Dectecting disease variants in case-parent trio studies using the Bioconductor software package `trio`

SUPPLEMENTARY MATERIAL

Holger Schwender[1] , Qing Li[2] , Christoph Neumann[3], Margaret A. Taub[4] , Samuel G. Younkin[5] , Philipp Berger[1], Robert B. Scharpf[6] , Terri H. Beaty[7] , Ingo Ruczinski[4]

[1]Mathematical Institute, Heinrich Heine University, Düsseldorf, Germany, [2]Inherited Disease Research Branch, National Human Genome Research Institute, Baltimore MD, USA, [3]Faculty of Statistics, TU Dortmund University, Dortmund, Germany, [4]Department of Biostatistics, Johns Hopkins University, Baltimore MD, USA, [5]Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison WI, USA, [6]Department of Oncology, Johns Hopkins University, Baltimore MD, USA, [7]Department of Epidemiology, Johns Hopkins University, Baltimore MD, USA.

1 Linkage Disequilibrium Block Estimation

The trio package provides functions for calculating the linkage disequilibrium (LD) measures r^2 and D' for pairs of SNPs based on the procedure of Hill (1974). The function getLD() computes LD measures for all pairs of SNPs in a genotype matrix, and thus should only be applied to a moderate number of markers. The function getLDlarge() allows for a genome-wide application since it calculates only the LD measures for each SNP with a pre-determined number of neighboring markers (which can be specified with the argument neighbors). Both getLD() and getLDlarge() are not restricted to trio data, and can also be applied to population-based designs such as case-control data. If the input is a matrix in genotype format, then the argument parentsOnly should be set to TRUE, to only use parental genotypes to determine the LD measures. Both the output of these functions and the actual genotype matrix can be passed to the function findLDblocks() to delineate the LD blocks based on a modified version of the procedure of Gabriel et al. (2002), as described in Wall and Pritchard (2003). For the estimation of the variance of D' we use the approximation proposed by Zapata et al. (1997) instead of a bootstrap approach.

For example, we calculate the r^2 and D' values for each of 28,585 SNPs from chromosome 8 and their respective 100 neighbors (50 in each direction):

```
> ldChr8 <- getLDlarge(matChr8, neighbors = 50, parentsOnly = TRUE)
```

For the 111 FGF SNPs, all pairwise r^2 and D' can be determined by calling

```
> ldFGF <- getLD(genoFGF, parentsOnly = TRUE, addVarN = TRUE)
```

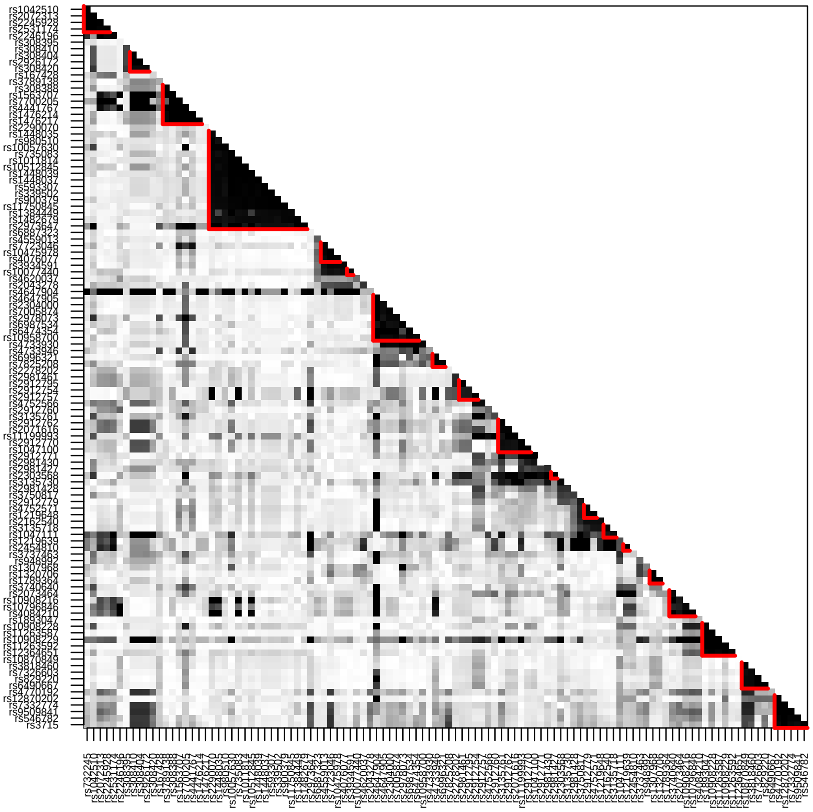
When the output of getLD() is used as input in findLDblocks(), the argument addVarN of getLD() (or getLDlarge()) has to be set to TRUE, to estimate the variance of D' required by findLDblocks(). The LD blocks can then be determined by the call

```
> blocksFGF <- findLDblocks(ldFGF)
```

The r^2 or D' values can be plotted by passing the output of `getLD()`, `getLDlarge()`, or `findLDblocks()` to the `S3` method `plot()`. For `findLDblocks()` the borders of the estimated LD blocks are also displayed (Supplementary Figure 1), for example by

```
> plot(blocksFGF, y = "Dprime", cexAxis = 0.5)
```

where `cexAxis` controls the relative size of the SNP names in the axes. Specific subregions can also be plotted by specifying the arguments `start` and `end`, by the column numbers, or the names of the SNPs specifying the beginning and the end of the subregion.



Supplementary Figure 1: LD blocks estimated with `findLDblocks()` in the analysis of the 111 SNPs from the FGF genes. In this analysis, 19 LD blocks containing between 2 and 16 SNPs are identified. 17 of the SNPs are not assigned to an LD block.

2 Gene-Environment Interactions

Below, the default output for `outGxE` from Section 3.1 in the main manuscript is displayed.

```
> outGxE

          Genotypic TDT for GxE Interactions with Binary E


Model Type: Additive


Top 5 GxE Interactions:
              Coef     OR  Lower Upper     SE Statistic   p-value Trios0 Trios1
rs876058    0.5365 1.7101 1.3663 2.140 0.1145     21.96 2.780e-06    679    382
rs6994211   0.4964 1.6428 1.3275 2.033 0.1087     20.85 4.977e-06    721    419
rs6989058   0.4861 1.6260 1.3027 2.029 0.1131     18.48 1.719e-05    678    395
rs12548247  0.6957 2.0051 1.4595 2.755 0.1620     18.44 1.757e-05    395    214
rs12682543 -0.4809 0.6182 0.4958 0.771 0.1127     18.22 1.968e-05    689    386



Effects of the SNPs in the Corresponding GxE Models:
              Coef     OR  Lower  Upper      SE Statistic   p-value
rs876058   -0.1531 0.8581 0.7506 0.9809 0.06824     5.032 0.0248855
rs6994211  -0.1393 0.8700 0.7651 0.9893 0.06557     4.511 0.0336693
rs6989058  -0.1102 0.8957 0.7842 1.0230 0.06783     2.640 0.1042344
rs12548247 -0.3141 0.7305 0.6049 0.8820 0.09620    10.659 0.0010952
rs12682543  0.1444 1.1553 1.0126 1.3182 0.06729     4.605 0.0318862



ORs for Exposed Cases:
               OR  Lower  Upper
rs876058   1.4673 1.2254 1.7570
rs6994211  1.4292 1.2058 1.6940
```

```
rs6989058  1.4563 1.2196 1.7389
rs12548247 1.4646 1.1344 1.8911
rs12682543 0.7143 0.5984 0.8527


2 df Likelihood Ratio Test, 2 df Wald Test, 1 df Likelihood Ratio Test:
           2df Stat 2df p-Value Wald Stat Wald p-value 1df Stat 1df p-Value
rs876058      22.77   1.137e-05     22.43    1.345e-05    22.23   2.418e-06
rs6994211     21.75   1.892e-05     21.47    2.178e-05    21.06   4.449e-06
rs6989058     20.21   4.092e-05     19.90    4.778e-05    18.67   1.552e-05
rs12548247    19.52   5.785e-05     19.23    6.682e-05    18.75   1.494e-05
rs12682543    18.68   8.774e-05     18.47    9.740e-05    18.39   1.795e-05
```

3 Higher-Order Interactions

3.1 Trio Logic Regression

The application of trio logic regression to case-parent trios with `trioLR()` requires a complete matrix of binary predictors (typically, SNPs as a pair of binary variables encoding dominant and recessive effects) for the observed affected proband and the pseudo-controls. This matrix can conveniently be generated with the functions `trio.check()` and `trio.prepare()`. First, `trio.check()` is applied to a data frame in ped format to check for Mendelian errors, which removes Mendelian errors by setting the three respective genotypes to `NA`. Next, `trio.prepare()` is applied to the output of `trio.check()` to generate the pseudo-controls, to impute missing values using the haplotype-based procedures proposed by Li et al. (2010), and to generate the response and the binary predictors coding for dominant and recessive effects of the SNPs. For linked markers it is highly recommended to use information about the haplotype structure in `trio.prepare()`, which has to specified via the argument `block`. Otherwise, all SNPs are treated as unlinked.

As an example, we apply trio logic regression to the SNPs from the FGF genes stored in the ped file `fgf.ped`. We prepare the data set for this analysis using the following commands:

```
> pedFGF <- read.pedfile("fgf.ped")
> outCheck <- trio.check(pedFGF)
> datTLR <- trio.prepare(outCheck, blocks = tabBlocks)
```

The haplotype information in `tabBlocks` can be generated as described in Supplementary Section 1, using the component `blocksFGF$blocks` of the object returned by function `findLDblocks()` in the FGF data analysis example. Because of massive computational complexity, the maximum number of SNPs per block is restricted to seven by Li et al. (2010). To split the LD blocks into sub-blocks of size seven or smaller, the function `splitBlocks` can be used.

```
> tabBlocks <- splitBlocks(blocksFGF)
```

Trio logic regression is then invoked by the following code:

```
> trioLR(datTLR)
```

This performs a trio regression analysis with the default setting (see Ruczinski et al., 2003, for details). Since the method is based on simulated annealing, a stochastic search algorithm, the performance can usually be improved by changing the default values of some of the parameters. The most important parameters are the number of iterations as well and the start and end “temperature” (on $\log_{10}$-scale) of the search algorithm. In our experience, employing 1 and -3 as start and end temperature usually works well in the analysis of trio data, and at least fifty thousand iterations should be used when analyzing a moderate number of SNPs, such as in the FGF example. We specify these temperatures and call for 50,000 iterations using

```
> myControl <- lrControl(start = 1, end = -3, iter = 50000)
```

The object `myControl` can then be used to specify the argument `control` in the `trioLR()` function. To save computing time, a greedy search algorithm instead of simulated annealing can also be used in trio logic regression (see Ruczinski et al., 2003) by setting search = "greedy" in trioLR(). However, as with any greedy algorithm, chances are much higher to only detect a local, not the global optimum. Another alternative to simulated annealing is a Bayesian logic regression (Kooperberg and Ruczinski, 2005) which can be applied to trio data by setting search = "mcmc" in trioLR().

To detect the best model of a pre-specified size (defined by the number of variables included in the model), we can specify the nleaves argument in trioLR() which by default is set to 5. For the prepared FGF data, the best model (according to the likelihood) of size four is found by typing the following code:

```
> tlrOut <- trioLR(datTLR, nleaves = 4, control = myControl)
```

The resulting object from this analysis is displayed below.

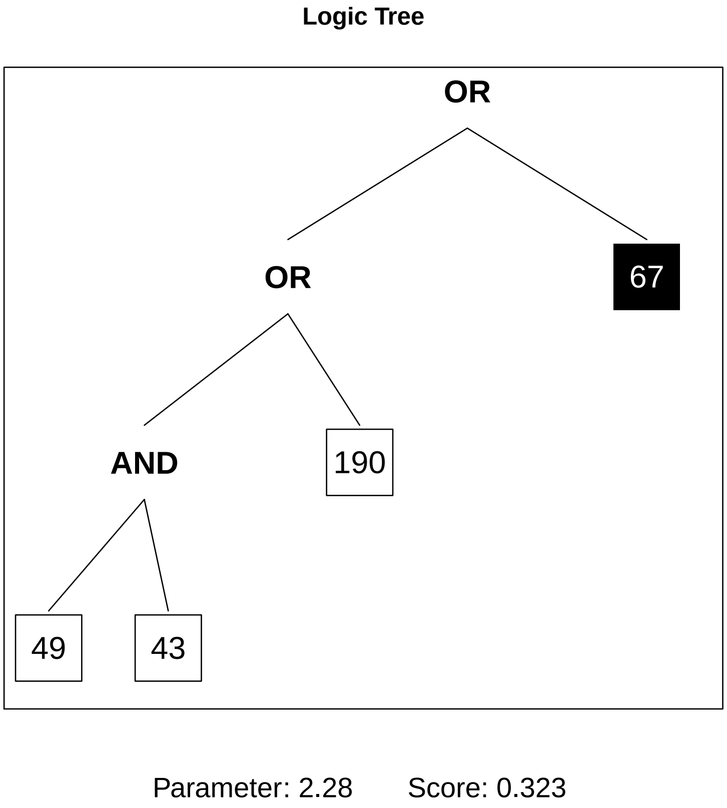
```
> tlrOut

         Trio Logic Regression


Search Algorithm: Simulated Annealing


A single model has been fitted:
2.28 * (((rs1011814.D & rs980510.D) | rs1893047.R) | !rs1482679.D)
Score:  0.323
```

For model interpretation and methods for model selection, see Ruczinski et al. (2003) and Li et al. (2010). For trio data, the latter is carried out via the function trio.permTest(). Note that for ease of interpretation it is also possible to display the above logic term in disjunctive normal form (a series of Boolean “and” statements, connected by “or” operators) by calling

```
> print(tlrOut, asDNF = TRUE)
```

For a graphical representation of the trio logic regression model, the logic expression can be plotted as a logic tree (Supplementary Figure 2) by typing

```
> plot(tlrOut)
```



Supplementary Figure 2: Logic tree representing the model found in the trio logic regression analysis of the SNPs from the FGF genes. See Ruczinski et al. (2003) for more logic tree details.

3.2 TrioFS

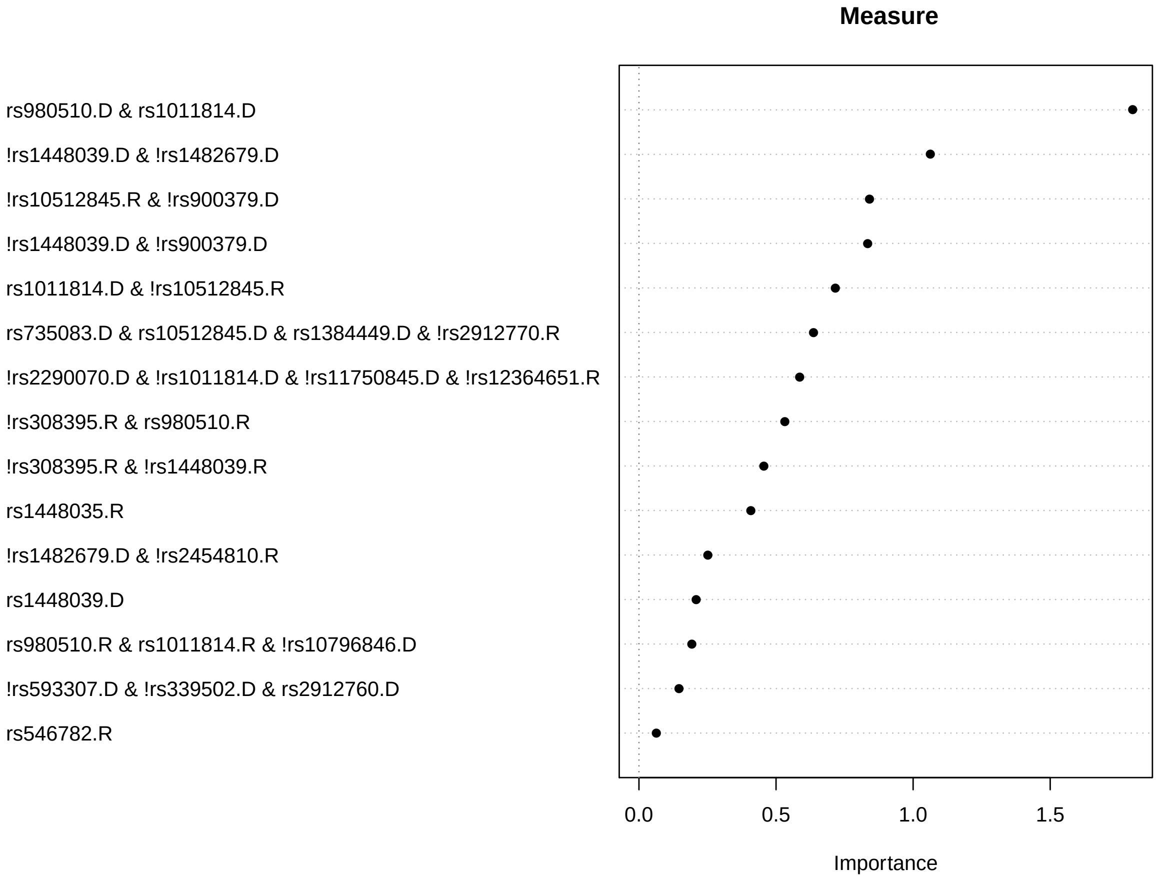
Since `trioLR()` is based on a stochastic search algorithm, repeated applications of the algorithm to the same data often yields different and competing models, for example consisting of different SNPs from the same LD block. To stabilize the search for interactions and to quantify the importance of these interactions on disease risk, Schwender et al. (2011) proposed a method called trio Feature Selection (trioFS). In this resampling-based procedure, trio logic regression is applied to several bootstrap or subsamples of the data, and the out-of-bag trios of each iteration are used to determine the importance of the detected interactions, and to rank these interactions by their influence on disease risk.

The approach is applied to the output of `trioPrepare()` in the same way as `trioLR()`, by invoking the function `trioFS()`. In `trioFS()`, the number of bootstrap or subsamples can be specified via the argument `B` (which by default is `B = 20`), and it is possible to choose between bootstrap samples (`replace = TRUE`, the default) and subsamples (`replace = FALSE`). In the latter case, the percentage of trios in the subsamples can be specified by the argument `sub.frac`, which by default is set to `0.632`, the expected proportion of different trios in the bootstrap samples. In our example, we apply trioFS to the FGF data.

```
> tfsOut <- trioFS(datTLR, nleaves = 4, control = myControl)
> tfsOut
Identification of Interactions Using Trio Logic Regression


Number of Iterations:  20
Sampling Method:       Bagging
Number of Trees:       1
Max. Number of Leaves: 4


The 5 Most Important Interactions:


  Importance Proportion                  Expression
1       1.80       0.10    rs980510.D & rs1011814.D
2       1.06       0.05 !rs1448039.D & !rs1482679.D
```

```
3       0.84       0.05 !rs10512845.R & !rs900379.D

4       0.83       0.05  !rs1448039.D & !rs900379.D

5       0.72       0.05 rs1011814.D & !rs10512845.R
```

Here, `rs980510.D & rs1011814.D` is identified as the interaction with the largest importance, and is included in 10% of the models. The `S3` method `plot()` can be employed to display these findings graphically (see Supplementary Figure 3).



Supplementary Figure 3: The 15 most important SNP interactions found in a trioFS analysis of the SNPs from the FGF genes.

References

Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Liu-Cordero SN, Rotimi C, Adeyemo A, Cooper R, Ward R, Lander ES, Daly MJ, Altshuler D. 2002. The structure of haplotype blocks in the human genome. Science 296:2225–2229.

Hill WO. 1974. Estimation of linkage disequilibrium in randomly mating populations. Heredity 33:229–239.

Kooperberg C, Ruczinski I. 2005. Identifying interacting SNPs using Monte Carlo logic regression. Genet Epidemiol 28:157–170.

Li Q, Fallin MD, Louis TA, Lasseter VK, McGrath JA, Avramopoulos D, Wolyniec PS, Valle D, Liang KY, Pulver AE, Ruczinski I. 2010. Detection of SNP-SNP interactions in trios of parents with schizophrenic children. Genet Epidemiol 34:396–406.

Ruczinski I, Kooperberg C, LeBlanc M. 2003. Logic regression. J Comput Graph Stat 12:475–511.

Schwender H, Bowers K, Fallin MD, Ruczinski I. 2011. Importance measures for epistatic interactions in case-parent trios. Ann Hum Genet 75:122–132.

Wall JD, Pritchard JK. 2003. Assessing the performance of the haplotype block model of linkage disequilibrium. Am J Hum Genet 73:502–515.

Zapata C, Alvarez G, Carollo C. 1997. Approximate variance of the standardized measure of gametic disequilibrium D’. Am J Hum Genet 61:771–774.