

Tutorial: Using *GeneticsBase*

Gregory Warnes
gregory_warnesurmc.rochester.edu
University of Rochester

Ross Lazarus
ross.lazarus@channing.harvard.edu
Channing Laboratory

April 22, 2010

```
> options(width = 90)
```

1 Introduction

This vignette was created as a tutorial for the 2007 BioConductor User's Conference held in Seattle, WA, USA during August 2007, and was presented by Dr. Warnes and Dr. Lazarus. The material is structured as a tutorial with a small example data set (8184 Markers x 180 Subjects belonging to 50 Families) .

2 Outline

1. Preliminaries
 - (a) Install the necessary packages
 - (b) Load the libraries
2. Loading Data
 - (a) Read data from files
 - (b) Error check the loaded data
3. Descriptive Statistics
 - (a) Summary of allele/genotype frequency
 - (b) HWE test
 - (c) Visualize Disequilibrium
4. Hypothesis Testing
 - (a) Armitage test
 - (b) Logistic with synthetic covariates
 - (c) GLM with synthetic covariates & outcome
5. Subsetting Results and Formatting Output
 - (a) Construct some output tables for top 100 significant markers
6. Study planning tools (power, sample size)

3 Preliminaries

3.1 Install RGenetics packages and dependencies

For MS-Windows it is necessary to manually install dependencies from CRAN

```
> install.packages(c("xtable", "combinat", "gdata", "gplots", "mvtnorm"),  
+   dep = TRUE)
```

Now to install the necessary packages:

```
> repos <- c("http://www.warnes.net/bioc2007/", "http://cran.fhcrc.org")  
> install.packages(c("GeneticsBase", "GeneticsDesign", "fbat"), repos = repos,  
+   type = "source", dep = TRUE)
```

3.2 Load the libraries

```
> library(GeneticsBase)  
> library(GeneticsDesign)  
> library(fbat)
```

4 Loading Data

4.1 Read data from files

Load the full data:

```
> hm.a <- readGenes(gfile = "hmCEU_YRI_chr22_ALLfbat.ped", gformat = "fbat")
```

```
Reading 8184 markers and 180 subjects from `hmCEU_YRI_chr22_ALLfbat.ped' ...  
generating 'geneSet' object...
```

```
100 200 300 400 500 600 700 800 900 1000  
1100 1200 1300 1400 1500 1600 1700 1800 1900 2000  
2100 2200 2300 2400 2500 2600 2700 2800 2900 3000  
3100 3200 3300 3400 3500 3600 3700 3800 3900 4000  
4100 4200 4300 4400 4500 4600 4700 4800 4900 5000  
5100 5200 5300 5400 5500 5600 5700 5800 5900 6000  
6100 6200 6300 6400 6500 6600 6700 6800 6900 7000  
7100 7200 7300 7400 7500 7600 7700 7800 7900 8000  
8100 Successfully read the pedigree file `hmCEU_YRI_chr22_ALLfbat.ped' .
```

```
Number of Markers: 8184  
Number of Subjects: 180  
Number of Families: 50
```

```
> print(hm.a)
```

```
geneSet object  
-----
```

```
Number of Markers:      8184  
Number of Observations: 180
```

Sample variables: family, pid, father, mother, sex, affected

Genetic data:

	1334.1	1334.10	1334.11	1334.12	1334.13	1334.2	74.3	77.1	77.2	
22_14884399_rs4911642	4/4	4/4	4/4	4/4	4/4	4/4	...	4/4	4/4	2/4
22_15298335_rs2027653	2/2	2/4	2/4	4/4	2/4	4/4	...	4/4	2/4	2/2
22_15412698_rs5747620	2/2	2/2	2/2	2/4	2/4	4/4	...	2/4	2/4	2/2
22_15434720_rs9605903	4/4	4/4	4/4	2/4	4/4	2/4	...	4/4	4/4	4/4
22_15447504_rs5747968	3/4	4/4	3/4	3/4	4/4	3/4	...	4/4	4/4	4/4
22_15452483_rs2236639	3/3	3/3	3/3	3/3	3/3	3/3	...	3/3	3/3	3/3

22_49497339_rs5770820	1/3	<NA>	1/1	3/3	3/3	3/3	...	3/3	3/3	3/3
22_49498590_rs6010061	2/4	2/2	4/4	2/2	2/2	2/2	...	2/4	2/4	4/4
22_49510004_rs715586	2/2	2/2	2/2	2/2	2/2	2/2	...	2/2	2/2	2/2
22_49512530_rs8137951	1/3	1/3	1/1	3/3	3/3	3/3	...	3/3	1/3	1/3
22_49518559_rs756638	3/3	3/3	3/3	1/3	3/3	3/3	...	1/3	3/3	1/3
22_49522492_rs3810648	1/1	1/1	1/1	1/1	1/1	1/1	...	1/3	1/1	1/1
	77.3	9.1	9.2							
22_14884399_rs4911642	2/4	<NA>	2/4							
22_15298335_rs2027653	4/4	4/4	4/4							
22_15412698_rs5747620	4/4	2/4	4/4							
22_15434720_rs9605903	4/4	4/4	4/4							
22_15447504_rs5747968	4/4	4/4	4/4							
22_15452483_rs2236639	3/3	3/3	3/3							
	.	.	.							
	.	.	.							
	.	.	.							
22_49497339_rs5770820	3/3	3/3	3/3							
22_49498590_rs6010061	2/4	4/4	4/4							
22_49510004_rs715586	2/2	2/2	2/2							
22_49512530_rs8137951	1/3	1/1	1/3							
22_49518559_rs756638	3/3	1/3	1/3							
22_49522492_rs3810648	1/1	1/3	1/1							

We'll also need just the founders later:

```
> hm.f <- readGenes(gfile = "hmCEU_YRI_chr22_Foundersfbat.ped", gformat = "fbat")
```

```
Reading 8184 markers and 120 subjects from `hmCEU_YRI_chr22_Foundersfbat.ped` ...  
generating 'geneSet' object...
```

```
100 200 300 400 500 600 700 800 900 1000  
1100 1200 1300 1400 1500 1600 1700 1800 1900 2000  
2100 2200 2300 2400 2500 2600 2700 2800 2900 3000  
3100 3200 3300 3400 3500 3600 3700 3800 3900 4000  
4100 4200 4300 4400 4500 4600 4700 4800 4900 5000  
5100 5200 5300 5400 5500 5600 5700 5800 5900 6000  
6100 6200 6300 6400 6500 6600 6700 6800 6900 7000  
7100 7200 7300 7400 7500 7600 7700 7800 7900 8000  
8100 Successfully read the pedigree file `hmCEU_YRI_chr22_Foundersfbat.ped`.
```

```
Number of Markers: 8184
Number of Subjects: 120
Number of Families: 50
```

For the purpose of speeding execution of examples, we'll also create a smaller subset of 100 markers from the original file.

```
> hm.a2 <- hm.a[1:100, ]
```

4.2 Error check the loaded data

Count frequencies of missing genotypes (requires 26 seconds on my MacBook Pro)

Number of missing genotypes per subject:

```
> mG <- missGFreq(hm.a2, founderOnly = TRUE, quiet = FALSE)
```

converting geneSet object to numerical matrix...

counting frequencies of missing genotypes...

```
> head(mG$nMissSubjects)
```

```
      00 0* *0
subject2 3 0 0
subject3 2 0 0
subject4 4 0 0
subject5 0 0 0
subject8 1 0 0
subject9 3 0 0
```

Column headers:

00 missing both alleles

0* 1st allele missing while the 2nd allele is not missing

*0 1st allele is not missing while the 2nd allele is missing

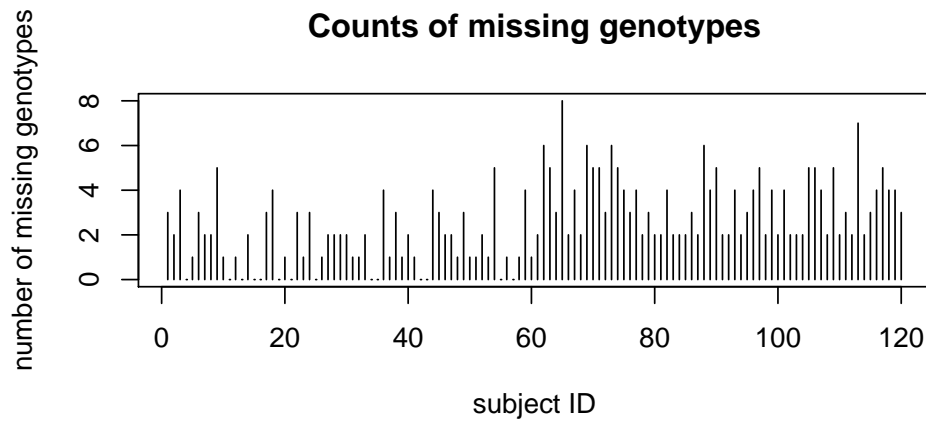
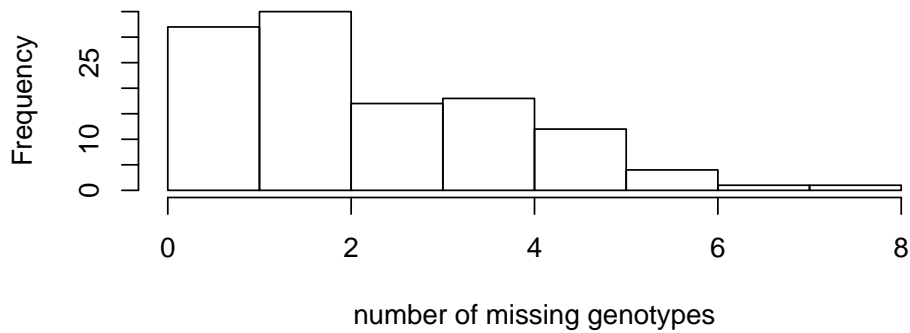
Number of missing genotypes per marker:

```
> head(mG$nMissMarkers)
```

```
      00 0* *0
22_14884399_rs4911642 13 0 0
22_15298335_rs2027653  1 0 0
22_15412698_rs5747620  0 0 0
22_15434720_rs9605903  0 0 0
22_15447504_rs5747968  0 0 0
22_15452483_rs2236639  0 0 0
```

Plot counts of missing genotypes:

```
> par(mfrow = c(2, 1))
> hist(mG$nMissSubjects[, 1], main = "", xlab = "number of missing genotypes")
> plot(1:nrow(mG$nMissSubjects), mG$nMissSubjects[, 1], xlab = "subject ID",
+      ylab = "number of missing genotypes", type = "h")
> title("Counts of missing genotypes")
> par(mfrow = c(1, 1))
```



5 Descriptive Statistics

1. Summary of allele/genotype frequency
2. HWE test
3. Visualize Disequilibrium

Basic data quality checks for markers. Column headings are:

ObsHET observed proportion of heterozygous genotypes per marker

PredHET predicted proportion of heterozygous genotypes per marker

HWpval pvalues of Hardy-Weinberg test per marker

pGeno percentage of non-missing genotypes for markers

MAF minor allele frequencies. missing allele are excluded from calculation

Rating 1 if passes HW test; 0 if failed HW test.

```
> cM <- checkMarkers(hm.a2)
> head(cM)
```

Name	Position	ObsHET	PredHET	Hwpval
22_14884399_rs4911642	22_14884399_rs4911642	? 0.30841121	0.2608525	0.05930340
22_15298335_rs2027653	22_15298335_rs2027653	? 0.34453782	0.3892734	0.20997430
22_15412698_rs5747620	22_15412698_rs5747620	? 0.50000000	0.4994444	0.99027790
22_15434720_rs9605903	22_15434720_rs9605903	? 0.16666667	0.2061111	0.03604633
22_15447504_rs5747968	22_15447504_rs5747968	? 0.23333333	0.2665278	0.17246954
22_15452483_rs2236639	22_15452483_rs2236639	? 0.09166667	0.1171875	0.01704962

	pGeno	MAF	Rating
22_14884399_rs4911642	89.16667	0.1542056	1
22_15298335_rs2027653	99.16667	0.2647059	1
22_15412698_rs5747620	100.00000	0.4833333	1
22_15434720_rs9605903	100.00000	0.1166667	0
22_15447504_rs5747968	100.00000	0.1583333	1
22_15452483_rs2236639	100.00000	0.0625000	0

Check Mendelian errors:

```
> cMend <- checkMendelian(hm.a2, quiet = FALSE)
```

```
converting geneSet object to numerical matrix...
checking Mendelian errors ...
checking compatibility ...
```

Number of Mendelian errors per marker:

```
> head(cMend$nMerrMarker)
```

22_14884399_rs4911642	22_15298335_rs2027653	22_15412698_rs5747620	22_15434720_rs9605903
8	1	0	0
22_15447504_rs5747968	22_15452483_rs2236639		
0	0		

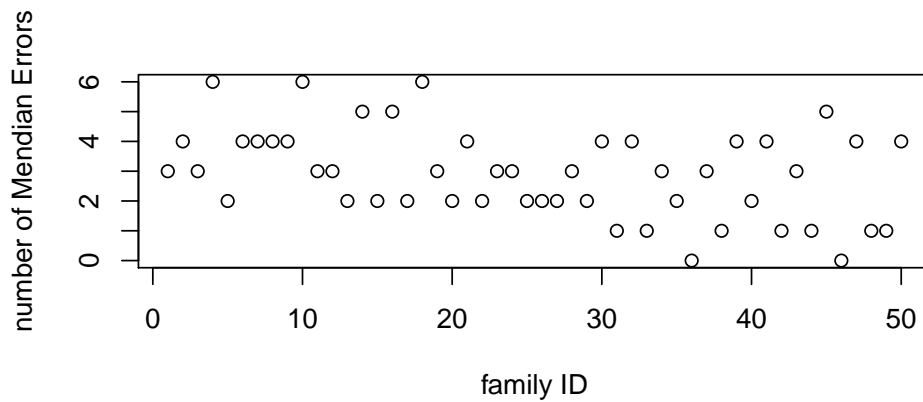
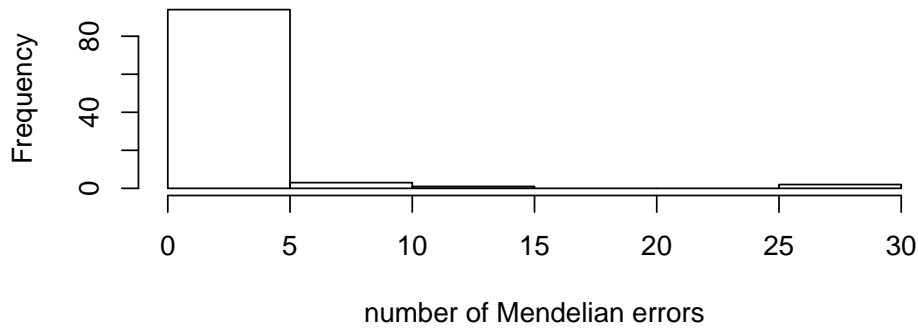
Number of Mendelian errors per family:

```
> head(cMend$nMerrFamily)
```

family1	family2	family3	family4	family5	family6
3	4	3	6	2	4

Plot counts of Mendelian errors

```
> par(mfrow = c(2, 1))
> hist(cMend$nMerrMarker, main = "", xlab = "number of Mendelian errors")
> plot(1:length(cMend$nMerrFamily), cMend$nMerrFamily, xlab = "family ID",
+      ylab = "number of Mendian Errors")
> par(mfrow = c(1, 1))
```



5.1 Summary of allele/genotype frequency based on GeneticsBase functions

Allele summary, including allele counts, allele frequencies, 95% CI of allele frequencies:

```
> t1 <- alleleSummary(hm.a2[1:10])
> t1
```

Gene Marker	Position	Group	Allele	Count	Freq	CI-Lower	CI-Upper
ALL 22_14884399_rs4911642	?	ALL	2	48	0.151	0.113	0.192
		ALL	4	270	0.849	0.808	0.887
22_15298335_rs2027653	?	ALL	2	94	0.264	0.219	0.312
		ALL	4	262	0.736	0.688	0.781
22_15412698_rs5747620	?	ALL	2	169	0.469	0.417	0.522
		ALL	4	191	0.531	0.478	0.583
22_15434720_rs9605903	?	ALL	2	43	0.119	0.086	0.153
		ALL	4	317	0.881	0.847	0.914

22_15447504_rs5747968	?	ALL	3	59	0.164	0.128	0.203
		ALL	4	301	0.836	0.797	0.872
22_15452483_rs2236639	?	ALL	1	26	0.072	0.047	0.100
		ALL	3	334	0.928	0.900	0.953
22_15455353_rs5747999	?	ALL	1	225	0.625	0.575	0.675
		ALL	2	135	0.375	0.325	0.425
22_15467656_rs11089263	?	ALL	1	219	0.608	0.558	0.658
		ALL	2	141	0.392	0.342	0.442
22_15474749_rs2096537	?	ALL	1	80	0.231	0.188	0.277
		ALL	2	266	0.769	0.723	0.812
22_15479107_rs9604959	?	ALL	2	249	0.808	0.763	0.851
		ALL	4	59	0.192	0.149	0.237

Footer:

Confidence intervals width is 95%, computed using the exact quantiles for the binomial distribution.

Same for genotypes:

```
> t2 <- genotypeSummary(hm.a2[1:10], founderOnly = TRUE)
> t2
```

Gene Marker	Position	Group	Genotype	Count	Freq	CI-Lower	CI-Upper
? 22_14884399_rs4911642	?	ALL	2/2	0	0.000		
			2/4	33	0.308	0.224	0.402
			4/4	74	0.692	0.598	0.776
			NA	13			
? 22_15298335_rs2027653	?	ALL	2/2	11	0.092	0.042	0.151
			2/4	41	0.345	0.261	0.429
			4/4	67	0.563	0.471	0.655
			NA	1			
? 22_15412698_rs5747620	?	ALL	2/2	28	0.233	0.158	0.308
			2/4	60	0.500	0.408	0.592
			4/4	32	0.267	0.192	0.350
			NA	0			
? 22_15434720_rs9605903	?	ALL	2/2	4	0.033	0.008	0.067
			2/4	20	0.167	0.100	0.233
			4/4	96	0.800	0.725	0.867
			NA	0			
? 22_15447504_rs5747968	?	ALL	3/3	5	0.042	0.008	0.083
			3/4	28	0.233	0.158	0.308
			4/4	87	0.725	0.642	0.800
			NA	0			

?	22_15452483_rs2236639	?	ALL	1/1	2	0.017	0.000	0.042
				1/3	11	0.092	0.042	0.150
				3/3	107	0.892	0.833	0.942
				NA	0			
?	22_15455353_rs5747999	?	ALL	1/1	49	0.408	0.325	0.500
				1/2	54	0.450	0.358	0.542
				2/2	17	0.142	0.083	0.208
				NA	0			
?	22_15467656_rs11089263	?	ALL	1/1	46	0.383	0.300	0.475
				1/2	52	0.433	0.342	0.525
				2/2	22	0.183	0.117	0.258
				NA	0			
?	22_15474749_rs2096537	?	ALL	1/1	0	0.000		
				1/2	51	0.440	0.353	0.534
				2/2	65	0.560	0.466	0.647
				NA	4			
?	22_15479107_rs9604959	?	ALL	2/2	68	0.642	0.547	0.736
				2/4	36	0.340	0.255	0.434
				4/4	2	0.019	0.000	0.047
				NA	14			

Expected	Obs-Exp	HWE	X ²	P-value
2.544	-2.544	3.557	0.0651	
27.911	5.089			
76.544	-2.544			

8.338	2.662	1.572	0.241
46.324	-5.324		
64.338	2.662		

28.033	-0.033	0.000	1
59.933	0.067		
32.033	-0.033		

1.633	2.367	4.395	0.0633
24.733	-4.733		
93.633	2.367		

3.008	1.992	1.861	0.305
31.983	-3.983		
85.008	1.992		

```
0.469    1.531    5.691    0.0501
14.062   -3.062
105.469  1.531
```

```
48.133    0.867    0.116    0.84
55.733   -1.733
16.133    0.867
```

```
43.200    2.800    1.134    0.344
57.600   -5.600
19.200    2.800
```

```
5.606    -5.606    9.210    0.003
39.789   11.211
70.606   -5.606
```

```
69.774   -1.774    1.266    0.343
32.453    3.547
3.774    -1.774
```

Footer:

Confidence intervals width is 95%, computed using the exact quantiles for the binomial distribution. As the true distribution is multinomial this is only approximately correct.

HWE test:

```
> hwe <- HWE(hm.a2[1:10])
> hwe
```

```
[[1]]
[1] "diseq"
```

```
$call
HWE(object = hm.a2[1:10])
```

```
$D
```

	Est	2.5%	97.5%	n	P-value
22_14884399_rs4911642	0.022783909	0.01353783	0.035599858	159	2.602556e-02
22_15298335_rs2027653	-0.025785886	-0.05535128	0.005057758	178	8.267860e-02
22_15412698_rs5747620	-0.001844136	-0.03876543	0.033456790	180	1.000000e+00
22_15434720_rs9605903	-0.019066358	-0.04258488	0.001859568	180	2.387117e-02
22_15447504_rs5747968	-0.023140432	-0.04684414	0.001111111	180	2.881018e-02
22_15452483_rs2236639	-0.005895062	-0.02128858	0.006049383	180	2.254611e-01
22_15455353_rs5747999	0.012847222	-0.01987654	0.045679012	180	5.265022e-01
22_15467656_rs11089263	-0.018819444	-0.05332562	0.017777778	180	2.780714e-01

```

22_15474749_rs2096537 0.053459855 0.03749708 0.072245982 173 7.796707e-06
22_15479107_rs9604959 0.010720611 -0.01239669 0.034333361 154 6.010626e-01

```

\$`D`´

	Est	2.5%	97.5%	n	P-value
22_14884399_rs4911642	0.177777778	0.1316726	0.232558140	159	2.602556e-02
22_15298335_rs2027653	-0.369850611	-0.7800000	0.024598028	178	8.267860e-02
22_15412698_rs5747620	-0.008368054	-0.1812500	0.133893281	180	1.000000e+00
22_15434720_rs9605903	-1.336398053	-2.7869822	0.018426485	180	2.387117e-02
22_15447504_rs5747968	-0.861534042	-1.7551020	0.006944444	180	2.881018e-02
22_15452483_rs2236639	-1.130177515	-4.4442344	0.084337349	180	2.254611e-01
22_15455353_rs5747999	0.054814815	-0.1337868	0.197900587	180	5.265022e-01
22_15467656_rs11089263	-0.122679946	-0.3591837	0.073075837	180	2.780714e-01
22_15474749_rs2096537	0.300751880	0.2401434	0.367588933	173	7.796707e-06
22_15479107_rs9604959	0.069226057	-0.3750000	0.201709296	154	6.010626e-01

\$r

	Est	2.5%	97.5%	n	P-value
22_14884399_rs4911642	-0.177777778	-0.232558140	-0.13167260	159	2.602556e-02
22_15298335_rs2027653	0.132694494	-0.024598028	0.27724541	178	8.267860e-02
22_15412698_rs5747620	0.007404195	-0.133893281	0.15513834	180	1.000000e+00
22_15434720_rs9605903	0.181277969	-0.018426485	0.36507937	180	2.387117e-02
22_15447504_rs5747968	0.168872121	-0.006944444	0.33751894	180	2.881018e-02
22_15452483_rs2236639	0.087977890	-0.084337349	0.29941618	180	2.254611e-01
22_15455353_rs5747999	-0.054814815	-0.197900587	0.08210097	180	5.265022e-01
22_15467656_rs11089263	0.078985718	-0.073075837	0.22285714	180	2.780714e-01
22_15474749_rs2096537	-0.300751880	-0.367588933	-0.24014337	173	7.796707e-06
22_15479107_rs9604959	-0.069226057	-0.201709296	0.08333333	154	6.010626e-01

\$`X^2`´

	Est	2.5%	97.5%	n	P-value
22_14884399_rs4911642	0.0316049383	NA	NA	159	2.602556e-02
22_15298335_rs2027653	0.0176078288	NA	NA	178	8.267860e-02
22_15412698_rs5747620	0.0000548221	NA	NA	180	1.000000e+00
22_15434720_rs9605903	0.0328617022	NA	NA	180	2.387117e-02
22_15447504_rs5747968	0.0285177933	NA	NA	180	2.881018e-02
22_15452483_rs2236639	0.0077401092	NA	NA	180	2.254611e-01
22_15455353_rs5747999	0.0030046639	NA	NA	180	5.265022e-01
22_15467656_rs11089263	0.0062387437	NA	NA	180	2.780714e-01
22_15474749_rs2096537	0.0904516931	NA	NA	173	7.796707e-06
22_15479107_rs9604959	0.0047922469	NA	NA	154	6.010626e-01

Defatult graphical display of LD:

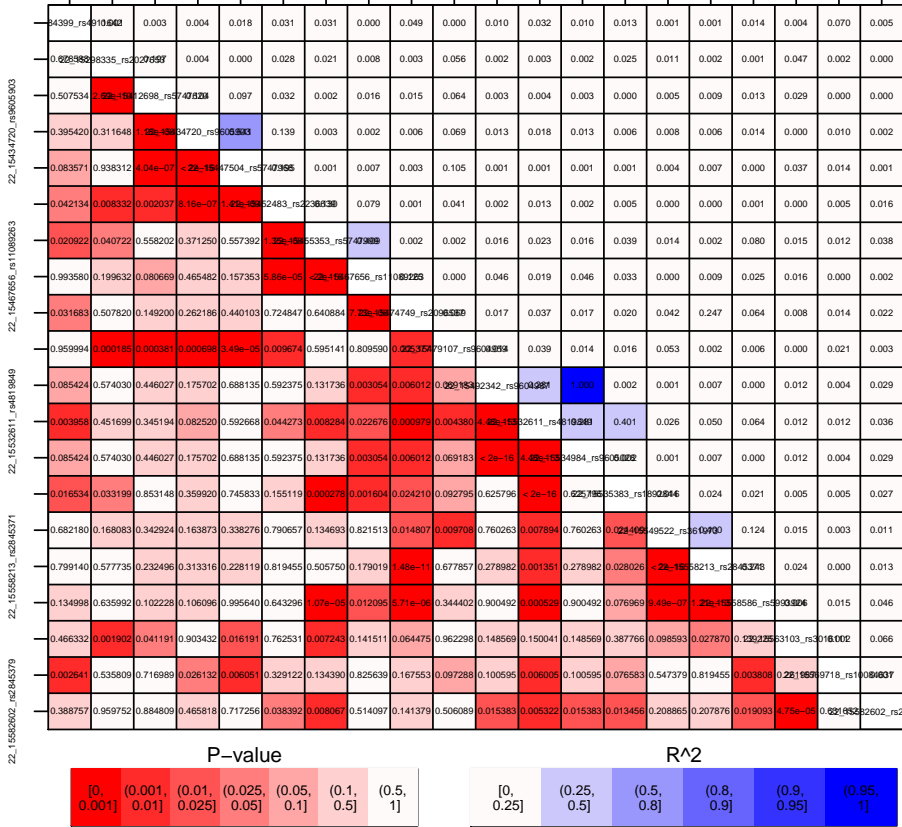
```

> ld.small <- LD(hm.a2[1:20])
> plot(ld.small)

```

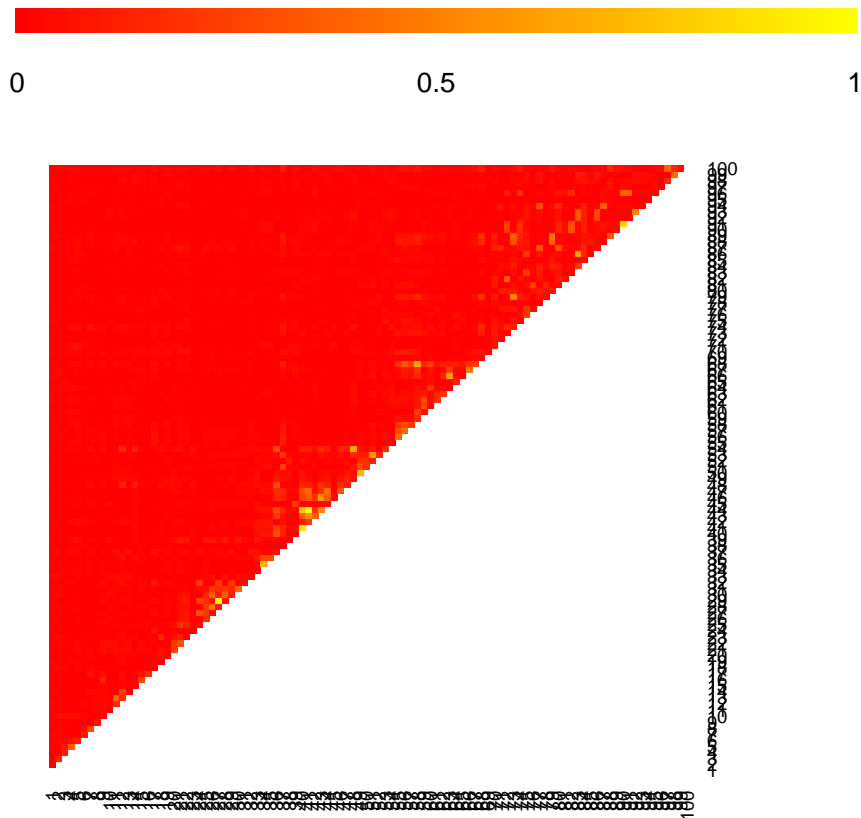
Linkage Disequilibrium

22_14884399_rs4911642 22_15434720_rs9605903 22_15455353_rs5747999 22_15479107_rs9604959 22_15534984_rs9605028 22_15558213_rs2845371 22_15569718_rs10084637



Alternative graphical display of LD:

```
> ld <- LD(hm.a2)
> LDView(ld@"X^2")
```



6 Hypothesis Testing

6.1 Armitage test

For the following examples, suppose 'A' is the minor allele, and 'a' is the major allele.

Armitage test using additive model to code genotype:

genotype	coding
AA	2
Aa	1
aa	0

```
> res.A <- Armitage(hm.a2, method = "A")
> head(res.A)
```

	stat	pvalue
22_14884399_rs4911642	4.38726430	3.620837e-02
22_15298335_rs2027653	8.62689991	3.312348e-03
22_15412698_rs5747620	0.01107079	9.162030e-01
22_15434720_rs9605903	34.00571358	5.495048e-09
22_15447504_rs5747968	35.11899027	3.101613e-09
22_15452483_rs2236639	2.43861135	1.183810e-01

	genotype	coding
Armitage test using recessive model to code genotype:	AA	1
	Aa	0
	aa	0

```
> res.R <- Armitage(hm.a2, method = "R")
> head(res.R)
```

	stat	pvalue
22_14884399_rs4911642	NA	NA
22_15298335_rs2027653	1.6258677	0.202275552
22_15412698_rs5747620	0.1285714	0.719917853
22_15434720_rs9605903	6.2068966	0.012725353
22_15447504_rs5747968	9.4736842	0.002084403
22_15452483_rs2236639	0.0000000	1.000000000

	genotype	coding
Armitage test using dominant model to code genotype:	AA	1
	Aa	1
	aa	0

```
> res.D <- Armitage(hm.a2, method = "D")
> head(res.D)
```

	stat	pvalue
22_14884399_rs4911642	4.3872643	3.620837e-02
22_15298335_rs2027653	10.0936094	1.487844e-03
22_15412698_rs5747620	0.2462380	6.197365e-01
22_15434720_rs9605903	37.0478170	1.152676e-09
22_15447504_rs5747968	35.8892308	2.088600e-09
22_15452483_rs2236639	3.0769231	7.941063e-02

6.2 Logistic regression

First, we need to construct some synthetic covariates on the founders

```
> sampleInfo(hm.f)$race <- sampleInfo(hm.f)$affected
> raceval <- sampleInfo(hm.f)$race - 1
> sampleInfo(hm.f)$Norm0.0 <- rnorm(nobs(hm.f), mean = 0 * raceval)
> sampleInfo(hm.f)$Norm0.5 <- rnorm(nobs(hm.f), mean = 0.5 * raceval)
> sampleInfo(hm.f)$Norm1.0 <- rnorm(nobs(hm.f), mean = 1 * raceval)
> sampleInfo(hm.f)$Norm1.5 <- rnorm(nobs(hm.f), mean = 1.5 * sampleInfo(hm.f)$race)
> doSample <- function(raceval, mult) {
+   prob <- c(0.33 - raceval * mult, 0.33 + (raceval * mult)/2, 0.33 +
+     (raceval * mult)/2)
+   factor(sample(x = c("Red", "Green", "Blue"), size = 1, p = prob,
+     rep = T))
+ }
> sampleInfo(hm.f)$Cat0.0 <- sapply(raceval, doSample, mult = 0)
> sampleInfo(hm.f)$Cat0.1 <- sapply(raceval, doSample, mult = 0.1)
> sampleInfo(hm.f)$Cat0.2 <- sapply(raceval, doSample, mult = 0.2)
> sampleInfo(hm.f)$Cat0.3 <- sapply(raceval, doSample, mult = 0.3)
```

Now, construct a function to fit the regression model and return the parameters and statistics that are of interest.

```

model <- function( markerName )
{
  # extract requested genetic marker
  genotype <- genotypes(hm.f,marker=markerName)

  # get data frame to use for fitting the model
  mframe <- model.frame(race ~ sex + Norm0.0 + Norm0.5 + Norm1.0 + Norm1.5 + genotype,
                        data=sampleInfo(hm.f) )

  # To test significance of a term, best method is to do anova of
  # the full model against a submodel omitting the particular term.
  # This avoids issues with changes in names of factor levels,
  # presence or absence of covariates, etc.
  result <- try(
    {
fit.with    <- glm( race==1 ~ sex + Norm0.0 + Norm0.5 + Norm1.0 + Norm1.5 + as.factor(genotype),
                    data=mframe, family="binomial")
fit.without <- glm( race==1 ~ sex + Norm0.0 + Norm0.5 + Norm1.0 + Norm1.5,
                    data=mframe, family="binomial")
                    anova(fit.with, fit.without, test="Chisq")$"P(>|Chi|)"[2]
    }
  )

  if(class(result)=="try-error")
    return(NA) # or return(result) to see the error messages
  else

    result # full result. Usually we want to specify exactly which
           # parameters and stats get returned so the format is consistent
           # across all markers.
}

```

```

> model <- function(markerName) {
+   genotype <- genotypes(hm.f, marker = markerName)
+   mframe <- model.frame(race ~ sex + Norm0.0 + Norm0.5 + Norm1.0 +
+     Norm1.5 + genotype, data = sampleInfo(hm.f))
+   result <- try({
+     fit.with <- glm(race == 1 ~ sex + Norm0.0 + Norm0.5 + Norm1.0 +
+       Norm1.5 + as.factor(genotype), data = mframe, family = "binomial")
+     fit.without <- glm(race == 1 ~ sex + Norm0.0 + Norm0.5 + Norm1.0 +
+       Norm1.5, data = mframe, family = "binomial")
+     anova(fit.with, fit.without, test = "Chisq")$"P(>|Chi|)"[2]
+   })
+   if (class(result) == "try-error")
+     return(NA)
+   else result
+ }

```

Fit to a subset of 50, then 100 and use this information to compute the expected run time for all markers:

```

> t1 <- unix.time(fits <- sapply(markerNames(hm.f)[1:50], model))[3]
> t1

```

```
elapsed
3.554
```

(This takes 5.365 seconds on my MacBook Pro, R 2.4.1)

```
> t2 <- unix.time(fits <- sapply(markerNames(hm.f)[1:100], model))[3]
> t2
```

```
elapsed
6.767
```

(This takes 11.878 seconds on my MacBook Pro, R 2.4.1)

Estimate total time to complete, in minutes

```
> t1 + (t2 - t1)/50 * (nmarker(hm.f) - 50)/60
```

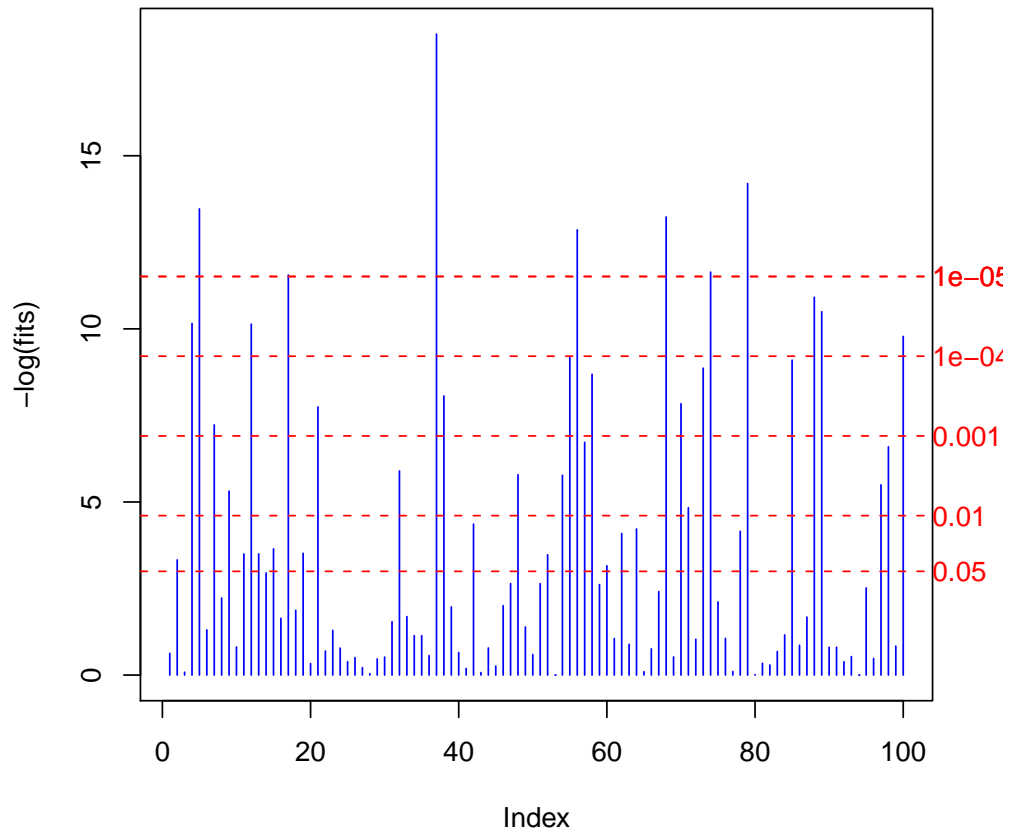
```
elapsed
12.26551
```

(This yields 23.02 minutes on my MacBook Pro, R 2.4.1)

Plot the p-values for the first 100 markers

```
> fits.sorted <- sort(fits)
> plot(-log(fits), type = "h", col = "blue")
> labels <- c(0.05, 0.01, 0.001, 1e-04, 1e-05, 1e-05)
> abline(h = -log(labels), lty = 2, col = "red")
> mtext(text = as.character(labels), side = 4, at = -log(labels), col = "red",
+       las = 1)
> title("Per-marker statistical significance")
```


Per-marker statistical significance



6.3 Family-Based Association Test ('fbat')

Do the fbat calculations:

```
> f <- fbat(hm.a2)
```

Show the p-values:

```
> summaryPvalue(f)
```

```
*****
              chisq rank    pvalue
22_14884399_rs4911642  1.6666667    1 0.19670560
22_15298335_rs2027653  0.0666667    1 0.79625341
22_15412698_rs5747620  0.57142857   1 0.44969180
22_15434720_rs9605903  1.0000000    1 0.31731051
22_15447504_rs5747968  1.0000000    1 0.31731051
22_15452483_rs2236639  0.0000000    1 1.00000000
22_15455353_rs5747999  1.05882353   1 0.30348366
22_15467656_rs11089263 4.5000000    1 0.03389485
22_15474749_rs2096537  0.75757576   1 0.38408825
```

22_15479107_rs9604959	2.77777778	1	0.09558070
22_15492342_rs9604967	1.00000000	1	0.31731051
22_15532611_rs4819849	1.00000000	1	0.31731051
22_15534984_rs9605028	1.00000000	1	0.31731051
22_15535383_rs1892844	0.00000000	0	0.00000000
22_15549522_rs361973	0.80645161	1	0.36917142
22_15558213_rs2845371	0.04761905	1	0.82725935
22_15558586_rs5993924	0.20000000	1	0.65472085
22_15563103_rs3016111	0.00000000	0	0.00000000
22_15569718_rs10084637	0.69230769	1	0.40538056
22_15582602_rs2845379	4.16666667	1	0.04122683
22_15583103_rs2845380	0.00000000	0	0.00000000
22_15594252_rs2845346	0.00000000	1	1.00000000
22_15608796_rs17433377	0.00000000	0	0.00000000
22_15634399_rs2190742	1.08695652	1	0.29714653
22_15636231_rs5748614	3.76923077	1	0.05220364
22_15644565_rs5748622	3.57142857	1	0.05878172
22_15644904_rs9605145	3.57142857	1	0.05878172
22_15645124_rs5748623	3.60000000	1	0.05777957
22_15645194_rs9605146	5.14285714	1	0.02334220
22_15647006_rs759235	0.60000000	1	0.43857803
22_15649076_rs2108585	2.90909091	1	0.08808151
22_15653728_rs9606468	0.00000000	1	1.00000000
22_15655394_rs5748636	0.36000000	1	0.54850624
22_15658762_rs4819535	0.69230769	1	0.40538056
22_15660822_rs5748648	0.69230769	1	0.40538056
22_15661931_rs738045	0.60000000	1	0.43857803
22_15665949_rs2385714	0.00000000	0	0.00000000
22_15668988_rs2072467	0.00000000	0	0.00000000
22_15669118_rs2072466	1.00000000	1	0.31731051
22_15674251_rs7291429	0.00000000	1	1.00000000
22_15681217_rs874835	0.03703704	1	0.84738966
22_15681843_rs874836	2.46153846	1	0.11666446
22_15684246_rs175139	2.46153846	1	0.11666446
22_15684887_rs983305	0.03703704	1	0.84738966
22_15686104_rs175140	0.66666667	1	0.41421618
22_15690741_rs175149	1.28571429	1	0.25683926
22_15692596_rs9606481	0.61538462	1	0.43276758
22_15695102_rs17363716	0.00000000	0	0.00000000
22_15695503_rs165757	0.11764706	1	0.73160059
22_15697233_rs175152	0.80645161	1	0.36917142
22_15698150_rs165611	0.92592593	1	0.33592381
22_15699156_rs165778	1.00000000	1	0.31731051
22_15706181_rs165810	0.00000000	0	0.00000000
22_15706432_rs2075120	0.00000000	0	0.00000000
22_15775610_rs737936	1.00000000	1	0.31731051
22_15776612_rs7288841	0.00000000	0	0.00000000
22_15777875_rs9606534	2.00000000	1	0.15729921
22_15778508_rs7292561	3.00000000	1	0.08326452
22_15778800_rs7293026	0.80645161	1	0.36917142
22_15778812_rs13058496	3.00000000	1	0.08326452

22_15779211_rs8136206	0.80000000	1	0.37109337
22_15785173_rs759081	1.00000000	1	0.31731051
22_15787349_rs5992587	1.80000000	1	0.17971249
22_15787566_rs11703901	2.66666667	1	0.10247043
22_15789897_rs12485066	3.00000000	1	0.08326452
22_15790373_rs5748744	3.85714286	1	0.04953461
22_15791899_rs9306242	3.24000000	1	0.07186064
22_15792216_rs9605179	4.00000000	1	0.04550026
22_15792806_rs5992590	0.61538462	1	0.43276758
22_15794103_rs5994097	2.13043478	1	0.14439979
22_15794640_rs9618937	0.50000000	1	0.47950012
22_15795572_rs5748748	0.69230769	1	0.40538056
22_15806401_rs5748755	2.66666667	1	0.10247043
22_15807037_rs2385785	2.00000000	1	0.15729921
22_15809384_rs1981707	5.53846154	1	0.01860293
22_15809434_rs1981708	1.60000000	1	0.20590321
22_15813210_rs4819923	2.46153846	1	0.11666446
22_15813888_rs5994105	0.40000000	1	0.52708926
22_15814084_rs5748760	1.00000000	1	0.31731051
22_15816846_rs2385786	0.12500000	1	0.72367361
22_15821524_rs5994110	1.00000000	1	0.31731051
22_15822154_rs17733785	0.33333333	1	0.56370286
22_15823131_rs7287116	0.20000000	1	0.65472085
22_15825502_rs5748765	0.03030303	1	0.86180443
22_15826157_rs1541529	0.00000000	0	0.00000000
22_15826914_rs5748766	0.81818182	1	0.36571230
22_15830515_rs2041607	0.69230769	1	0.40538056
22_15832966_rs757630	0.00000000	0	0.00000000
22_15842185_rs4819932	1.00000000	1	0.31731051
22_15847411_rs4819934	0.22222222	1	0.63735189
22_15847684_rs4819936	0.22222222	1	0.63735189
22_15850779_rs9618954	1.60000000	1	0.20590321
22_15855921_rs2399152	2.27272727	1	0.13166802
22_15868195_rs5748798	0.00000000	0	0.00000000
22_15869577_rs7291404	0.06666667	1	0.79625341
22_15869890_rs11913227	0.18181818	1	0.66981536
22_15870932_rs5994128	0.14285714	1	0.70545699
22_15872203_rs5994129	1.50000000	1	0.22067136
22_15872452_rs917838	0.18181818	1	0.66981536
22_15872533_rs2192155	0.00000000	0	0.00000000

Look at the fit details for a specific marker:

```
> viewstat(f, "22_14884399_rs4911642")
```

```
50 pedigree 180 persons
13 informative families at marker 22_14884399_rs4911642
The alleles of marker 22_14884399_rs4911642 >>
[1] 1 2
Score for marker 22_14884399_rs4911642 >>
[1] 5 21
```

```

Expected score for marker 22_14884399_rs4911642 >>
[1] 7.5 18.5
Covariance matrix of the score for marker 22_14884399_rs4911642 >>
      [,1] [,2]
[1,] 3.75 -3.75
[2,] -3.75 3.75
Moore-Penrose generalized inverse of covariance matrix
      [,1] [,2]
[1,] 0.06666667 -0.06666667
[2,] -0.06666667 0.06666667
test statistics for marker 22_14884399_rs4911642 >>
      chisq      rank      pvalue
1.6666667 1.0000000 0.1967056
*****

```

7 Study planning tools (GeneticsDesign package)

7.1 Power to detect a low-frequency allele

Compute the probability of missing an allele with frequency 0.15 when 20 genotypes are sampled:

```
> gregorius(freq = 0.15, N = 20)
```

```
$call
```

```
gregorius(freq = 0.15, N = 20)
```

```
$method
```

```
[1] "Compute missprob given N and freq"
```

```
$freq
```

```
[1] 0.15
```

```
$N
```

```
[1] 20
```

```
$missprob
```

```
[1] 0.1938351
```

Determine what sample size is required to observe all alleles with true frequency 0.15 with probability 0.95

```
> gregorius(freq = 0.15, missprob = 1 - 0.95)
```

```
$call
```

```
gregorius(freq = 0.15, missprob = 1 - 0.95)
```

```
$method
```

```
[1] "Determine minimal N given missprob and freq"
```

```
$freq
```

```
[1] 0.15
```

```
$N
```

```
[1] 29
```

```
$missprob  
[1] 0.04520557
```

7.2 Power for a genetics study using a quantitative outcome

Calculate power for genetics study using a quantitative outcome:

```
> GeneticPower.Quantitative.Numeric(N = 50, freq = 0.1, minh = "recessive",  
+   alpha = 0.05)
```

```
[1] 0.0600883
```

```
> GeneticPower.Quantitative.Factor(N = 50, freq = 0.1, minh = "recessive",  
+   alpha = 0.05)
```

```
[1] 0.08666032
```

For a range of sample sizes:

```
> power.range <- function(N, ...) {  
+   sapply(N, function(n) GeneticPower.Quantitative.Numeric(N = n, ...))  
+ }  
> power.range(N = c(25, 50, 100, 200, 500), freq = 0.1, minh = "recessive",  
+   alpha = 0.05)
```

```
[1] 0.05492509 0.06008830 0.07049383 0.09160716 0.15665828
```

Create a power table:

```
> fun <- function(p) power.range(freq = p, N = seq(100, 1000, by = 100),  
+   alpha = 0.05, minh = "recessive")  
> m <- sapply(X = seq(0.1, 0.9, by = 0.1), fun)  
> colnames(m) <- seq(0.1, 0.9, by = 0.1)  
> rownames(m) <- seq(100, 1000, by = 100)  
> print(round(m, 2))
```

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
100	0.07	0.20	0.47	0.76	0.93	0.98	0.99	0.99	0.97
200	0.09	0.35	0.77	0.97	1.00	1.00	1.00	1.00	1.00
300	0.11	0.49	0.91	1.00	1.00	1.00	1.00	1.00	1.00
400	0.13	0.61	0.97	1.00	1.00	1.00	1.00	1.00	1.00
500	0.16	0.70	0.99	1.00	1.00	1.00	1.00	1.00	1.00
600	0.18	0.78	1.00	1.00	1.00	1.00	1.00	1.00	1.00
700	0.20	0.84	1.00	1.00	1.00	1.00	1.00	1.00	1.00
800	0.22	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00
900	0.24	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1000	0.27	0.94	1.00	1.00	1.00	1.00	1.00	1.00	1.00

7.3 Power calculator for genetic linear trend association studies.

The power is for the test that disease is associated with a marker, given high risk allele frequency ('A'), disease prevalence, genotype relative risk ('Aa'), genotype relative risk ('AA'), LD measure (D' or R^2), marker allele frequency ('B'), number of cases, control:case ratio, and probability of the Type I error. The linear trend test (Cochran 1954; Armitage 1955) is used.

Using R^2 as the measuer of LD:

```
> res1 <- GPC(pA = 0.05, pD = 0.1, RRAa = 1.414, RRAA = 2, r2 = 0.9, pB = 0.06,
+ nCase = 500, ratio = 1, alpha = 0.05, quiet = FALSE)
```

```
Case-control parameters>>
```

```

                                     [,1]
Number of cases                      500.00000000
Number of controls                   500.00000000
High risk allele frequency (A)       0.05000000
Prevalence                           0.10000000
Genotypic relative risk Aa           1.41400000
Genotypic relative risk AA           2.00000000
Genotypic risk for aa (baseline)     0.09598495
```

```
Marker locus B>>
```

```

                                     [,1]
High risk allele frequency (B)       0.06000000
Linkage disequilibrium (D')          0.99723021
Penetrance at marker genotype bb     0.09599596
Penetrance at marker genotype Bb     0.12902094
Penetrance at marker genotype BB     0.17344738
Genotypic odds ratio Bb              1.39498627
Genotypic odds ratio BB              1.97612611
```

```
Expected allele frequencies>>
```

```

      Case   Control
B 0.07901192 0.05788756
b 0.92098808 0.94211244
```

```
Expected genotype frequencies>>
```

```

      Case   Control
BB 0.006244106 0.003306210
Bb 0.145535624 0.109162708
bb 0.848220271 0.887531081
```

```
Case-control statistics>>
```

```

Alpha   Power
0.100 0.58900688
0.050 0.46393515
0.010 0.23992694
0.001 0.07762229
0.050 0.46393515
```

```
power (alpha= 0.05 )= 0.4639352 ncp= 3.494199
```

```
Using  $D'$  as the measure of LD:
```

```
> res2 <- GPC.default(pA = 0.05, pD = 0.1, RRAa = 1.414, RRAA = 2, Dprime = 0.9,
+ pB = 0.06, nCase = 500, ratio = 1, alpha = 0.05, quiet = FALSE)
```

```
Case-control parameters>>
```

```

                                     [,1]
Number of cases                      500.00000000
Number of controls                   500.00000000
```

High risk allele frequency (A) 0.05000000
Prevalence 0.10000000
Genotypic relative risk Aa 1.41400000
Genotypic relative risk AA 2.00000000
Genotypic risk for aa (baseline) 0.09598495

Marker locus B>>

[,1]
High risk allele frequency (B) 0.06000000
Linkage disequilibrium (D') 0.90000000
Penetrance at marker genotype bb 0.09638274
Penetrance at marker genotype Bb 0.12624798
Penetrance at marker genotype BB 0.16539976
Genotypic odds ratio Bb 1.35463248
Genotypic odds ratio BB 1.85798248

Expected allele frequencies>>

	Case	Control
B	0.07715825	0.05809353
b	0.92284175	0.94190647

Expected genotype frequencies>>

	Case	Control
BB	0.005954391	0.003338401
Bb	0.142407718	0.109510254
bb	0.851637891	0.887151345

Case-control statistics>>

Alpha	Power
0.100	0.52110203
0.050	0.39631201
0.010	0.18968278
0.001	0.05549084
0.050	0.39631201

power (alpha= 0.05)= 0.396312 ncp= 2.878885

The End.