riboSeq

Thomas J. Hardcastle, Betty Y.W. Chung

August 21, 2014

Introduction

Ribosome profiling extracts those parts of a coding sequence currently bound by a ribosome (and thus, are likely to be undergoing translation). Ribosomes typically cover between 20-30 bases of the mRNA (dependant on conformational changes) and move along the mRNA three bases at a time. Sequenced reads of a given length are thus likely to lie predominantly in a single frame relative to the start codon of the coding sequence. This package presents a set of methods for parsing ribosomal profiling data from multiple samples and aligned to coding sequences, inferring frameshifts, and plotting the average and transcript-specific behaviour of these data. Methods are also provided for extracting the data in a suitable form for differential translation analysis.

Getting Data

`riboSeq` currently reads alignment data from flat text files that contain (as a minimum), the sequence of the read, the name of the sequence to which the read aligns, the strand to which it aligns, and the starting position of alignment. A *Bowtie* alignment (note that *Bowtie*, rather than *Bowtie2*, is recommended for short reads, which ribosome footprints are) using the option "–suppress 1,6,7,8" will generate this minimal data. It is by default assumed that the data are generated in this way, and the default columns specification for the default `readRibodata` function (see below) reflects this.

Workflow Example

Begin by loading the riboSeq library.

```
> library(riboSeq)
```

Identify the data directory for the example data.

```
> datadir <- system.file("extdata", package = "riboSeq")
```

The `fastaCDS` function can be used to guess at potential coding sequences from a (possibly compressed; see `base::file`) fasta file containing mRNA transcripts (note; do not use this on a genome!). These can also be loaded into a *GRanges* object from an annotation file.

```
> chlamyFasta <- paste(datadir, "/rsem_chlamy236_deNovo.transcripts.fa", sep = "")
> fastaCDS <- findCDS(fastaFile = chlamyFasta,
+                      startCodon = c("ATG"),
+                      stopCodon = c("TAG", "TAA", "TGA"))
```

The ribosomal and RNA (if available) alignment files are specified.

```
> ribofiles <- paste(datadir,
+                     "/chlamy236_plus_deNovo_plusOnly_Index", c(17,3,5,7), sep = "")
> rnafiles <- paste(datadir,
+                    "/chlamy236_plus_deNovo_plusOnly_Index", c(10,12,14,16), sep = "")
```

The aligned ribosomal (and RNA) data can be read in using the `readRibodata` function. The columns can be specified as a parameter of the `readRibodata` function if the data in the alignment files are differently arranged.

```
> riboDat <- readRibodata(ribofiles, rnafiles, replicates = c("WT", "WT", "M", "M"))
```

The alignments can be assigned to frames relative to the coding coordinates with the `frameCounting` function.

```
> fCs <- frameCounting(riboDat, fastaCDS)
```

The predominant reading frame, relative to coding start, can be estimated from the frame calling (or from a set of coordinates and alignment data) for each n-mer. The weighting decribes the proportion of n-mers fitting with the most likely frameshift. The reading frame can also be readily visualised using the `plotFS` function.

```
> fS <- readingFrame(rC = fCs); fS

            26    27    28   29   30
          1030  8261 16355 2379 1346
          2847 36011  3582 1634  436
          3352  1687  3331  701  609
frame.ML     2     1     0    0    0
```

```
> plotFS(fS)
```

These can be filtered on the mean number of hits and unique hits within replicate groups to give plausible candidates for coding. Filtering can be limited to given lengths and frames, which may be inferred from the output of the `readingFrame` function.

```
> ffCs <- filterHits(fCs, lengths = c(27, 28), frames = list(0, 2),
+                     hitMean = 50, unqhitMean = 10)
```

We can plot the total alignment at the 5' and 3' ends of coding sequences using the `plotCDS` function. The frames are colour coded; frame-0 is red, frame-1 is green, frame-2 is blue.

```
> plotCDS(coordinates = ffCs@CDS, riboDat = riboDat, lengths = 27)
```

Note the frameshift for 28-mers.

```
> plotCDS(coordinates = ffCs@CDS, riboDat = riboDat, lengths = 28)
```

We can plot the alignment over an individual transcript sequence using the `plotTranscript` function. Observe that one CDS (on the right) contains the 27s in the same phase as the CDS (they are both red) while the putative CDSes to the left are not in phase with the aligned reads, suggesting either a sequence error in the transcript or a misalignment. The coverage of RNA sequenced reads is shown as a black curve (axis on the right).

```
> plotTranscript("CUFF.37930.1", coordinates = ffCs@CDS,
+                 riboData = riboDat, length = 27, cap = 200)

NULL
```

We can extract the counts from a *riboCoding* object using the `sliceCounts` function

```
> riboCounts <- sliceCounts(ffCs, lengths = c(27, 28), frames = list(0, 2))
```

Counts for RNA-sequencing can be extracted using from the riboData object and the coding coordinates using the `rnaCounts` function. This is a relatively crude counting function, and alternatives have been widely described in the literature on mRNA-Seq.

```
> rnaCounts <- rnaCounts(riboDat, ffCs@CDS)
```

These data may be used in an analysis of differential translation through comparison with the RNA-seq data. See the description of a beta-binomial analysis in the *baySeq* vignettes for further details.

```
> library(baySeq)
> pD <- new("pairedData", replicates = ffCs@replicates,
+            data = riboCounts, pairData = rnaCounts,
+            groups = list(NDT = c(1,1,1,1), DT = c("WT", "WT", "M", "M")),
+            annotation = as.data.frame(ffCs@CDS))
> libsizes(pD) <- getLibsizes(pD)
```

```
            26    27    28   29   30
          1030  8261 16355 2379 1346
          2847 36011  3582 1634  436
          3352  1687  3331  701  609
frame.ML     2     1     0    0    0
```

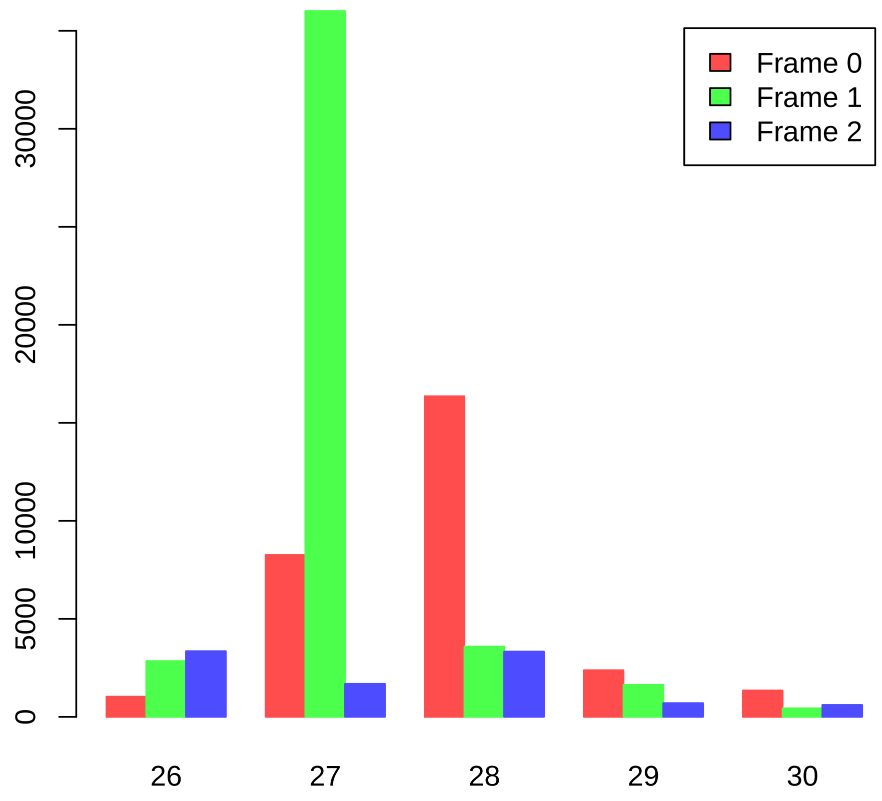

Figure 1: Number of n-mers in each frame relative to coding start. 27-mers are predominantly in frame-1, while 28-mers are chiefly in frame-0.

```
> pD <- getPriors.BB(pD, cl = NULL)
> pD <- getLikelihoods.BB(pD, cl = NULL)

.

> topCounts(pD, "DT", normaliseData = TRUE)

            seqnames start  end width strand frame    WT.1    WT.2     M.1     M.2
1        CUFF.28790.1   172  408   237      *     0  83:108  68:104  29:109  46:114
2        CUFF.37930.1   541  729   189      *     0 374:328 353:344 511:320 419:320
3        CUFF.37930.1   367  489   123      *     0 201:220 213:229 165:248 164:233
4        CUFF.37930.1  2021 2035    15      *     1   48:77   89:66   70:59   86:68
5        CUFF.43721.1   389  571   183      *     1  150:66  103:58   95:64   93:57
6          g17763.t1  1162 1422   261      *     0   63:40   64:31  187:44   52:42
7  Cre06.g281600.t1.2  2058 2657   600      *     2 374:556 136:531 529:244 522:664
8        CUFF.37930.1  1775 1807    33      *     1 147:104 124:111   99:86 188:118
9  Cre09.g396400.t1.2   203  259    57      *     1 249:416 233:281 275:416 248:317
10           g1272.t1  2247 2579   333      *     2   53:56   68:45   55:46   31:28
```

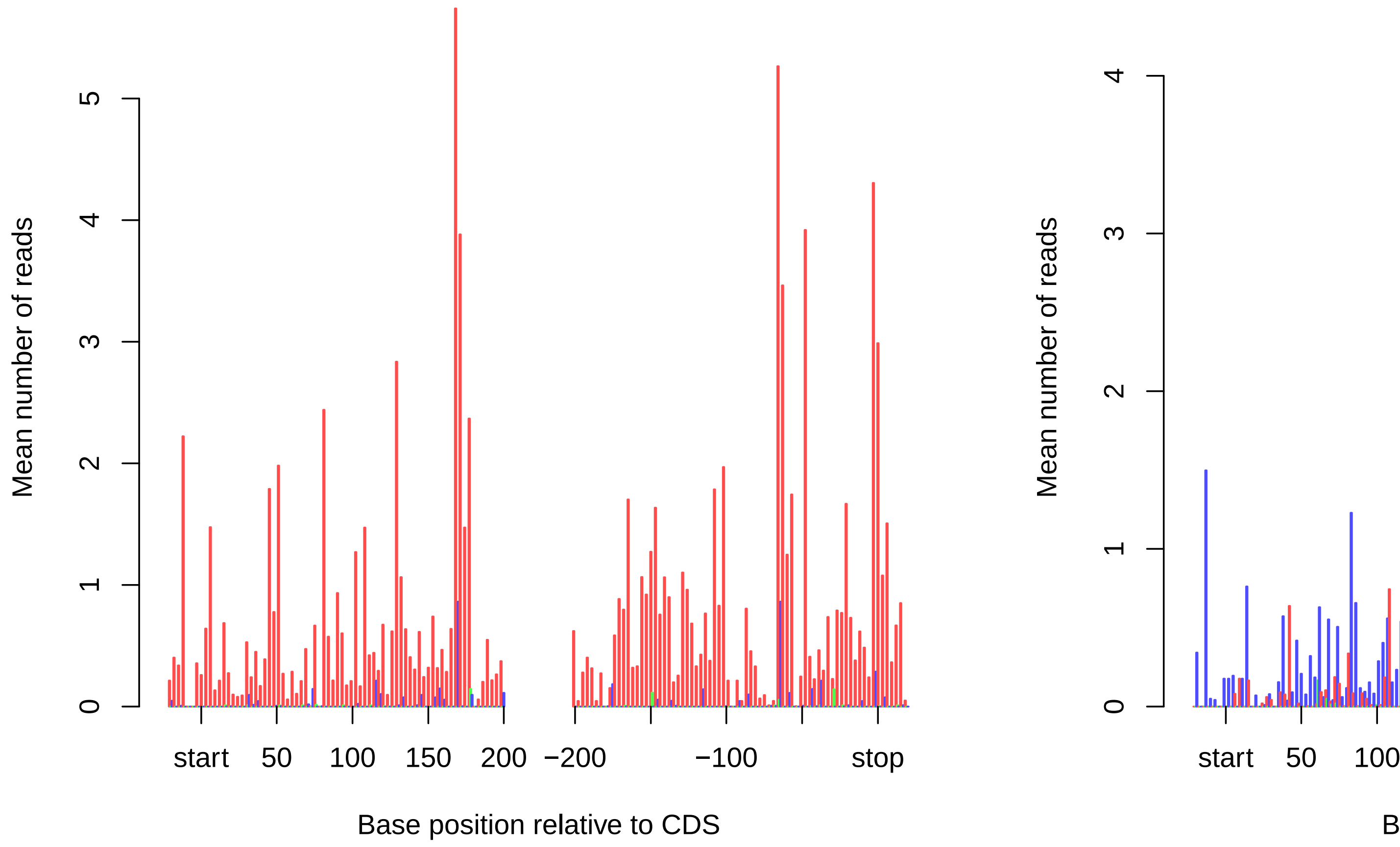

Figure 2: Sum alignment of 27-mers to 5' and 3' ends of coding sequences.

```
   Likelihood   DT    FDR.DT   FWER.DT
1  0.78824073 WT>M 0.2117593 0.2117593
2  0.65975998 M>WT 0.2759996 0.4799503
3  0.55334865 WT>M 0.3328835 0.7122312
4  0.25165270 M>WT 0.4367495 0.9275822
5  0.17403399 WT>M 0.5145928 0.9873968
6  0.14154353 M>WT 0.5719034 0.9982161
7  0.12880985 M>WT 0.6146587 0.9997702
8  0.11584903 M>WT 0.6483452 0.9999734
9  0.10927530 M>WT 0.6752762 0.9999971
10 0.08160985 WT>M 0.6995876 0.9999998
```

Session Info

```
> sessionInfo()
```

```
R version 3.1.1 (2014-07-10)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
```

```
NULL
```

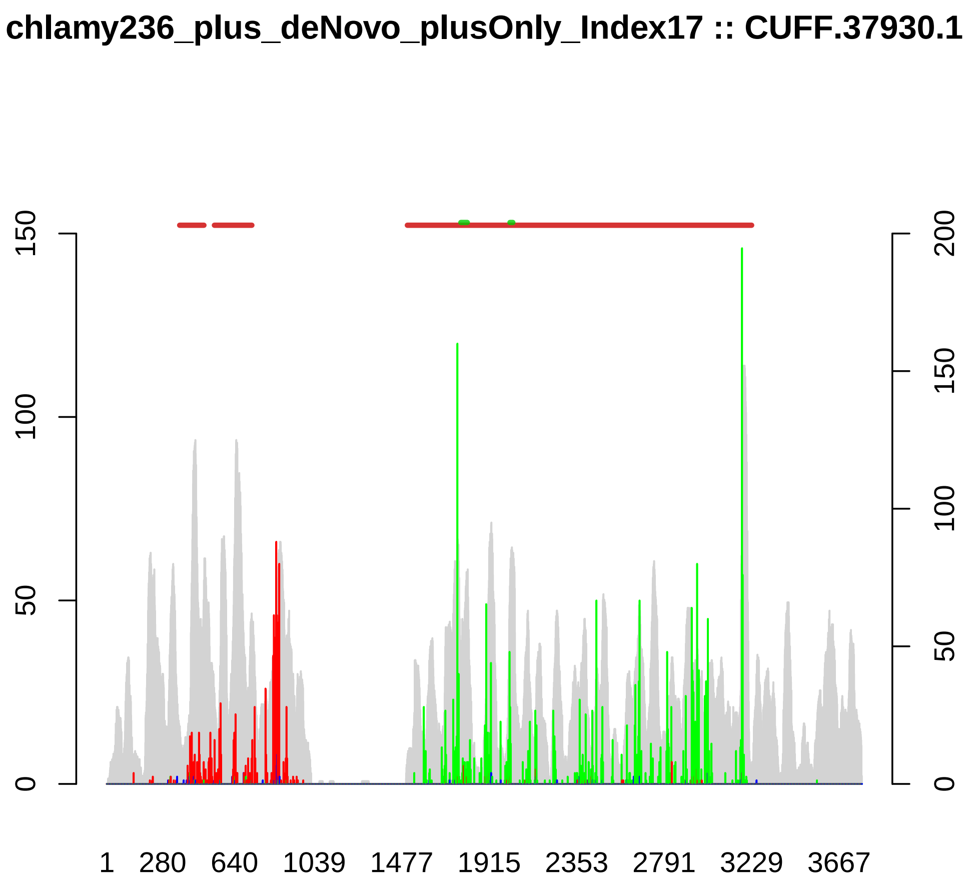

Figure 3: Alignment to individual transcript.

```
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=C               LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] baySeq_1.19.1         riboSeq_0.99.14       abind_1.4-0
[4] GenomicRanges_1.17.35 GenomeInfoDb_1.1.18   IRanges_1.99.24
[7] S4Vectors_0.1.2       BiocGenerics_0.11.4

loaded via a namespace (and not attached):
[1] BiocStyle_1.3.9 XVector_0.5.7   stats4_3.1.1    tools_3.1.1
```