

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, Robert Scharpf, Holger Schwender, Ingo Ruczinski

March 8, 2013

## 1 Packages

```
> library("trioClasses")
```

## 2 Sample Data

```
> data("sample")
> 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: 511 968
exptData(0):
assays(1): geno
rownames(511): chr14:54383433 chr14:54383470 ...
  chr14:54444385 chr14:54444750
rowData metadata column names(0):
colnames(968): 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
```

## 3 Methods

### 3.1 Holger-style Genotype Matrix

Coercion to matrix for *trio*

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

```
$stat
[1] 1.000000 0.000000 0.000000 8.727273 19.565217
```

```
$pval
[1] 3.173105e-01 1.000000e+00 1.000000e+00 3.134851e-03 9.722321e-06
```

```
$transMinor
[1] 0 9 2 108 8
```

```
$transMajor
[1] 1 9 2 156 38
```

Or apply the aTDT method to the SNP Trio Experiment directly.

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

```
$stat
[1] 1.000000 0.000000 0.000000 8.727273 19.565217
```

```
$pval
[1] 3.173105e-01 1.000000e+00 1.000000e+00 3.134851e-03 9.722321e-06
```

```
$transMinor
[1] 0 9 2 108 8
```

```
$transMajor
[1] 1 9 2 156 38
```

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

First we define the world and a block.

```
> world <- range(gr)
> block <- GRanges(seqnames = "chr14", IRanges(start(gr[100]),
  width = 10000), strand = "*")
```

#### 3.2.1 In a block

```
> block
```

GRanges with 1 range and 0 metadata columns:

seqnames	ranges	strand
<Rle>	<IRanges>	<Rle>

```

[1] chr14 [54395802, 54405801] *
---
seqlengths:
chr14
NA
> TransCount(ste[MAF(ste) <= 0.01, ], block)

[1] 26

```

### 3.2.2 Not In the block

```

> setdiff(world, block)
GRanges with 2 ranges and 0 metadata columns:
      seqnames      ranges strand
      <Rle>        <IRanges> <Rle>
[1] chr14 [54383433, 54395801] *
[2] chr14 [54405802, 54444750] *
---
seqlengths:
chr14
NA
> TransCount(ste[MAF(ste) <= 0.01, ], setdiff(world, block))

[1] 27 89

```

### 3.2.3 In the “world”

```

> world
GRanges with 1 range and 0 metadata columns:
      seqnames      ranges strand
      <Rle>        <IRanges> <Rle>
[1] chr14 [54383433, 54444750] *
---
seqlengths:
chr14
NA
> TransCount(ste[MAF(ste) <= 0.01, ], world)

[1] 142

```

## 3.3 Iteratively

```

> trans.vec <- TransCount(ste[MAF(ste) <= 0.01, ], gr[1:10] +
  1000)

> table(trans.vec)

trans.vec
2 6
5 5

```

## 4 Classes

### 4.1 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 968 rows and 511 columns
Row names: H_ME-DS10776_2-DS10776_2 ... H_ME-DS11313_1-DS11313_1
Col names: chr14:54383433 ... chr14:54444750
```

```
> 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 511 ranges and 0 metadata columns:

	seqnames	ranges	strand
	<Rle>	<IRanges>	<Rle>
chr14:54383433	chr14	[54383433, 54383433]	*
chr14:54383470	chr14	[54383470, 54383470]	*
chr14:54383819	chr14	[54383819, 54383819]	*
chr14:54383945	chr14	[54383945, 54383945]	*
chr14:54384030	chr14	[54384030, 54384030]	*
chr14:54384232	chr14	[54384232, 54384232]	*
chr14:54384251	chr14	[54384251, 54384251]	*
chr14:54384291	chr14	[54384291, 54384291]	*
chr14:54384382	chr14	[54384382, 54384382]	*
...	...	...	...
chr14:54443495	chr14	[54443495, 54443495]	*
chr14:54443499	chr14	[54443499, 54443499]	*
chr14:54443514	chr14	[54443514, 54443514]	*
chr14:54443831	chr14	[54443831, 54443831]	*
chr14:54443880	chr14	[54443880, 54443880]	*
chr14:54444016	chr14	[54444016, 54444016]	*
chr14:54444124	chr14	[54444124, 54444124]	*
chr14:54444385	chr14	[54444385, 54444385]	*
chr14:54444750	chr14	[54444750, 54444750]	*

```

---
seqlengths:
  chr14
    NA

```

## 4.2 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)

```

## 4.3 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: 511 968
exptData(0):
assays(1): geno
rownames(511): chr14:54383433 chr14:54383470 ...
  chr14:54444385 chr14:54444750
rowData metadata column names(0):
colnames(968): 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"
```

## 4.4 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	



## 4.5 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 968 rows and 511 columns
Row names:  H_ME-CSc175_1-CSc175_1_a.1 ... H_ME-DS11418_3-DS11418_3
Col names:  chr14:54383433 ... chr14:54444750
```

## 4.6 RowData

```
> rowData(ste)

GRanges with 511 ranges and 0 metadata columns:
      seqnames      ranges strand
      <Rle>        <IRanges> <Rle>
chr14:54383433 chr14 [54383433, 54383433] *
chr14:54383470 chr14 [54383470, 54383470] *
chr14:54383819 chr14 [54383819, 54383819] *
chr14:54383945 chr14 [54383945, 54383945] *
chr14:54384030 chr14 [54384030, 54384030] *
chr14:54384232 chr14 [54384232, 54384232] *
chr14:54384251 chr14 [54384251, 54384251] *
chr14:54384291 chr14 [54384291, 54384291] *
chr14:54384382 chr14 [54384382, 54384382] *
      ...      ...      ...
chr14:54443495 chr14 [54443495, 54443495] *
chr14:54443499 chr14 [54443499, 54443499] *
chr14:54443514 chr14 [54443514, 54443514] *
```

```

chr14:54443831    chr14 [54443831, 54443831]    *
chr14:54443880    chr14 [54443880, 54443880]    *
chr14:54444016    chr14 [54444016, 54444016]    *
chr14:54444124    chr14 [54444124, 54444124]    *
chr14:54444385    chr14 [54444385, 54444385]    *
chr14:54444750    chr14 [54444750, 54444750]    *
---
seqlengths:
chr14
NA

```

## 4.7 ColData

## 5 Validity