`GenomicTuples` and DNA methylation patterns

Peter Hickey (@PeteHaitch) - Walter and Eliza Hall Institute of Medical Research

European Bioconductor Developers' Meeting, 12 January 2015

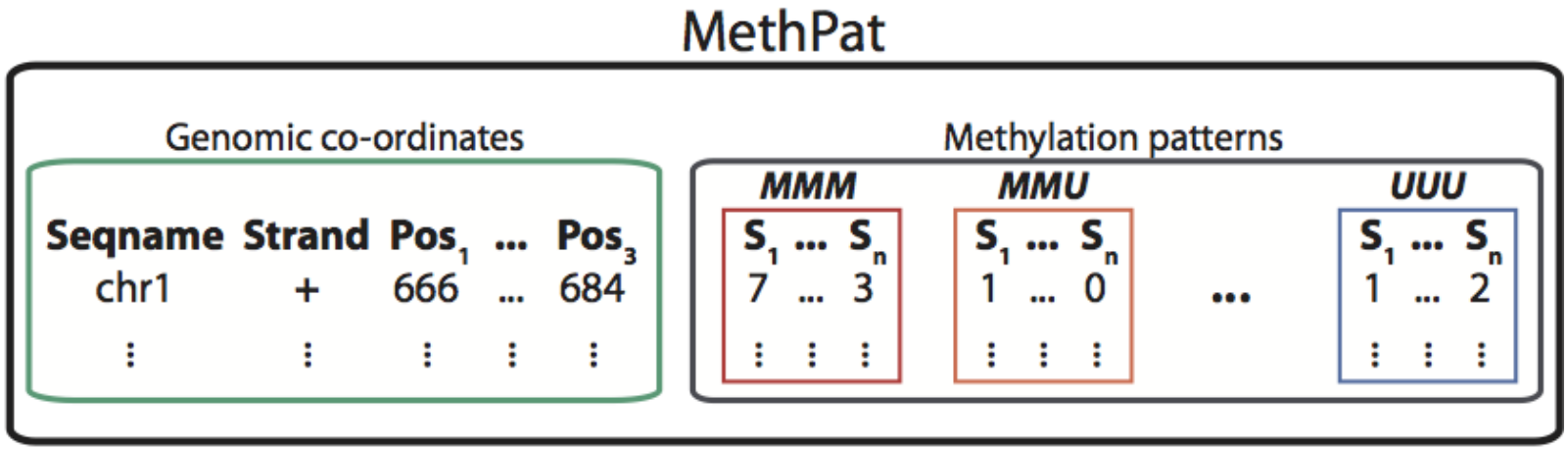
Motivation

- Analysing counts of methylation patterns at genomic tuples
- Counts extracted from `BAM` file using `methtuple` (https://github.com/PeteHaitch/methtuple; Python)

Example output of `methtuple` for 3-tuples

chr	strand	pos1	pos2	pos3	MMM	MMU	MUM	MUU	UMM	UMU	UUM	UUU
chr1	+	781154	781161	781190	4	1	0	0	0	0	0	0
chr1	+	781362	781406	781455	0	0	1	1	0	0	0	0
chr1	+	781616	781720	781732	0	0	1	0	0	1	1	1
chr1	+	781616	781763	781795	0	0	0	0	1	0	0	0
chr1	+	781720	781732	781738	0	1	2	1	4	0	1	0
chr1	+	781732	781738	781763	3	0	0	1	0	2	1	0
chr1	+	781738	781763	781795	0	0	0	0	0	1	0	0
chr1	+	781738	781763	781912	0	1	0	0	0	0	0	0
chr1	+	781763	781795	781912	0	0	0	1	0	0	1	0
chr1	+	781912	781989	782013	1	0	1	1	0	0	1	0
chr1	+	781912	782013	782024	3	0	0	0	0	0	0	0
chr1	+	781989	782013	782024	2	0	3	0	3	0	3	0
chr1	+	782013	782024	782048	2	2	0	0	3	2	0	0
chr1	+	782236	782243	782268	1	0	1	0	0	1	0	0

Aim



`MethPat` implemented in **`MethylationTuples`**

- `MethPat` extends `GenomicRanges::SummarizedExperiment`

Genomic tuples

chr	strand	pos1	pos2	pos3
chr1	+	781154	781161	781190
chr1	+	781362	781406	781455
chr1	+	781616	781720	781732
chr1	+	781616	781763	781795
chr1	+	781720	781732	781738

`GenomicTuples`

- Extend `GenomicRanges` to *genomic tuples*
- Retains a familiar interface

GTuples

```
library(GenomicTuples)
# Create a GTuples object with two 3-tuples
seqinfo <- Seqinfo("chr1", 1000, NA, "toy")
gt <- GTuples(seqnames = 'chr1',
              tuples = matrix(c(1L, 5L, 5L, 10L, 10L, 20L), ncol = 3),
               strand = "+",
               seqinfo = seqinfo)
gt
```

```
># GTuples object with 2 x 3-tuples and 0 metadata columns:
>#       seqnames pos1 pos2 pos3 strand
>#   [1]     chr1    1    5   10      +
>#   [2]     chr1    5   10   20      +
>#   ---
>#   seqinfo: 1 sequence from toy genome
```

GTuples extends GRanges

```
setClass("GTuples",
         contains = "GRanges",
         representation(
           internalPos = "matrixOrNULL",
           size = "integer"),
         prototype(
           internalPos = NULL,
           size = NA_integer_)
)

# Ensure the internalPos slot "sticks" during subsetting, etc.
setMethod(GenomicRanges:::extraColumnSlotNames,
          "GTuples",
          function(x) {
            c("internalPos")
          }
)
```

Useful `GTuples` methods (inherited)

```
seqnames(gt)
```

```
>#  factor-Rle of length 2 with 1 run
>#    Lengths:    2
>#    Values : chr1
>#  Levels(1): chr1
```

```
strand(gt)
```

```
>#  factor-Rle of length 2 with 1 run
>#    Lengths: 2
>#    Values : +
>#  Levels(3): + - *
```

Useful **`GTuples`** methods (new)

```
size(gt)

>#  [1] 3

tuples(gt)

>#       pos1 pos2 pos3
>#  [1,]    1    5   10
>#  [2,]    5   10   20

IPD(gt) # IPD = intra-pair distances

>#       [,1] [,2]
>#  [1,]    4    5
>#  [2,]    5   10
```

Ill-defined `GTuples` methods

These return errors

- `coverage`
- `flank`, `promoters`, `resize`, `narrow`
- `disjoin`, `gaps`, `isDisjoint`, `range`, `reduce`
- `mapCoords`
- `Ops`, `intersect`, `pgap`, `pintersect`, `psetdiff`, `punion`, `setdiff`, `union`, `tile`

Meaningful definitions (and pull requests) are welcomed!

GTuples comparison and sorting

```
# Sorted first by seqnames, then by strand, then by tuples
sort(gt3)
```

```
>#  GTuples object with 7 x 3-tuples and 0 metadata columns:
>#        seqnames pos1 pos2 pos3 strand
>#    [1]     chr1    5   20   30      +
>#    [2]     chr1   10   20   30      +
>#    [3]     chr1   10   20   35      +
>#    [4]     chr1   10   25   30      +
>#    [5]     chr1   10   20   30      -
>#    [6]     chr1   10   20   35      *
>#    [7]     chr2   10   20   30      +
>#    ---
>#    seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

`findOverlaps`-based methods

```
if (size < 3) {
  # Treat GTuples as GRanges
} else {
  if (type == "equal") {
    # Call .findEqual.GTuples()
  } else {
    # Treat GTuples as GRanges
  }
}
```

`GenomicTuples` summary

A drop in replacement for `GenomicRanges` when you have genomic *tuples* rather than *ranges*.

Limitations

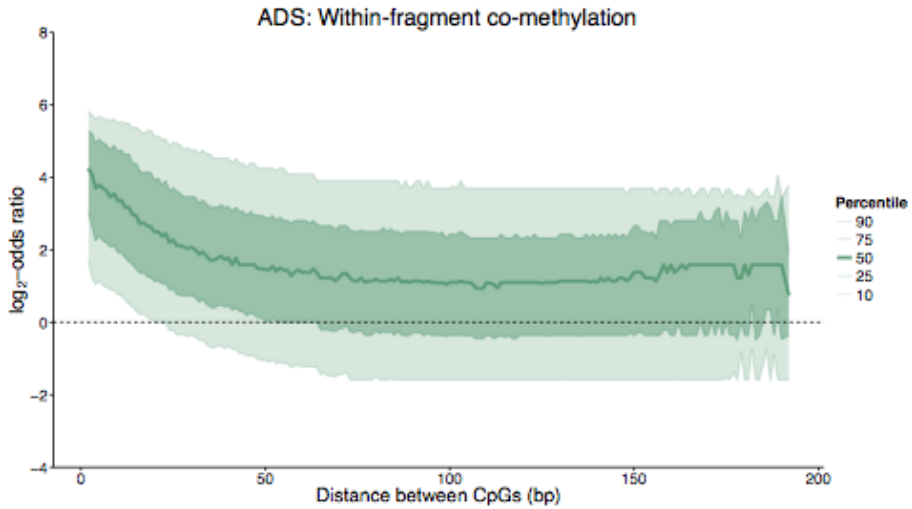
- All tuples in a `GTuples` object must have same `size`
- Room for improvement with `findOverlaps(x, y, type = 'equal')`
  - Performance
  - Not all options supported (e.g., `maxgap` and `minoverlap`)

MethylationTuples

An R package for analysing, managing and visualising methylation patterns at genomic tuples.

Analyses

- Epialleles
- Methylation entropy
- Allele-specific methylation
- Co-methylation



`MethylationTuples` development

- Adding additional features and tests, improving documentation and adding vignette
- **Performance**: `MethPat` objects become increasingly sparse as `size` increases (and as $n_{samples}$ increases)

12 whole-genome bisulfite-sequencing samples

	`pryr::object_size(x)`	`nrow`	Number of assays	Percentage of `NA` and 0 values
1-tuples	5.9 GB	56,348,522	2	28%
2-tuples	20.1 GB	100,586,237	4	80%
3-tuples	43.3 GB	109,376,348	8	93%
4-tuples	80.5 GB	102,625,758	16	97%

Thanks

PhD advisors

- Terry Speed
- Peter Hall

Programming

- Hervé Pagès
- Martin Morgan
- Michael Lawrence
- R/BioC community

Funding

- Edith Moffat Travel Award

Links

- Slides.Rmd (https://github.com/PeteHaitch/BiocEurope_2015_presentation)
- GitHub: @PeteHaitch
    - `GenomicTuples` (release)
    - `GenomicTuples` (GitHub devel)
    - `MethylationTuples` (GitHub devel)
- Twitter: @PeteHaitch