An Introduction to the GenomicAlignments Package

Hervé Pagès

August 18, 2017

Contents

1 Introduction

The *GenomicAlignments* package serves as the foundation for representing genomic alignments within the Bioconductor project. In the Bioconductor package hierarchy, it builds upon the *GenomicRanges* (infrastructure) package and provides support for many Bioconductor packages.

This package defines three classes: *GAlignments*, *GAlignmentPairs*, and *GAlignmentsList*), which are used to represent genomic alignments, pairs of genomic alignments, and groups of genomic alignments.

The *GenomicAlignments* package is available at bioconductor.org and can be downloaded via `biocLite`:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("GenomicAlignments")
```

```
> library(GenomicAlignments)
```

2 *GAlignments*: Genomic Alignments

The *GAlignments* class which is a container for storing a set of genomic alignments. The class is intended to support alignments in general, not only those coming from a 'Binary Alignment Map' or 'BAM' files. Also alignments with gaps in the reference sequence (a.k.a. *gapped alignments*) are supported which, for example, makes the class suited for storing junction reads from an RNA-seq experiment.

More precisely, a *GAlignments* object is a vector-like object where each element describes an *alignment*, that is, how a given sequence (called *query* or *read*, typically short) aligns to a reference sequence (typically long).

As shown later in this document, a *GAlignments* object can be created from a 'BAM' file. In that case, each element in the resulting object will correspond to a record in the file. One important thing to note though

is that not all the information present in the BAM/SAM records is stored in the object. In particular, for now, we discard the query sequences (SEQ field), the query ids (QNAME field), the query qualities (QUAL), the mapping qualities (MAPQ) and any other information that is not needed in order to support the basic set of operations described in this document. This also means that multi-reads (i.e. reads with multiple hits in the reference) don't receive any special treatment i.e. the various SAM/BAM records corresponding to a multi-read will show up in the *GAlignments* object as if they were coming from different/unrelated queries. Also paired-end reads will be treated as single-end reads and the pairing information will be lost. This might change in the future.

2.1 Load a 'BAM' file into a *GAlignments* object

First we use the `readGAlignments` function from the *GenomicAlignments* package to load a toy 'BAM' file into a *GAlignments* object:

```
> library(GenomicAlignments)
> aln1_file <- system.file("extdata", "ex1.bam", package="Rsamtools")
> aln1 <- readGAlignments(aln1_file)
> aln1

GAlignments object with 3271 alignments and 0 metadata columns:
         seqnames strand       cigar    qwidth     start       end
            <Rle>  <Rle> <character> <integer> <integer> <integer>
     [1]     seq1      +         36M        36         1        36
     [2]     seq1      +         35M        35         3        37
     [3]     seq1      +         35M        35         5        39
     [4]     seq1      +         36M        36         6        41
     [5]     seq1      +         35M        35         9        43
     ...      ...    ...         ...       ...       ...       ...
  [3267]     seq2      +         35M        35      1524      1558
  [3268]     seq2      +         35M        35      1524      1558
  [3269]     seq2      -         35M        35      1528      1562
  [3270]     seq2      -         35M        35      1532      1566
  [3271]     seq2      -         35M        35      1533      1567
             width     njunc
         <integer> <integer>
     [1]        36         0
     [2]        35         0
     [3]        35         0
     [4]        36         0
     [5]        35         0
     ...       ...       ...
  [3267]        35         0
  [3268]        35         0
  [3269]        35         0
  [3270]        35         0
  [3271]        35         0
  -------
  seqinfo: 2 sequences from an unspecified genome

> length(aln1)

[1] 3271
```

3271 'BAM' records were loaded into the object.

Note that `readGAlignments` would have discarded any 'BAM' record describing an unaligned query (see description of the <flag> field in the SAM Format Specification [1] for more information). The reader interested in tracking down these events can always use the `scanBam` function but this goes beyond the scope of this document.

2.2 Simple accessor methods

There is one accessor per field displayed by the `show` method and it has the same name as the field. All of them return a vector or factor of the same length as the object:

```
> head(seqnames(aln1))

factor-Rle of length 6 with 1 run
  Lengths:    6
  Values : seq1
Levels(2): seq1 seq2

> seqlevels(aln1)

[1] "seq1" "seq2"

> head(strand(aln1))

factor-Rle of length 6 with 1 run
  Lengths: 6
  Values : +
Levels(3): + - *

> head(cigar(aln1))

[1] "36M" "35M" "35M" "36M" "35M" "35M"

> head(qwidth(aln1))

[1] 36 35 35 36 35 35

> head(start(aln1))

[1]  1  3  5  6  9 13

> head(end(aln1))

[1] 36 37 39 41 43 47

> head(width(aln1))

[1] 36 35 35 36 35 35

> head(njunc(aln1))

[1] 0 0 0 0 0 0
```

[1] http://samtools.sourceforge.net/SAM1.pdf

2.3 More accessor methods

[coming soon...]

3 *GAlignmentPairs*: Pairs of Genomic Alignments

[coming soon...]

4 *GAlignmentsList*: Groups of Genomic Alignments

[coming soon...]

5 Session Information

All of the output in this vignette was produced under the following conditions:

```
> sessionInfo()
```

```
R version 3.4.1 (2017-06-30)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

Matrix products: default

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils
[7] datasets  methods   base

other attached packages:
 [1] GenomicAlignments_1.12.2   Rsamtools_1.28.0
 [3] Biostrings_2.44.2          XVector_0.16.0
 [5] SummarizedExperiment_1.6.3 DelayedArray_0.2.7
 [7] matrixStats_0.52.2         Biobase_2.36.2
 [9] GenomicRanges_1.28.4       GenomeInfoDb_1.12.2
[11] IRanges_2.10.2             S4Vectors_0.14.3
[13] BiocGenerics_0.22.0

loaded via a namespace (and not attached):
 [1] lattice_0.20-35         bitops_1.0-6
 [3] grid_3.4.1              zlibbioc_1.22.0
 [5] Matrix_1.2-11           BiocParallel_1.10.1
 [7] tools_3.4.1             RCurl_1.95-4.8
 [9] compiler_3.4.1          GenomeInfoDbData_0.99.0
```