

Using Iterators in SeqVarTools

Stephanie M. Gogarten

May 2, 2019

Contents

1	Introduction	1
2	Block iterators.	1
3	Range iterators	3
4	Window iterators	4
5	List iterators	6

1 Introduction

Iterators can be used to apply a user function to a *SeqVarData* object. Creating the iterator defines the sets of variants returned on every subsequent call to `iterateFilter`. `iterateFilter` returns TRUE if there are more variants remaining, and FALSE when all variants have been returned.

2 Block iterators

The simplest type of iterator, a *SeqVarBlockIterator*, returns variants in consecutive blocks.

```
> library(SeqVarTools)
> gds <- seqOpen(seqExampleFileName("gds"))
> seqData <- SeqVarData(gds)
> iterator <- SeqVarBlockIterator(seqData, variantBlock=500)

# of selected variants: 500

> var.info <- list(variantInfo(iterator))
> i <- 2
```

Using Iterators in SeqVarTools

```
> while(iterateFilter(iterator)) {  
+   var.info[[i]] <- variantInfo(iterator)  
+   i <- i + 1  
+ }  
  
# of selected variants: 500  
# of selected variants: 348  
# of selected variants: 0  
  
> lapply(var.info, head)  
  
[[1]]  
  variant.id chr      pos ref alt  
1          1  1 1105366   T   C  
2          2  1 1105411   G   A  
3          3  1 1110294   G   A  
4          4  1 3537996   T   C  
5          5  1 3538692   G   C  
6          6  1 3541597   C   T  
  
[[2]]  
  variant.id chr      pos ref alt  
1         501  7 48109509   C   T  
2         502  7 72486435   G   C  
3         503  7 72487157   A   C  
4         504  7 72487452   G   A  
5         505  7 94872711   G   C  
6         506  7 94878932   G   A  
  
[[3]]  
  variant.id chr      pos ref alt  
1        1001 17 30793147   G   A  
2        1002 17 30793151   C   A  
3        1003 17 30795794   G   A  
4        1004 17 30796109   G   A  
5        1005 17 30796359   C   T  
6        1006 17 30796663   C   T  
  
> seqResetFilter(seqData)  
  
# of selected samples: 90  
# of selected variants: 1,348
```

A filter can be applied before the iterator is created, and only variants included in the filter will be returned by the iterator.

Using Iterators in SeqVarTools

```
> seqSetFilter(seqData, variant.sel=1:100)
# of selected variants: 100
> iterator <- SeqVarBlockIterator(seqData, variantBlock=500)
# of selected variants: 100
> var.info <- variantInfo(iterator)
> nrow(var.info)
[1] 100
> iterateFilter(iterator)
# of selected variants: 0
[1] FALSE
> seqResetFilter(seqData)
# of selected samples: 90
# of selected variants: 1,348
```

3 Range iterators

A *GRanges* object can be used to create a *SeqVarRangeIterator*, where every iteration returns the next range.

```
> library(GenomicRanges)
> gr <- GRanges(seqnames=rep(1,3),
+               ranges=IRanges(start=c(1e6, 2e6, 3e6), width=1e6))
> iterator <- SeqVarRangeIterator(seqData, variantRanges=gr)
# of selected variants: 3
> var.info <- list(variantInfo(iterator))
> i <- 2
> while(iterateFilter(iterator)) {
+   var.info[[i]] <- variantInfo(iterator)
+   i <- i + 1
+ }
# of selected variants: 0
# of selected variants: 4
# of selected variants: 0
> lapply(var.info, head)
[[1]]
  variant.id chr      pos ref alt
```

Using Iterators in SeqVarTools

```
1      1  1 1105366  T  C
2      2  1 1105411  G  A
3      3  1 1110294  G  A
```

```
[[2]]
[1] variant.id chr      pos
<0 rows> (or 0-length row.names)
```

```
[[3]]
      variant.id chr      pos ref alt
1          4    1 3537996   T    C
2          5    1 3538692   G    C
3          6    1 3541597   C    T
4          7    1 3541652   G    A
```

```
> seqResetFilter(seqData)
```

```
# of selected samples: 90
```

```
# of selected variants: 1,348
```

4 Window iterators

Window iterators (*SeqVarWindowIterator*) are a special class of range iterators. When the object is created, the ranges are generated automatically with a specified width and step size, covering the entire genome.

```
> seqSetFilterChrom(seqData, include="22")
```

```
# of selected variants: 23
```

```
> iterator <- SeqVarWindowIterator(seqData, windowSize=10000,
+                                 windowShift=5000)
```

```
# of selected variants: 2
```

```
> var.info <- list(variantInfo(iterator))
```

```
> i <- 2
```

```
> while(iterateFilter(iterator)) {
+   var.info[[i]] <- variantInfo(iterator)
+   i <- i + 1
+ }
```

```
# of selected variants: 3
```

```
# of selected variants: 1
```

```
# of selected variants: 1
```

```
# of selected variants: 2
```

Using Iterators in SeqVarTools

```
# of selected variants: 4
# of selected variants: 1
# of selected variants: 2
# of selected variants: 1
# of selected variants: 2
# of selected variants: 1
# of selected variants: 1
# of selected variants: 4
# of selected variants: 1
# of selected variants: 0

> lapply(var.info, head)

[[1]]
  variant.id chr      pos ref alt
1      1326  22 16042444  C   G
2      1327  22 16042793  A   G

[[2]]
  variant.id chr      pos ref alt
1      1326  22 16042444  C   G
2      1327  22 16042793  A   G
3      1328  22 16049306  T   C

[[3]]
  variant.id chr      pos ref alt
1      1328  22 16049306  T   C

[[4]]
  variant.id chr      pos ref alt
1      1329  22 17729354  G   A

[[5]]
  variant.id chr      pos ref alt
1      1330  22 18338811  C   T
2      1331  22 18338829  G   A

[[6]]
  variant.id chr      pos ref alt
1      1332  22 18348971  G   A
2      1333  22 18349075  A   G
3      1334  22 18349106  A   G
4      1335  22 18349495  G   T

[[7]]
  variant.id chr      pos ref alt
```

Using Iterators in SeqVarTools

```
1      1336  22 20328280   G   A

[[8]]
  variant.id chr      pos ref alt
1      1337  22 32000584   G   A
2      1338  22 32003125   C T,AT

[[9]]
  variant.id chr      pos ref alt
1      1339  22 32330460   G   A

[[10]]
  variant.id chr      pos ref alt
1      1340  22 38747766   A   G
2      1341  22 38747889   G   A

[[11]]
  variant.id chr      pos ref alt
1      1342  22 43657667   G   A

[[12]]
  variant.id chr      pos ref alt
1      1343  22 43670607   C   A

[[13]]
  variant.id chr      pos ref alt
1      1344  22 43690908   G   A
2      1345  22 43690970   C   T
3      1346  22 43691009   C   T
4      1347  22 43691073   G   A

[[14]]
  variant.id chr      pos ref alt
1      1348  22 48958933   A   G

> seqResetFilter(seqData)

# of selected samples: 90
# of selected variants: 1,348
```

5 List iterators

A *SeqVarListIterator* can be used to specify particular variants to include in each iteration. The input is a *GRangesList*, and each list element defines an iteration set.

Using Iterators in SeqVarTools

```
> gr <- GRangesList(
+   GRanges(seqnames=rep(22,2),
+     ranges=IRanges(start=c(16e6, 17e6), width=1e6)),
+   GRanges(seqnames=rep(22,2),
+     ranges=IRanges(start=c(18e6, 20e6), width=1e6)))
> iterator <- SeqVarListIterator(seqData, variantRanges=gr)

# of selected variants: 4

> var.info <- list(variantInfo(iterator))
> i <- 2
> while(iterateFilter(iterator)) {
+   var.info[[i]] <- variantInfo(iterator)
+   i <- i + 1
+ }

# of selected variants: 7
# of selected variants: 0

> lapply(var.info, head)

[[1]]
  variant.id chr      pos ref alt
1      1326  22 16042444   C   G
2      1327  22 16042793   A   G
3      1328  22 16049306   T   C
4      1329  22 17729354   G   A

[[2]]
  variant.id chr      pos ref alt
1      1330  22 18338811   C   T
2      1331  22 18338829   G   A
3      1332  22 18348971   G   A
4      1333  22 18349075   A   G
5      1334  22 18349106   A   G
6      1335  22 18349495   G   T
```

After the last iteration, any methods used on the iterator object will return 0 variants. The `resetIterator` method can be used to reset an iterator back to the beginning.

```
> variantInfo(iterator)

[1] variant.id chr      pos
<0 rows> (or 0-length row.names)

> resetIterator(iterator)

# of selected variants: 4

> variantInfo(iterator)
```

Using Iterators in SeqVarTools

	variant.id	chr	pos	ref	alt
1	1326	22	16042444	C	G
2	1327	22	16042793	A	G
3	1328	22	16049306	T	C
4	1329	22	17729354	G	A

```
> seqClose(gds)
```