

# Sequences, Reads, Ranges, and Alignments

Martin Morgan<sup>1</sup>

June 23 – 28, 2013

---

<sup>1</sup>[mtmorgan@fhcrc.org](mailto:mtmorgan@fhcrc.org)

# Overview

1. Sequences – *Biostrings*, *BSgenome*, (*Rsamtools*)
2. Reads – *ShortRead*
3. Ranges – *GenomicRanges*
4. Alignments – *Rsamtools*
5. Working with large data

## Sequences – *Biostrings*

One	Many
<i>DNASTring</i>	<i>DNASTringSet</i>
<i>RNAString</i>	<i>RNAStringSet</i>
<i>AAString</i>	<i>AAStringSet</i>
<i>BString</i>	<i>BStringSet</i>

Operation	Examples
Access	length, nchar
Compare	match, sort
Edit	subseq, translate
Count	alphabetFrequency, consensusMatrix
Match	stringDist, matchPattern pairwiseAlignment, matchPWM
I/O	readDNASTringSet, writeXStringSet

# Whole-genome sequences

## *BSgenome*

- ▶ Pre-built or custom-made – `vignette("BSgenomeForge")`
- ▶ Load chromosome-at-a-time or selected sequences into memory

## *Rsamtools*: `FaFile`

- ▶ Point to standard FASTA file on disk
- ▶ Index – `indexFa`
- ▶ Query – `scanFa`

Tasks?

# Reads – *ShortRead*

```
HWI-EAS299_4_30M2BAAXX:5:1:1513:1024 length=37
GTTTTGTCCAAGTTCTGGTAGCTGAATCCTGGGGCGC
+HWI-EAS299_4_30M2BAAXX:5:1:1513:1024 length=37
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII+HIIII<IE
```

## Input

- ▶ `readFastq`
- ▶ `FastqSampler & yield; FastqStreamer & yield`
- ▶ `trimTails, ...`

## Manipulation

- ▶ `sread, quality, ...`
- ▶ `frequentSequences, alphabetByCycle, ...`
- ▶ `qa, report`

# Ranges – *GenomicRanges*

## *GRanges*

- ▶ Ranges in genomic space – seqname, start / end / width, strand (+, -, \*)
- ▶ Annotation (e.g., regulatory regions, genes, exons) or data (e.g., reads, peaks, variants)

## Operations

**Intra-range** flank, narrow, ...

**Inter-range** range, disjoint, gaps, coverage, ...

**Between** countOverlaps, findOverlaps, summarizeOverlaps  
(*GenomicRanges*), ...

# Ranges – *GenomicRanges*

## *GRangesList*

- ▶ List-like, where all elements are *GRanges*
- ▶ Annotation (e.g., exons within genes) or data (e.g., gapped alignments)

Advanced: derived from *IRanges* package *CompressedList* class

- ▶ *CompressedList*: *many* homogenous elements
- ▶ Internally, a single *GRanges* and a 'partitioning' describing how elements of the *GRanges* are separated into groups.
- ▶ Some operations (e.g., `unlist`, `relist`) very fast.
- ▶ Common pattern for implementation: `unlist`, `transform`, `relist`.

Tasks?

# Alignments – *Rsamtools*

## *GappedAlignments* (*GAlignments*)

- ▶ `readGappedAlignments` (`readGAlignments`)
- ▶ (`Bioc-devel`: `readGAlignmentsList`)
- ▶ Also: `scanBam`, `asBam`, `indexBam`, `quickCountBam`, ...

## *BamFile* / *BamFileList*

- ▶ A *type* (for method selection, e.g., `coverage`) and `yieldSize`

## *ScanBamParam*: optional argument to input functions

- ▶ `what`: (additional) fields to return
- ▶ `which`: restrict query to specific *GRanges*
- ▶ `flag`: restrict read input, e.g., marked as passing quality control, not optical duplicates, plus strand
- ▶ ...

Tasks?



# Working with large data

1. **Restriction**
2. Sampling
3. Iteration
4. Parallel evaluation

```
library(Rsamtools)
gr <- GRanges("chr7",
              IRanges(100000, width=100))
param <- ScanBamParam(
  what = c("rname", "pos", "cigar"),
  which = gr)
scanBam("a.bam", param = param)
```

# Working with large data

1. Restriction

2. **Sampling**

3. Iteration

4. Parallel evaluation

```
library(ShortRead)
samp <- FastqSampler("end1.fastq")
yield(samp)
set.seed(123); yield(samp)
set.seed(123); yield(samp)
```

# Working with large data

1. Restriction
2. Sampling
3. **Iteration**
4. Parallel evaluation

```
library(Rsamtools)
bf <-
  BamFile("a.bam", yieldSize=1e6)
open(bf)
repeat {
  gl <- readGAlignments(bf)
  if (length(gl) == 0)
    break;
  ## do work
}
close(bf)
```

# Working with large data

1. Restriction

2. Sampling

3. Iteration

4. **Parallel evaluation**

```
library(parallel)
options(mc.cores=detectCores)
fls <- c("a.bam", "b.bam")
```

```
## sequential
x0 <- lapply(fls, countBam)
```

```
## parallel, 1 core per BAM file
x1 <- mclapply(fls, countBam)
```

```
identical(x0, x1) # TRUE
```

Other parallel solutions possible, e.g.,  
clusters, Windows

# Acknowledgements

*IRanges, GenomicRanges*

- ▶ Michael Lawrence
- ▶ Hervé Pagès
- ▶ Patrick Aboyoun
- ▶ Valerie Obenchain

*Biostrings, BSgenome*

- ▶ Hervé Pagès

And...

- ▶ Marc Carlson
- ▶ Paul Shannon
- ▶ Dan Tenenbaum

*Bioconductor* mailing list