

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

Samuel G. Younkin

May 1, 2013

## 1 Packages & Data

```
> library("trioClasses")
> data("sample")
> data("8q24-european-all.sm")
```

## 2 SummarizedExperiment

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

## 3 Pedigree

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

## 4 FamilyExperiment

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

```
class: FamilyExperiment
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-DS10776_2-DS10776_2
  H_ME-DS10776_3-DS10776_3 ... H_ME-DS11313_3-DS11313_3
  H_ME-DS11313_1-DS11313_1
colData names(1): id
pedigree(4139): famid id fid mid sex dx
complete trios(320):
```

## 5 Methods

### 5.1 ScanTrio

```
> (ste.rare <- ste[!(MAF(ste) >= 0.01 | is.na(MAF(ste))))])
```

```
class: FamilyExperiment
dim: 6397 960
exptData(0):
assays(1): geno
rownames(6397): chr8:129296113 chr8:129296185 ...
  chr8:130354703 chr8:130354790
rowData metadata column names(0):
colnames(960): H_ME-DS10776_2-DS10776_2
  H_ME-DS10776_3-DS10776_3 ... H_ME-DS11313_3-DS11313_3
  H_ME-DS11313_1-DS11313_1
colData names(1): id
pedigree(4139): famid id fid mid sex dx
complete trios(320):
```

```
GRanges with 100 ranges and 0 metadata columns:
```

	seqnames	ranges	strand
	<Rle>	<IRanges>	<Rle>
[1]	chr8	[129296113, 129305165]	*
[2]	chr8	[129305476, 129313104]	*
[3]	chr8	[129313362, 129322866]	*
[4]	chr8	[129324558, 129333656]	*
[5]	chr8	[129334625, 129343161]	*
...	...	...	...
[96]	chr8	[130298273, 130307946]	*
[97]	chr8	[130307953, 130319387]	*
[98]	chr8	[130319526, 130328613]	*
[99]	chr8	[130328617, 130345350]	*
[100]	chr8	[130345517, 130354790]	*

```
---
```

```
seqlengths:
```

```
chr8
NA
```

```
[1] 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
[22] 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
[43] 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
[64] 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
[85] 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 61
```

```
> system.time(scan.trio <- ScanTrio(object = ste.rare,
  window = window, block = range(rowData(ste.rare))))
```

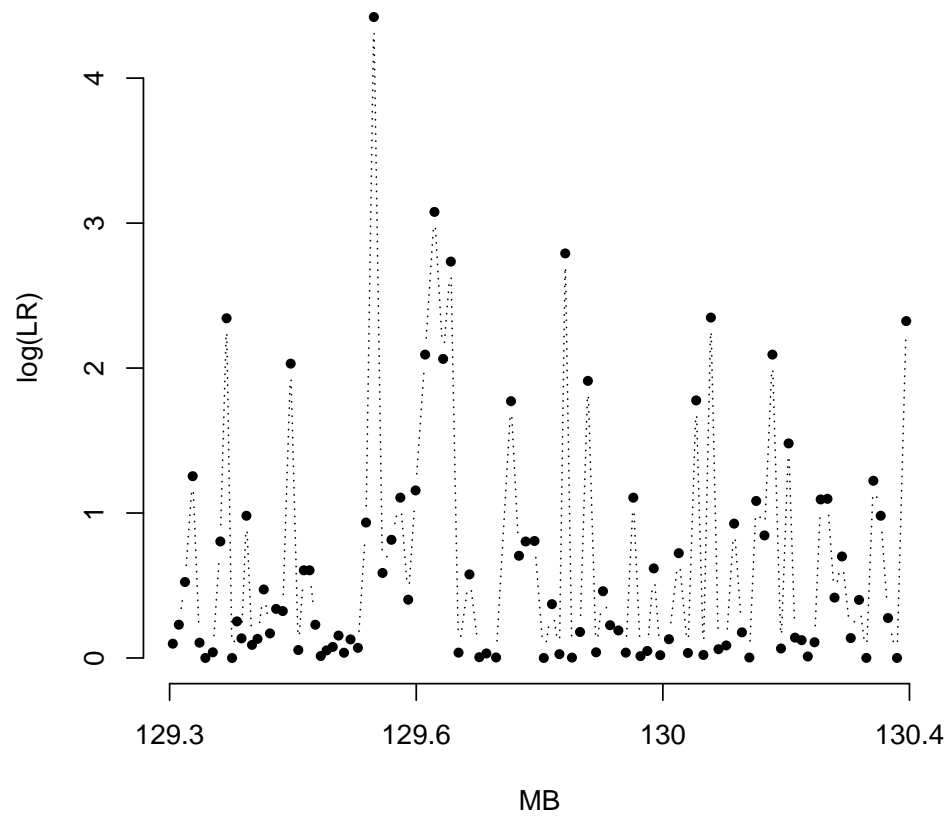
```
user system elapsed
12.057 0.028 12.196
```

```
> scan.trio
```

DataFrame with 100 rows and 7 columns

	lr	minor.in	major.in	minor.out	major.out	mendel.in
	<numeric>	<integer>	<integer>	<integer>	<integer>	<integer>
1	1.103716	58	59	5390	5955	0
2	1.258080	46	58	5402	5956	0
3	1.689351	42	57	5406	5957	0
4	3.504761	41	62	5407	5952	0
5	1.111341	42	51	5406	5963	0
...	...	...	...	...	...	...
96	3.395825	49	72	5399	5942	0
97	2.668222	34	51	5414	5963	0
98	1.317561	53	67	5395	5947	0
99	1.000407	71	78	5377	5936	0
100	10.219049	39	66	5409	5948	0
	mendel.out					
	<integer>					
1	0					
2	0					
3	0					
4	0					
5	0					
...	...					
96	0					
97	0					
98	0					
99	0					
100	0					

```
> save(scan.trio, file = "../data/scan-trio.RData")
```



## A Holger-style Genotype Matrix

Coercion to matrix for *trio*

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

Or apply the aTDT method to the FamilyExperiment directly.

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

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

### B.1 Window

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

> TransCount(ste, window)
```

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

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

### B.3 In the whole block

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

### B.4 In both windows

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

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

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

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

### C.1 In both windows

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

## D Classes

### D.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
> ped.DF
> gr
```

### D.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)
```

### D.3 FamilyExperiment

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

```
> ped <- PedClass(ped.DF)
> ste <- FamilyExperiment(se, pedigree = ped)
```

Here is the show method.

```
> ste
```

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

```
> getClass("FamilyExperiment")
```

### D.4 PedClass

Now we investigate the pedigree slot of the `FamilyExperiment` object.

```
> class(pedigree(ste))
> getClass("PedClass")
> pedigree(ste)
```

### D.5 geno accessor

```
> class(geno(ste))
> getClass("SnpMatrix")
> geno(ste)
```

### D.6 RowData

```
> rowData(ste)
```

### D.7 ColData

## E Validity