

The R package *trioClasses* for definition of the class *SNPTrioExperiment*, an extension of *SummarizedExperiment*, for use in trio based analyses of genetic data.

Samuel G. Younkin

March 12, 2013

## 1 Packages

```
> library("trioClasses")
> data("sample")
> data("8q24-european-all.sm")
> pos <- as(do.call("rbind", strsplit(colnames(sm), split = ":"))[,
  2], "integer")
> chr <- do.call("rbind", strsplit(colnames(sm), split = ":"))[,
  1]
> gr <- GRanges(seqnames = chr, ranges = IRanges(start = pos,
  width = 1), strand = "*")
> names(gr) <- colnames(sm)
> col.DF <- col.DF[col.DF$id %in% rownames(sm), ]
> table(as(sm, "numeric"), useNA = "ifany")
```

	0	1	2	<NA>
7630997	519922	265944	176097	

## 2 Sample Data

```
> se <- SummarizedExperiment(assays = SimpleList(geno = t(sm)),
  colData = col.DF, rowData = gr)
```

### 2.1 Pedigree

```
> ped <- PedClass(ped.DF)
```

### 2.2 SNPTrioExperiment

```
> ste <- SNPTrioExperiment(se, pedigree = ped)
```

```
> ste
```

```
class: SNPTrioExperiment
dim: 8951 960
```

```

exptData(0):
assays(1): geno
rownames(8951): chr8:129296000 chr8:129296113 ...
chr8:130354703 chr8:130354790
rowData metadata column names(0):
colnames(960): H_ME-CSc175_1-CSc175_1_a.1
H_ME-CSc175_2-CSc175_2 ... H_ME-DS11418_2-DS11418_2
H_ME-DS11418_3-DS11418_3
colData names(28): vcfid jb_famid ... id ped
pedigree(4139): famid id fid mid sex dx

```

```

> gtrio <- trioClasses:::GenoTrio(ste)
> F <- as(with(gtrio, F), "numeric")
> M <- as(with(gtrio, M), "numeric")
> O <- as(with(gtrio, O), "numeric")
> table(paste0(F, M, O))

```

	000	001	00NA	011	012	01NA	0NA0	0NA1
2418678	39837	13147	40206	13162	2967	3230	2292	
	0NA2	0NANA	100	101	10NA	110	111	112
1781	8806	39815	11931	1297	42773	27822	12971	
	11NA	121	122	12NA	1NA0	1NA1	1NA2	1NANA
2691	13050	12212	1353	2443	1830	1169	1932	
	210	211	212	21NA	221	222	22NA	2NA0
12460	12213	1	967	13432	38076	3554	1664	
	2NA1	2NA2	2NANA	NA00	NA01	NA0NA	NA10	NA11
1121	1339	3153	12224	1256	3207	2902	2392	
	NA12	NA1NA	NA21	NA22	NA2NA	NANA0	NANA1	NANA2
917	1376	1348	3360	3384	9310	2083	3207	
	NANANA							
11979								

## 3 Methods

### 3.1 ScanTrio

```

> (scan.trio <- ScanTrio(object = ste, window = rowData(ste[1:10]),
  block = range(rowData(ste))))

```

	minor.win	major.win	mendel.win	minor.out	major.out
chr8:129296000	82	82	0	162463	156438
chr8:129296113	2	0	0	162543	156520
chr8:129296185	2	0	0	162543	156520
chr8:129296191	0	1	0	162545	156519
chr8:129296198	93	80	0	162452	156440
chr8:129296209	0	1	0	162545	156519
chr8:129296289	2	0	0	162543	156520
chr8:129296343	1	0	0	162544	156520
chr8:129296434	20	21	0	162525	156499
chr8:129296732	0	1	0	162545	156519
	mendel.out				

```
chr8:129296000      1
chr8:129296113      1
chr8:129296185      1
chr8:129296191      1
chr8:129296198      1
chr8:129296209      1
chr8:129296289      1
chr8:129296343      1
chr8:129296434      1
chr8:129296732      1
```

## 3.2 Holger-style Genotype Matrix

Coercion to matrix for *trio*

```
> geno <- as(ste, "matrix")
> aTDT(geno[, 1:5])
```

```
$stat
[1] 24.64103 37.09756 30.11765 32.40000 14.22581
```

```
$pval
[1] 6.906563e-07 1.123642e-09 4.066142e-08 1.254865e-08 1.621318e-04
```

```
$transMinor
[1] 4 1 1 2 5
```

```
$transMajor
[1] 35 40 33 38 26
```

Or apply the aTDT method to the SNP TrioExperiment directly.

```
> aTDT(ste[1:5])
```

```
$stat
[1] 16.33333 11.20000 19.00000 28.59459 28.80000
```

```
$pval
[1] 5.312128e-05 8.179733e-04 1.307185e-05 8.923093e-08 8.025111e-08
```

```
$transMinor
[1] 20 21 19 14 16
```

```
$transMajor
[1] 55 49 57 60 64
```

## A Count of Transmission of Variants (*TransCount()*)

### A.1 Window

```
> window <- gr[100] + 10000
> window2 <- gr[100] + 1000
> block <- range(rowData(ste))

> TransCount(ste, window)
```

```
$minor
[1] 4007
```

```
$major
[1] 4249
```

```
$mendel
[1] 0
```

### A.2 Not in window, but in block

```
> TransCount(ste, setdiff(block, window))
```

```
$minor
[1] 158538
```

```
$major
[1] 152271
```

```
$mendel
[1] 1
```

### A.3 In the whole block

```
> TransCount(ste, block)
```

```
$minor
[1] 162545
```

```
$major
[1] 156520
```

```
$mendel
[1] 1
```

### A.4 In both windows

```
> TransCount(ste, GRangesList(window, window2))
```

```
$minor
[1] 4007 468
```

```
$major  
[1] 4249 521
```

```
$mendel  
[1] 0 0
```

## A.5 Not “In both windows,” but in block

```
> TransCount(ste, GRangesList(setdiff(block, window), setdiff(block,  
  window2)))
```

```
$minor  
[1] 158538 162077
```

```
$major  
[1] 152271 155999
```

```
$mendel  
[1] 1 1
```

## B Count of Transmission of Rare Variants (*TransCount()*)

```
> ste.rare <- ste[MAF(ste) <= 0.01]
```

### B.1 In both windows

```
> TransCount(ste.rare, GRangesList(window, window2))
```

```
$minor  
[1] 105 8
```

```
$major  
[1] 131 15
```

```
$mendel  
[1] 0 0
```

### B.2 Not “In both windows,” but in block

```
> TransCount(ste.rare, GRangesList(setdiff(block, window),  
  setdiff(block, window2)))
```

```
$minor  
[1] 5356 5453
```

```
$major  
[1] 6166 6282
```

```
$mendel  
[1] 0 0
```

## C Classes

### C.1 SnpMatrix, DataFrame, GRanges

The four key ingredients are the SNP matrix, the pedigree information as a `DataFrame`, position of the SNPs given by a `GRanges` object, and covariate data given as a `DataFrame`.

```
> sm
```

```
A SnpMatrix with 960 rows and 8951 columns
Row names: H_ME-DS10776_2-DS10776_2 ... H_ME-DS11313_1-DS11313_1
Col names: chr8:129296000 ... chr8:130354790
```

```
> ped.DF
```

```
DataFrame with 4139 rows and 6 columns
```

	famid		id
	<character>		<character>
1	4778		H_ME-4778_1-4778_1.2
2	4778		H_ME-4778_2-4778_2.2
3	4778		H_ME-4778_3-4778_3.2
4	4783		H_ME-4783_1-4783_1.1
5	4783		H_ME-4783_2-4783_2.1
6	4783		H_ME-4783_3-4783_3.1
7	20000492	H_ME-20000492_1-20000492_1.1	
8	20000492	H_ME-20000492_2-20000492_2.a.1	
9	20000492	H_ME-20000492_3-20000492_3.1	
...	...		...
4131	DS12332		H_ME-DS12332_1-DS12332_1
4132	DS12332		H_ME-DS12332_2-DS12332_2
4133	DS12332		H_ME-DS12332_3-DS12332_3
4134	DS99998	H_ME-DS10193_5-DS10193_5.2	
4135	DS99998	H_ME-DS10193_6-DS10193_6	
4136	DS99998	H_ME-DS10193_7-DS10193_7	
4137	DS99999	H_ME-DS10707_4-DS10707_4.2	
4138	DS99999	H_ME-DS10707_5-DS10707_5	
4139	DS99999	H_ME-DS10707_6-DS10707_6	
	famid		mid
	<character>		<character>
1	H_ME-4778_2-4778_2.2		H_ME-4778_3-4778_3.2
2	NA		NA
3	NA		NA
4	H_ME-4783_2-4783_2.1		H_ME-4783_3-4783_3.1
5	NA		NA
6	NA		NA
7	H_ME-20000492_2-20000492_2.a.1	H_ME-20000492_3-20000492_3.1	
8	NA		NA
9	NA		NA
...	...		...
4131	H_ME-DS12332_2-DS12332_2	H_ME-DS12332_3-DS12332_3	
4132	NA		NA
4133	NA		NA

4134	H_ME-DS10193_6-DS10193_6	H_ME-DS10193_7-DS10193_7
4135	NA	NA
4136	NA	NA
4137	NA	NA
4138	H_ME-DS10707_4-DS10707_4.2	H_ME-DS10707_6-DS10707_6
4139	NA	NA

	sex	dx
	<numeric>	<numeric>
1	1	1
2	1	0
3	2	0
4	2	1
5	1	0
6	2	1
7	2	1
8	1	0
9	2	0
...	...	...
4131	1	1
4132	1	0
4133	2	0
4134	2	1
4135	1	0
4136	2	0
4137	1	0
4138	1	1
4139	2	0

> gr

GRanges with 8951 ranges and 0 metadata columns:

	seqnames	ranges	strand
	<Rle>	<IRanges>	<Rle>
chr8:129296000	chr8 [129296000, 129296000]	*	
chr8:129296113	chr8 [129296113, 129296113]	*	
chr8:129296185	chr8 [129296185, 129296185]	*	
chr8:129296191	chr8 [129296191, 129296191]	*	
chr8:129296198	chr8 [129296198, 129296198]	*	
chr8:129296209	chr8 [129296209, 129296209]	*	
chr8:129296289	chr8 [129296289, 129296289]	*	
chr8:129296343	chr8 [129296343, 129296343]	*	
chr8:129296434	chr8 [129296434, 129296434]	*	
...	...	...	
chr8:130353593	chr8 [130353593, 130353593]	*	
chr8:130353671	chr8 [130353671, 130353671]	*	
chr8:130354142	chr8 [130354142, 130354142]	*	
chr8:130354182	chr8 [130354182, 130354182]	*	
chr8:130354239	chr8 [130354239, 130354239]	*	
chr8:130354240	chr8 [130354240, 130354240]	*	
chr8:130354296	chr8 [130354296, 130354296]	*	
chr8:130354703	chr8 [130354703, 130354703]	*	
chr8:130354790	chr8 [130354790, 130354790]	*	

```

---
seqlengths:
  chr8
  NA

```

## C.2 SummarizedExperiment

We combine three of the key ingredients when we create the SummarizedExperiment object.

```

> se <- SummarizedExperiment(assays = SimpleList(geno = t(sm)),
  colData = col.DF, rowData = gr)

```

## C.3 SNP TrioExperiment

Now, we include the pedigree information as an object of class PedClass. We keep PedClass independent of SNP TrioExperiment for flexibility.

```

> ped <- PedClass(ped.DF)
> ste <- SNP TrioExperiment(se, pedigree = ped)

```

Here is the show method.

```

> ste

class: SNP TrioExperiment
dim: 8951 960
exptData(0):
assays(1): geno
rownames(8951): chr8:129296000 chr8:129296113 ...
  chr8:130354703 chr8:130354790
rowData metadata column names(0):
colnames(960): H_ME-CSc175_1-CSc175_1_a.1
  H_ME-CSc175_2-CSc175_2 ... H_ME-DS11418_2-DS11418_2
  H_ME-DS11418_3-DS11418_3
colData names(28): vcfid jb_famid ... id ped
pedigree(4139): famid id fid mid sex dx

```

And now we verify that it is indeed an extension of SummarizedExperiment.

```

> getClass("SNP TrioExperiment")

```

```

Class "SNP TrioExperiment" [package "trioClasses"]

```

Slots:

Name:	pedigree	exptData
Class:	PedClass	SimpleList
Name:	rowData	colData
Class:	GenomicRangesORGRangesList	DataFrame
Name:	assays	



```
Class: Assays
```

```
Extends: "SummarizedExperiment"
```

## C.4 PedClass

Now we investigate the pedigree slot of the SNP Trio Experiment object.

```
> class(pedigree(ste))
```

```
[1] "PedClass"
attr(,"package")
[1] "trioClasses"
```

```
> getClass("PedClass")
```

```
Class "PedClass" [package "trioClasses"]
```

```
Slots:
```

```
Name:      rownames      nrows      listData
Class: characterORNULL    integer      list
```

```
Name:      elementType elementMetadata      metadata
Class:      character DataTableORNULL      list
```

```
Extends:
```

```
Class "DataFrame", directly
Class "DataTable", by class "DataFrame", distance 2
Class "SimpleList", by class "DataFrame", distance 2
Class "DataTableORNULL", by class "DataFrame", distance 3
Class "List", by class "DataFrame", distance 3
Class "Vector", by class "DataFrame", distance 4
Class "Annotated", by class "DataFrame", distance 5
```

```
> pedigree(ste)
```

```
PedClass with 4139 rows and 6 columns
```

	famid	id
	<factor>	<factor>
1	4778	H_ME-4778_1-4778_1.2
2	4778	H_ME-4778_2-4778_2.2
3	4778	H_ME-4778_3-4778_3.2
4	4783	H_ME-4783_1-4783_1.1
5	4783	H_ME-4783_2-4783_2.1
6	4783	H_ME-4783_3-4783_3.1
7	20000492	H_ME-20000492_1-20000492_1.1
8	20000492	H_ME-20000492_2-20000492_2_a.1
9	20000492	H_ME-20000492_3-20000492_3.1
...	...	...
4131	DS12332	H_ME-DS12332_1-DS12332_1
4132	DS12332	H_ME-DS12332_2-DS12332_2

4133	DS12332	H_ME-DS12332_3-DS12332_3	
4134	DS99998	H_ME-DS10193_5-DS10193_5.2	
4135	DS99998	H_ME-DS10193_6-DS10193_6	
4136	DS99998	H_ME-DS10193_7-DS10193_7	
4137	DS99999	H_ME-DS10707_4-DS10707_4.2	
4138	DS99999	H_ME-DS10707_5-DS10707_5	
4139	DS99999	H_ME-DS10707_6-DS10707_6	
		fid	mid
		<factor>	<factor>
1		H_ME-4778_2-4778_2.2	H_ME-4778_3-4778_3.2
2		NA	NA
3		NA	NA
4		H_ME-4783_2-4783_2.1	H_ME-4783_3-4783_3.1
5		NA	NA
6		NA	NA
7	H_ME-20000492_2-20000492_2_a.1	H_ME-20000492_3-20000492_3.1	
8		NA	NA
9		NA	NA
...		...	...
4131		H_ME-DS12332_2-DS12332_2	H_ME-DS12332_3-DS12332_3
4132		NA	NA
4133		NA	NA
4134		H_ME-DS10193_6-DS10193_6	H_ME-DS10193_7-DS10193_7
4135		NA	NA
4136		NA	NA
4137		NA	NA
4138		H_ME-DS10707_4-DS10707_4.2	H_ME-DS10707_6-DS10707_6
4139		NA	NA
	sex	dx	
	<factor>	<factor>	
1	1	1	
2	1	0	
3	2	0	
4	2	1	
5	1	0	
6	2	1	
7	2	1	
8	1	0	
9	2	0	
...	...	...	
4131	1	1	
4132	1	0	
4133	2	0	
4134	2	1	
4135	1	0	
4136	2	0	
4137	1	0	
4138	1	1	
4139	2	0	

## C.5 geno accessor

```
> class(geno(ste))

[1] "SnpMatrix"
attr(,"package")
[1] "snpStats"

> getClass("SnpMatrix")

Class "SnpMatrix" [package "snpStats"]

Slots:

Name:      .Data
Class: matrix

Extends:
Class "matrix", from data part
Class "array", by class "matrix", distance 2
Class "mMatrix", by class "matrix", distance 2
Class "structure", by class "matrix", distance 3
Class "vector", by class "matrix", distance 4, with explicit coerce

Known Subclasses: "XSnpMatrix"

> geno(ste)

A SnpMatrix with 960 rows and 8951 columns
Row names:  H_ME-CSc175_1-CSc175_1_a.1 ... H_ME-DS11418_3-DS11418_3
Col names:  chr8:129296000 ... chr8:130354790
```

## C.6 RowData

```
> rowData(ste)

GRanges with 8951 ranges and 0 metadata columns:
      seqnames          ranges strand
      <Rle>           <IRanges> <Rle>
chr8:129296000 chr8 [129296000, 129296000] *
chr8:129296113 chr8 [129296113, 129296113] *
chr8:129296185 chr8 [129296185, 129296185] *
chr8:129296191 chr8 [129296191, 129296191] *
chr8:129296198 chr8 [129296198, 129296198] *
chr8:129296209 chr8 [129296209, 129296209] *
chr8:129296289 chr8 [129296289, 129296289] *
chr8:129296343 chr8 [129296343, 129296343] *
chr8:129296434 chr8 [129296434, 129296434] *
      ...      ...      ...
chr8:130353593 chr8 [130353593, 130353593] *
chr8:130353671 chr8 [130353671, 130353671] *
chr8:130354142 chr8 [130354142, 130354142] *
```

```

chr8:130354182      chr8 [130354182, 130354182]      *
chr8:130354239      chr8 [130354239, 130354239]      *
chr8:130354240      chr8 [130354240, 130354240]      *
chr8:130354296      chr8 [130354296, 130354296]      *
chr8:130354703      chr8 [130354703, 130354703]      *
chr8:130354790      chr8 [130354790, 130354790]      *
---
seqlengths:
chr8
NA

```

## C.7 ColData

## D Validity