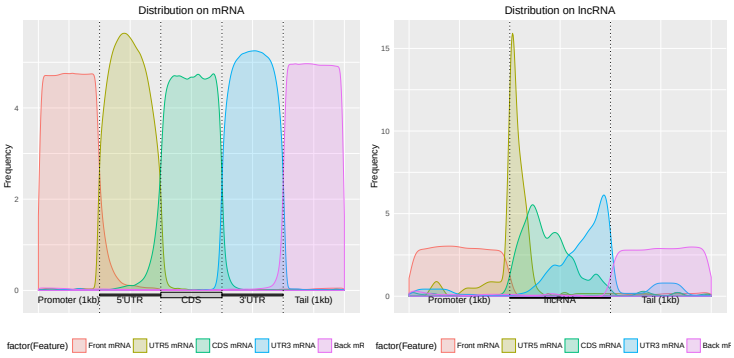


Alternatively, we can extract the RNA components

```
> GuitarCoords <- gc_txdb
> type <- paste(mcols(GuitarCoords)$comp,mcols(GuitarCoords)$category)
> key <- unique(type)
> landmark <- list(1,2,3,4,5,6,7,8)
> names(landmark) <- key
> for (i in 1:length(key)) {
+   landmark[[i]] <- GuitarCoords[type==key[i]]
+ }
```

Check the distribution of mRNA components in the transcriptome

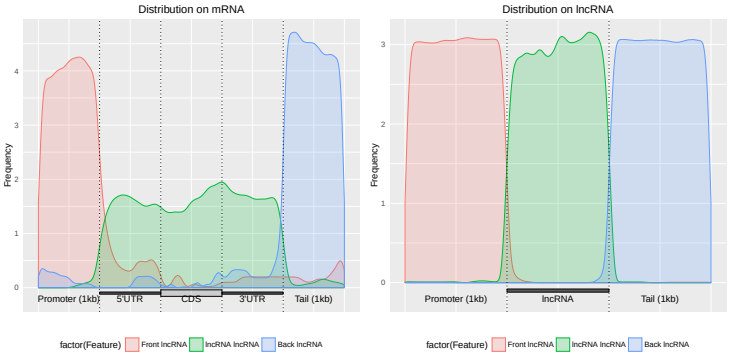
```
> GuitarPlot(gfeatures=landmark[1:5],
+            GuitarCoordsFromTxDb = gc_txdb,
+            includeNeighborDNA =TRUE,
+            rescaleComponent=FALSE)

[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```



Check the distribution of lncRNA components in the transcriptome

```
> GuitarPlot(gfeatures=landmark[6:8],
+             GuitarCoordsFromTxDb = gc_txdb,
+             includeNeighborDNA =TRUE,
+             rescaleComponent=FALSE)
```

```
[1] "Using provided Guitar Coordinates"
[1] "resolving ambiguious features ..."
[1] "no figure saved ..."
```



6 Session Information

```
> sessionInfo()
```

```
R version 3.3.0 (2016-05-03)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.4 LTS

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] grid      parallel  stats4    stats     graphics
 [6] grDevices utils     datasets  methods   base

other attached packages:
 [1] Guitar_1.10.0               ggplot2_2.1.0
 [3] GenomicAlignments_1.8.0     SummarizedExperiment_1.2.0
 [5] rtracklayer_1.32.0          GenomicFeatures_1.24.0
 [7] AnnotationDbi_1.34.0        Biobase_2.32.0
 [9] Rsamtools_1.24.0            Biostrings_2.40.0
[11] XVector_0.12.0              GenomicRanges_1.24.0
[13] GenomeInfoDb_1.8.0          IRanges_2.6.0
[15] S4Vectors_0.10.0            BiocGenerics_0.18.0

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.4.5       zlibbioc_1.18.0
 [3] munsell_0.4.3       BiocParallel_1.6.0
 [5] colorspace_1.2-6    plyr_1.8.3
 [7] tools_3.3.0         gtable_0.2.0
 [9] DBI_0.4             digest_0.6.9
[11] bitops_1.0-6        RCurl_1.95-4.8
[13] biomaRt_2.28.0      RSQLite_1.0.0
[15] labeling_0.3        scales_0.4.0
[17] XML_3.98-1.4
```