

Parent Offspring Data

Rodney J. Dyer

Department of Biology

Virginia Commonwealth University

<http://dyerlab.bio.vcu.edu>

Synopsis

There are several cases where you have data that consists of both adults and offspring. With these kinds of data, there are some interesting kinds of analyses available for examining structure and diversity. The functionality that **gstudio** provides focuses on translations of offspring data into common formats that can be analyzed using regular routines.

Getting Data

The use of GoogleDocs as a repository for your data is not unique to parent-offspring data and is used here to demonstrate how to utilize this options. There is a more detailed discussion of how to set up your GoogleSpreadsheets so that you can access them in the **DataImport** vignette. In what follows, I will use the *Cornus florida* data. I split URL (see Data Import) because it was so long it trailed off the page...

```
> require(gstudio)
> baseURL <- "https://docs.google.com/spreadsheet/pub?hl=en_US&hl=en_US&"
> key <- "key=0Aq-lsUWPDuZtdF9xMXZGQWNTbk1FNTVWd3F3U0FDdXc&single=true&gid=0&"
> range <- "range=A1%3AI63&output=csv"
> cornusURL <- paste(baseURL,key,range,sep="")
> cornus <- read.population( googleURL=cornusURL)
```

The structure of adult/offspring data is just like any other kind of data and can consist of covariates such as physical location, size, etc. along with strata and loci. The distinction is that there must be *at least* two strata columns:

Individual ID There should a column in the dataset that has identification number or names that are unique to adults. Every adult *must* have a unique identification numver.

Offspring ID To differentiate offspring from adults, the Offspring ID column should have the maternal individual (or paternal if you like) equal to 0. Offspring from this individual have non-zero values for the Offspring ID column. Offspring do not need to all have unique Offspring ID designations, just unique ones within the set of offspring with the same Individual ID.

Here is an example to show the distinctions.

```
> family <- offspring.array(cornus,474)
> family
```

\$mom

	IndID	OffID	X	Y	G8	H18	N5	N10	O5
1	474	0	1545	2234	156:164	104:112	126:126	198:200	185:193

```

$offspring
  IndID OffID    X    Y    G8    H18    N5    N10    O5
1    474     1 1545 2234 164:168 104:112 126:126 198:202 185:195
2    474     2 1545 2234 156:156 102:112 126:126 198:198 179:185
3    474     3 1545 2234 162:164 112:114 124:126 192:198 185:193
4    474     4 1545 2234 164:188 110:112 126:126 194:198 185:193
5    474     5 1545 2234 156:158 112:112 126:128 192:198 185:193
6    474     6 1545 2234 164:180 108:112 126:126 188:198 177:193
7    474     7 1545 2234 164:188 110:112 126:126 190:198 177:185
8    474     8 1545 2234 164:168 104:112 126:126 200:202 193:193
9    474     9 1545 2234 156:164 112:112 126:126 190:198 185:193
10   474    10 1545 2234 164:188 110:112 126:126 190:198 177:185
11   474    11 1545 2234 164:168 110:112 126:126 188:200 181:185
12   474    12 1545 2234 164:180 112:112 126:126 190:198 179:193
13   474    13 1545 2234 164:164 112:114 126:126 188:198 181:185
14   474    14 1545 2234 156:180 112:118 126:126 188:200 185:193
15   474    15 1545 2234 156:180 112:114 126:126 190:200 179:185
16   474    16 1545 2234 156:168 104:112 126:126 192:198 193:193
17   474    17 1545 2234 164:164 112:112 126:126 198:200 193:193
18   474    18 1545 2234 156:164 104:114 126:126 198:202 181:193
19   474    19 1545 2234 164:164 112:112 126:128 198:200 185:193
20   474    20 1545 2234 164:168 112:112 126:126 192:198 193:193

```

Notice that all the offspring from mom '474' have the same IndID and she is differentiated from the offspring by having 'OffID=0'. In fact, all the adults in the dataset can be found as:

```

> adults <- cornus[cornus$OffID==0,]
> adults
  IndID OffID    X    Y    G8    H18    N5    N10    O5
1    226     0 1392 3534 162:180 114:114 124:126 192:192 185:195
2    232     0 1656 3414 158:180 112:112 124:126 184:192 185:185
3    234     0 1718 3330 158:180 112:96 128:128 184:192 185:185
4    300     0 1175 3114 180:188 112:116 126:126 198:200 191:195
5    305     0 1529 3237 154:170 122:124 124:126 188:192 181:195
6    432     0 1336 2748 164:180 114:116 124:126 198:202 185:193
7    433     0 1337 2749 180:188 112:114 126:126 198:202 179:193
8    468     0 1588 2233 164:164 110:116 124:124 198:202 181:193
9    474     0 1545 2234 156:164 104:112 126:126 198:200 185:193
10   484     0 1514 2302 160:168 112:116 126:126 192:192 193:193
11   487     0 1517 2305 164:176 110:112 126:126 192:202 179:181
12   489     0 1519 2307 160:164 104:112 126:126 192:202 179:181
13   490     0 1520 2308 164:176 112:112 128:128 192:202 179:181
14   493     0 1523 2311 168:168 104:112 124:126 192:202 193:195
15   512     0 1174 2279 156:174 108:114 126:126 200:202 179:195
16   513     0 1239 2276 156:180 102:114 126:126 190:198 179:179
17   516     0 1299 2135 156:180 102:114 124:126 190:198 179:179
18   519     0 1357 2148 156:180 104:114 126:126 182:188 181:181
19   520     0 1412 2041 164:172 114:124 126:126 188:198 195:195
20   521     0 1511 1949 160:164 112:112 128:128 198:200 185:193
21   590     0 1880 1040 164:168 112:118 124:126 192:198 177:193
22   607     0 2286 2888 154:164 114:114 126:126 188:202 181:181

```

Pollen Pools

Since my research is primarily focused on the analysis of plant populations and mother/offspring combinations provide information about pollen donors, naturally these kinds of analyses will be the first kind to have functionality.

Minus Mom

If you have the collection of offspring and a mother, you can estimate pollen pool allele frequencies as by subtracting the maternal contribution to each genotype and then estimating the allele frequencies of the paternal components (this could be reversed if you have father/offspring data and need to estimate maternal genotype frequencies just as easily).

```
> offs <- minus.mom( cornus )
> offs
```

	IndID	OffID	X	Y	G8	H18	N5	N10	O5
1	468	1	1588	2233	156	112	126	200	185
2	468	2	1588	2233	180	112	126	198:202	193
3	468	3	1588	2233	188	114	126	198	179
4	468	4	1588	2233	180	112	126	192	185
5	468	5	1588	2233	154	104	126	202	195
6	468	6	1588	2233	154	104	124	188	193
7	468	7	1588	2233	162	114	124	192	185
8	468	8	1588	2233	180	114	126	192	195
9	468	9	1588	2233	180	114	126	202	193
10	468	10	1588	2233	182	114	124	184	195
11	468	11	1588	2233	158	112	128	192	185
12	468	12	1588	2233	158	112	128	192	185
13	468	13	1588	2233	164	108	126	188	191
14	468	14	1588	2233	182	108	126	198:202	181:193
15	468	15	1588	2233	180	112	126	198:202	181:193
16	468	16	1588	2233	164	102	126	198	181:193
17	468	17	1588	2233	164	104	126	188	193
18	468	18	1588	2233	182	108	126	198:202	181:193
19	468	19	1588	2233	182	104	126	202	181:193
20	468	20	1588	2233	182	110	126	198:202	179
21	474	1	1545	2234	168	104:112	126	202	195
22	474	2	1545	2234	156	102	126	198	179
23	474	3	1545	2234	162	114	124	192	185:193
24	474	4	1545	2234	188	110	126	194	185:193
25	474	5	1545	2234	158	112	128	192	185:193
26	474	6	1545	2234	180	108	126	188	177
27	474	7	1545	2234	188	110	126	190	177
28	474	8	1545	2234	168	104:112	126	202	193
29	474	9	1545	2234	156:164	112	126	190	185:193
30	474	10	1545	2234	188	110	126	190	177
31	474	11	1545	2234	168	110	126	188	181
32	474	12	1545	2234	180	112	126	190	179
33	474	13	1545	2234	164	114	126	188	181
34	474	14	1545	2234	180	118	126	188	185:193
35	474	15	1545	2234	180	114	126	190	179
36	474	16	1545	2234	168	104:112	126	192	193
37	474	17	1545	2234	164	112	126	198:200	193

```

38  474    18 1545 2234 156:164    114 126    202    181
39  474    19 1545 2234    164    112 128 198:200 185:193
40  474    20 1545 2234    168    112 126    192    193

```

```

> freqs.G8 <- Frequencies(offss$G8)
> freqs.G8

```

Allele Frequencies:

```

156 = 0.0952381
180 = 0.2142857
188 = 0.0952381
154 = 0.04761905
162 = 0.04761905
182 = 0.1190476
158 = 0.07142857
164 = 0.1904762
168 = 0.1190476

```

Now the distinction should be made that these are the pollen donor allele frequencies since the contribution of the maternal individual has been removed from each offspring, the differences you can see as by comparing the above to:

```

> unred.offs <- cornus[cornus$OffID!=0,]
> freqs.unred.G8 <- Frequencies( unred.offs$G8 )
> freqs.unred.G8

```

Allele Frequencies:

```

156 = 0.1125
164 = 0.5125
180 = 0.1125
188 = 0.05
154 = 0.025
162 = 0.025
182 = 0.0625
158 = 0.0375
168 = 0.0625

```

where the genotype of each offspring has 50% of the mother's genotype.

Genetic Distances and Structure (e.g., 2Gener)

The reduced genotypes can be used in traditional genetic analyses as any other type of genetic data. For example, the Two-Generation Analysis of Pollen Structure (hereafter 2Gener; Smouse *et al.* 2001, Dyer *et al.* 2004) is essentially an AMOVA analysis on pollen donor genotypes. This is a bit of a manual version of it but it can be conducted as (in the next version I'll add a the AMOVA/2Gener options to the `genetic.structure` function).

```

> require(pegas,quietly=TRUE,warn.conflicts=FALSE)
> D <- genetic.distance(offss,mode="AMOVA")[[1]]
> D <- as.dist(D)
> Moms <- as.factor( offss$IndID )
> amova(D ~ Moms)

```

Analysis of Molecular Variance

Call: amova(formula = D ~ Moms)

	SSD	MSD	df
Moms	8.10625	8.106250	1
Error	324.48750	8.539145	38
Total	332.59375	8.528045	39

Variance components:

	sigma2	P.value
Moms	-0.021645	0.474
Error	8.539145	

Variance coefficients:

a
20

Paternity

The `gstudio` package has some basic functionality regarding estimating paternity (or maternity if you have those kinds of data). Thus far, only fractional paternity is implemented and only basically.

Initially,

```
> pollen.freqs <- allele.frequencies( offs )
> Pexcl <- lapply( pollen.freqs, exclusion.probability)
> Pexcl
```

\$G8

Pe
0.7225813

\$H18

Pe
0.5868707

\$N5

Pe
0.17702

\$N10

Pe
0.6501088

\$O5

Pe
0.5702125

The multilocus exclusion probability is given by:

$$P_{excl} = 1 - \prod_{i=1}^{\ell} (1 - P_{excl,i})$$

which in R can be found as:

```
> 1- prod((1-unlist(Pexcl)))
[1] 0.985816
```

Which means that on average, these loci are expected to be able to exclude 98.6% of potential fathers for an mother/offspring pair.

The function `paternity` estimates fractional paternity for a particular mother and set of offspring. Fractional paternity is estimated using multilocus Mendelian transition probabilities for triplet of male parent (MP), female parent (FP), and offspring (O) standardized by the likelihood of all potential fathers.

$$\lambda_i = \frac{T(O|FP, MP_i)}{\sum_{\forall k} T(O|FP, MP)_k}$$

This ensures that $\sum \lambda = 1$. The function `paternity` estimates this for all the offspring within a single family providing the subset of offspring that have potential fathers in the population, the identity of each father, and the fractional likelihood of each father.

```
> pat <- paternity(cornus,474)
> print(pat)
```

Paternity Analysis:

```
Family ID: 474
Number of Offspring: 20
Offspring Assigned Paternity: 12
Fractional Paternity (off: dad(prob) ):
1:493(1)
13:607(1)
15:513(0.6666666666666667) 516(0.3333333333333333)
16:484(0.615384615384615) 493(0.307692307692308) 590(0.0769230769230769)
17:474(0.8) 590(0.2)
18:607(1)
19:521(1)
2:513(0.6666666666666667) 516(0.3333333333333333)
20:484(0.727272727272727) 493(0.181818181818182) 590(0.0909090909090909)
3:226(1)
5:234(1)
8:493(1)
```

You can visualize the results using the `paternity.spiderplot` function that plots the location of all the individuals and indicates putative paternity by connecting mothers and indicated fathers.

```
> paternity.spiderplot(cornus,pat,X="X",Y="Y", bty="n", xlab="X", ylab="Y")
```

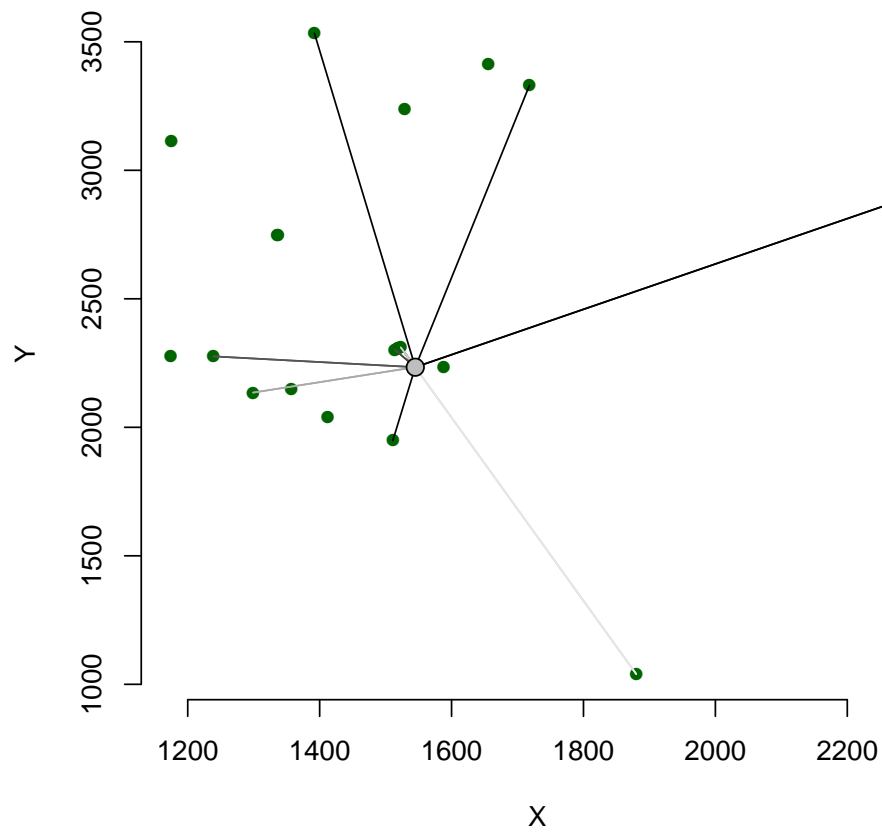


Figure 1: Estimated parentage for family 474 in the *Cornus florida* dataset.