QMS2 How-To

A step-by-step guide for using the QTL Macro for SAS® Software[1]

Dr. Jeffrey Lessem
Research Associate
University of Colorado
Institute for Behavioral Genetics

447 CB
Boulder CO, 80309-0447
USA
Jeff.Lessem@Colorado.EDU

Dr. Stacey Cherny
Head of Statistical Genetics Applications
University of Oxford
Wellcome Trust Centre for Human Genetics

Roosevelt Drive
Oxford, OX3 7BN
United Kingdom
cherny@well.ox.ac.uk

QMS2 How-To: A step-by-step guide for using the QTL Macro for SAS® Software[1]

by Dr. Jeffrey Lessem and Dr. Stacey Cherny

Revision History

Revision 0.10 03.25.2001

Table of Contents

List of Examples

Chapter 1. Introduction

This is a SAS(r) Software macro package for performing multipoint QTL mapping using the DeFries-Fulker multiple regression approach to unselected and selected samples (basic and augmented) [1] [2] [3], the Haseman-Elston approach [4], the New Haseman-Elston approach [5], and the Sham-Purcell [6] update of the Haseman-Elston method.

About the Document

The latest version of this document is always available at http://qms2.sourceforge.net/ in multipage (http://qms2.sourceforge.net/How-To/), single page (http://qms2.sourceforge.net/How-To.html), PostScript (http://qms2.sourceforge.net/How-To.ps), Portable Document Format (http://qms2.sourceforge.net/How-To.pdf), and text (http://qms2.sourceforge.net/How-To.txt) formats. The Portable Document Format version is recommended for printing. It requires the freely available Adobe Acrobat (http://www.adobe.com/acrobat/) software to view.

This document was created using DocBook (http://www.docbook.org/).

License Terms

Reporting Bugs

Reporting bugs is highly encouraged, as this may be the only way they will come to the authors' attention and get fixed. Bug reports should be sent to <Jeff.Lessem@Colorado.EDU>.

Obtaining and Extracting QMS2

The homepage for QMS2 is http://qms2.sourceforge.net/. The latest version can always be downloaded from http://sourceforge.net/projects/qms2/.

Once you have downloaded QMS2, you should have a file called, for example, `qms2-0.07.tar.gz`. To extract the file use the command **gunzip -c qms2-0.07.tar.gz | tar -xvf -** which will create the directory `qms2/` containing the elements of this package.

Requirements for using QMS2

QMS2 is not a stand alone package. Several other pieces of software, most importantly the SAS Software System, are required.

- *SAS Software* This software is a macro which runs under SAS Software. If you do now know what SAS Software is, then this package is not for you. The software has been tested with SAS Software versions 6.12, 8, and 8.1 under Compaq Tru64 and Version 8.2 under Linux. There are no known reasons why the software will not work with SAS Software versions 6.12 or higher on any Unix or Unix-like operating system.

  A version of this software labeled as "for Windows" is also released, but it is completely untested. It merely contains the Unix version of the software with the end-of-line characters changed to the DOS standard. As neither author works with SAS Software for Windows, it has not even been verified as working.

- *Genehunter 2* Genehunter 2 is not strictly required, as any data which contains IBD information can be used, but as written, the import macro `ghimport.sas` expects files that are produced by Genehunter 2. Genehunter 2 is available from http://www-genome.wi.mit.edu/ftp/distribution/software/genehunter/.
- *bash* The `pre-sas.sh` script uses bash. Any sufficiently configured Unix or Unix-like system should contain bash. It is freely available from http://www.gnu.org/software/bash/bash.html.
- *grep* The `pre-sas.sh` also requires grep. If you are using a Unix or Unix-like operating system, you have grep.
- *expand* The `pre-sas.sh` also requires expand, which converts tabs to spaces. Unix and Unix-like systems should have this, but if not it is freely available as part of the GNU Textutils package at http://www.gnu.org/software/textutils/textutils.html.
- *sed* The `pre-sas.sh` also requires sed. If you are using a Unix or Unix-like operating system, you have sed.

Notes

1. DeFries, J. C., & Fulker, D. W. (1985). Multiple regression analysis of twin data. Behavior Genetics, 15, 467-473.

2. LaBuda, M. C., DeFries, J. C., & Fulker, D. W. (1986). Multiple regression analysis of twin data obtained from selected samples. Genetic Epidemiology, 3, 425-433.
3. DeFries, J. C., & Fulker, D. W. (1988). Multiple regression analysis of twin data: Etiology of deviant scores versus individual differences. Acta Geneticae Medicae et Gemellologiae, 37, 205-216.
4. Haseman, J. K., & Elston, R. C. (1972). The investigation of linkage between a quantitative trait and a marker locus. Behavior Genetics, 2, 3-19.
5. Haseman, J. K., & Elston, R. C. (in press). Haseman and Elston revisited. Genetic Epidemiology.
6. Sham, P. C., & Purcell, S. (in press). Equivalance between Haseman-Elston and variance components linkage analyses for sib pairs. American Journal of Human Genetics.

Chapter 2. I just want to analyze my data!

If you have already obtained QMS2 and are reasonably familiar with Unix or Unix-like systems and SAS Software, then you might only need the information in the Section called *Putting it all together*. This section give a brief description of exactly what to type to go through a complete analysis using Genehunter 2 and QMS2. Each step will be described in more detail in the subsequent chapters. If you are not comfortable writing your own SAS Software scripts and using a Unix or Unix-like command line, then you should read this entire document.

Warning

This document does *not*:

- Give a complete tutorial on all of the options available with QMS2.
- Explain how to format raw data for use with Genehunter 2.
- Describe how to interpret the results of QMS2.

Because QMS2 was primarily written to perform the DF analysis, the default settings are selected to correctly handle data destined for a DF analysis. This includes expecting to have probands specified on the input file and double entering the data. This is probably not appropriate for the DF Augmented, HE, New HE, and SP models.

The `qms2.sas` macro is configured to produce no output by default. Though a limited and probably ineffectual safety measure it was designed by the authors to require active intervention by the user before any files or datasets are overwritten.

Warning

Using `proc print;` procedures on the output statistics and influence statistics datasets can produce a huge amount of data. Influence statistics are produced for each sib pair at each position, so even moderately sized datasets can produce listings of 10s or 100s of megabytes. It is strongly recommended that some post-processing be performened on the influence statistics, such as only printing the top 10 most influentual cases.

Data analysis considerations

As it is written, QMS2 uses all five models:

- DeFries-Fulker Basic model (DF)
- De-Fries-Fulker Augmented (DF Augmented)
- Haseman-Elston (HE)
- New Haseman-Elston (New HE)

- Sham-Purcell (SP

every time it is run. Because these models have different properties it is unlikely that it is appropriate to use all of them on the same dataset. Some steps have been taken to make the data better conform to the assumptions of all of the models. Even when requested, double entry is not used for the HE and SP models, because it would always be inappropriate. This does allow for comparison of the DF, HE, and SP models from the same run of QMS2, but does not substitute for having an understanding of the different analyses being performed.

In QMS2 version 0.10 and later single entered data is used for the HE and SP models even if double entry is requested, because double entry is not needed for these models. These models uses a squared sib-pair difference test, so sibling order is not important---A, B will produce the same results as B, A.

Another important factor to take into consideration is the direction of the *t*. Under the HE model the *t*s of interest will have negative values, while under the DF Augmented and the New HE the *t*s of interest will be positive. The direction of the *t* of the DF Basic model depends on the scaling of the data. If the mean value of the selected group is lower than the mean value of their co-sibs (i.e. selection on a low score) the *t*s of interest will have a negative value because the regression line has a negative slope. If the mean value of the selected group is higher than the mean value of their co-sibs (i.e. selection on a high score) the *t*s of interest will have a positive value because the regression line has a positive slope.

These things must be taken into account when considering whether to get excited about the peaks or the troughs of the graphs. Genehunter 2 and other programs may truncate positive *t*s for the HE model, but QMS2 does no filtering of the *t-values*.

QMS2 (in version 0.10 and later) does filter the *p-value* that is reported for all of the models. For the DF Augmented, New HE, HE, and SP models, where the interesting tail is known ahead of time, the *p-values* for the improper tail are set to 1. The interpretation of this would be *based on the observed pattern of data, the null hypothesis could never be rejected*.

Under the DF model, QMS2 has no way of knowing ahead of time which tail is interesting, so this can be specified when calling the macro. An explanation of the commands used are in Chapter qms2.sas. Generally, the tail of interest is the one in which selection was performed. For example, if the probands are selected from the low end of the distribution, then the low (negative) tail is the one of interest. However, if you are using transformed scores then your transformation will effect which tail is interesting. For example, if you are estimating the heritability of the QTL by dividing the scores by the difference between the proband mean and the population mean, then the high tail is interesting, regardless of which tail was selected.

Putting it all together

This section is put at the beginning to both provide a quick reference to remind the user of the commands to type and to provide an introduction to what will come in later sections. Unless you are already familiar QMS2 this section will probably not be enough to get the data analysis started.

Example 2-1. What to type

```
$ gunzip -c qms2-0.07.tar.gz | tar -xvf - ❶
qms2/qms2.sas
qms2/ghimport.sas
```

```
qms2/pre-sas.sh
$ cd qms2
$ mv dump.ibd dump.ibd.old ❷
$ gh2 < my_genehunter.in
analyzing pedigree 649... ❸
using non-originals:  3 4 5 6

analyzing pedigree 650...
using non-originals:  3 4 5 6

npl:6> file to store IBD distribution [ibd_dist.out]:
npl:7>
        ...goodbye...
$ pre-sas.sh dump.ibd
Converting "dump.ibd" to "dump-sas.ibd"...done.
$ sas myanalyzer.sas
$ less myanalyzer.log ❹ ❺
$ less myanalyzer.lst ❻
$ gv DF_Augmented.eps & ❼
$ gv DeFries-Fulker.eps &
$ gv Haseman-Elston.eps &
$ gv New-HE.eps &
$ gv Sham-Purcell.eps &
```

❶ In this example the shell prompt is represented by "`$`", so any lines where the user types are preceded by a `$`.

❷ When redirecting commands into Genehunter 2 (as is done on the next line), Genehunter 2 will not overwrite an existing output file, so any existing file needs to be moved out of the way.

❸ Genehunter 2 will produce many more lines of output, but only the last few lines are shown here.

❹ less (http://home.flash.net/~marknu/less/) is a pager (a program which displays a text file on the screen). There is no particular reason to use less, other than that it is a very good pager, more, a favorite editor, or any other program that allows you to view text could be substituted here.

❺ The log should be checked to make sure that the SAS Software ran correctly. Searching the `log` for the string `error` will show any problems which prevented SAS Software from completing the script.

❻ The output from the SAS Software will be saved into a `lst` file. Viewing the file will show the results of the analyses for each model.

❼ gv (http://wwwthep.physik.uni-mainz.de/~plass/gv/) will display the PostScript plots on the screen. gv is a front end to Ghostscript (http://www.cs.wisc.edu/~ghost/), a PostScript interpreter. The "`&`" causes the program to be run in the background, returning a prompt immediately.

Example 2-2. `my_genehunter.in`

```
disp score off
load mark linkage.dat
increment distance 2
scan ped genehunter.ped
dump ibd
```

```
dump.ibd
quit
```

Example 2-3. `myanalyzer.sas`

```
/* Tell SAS to look in the current directory for any macros */
options mautosource sasautos=('.');

/* read in data files */
%ghimport(ibdfile=dump-sas.ibd,
          phenfile=genehunter.ped,
          markers=10,
          missing=-99);

/* analyze the imported data and produce plots */
%qms2(infce=1,
      plot=1);

/* print out the results of the analysis */
proc print data=stats;

/* print out the influence statistics */
proc print data=infce
```

Chapter 3. A step-by-step walk-through

This section contains a step-by-step walk-through of moving data through Genehunter 2 and QMS2

Genehunter 2

Initially the data will have to be prepared in a format suitable for Genehunter 2. You should have two files:

- A file containing the marker information and distances. In this example, the file will be referred to as `linkage.dat`.
- A file containing the genotypes and phenotypic scores for each individual in the sample. In this example, the file will be referred to as `genehunter.ped`.

Once these files are ready, they need to be run through Genehunter 2 to produce IBD data. The commands in `my_genehunter.in` will create the file `dump.ibd` which contains the IBD values for each pairing.

Example 3-1. my_genehunter.in

```
disp score off
load mark linkage.dat
increment distance 2
scan ped genehunter.ped
dump ibd
dump.ibd
quit
```

The commands in `my_genehunter.in` can either be typed directly into Genehunter 2, or can be redirected from the command line:

> **Warning**
>
> When using redirection, if `dump.ibd`, or whatever file is specified after **dump ibd**, exists, then Genehunter 2 will not save the IBDs.

```
$ gh2 < my_genehunter.in
```

...

```
analyzing pedigree 649...
using non-originals:  3 4 5 6

analyzing pedigree 650...
using non-originals:  3 4 5 6
```

```
npl:6> file to store IBD distribution [ibd_dist.out]:
npl:7>
        ...goodbye...
$
```

pre-sas.sh

The `pre-sas.sh` script is used to convert the **dump ibd** output of Genehunter 2 into a format that can be easily read by SAS Software. It is fully documented in Chapter pre-sas.sh. Run `pre-sas.sh` with the format **./pre-sas.sh *file-to-convert file-to-create***. If no *file-to-create* is specified, a filename based on the *file-to-convert* will be created.

Example 3-2. Running `pre-sas.sh`.

```
$ ./pre-sas.sh dump.ibd
Converting "dump.ibd" to "dump-sas.ibd"...done.
$
```

ghimport.sas

The SAS Software macro `ghimport.sas` reads in the `dump-sas.ibd` and `genehunter.ped` files (as named in this example) and creates a SAS Software dataset suitable for analysis with `qms2.sas`. `ghimport.sas` is fully documented in Chapter ghimport.sas. The easiest way to use `ghimport.sas` is calling it using the SAS Software option mautosource. This allows the replacement of the `ghimport.sas` macro without changing the SAS Software script.

Example 3-3. Setting mautosource

```
options mautosource sasautos=('$HOME/sas/qms2');
```

The `ghimport.sas` is then called using a SAS Software macro reference. Macros are called by preceding the name of the macro with a "`%`". Any variables to be passed to the macro are placed inside "`( )`" and separated by a "`,`". To allow for default values, all macro variables in `ghimport.sas` are called using the form `variable=`*`value`*. To improve readability, a macro call can be broken across multiple lines.

In Example 3-4 the macro is called with all of the available arguments. IBDs will be read from the file `dump-sas.ibd`, and the phenotypes will be read from `genehunter.ped`. The phenotype file contains 10 markers and uses the value "-99" to represent missing data. The data will NOT be read in as-is (i.e. there are probands in the dataset and double entry will be performed). The output will be saved the the dataset `sample` in the `qtl` library.

Example 3-4. Calling `ghimport.sas`

```
%ghimport(ibdfile=dump-sas.ibd,
          phenfile=genehunter.ped,
          markers=10,
          missing=-99,
          asis=0,
          outdata=qtl.sample);
```

The code snippet in Example 3-4 will import data from the files `dump-sas.ibd` and `genehunter.ped` and save it into the temporary dataset `ibdphen`.

`qms2.sas`

The same macro calling rules as defined for ghimport.sas apply to `qms2.sas`. The mautosource command should be used to reference the `qms2.sas` macro. The code snippet in Example 3-5 shows an example of the `qms2.sas` macro being called with all of the required and optional arguments specified. Data will be read from the `sample` dataset in the `qtl` library, and the output will be saved to the working dataset `stats`. The influence statistics will be saved to the working dataset `infce`. Plots will be produced showing *t* by *pos* and output as color encapsulated PostScript files. The vertical axis will be labeled "t-score" and horizontal axis will be labeled "Position on Chromosome 6 (cM)".

Example 3-5. Calling `qms2.sas`

```
%macro qms2(data=qtl.sample,
            output=stats,
            infce=1,
            infout=infce,
            plot=1,
            vars=t*pos,
            format=pslepsfc,
            axis1=t-score,
            axis2=Position on chromosome 6 (cM),
                    tail=high);
```

Chapter 4. pre-sas.sh

The `pre-sas.sh` script is used to convert the IBD file from Genehunter 2 into a form that SAS Software can easily read. Genehunter 2 produces an odd file with a mix of tabs, spaces, and commas as delimiters.

Simply run the `pre-sas.sh` script on the IBD file: **./pre-sas.sh dump.ibd dump-sas.ibd**. The first argument, `dump.ibd` in this example, is the name of the file which Genehunter 2 placed the IBDs into. The second argument, `dump-sas.ibd` in this example, is the file into which `pre-sas.sh` should save the output into. If second argument is not specified, the output will automatically be written to a file named the same as the input file, except the extension `-sas` is added before the `.` in the original filename.

Example 4-1. Running `pre-sas.sh`.

```
$ ./pre-sas.sh dump.ibd
Converting "dump.ibd" to "dump-sas.ibd"...done.
$
```

Chapter 5. `ghimport.sas`

`ghimport.sas` is used to import an IBD file from Genehunter 2 and phenotypic data into a SAS Software dataset.

The macro itself

The best documentation for using `ghimport.sas` are the comments at the top of the macro.

```
  /*****************************************************************
%ghimport 0.11

Copyright (c) 2000 & 2001 Jeff Lessem, Institute for Behavioral Genetics,
    University of Colorado; and Stacey Cherny, Wellcome Trust Centre
    for Human Genetics, University of Oxford

See http://ibgwww.Colorado.EDU/~lessem/software/qms2.html#license for
the terms under which this software may be used.

Import a GENEHUNTER ibd file into a SAS Software dataset.

Usage:
       ibdfile= ibd file to analyze.  This is the file that is output
by "pre-sas.sh".  REQUIRED

       phenfile= The file used to read phenotype scores from.  REQUIRED

       markers= The number of loci at which you have markers.  This is
                needed so that when reading in the phenotype file the
                macro knows how many columns to skip before reading
                the phenotype values.  REQUIRED

       missing= The number used to represent missing phenotype
values.  This will be converted to the SAS Software
standard "." notation.  This will default to "." if
not specified.  OPTIONAL, DEFAULT="."

       asis=  Set to 1 to use the data exactly as read in, no
               selection on the affection status or double entry is
               performed.  This assumes that the first sib is the
               proband.  This may produce a datafile which is
               statistically inapropriate for some analysis.  0
               selects probands based on the affection status code and
               performs double entry when there are multiple probands
               in a family.  The default of reading the data in with
               selection and double entry is that the QMS2 macro was
               primarily written to perform the DF model, so it is
               setup by default to be correct for a DF analysis.
               OPTIONAL, DEFAULT=0
```

```
       outdata= The dataset to save the imported data into.  This
defaults to "data.ibdphen" (the temporary dataset
named "ibdphen").  OPTIONAL, DEFAULT="ibdphen"

For example to call this from within a SAS Software script the
following commands would work:

%ghimport(ibdfile=dump-sas.ibd,
   phenfile=pheno.dat,
           markers=10,
   missing=99.000,
           asis=0,
   outdata=mydata.wave1);

or

%ghimport(ibdfile=dump-sas.ibd,
   phenfile=pheno.dat,
           markers=10);

**********************************************************************/

%macro ghimport(
    ibdfile=macro-error,
    phenfile=macro-error,
    markers=macro-error,
    missing=.,
    asis=0,
    outdata=ibdphen);

data ibd; /* the dataset ibd will contain the ibd probabilities */
    infile "&ibdfile";

/* The columns are assigned to the following variables, where pos is
    the position on the chromosome, pedigree is the ID number of the
    family, indnum1 is the ID number of the first family member of the
    pairing and indnum2 is the ID number of the second member of the
    pairing, ibd0 ibd1 ibd2 represent the probability of the
    pairing having ibd 0, .5 or 1 at the pos(ition).  This routine
    could be modified to import ibd files from other programs than
    GENEHUNTER by changing the order of the variables read and adding
    or removing variables as necessary.
    */
   input pos pedigree indnum1 indnum2 prior0 prior1 prior2 ibd0 ibd1 ibd2 ;
run;

data phen; /* the dataset phen will contain the phenotype scores */
    infile "&phenfile";
    /* The phenotypes are read from a GENEHUNTER input file.  The
        variable "proband" designates the affection status with
        0=unknown 1=unaffected 2=affected.  For the purposes of this
        macro 0,1=not a proband; 2=proband */
    input pedigree indnum1 mothnum fathnum sex1 proband1
```

```
        x1-x%eval(&markers*2) phen1;
    drop x1-x%eval(&markers*2) mothnum fathnum;
    if phen1 eq &missing then phen1=.;
run;

/* sort the datasets in preparation to merge them */
proc sort data=ibd;
    by pedigree indnum1;
proc sort data=phen;
    by pedigree indnum1;

/* merge the phenotype of the first individual into the ibd file */
data ibdphen;
    merge ibd phen ;
    by pedigree indnum1;
run;

/* rename the variables in the pheno dataset to be set for individual
    2 */
proc datasets library=work;
    modify phen;
    rename sex1=sex2 phen1=phen2 indnum1=indnum2 proband1=proband2;
run;

/* sort ibdphen by indnum2 for next merge */

proc sort data=ibdphen;
    by pedigree indnum2;

data ibdphen;
    merge ibdphen phen;
    by pedigree indnum2;
/* drop pairings with missing phenotypes */
        if phen1 ne . and phen2 ne .;
/* drop the parents and lines with missing individuals */
/* no assumption of the parents id number, use the prior probability
    score */
        if indnum1 ne . and indnum2 ne . and
        (prior0 eq .25 and prior0 ne .) and
        (prior1 eq .5 and prior1 ne .) and
        (prior2 eq .25 and prior2 ne .);
    df=1; /* all forward pairings are an independent observation */
    dbl=0; /* is this the original or the double entered version? */
        drop prior0 prior1 prior2;
run;

/* Double enter the data */
data backward;
    set ibdphen;
    /* save into dummy variables */
    indnumx=indnum1;
    sexx=sex1;
    phenx=phen1;
```

```
    probandx=proband1;
    /* move 1 to 2 */
    indnum1=indnum2;
    sex1=sex2;
    phen1=phen2;
    proband1=proband2;
    /* move dummies back to 1 */
    indnum2=indnumx;
    sex2=sexx;
    phen2=phenx;
    proband2=probandx;
    /* drop dummy variables */
    drop indnumx sexx phenx probandx;
    /* double entered cases do not add new information */
    df=0;
    /* set double entered to true */
    dbl=1;
run;

data ibdphen;
/* Combine the single and the double entered datasets */
    set ibdphen backward;
/* compute pihat from the ibd probabilities */
    pihat = .5 * ibd1 + ibd2 ;
run;

proc sort data=ibdphen;
    by pos;

/* This part handles selection */
%if &asis=0 %then %do; /* do selection */
        data &outdata;
           set ibdphen;
/* This removes non-proband predictor phenotypes */
              if proband1 eq 2;
/* If the first instance of a pairing has been dropped then add back
    a degree of freedom */
                  if proband1 eq 2 and (proband2 eq 0 or proband2 eq
                      1) and df eq 0 then df=1;
run;
%end; /* selection */
    %else %do; /* No selection */
        data &outdata;
            set ibdphen;
            if dbl ne 1;
run;
%end; /* No selection */

%mend ghimport; /* ghimport */
```

A sample SAS Software script with `ghimport.sas`

This example calls the `ghimport` macro. The macro is told to read IBD information from the file `dump-sas.ibd`, which would have been generated using `pre-sas.sh`. The phenotype information is in the file `genehunter.ped`. The data used has 10 markers, and codes missing phenotypes as `-99`. The imported data will be saved into the library `mydata` and the dataset `ibdphen`.

Example 5-1. `myimporter.sas`

```
libname mydata '$HOME/sas/qms2';
options mautosource sasautos=('$HOME/sas/qms2');        ❶
options mprint;                                         ❷

%ghimport(ibdfile=dump-sas.ibd,                         ❸
          phenfile=genehunter.ped,
          markers=10,
          missing=-99,
                  outdate=mydata.ibdphen);
```

❶ The option *mautosource* tells the SAS Software to look in the specified directory for any macros which are referenced in the script. In this example, the directory `sas/qms2/` under the user's home directory will be searched.

❷ The option *mprint* will print out the SAS Software statements which are created from the macro. This option isn't necessary for normal usage, but it makes debugging much easier.

❸ A SAS Software macro is called using the convention `%macroname(macroargument1,macroargument2);`. For readability, the macro arguments can be split across lines.

After running this SAS Software script a permanent dataset will be created. That dataset can then be used with the `qms2` macro or other SAS Software scripts.

Chapter 6. qms2.sas

This chapter explains how to use the qms2.sas macro.

The macro itself

The best documentation for using qms2.sas are the comments at the top of the macro.

```
    /*****************************************************************
%qms2 0.11

Copyright (c) 2000 & 2001 Jeff Lessem, Institute for Behavioral Genetics,
    University of Colorado; and Stacey Cherny, Wellcome Trust Centre
    for Human Genetics, University of Oxford

See http://ibgwww.Colorado.EDU/~lessem/software/qms2.html#license for
the terms under which this software may be used.

Perform a DeFries-Fulker or Haseman-Elston regression on the data
imported by %ghimport, or a compatible script.

Usage:

    data= The dataset to read records for analysis from.  The dataset
          produced by %ghimport meets these requirements, but any user
          supplied dataset in the same arrangement would also work.
          The dataset should have the following variables:
            POS= The position on the chromosome in cM
            Pedigree= A unique number used to identify a family
            Indnum1= A unique number used to identify the first
                   sibling from a family in a pairing
            Indnum2= A unique number used to identify the second
                   sibling from a family in a pairing
            IBD0= The probability of indnum1 and indnum2 being IBD 0
                  at POS
            IBD1= The probability of indnum1 and indnum2 being IBD 1
                  at POS
            IBD2= The probability of indnum1 and indnum2 being IBD 2
                  at POS
            SEX1= Sex of indnum1 (typically 1 or 2, but this macro makes
                no assumption about the meaning of the values)
            SEX2= Sex of indnum2
            Proband1= Proband (or affection) status of indnum1, 1 is
                    not affected and 2 is affected (proband)
            Proband2= Proband (or affection) status of indnum2
            Phen1= The phenotypic score of indnum1
            Phen2= The phenotypic score of indnum1
            DF= The number of unique observations provided by this
              record (either 1 if this is the first appearance of this
              sib pairing, or 0 if this is a subsequent, double entered,
```

```
          appearance of the pairing)
        DBL= Designates the current record as being the first
           pairing of the two individuals (0) or the second,
           double-entered pairing (1)
        Pihat= .5 * ibd1 + ibd2
      OPTIONAL, DEFAULT=ibdphen

output= The dataset to save the output statistics to.  This dataset
        contains the following variables:
        POS= position on the chromosome in cM
        N= The total number of records used in the analyses
          (includes double-entered pairings)
        Real_N= N after correcting for double entry
        ModelDF= Model degrees of freedom, which is the number of
          independent variables.
        Intercep= Mean of the phenotype for ibd 0 (DF model)
          Mean of sibpair difference for ibd 0 (HE model)
        Phen1= The regression beta for the phenotype (DF model)
        b_Aug= The regression beta for the interaction term (DF
          augmented model B5)
        Shapiro= The Shapiro-Wilk W statistic, a measure of
          normality, for the residuals from the regression model
        ShapiroP= The probability that the residuals depart from
          normality
        T= t score, adjusted for double entry
        P= p value, adjusted for double entry
        RMSE= root mean squared error
        B_Pihat= Beta coefficient of pihat (DF model B2 or the HE
            regression coefficient)
        Model= Which model used, either Haseman-Elston,
          DeFries-Fulker, or DF Augmented
        OPTIONAL, DEFAULT=stats

infce= Output statistics detailing how much influence each
       family has on the regression, 1=yes, 0=no
       OPTIONAL, DEFAULT=0 (no)

infout= The dataset to save the influence statistics to.  A
    complete explanation of the influence statistics can be found
    in the PROC REG documentation for SAS Software.  The following
    variables are saved:  OPTIONAL, DEFAULT=infce
        POS= position on the chromosome in cM
        PEDIGREE= Pedigree number
        INDNUM1= ID number of the proband
        INDNUM2= ID number of the co-sib
        IBD0= Probability that the pair shares no alleles IBD at
            the locus
        IBD1= Probability that the pair shares one allele IBD at
            the locus
        IBD2= Probability that the pair shares both alleles IBD at
            the locus
        SEX1= Sex of the proband
        PROBAND1= 2 indicates individual number 1 is a proband,
```

```
                this should always be the case in a selected sample
            PHEN1= Phenotypic score of the proband
            SEX2= Sex of the co-sib
            PROBAND2= 2 indicates that the co-sib is also a proband, a
                1 indicates the co-sib is not a proband
            PHEN2= Phenotypic score of the co-sib
            DF= The number of unique statistics provided by this entry
                (1 or 0)
            DBL= A 1 indicates this is a double entered pairing
            PIHAT= The pihat of the pairing
            AUGMENT= The interaction term of the pair in the DF
                augmented model
            H= Belsley, Kuh, and Welsch (1980) hat matrix statistic
            RSTUDENT= The studentized residual of the observation.
                 Values with an absolute value greater than 2 may
                 warrant investigation.
            DFFITS= A scaled measure of the change in the predicted
                 value for the observation.  Values of DFFITS over 2
                 may warrant investigation.
            MODEL= Which model the results came from

    plot= Produce a plot of the results, 1=yes, 0=no
          OPTIONAL, DEFAULT=0 (no)

    vars= Variables to plot, in the form "Y-Axis*X-Axis".  Useful
         examples are "t*pos" or "p*pos"
         OPTIONAL, DEFAULT="t*pos" (t-score by position)

    format= Output format of the plot.  OPTIONAL, DEFAULT="pslepsfc"
          (full color encapsulated postscript) see the SAS Software
          manual for more options

    axis1= The title for axis1, the vertical axis
          OPTIONAL, DEFAULT=t Score

    axis2= The title for axis2, the horizontal axis
          OPTIONAL, DEFAULT=Position on chromosome (cM)

    tail= Set which tail of the distribution is of interest for the
          one-tailed DF basic test.  The p-values for the
          uninteresting tail are all set to 1.  The high tail is
          interesting if the probands were selected from the high end
          of the distribution, and vice-versa for "low".  Any
          transformations you have performed on the data may effect
          which tail is interesting.
          OPTIONAL, DEFAULT=do nothing, use "high" or "low"
          to set the tail of interest.  Any value other than "high" or
          "low" leaves the p-values unchanged.

The plot files are named "<model>.eps" with the name of the model in
    the filename.  The graphs are titled "Plot by <model> method".  At
    this time the only way to change the filenames or graph titles is
    to edit the macro at the lines "filename t..." and "title1
```

```
    ..."  In both of these places, the term "&model" is replaced by
    the name of the model, so it is strongly suggested that &model be
    preserved in the filename so that the plots from a single run
    do not overwrite each other.

**********************************************************************/
%macro qms2(data=ibdphen,
    output=stats,
    infce=0,
    infout=infce,
    plot=0,
    vars=t*pos,
    format=pslepsfc,
    axis1=t-score,
    axis2=Position on chromosome (cM),
    tail=nothing,
    spr=compute);


/* create the dataset to contain the results if it does not already
    exist */
%if %sysfunc(exist(&output))=0 %then %do;
    data &output;
    run;
%end; /* existence check */

/* create the dataset to contain the influence statistics if it does
    not already exist */
%if %sysfunc(exist(&infout))=0 %then %do;
    data &infout;
    run;
%end; /* existence check */

/* This macro handles the data produced by the regressions */
/* It is inefficient to define one macro within another, but it works */
%macro output(model);

proc univariate normal data=resid noprint;
    output out=compr normal=shapiro probn=shapirop;
    var compr;
    by pos;

/* get the proper degrees of freedom */
proc means data=&data noprint;
    var df;
    by pos;
    output out=df mean=m n=n;
data modeldf;
    set results (keep=_in_);
    if _in_ ne .;
    dummy=0;
run;
data df;
```

```
    set df;
    /* real_n is the number of unique statistics */
    real_n=m*n;
    /* Hasemant-Elston is never double entered */
    %if "&model"="Haseman-Elston" | "&model"="Sham-Purcell" %then n=real_n;;
    dummy=0;
    drop _type_ _freq_ m;
run;
/* the model degrees of freedom is the number of independent variables */
data df;
    merge df modeldf (rename=(_IN_=modeldf));
    by dummy;
    drop dummy;
run;

/* compute fit statistics based on the corrected degrees of freedom */
data fits;
    merge df results compr;
    by pos;
    if _type_="T" then do;
    %if "&model"="DF_Augmented" %then %do;
        t=augment*sqrt(real_n/n);
        %end;
    %else %do;
        t=pihat*sqrt(real_n/n);
        %end;
    p=(1-probf(t**2,1,real_n-modeldf-1))/2;
    t2=t**2;
    /* set p to 1 for the wrong tail */
    %if "&model"="DeFries-Fulker" %then %do;
        %if "&tail"="high" %then %do;
            if t lt 0 then p=1;
            %end;
        %else %if "&tail"="low" %then %do;
            if t gt 0 then p=1;
            %end;
        %end;
    %else %if "&model"="Haseman-Elston" %then %do;
        if t gt 0 then p=1;
        %end;
    %else %do; /* New-He DF_Augmented and Sham-Purcell */
        if t lt 0 then p=1;
        %end;
    end;
    if _type_="T";
    drop _model_ _depvar_ _rmse_ pihat comp _in_ _p_ _edf_ _rsq_ phen2 depvar
        %if "&model"="DF_Augmented" %then augment;
run;

/* get a datafile with the fit statistics and betas */
data parms;
    set results (keep=_type_ pos _rmse_ pihat
        %if "&model"="DF_Augmented" %then augment;
```

```
    );
    if _type_="PARMS";
run;
data tmprslt;
    merge fits (drop=_type_)
        parms (rename=( _rmse_=rmse pihat=b_pihat
        %if "&model"="DF_Augmented" %then augment=b_aug;
    ) drop=_type_ );
    model="&model";
run;

/* merge the results with the output dataset */
data &output;
    set &output tmprslt;
    /* drop the blank line created in the empty dataset */
    if N ne .;
    run;

/* handle the influence statistics, if requested */
%if &infce=1 %then %do;
data tmpinf;
    set tmpinf;
    model="&model";
run;
data &infout;
    set &infout tmpinf;
    /* drop the blank line created in the empty dataset */
    if pos ne .;
run;
%end; /* influence statistics */

%if &plot=1 %then %do;
/* graph the results */

/* filename to save graph under */
filename graph "&model..eps";

/* Graphing options */
goptions
    /* reset all options to default */
    reset=all
    /* select the font (helvetica) */
    ftext=hwpsl009
    /* full color encapsilated postscript output */
    device=&format
    /* replace the output file if it already exists */
    gsfmode=replace
    /* make the graph print in landscape(!) */
    rotate=portrait
    /* select the size of the graph.  It is eps so it will scale */
    vsize=3.5in hsize=6.0in
     /* fileref to use for graph */
    gsfname=graph;
```

```
/* set the graph main title */
title1 "Plot by &model method";

/* add axis labels */
axis1 label=(
    /* make the text vertical */
    angle=90
    /* center justify */
    justify=c
    "&axis1");
/* define another axis */
axis2 label=(justify=c "&axis2");

proc gplot data=tmprslt;
    symbol
    /* method of connecting points, use join for straight lines */
    interpol=join
    /* make the line a bit thicker */
    width=3;
    plot &vars /
        /* add vertical and horizontal reference lines */
        grid
        /* use the axis statements set earlier */
        vaxis=axis1 haxis=axis2;
%end; /* end plotting */

%mend output;

/* Haseman-Elston Regression */
data he;
    set &data;
    /* drop double entered records because twin ordering is irrelevant
        for HE */
    if df=1;
    /* create the dependent variable, squared sib difference */
    comp=(phen1 - phen2)**2;
run;

proc reg data=he tableout edf outest=results noprint;
    model comp = pihat %if &infce=1 %then / influence ;;
    output out=resid r=compr;
    %if &infce=1 %then
    output out=tmpinf dffits=dffits h=h rstudent=rstudent;;
    by pos;

%output(Haseman-Elston);

/* New Haseman-Elston Regression */
/* Get the means of the phenotypes */
proc means data=&data noprint;
    var phen1 phen2;
    output out=means mean=mean1 mean2;
```

```
data means;
    set means;
    dummy=1;
run;
data nhe;
    set &data;
    dummy=1;
run;

data nhe;
    merge nhe means(drop=_type_ _freq_);
    by dummy;
    /* create the dependent variable */
    comp=(phen1 - mean1)*(phen2 - mean2);
    drop=dummy;
run;

proc reg data=nhe tableout edf outest=results noprint;
    model comp = pihat %if &infce=1 %then / influence ;;
    output out=resid r=compr;
    %if &infce=1 %then
    output out=tmpinf dffits=dffits h=h rstudent=rstudent;;
    by pos;

%output(New-HE);

/* DeFries-Fulker Basic Regression */
proc reg data=&data tableout edf outest=results noprint;
    model phen2 = phen1 pihat %if &infce=1 %then / influence ;;
    output out=resid r=compr;
    %if &infce=1 %then
    output out=tmpinf dffits=dffits h=h rstudent=rstudent;;
    by pos;
%output(DeFries-Fulker);

/* DeFries-Fulker Augmented Regression */
data dfaug;
    set &data;
    augment=phen1*pihat;
    run;

proc reg data=dfaug tableout edf outest=results noprint;
    model phen2 = phen1 pihat augment %if &infce=1 %then / influence ;;
    output out=resid r=compr;
    %if &infce=1 %then
    output out=tmpinf dffits=dffits h=h rstudent=rstudent;;
    by pos;
%output(DF_Augmented);

/* Sham-Purcell Regression Approach */
/* This requires a bi-variate normal with mean 0 and SD 1 in the original population */
data sham;
    set &data;
```

```
    /* Is this method valid for double entered data? */
    if df=1;
    dummy=1;

/* compute the sib-pair correlation if it isn't provided */
%if "&spr"="compute" %then %do;
/* create the correlation used in the dependent variable */
    proc corr data=sham nosimple noprint outp=shamr;
        var phen1 phen2;

/* get r into a dataset with a dummy variable for merging */
    data shamr;
        set shamr;
        if _type_="CORR" and _name_="phen1";
        r=phen2;
        dummy=1;
        keep r dummy;
%end;
%else %do;
    data shamr;
        r=&spr;
        dummy=1;
%end;

/* setup the dependent variable */
data sham;
    merge sham shamr;
    by dummy;
    depvar=((phen1+phen2)**2/(1+r)**2) - ((phen1-phen2)**2/(1-r)**2)
        + ( (4*r)/(1-r**2));
    pihat=pihat-.5;
    drop dummy r;

proc reg data=sham tableout edf outest=results noprint;
    model depvar = pihat / noint %if &infce=1 %then influence ;;
    output out=resid r=compr;
    %if &infce=1 %then
    output out=tmpinf dffits=dffits h=h rstudent=rstudent;;
    by pos;
%output(Sham-Purcell);

/* Reset the title to be blank */
title 'QMS2 Output';
footnote1 "Lessem, J. M., and Cherny, S. S. (2001) DeFries-Fulker";
footnote2 "multiple regression analysis of sibship data: a SAS macro.";
footnote3 "Bioinformatics, 17, 371-372.";

%mend qms2;
```

A sample SAS Software script with `qms2.sas`

This example calls the `qms2` macro. The macro is told to read IBD information from the file `dump-sas.ibd`, which would have been generated using `pre-sas.sh`. The phenotype information is in the file `genehunter.ped`. The data used has 10 markers, and codes missing phenotypes as `-99`. The imported data will be saved into the library `mydata` and the dataset `ibdphen`.

Example 6-1. `myanalyzer.sas`

```
libname mydata '$HOME/sas/qms2';
options mautosource sasautos=('$HOME/sas/qms2');        ❶
options mprint;                                         ❷

%qms2(data=mydata.ibdphen,                              ❸
      output=mydata.qtl,
      infce=1,
      infout=mydata.inf);

 proc print data=mydata.qtl;                            ❹
 proc print data=mydata.inf;
```

❶ The option *mautosource* tells the SAS Software to look in the specified directory for any macros which are referenced in the script. In this example, the directory `sas/qms2/` under the user's home directory will be searched.

❶ The option *mprint* will print out the SAS Software statements which are created from the macro. This option isn't necessary for normal usage, but it makes debugging much easier.

❸ A SAS Software macro is called using the convention `%macroname(macroargument1,macroargument2);`. For readability, the macro arguments can be split across lines.

❹ This line will print out the results of the QTL analyses. The `qms2` macro does not print anything out on its own. Under normal SAS Software conventions, the output will be saved in, for this example, the file `myanalyzer.lst`.

After running this SAS Software script two permanent datasets will be created. That dataset can then be used with the `qms2` macro or other SAS Software scripts.