Chapter 22

Gene Segregation and Linkage Analysis

Jinsheng Liu
Todd C. Wehner
Sandra B. Donaghy

Purpose

To calculate single-gene goodness-of-fit testing to analyze gene linkage relationships, including calculations of chi-square, probability value, and two-locus-combined phases, for all gene pairs in segregation for the F_2, BC_{1P1}, and BC_{1P2} generations. Recombination frequency and standard error are calculated according to the linkage phase.

Genetic Analysis

Linkage is estimated using the chi-square method, a widely used standard for genetic data analysis (although it may produce inaccurate results in some cases). Recombination frequency (RF) and standard error (SE) are calculated according to phase (coupling or repulsion), using the following formulas (Sinnott and Dunn, 1939; Weir, 1994).

Definitions

F_2 (repulsion):

$$RF = p = \sqrt{\frac{-(bc+ad)+\sqrt{(bc+ad)^2+ad(bc-ad)}}{(bc-ad)}}$$

F_2 (coupling):

$$RF = 1 - p$$
$$SE = \sqrt{(1-p^2)(2+p^2)/2n(1+2p^2)}$$

BC_1 (only coupling accepted):

$$RF = (b+c)/n$$
$$SE = \sqrt{RF(1-RF)/n}$$

where a ($A\_B\_$), b (A_bb), c ($aaB\_$) and d ($aabb$) are genotype segregation ratios in F_2 or BC_1.

Originators

Sinnott, E.W. and Dunn, L.C. (1939). *Principles of Genetics*. McGraw-Hill, New York.
Weir, B.S. (1994). *Genetic Data Analysis: Methods for Discrete Population Data*. Sinauer, Sunderland, MA.

Software Available

Files can be found on the World Wide Web at <http://cuke.hort.ncsu.edu/cucurbit/Wehner/software.html>. Or, send a 3.5" floppy disk to Todd C. Wehner, Department of Horticultural Science, North Carolina State University, Raleigh, NC 27695-7609.

Publication

Liu, J.S., Wehner, T.C., and Donaghy, S.B. (1997). SASGENE: A SAS computer program for genetic analysis of gene segregation and linkage. *Journal of Heredity* 88: 253-254.

Some References Using the Software

Wehner, T.C., Liu, J.S., Staub, J.E., and Fazio, G. (2003). Segregation and linkage of 14 loci in cucumber. *Journal of American Society of Horticulture Science.*

Contact

Dr. Todd Wehner, Department of Horticultural Science, North Carolina State University, Raleigh, NC 27695-7609, USA. E-mail: <todd_wehner@ncsu.edu>; Web site: <http://cuke.hort.ncsu.edu>.

Revisions That Have Been Made

SASGene1.0 and 1.1 had an error in the formula for calculation of SE for RF in coupling. F_2 (coupling):

$$RF = 1 - p$$
$$SE = \sqrt{(1-p^2)(1+p^2)/2n(1+2p^2)}$$

SASGene1.2 has been corrected F_2 (coupling):

$$RF = 1 - p$$
$$SE = \sqrt{(1-p^2)(2+p^2)/2n(1+2p^2)}$$

EXAMPLE

Data to be analyzed:

Plot	Rep	Fam	Gen	Plnt	Bi	Rc	Dv	Sp	Ll	Df	F	B	D	U	Tu
1	1	28	1	1	B	N	N	N	L	N	M	W	D	N	W
2	1	28	1	2	B	N	N	N	L	N	M	W	D	N	W
3	1	28	1	3	B	N	N	N	L	N	M	W	D	N	W
4	1	28	1	4	B	N	N	N	L	N	M	W	D	N	W
5	1	28	1	5	B	N	N	N	L	N	M	W	D	N	W
6	1	28	2	1	N	N	N	N	N	D	G	W	D	U	S
7	1	28	2	2	N	N	N	N	N	D	G	W	S	U	S
8	1	28	2	3	N	N	N	N	N	D	G	W	S	U	S
9	1	28	2	4	N	N	N	N	N	D	G	W	S	U	S
10	1	28	2	5	N	N	N	N	N	D	G	W	S	U	S
11	1	28	3	1	B	N	N	N	N	N	G	W	D	N	W
12	1	28	3	2	B	N	N	N	N	N	G	W	D	N	W
13	1	28	3	3	B	N	N	N	N	N	G	W	D	N	W
14	1	28	3	4	B	N	N	N	N	N	G	W	D	N	W
15	1	28	3	5	B	N	N	N	N	N	M	W	D	N	W
16	1	28	3	6	B	N	N	N	N	N	G	W	D	N	W
17	1	28	4	1	B	.	N	N	N	D	G	W	D	U	S
18	1	28	4	2	N	.	N	N	L	D	G	W	.	U	S
19	1	28	4	3	B	N	N	N	N	D	0	W	.	U	S
20	1	28	4	4	B	N	N	N	N	N	G	W	D	N	W
21	1	28	4	5	B	N	N	N	N	N	M	W	D	N	W
22	1	28	4	6	B	N	N	N	N	N	G	W	D	N	W
23	1	28	4	7	B	N	N	N	L	N	G	W	D	N	W
24	1	28	4	8	B	N	N	N	N	N	G	W	D	U	W
25	1	28	4	9	B	N	N	N	N	D	G	W	D	N	W
26	1	28	4	10	N	N	N	N	N	D	G	W	D	U	S
27	1	28	4	11	N	N	N	N	N	N	G	W	D	N	W

28	1	28	4	12	B	N	N	N	N	N	G	W	D	N	W
29	1	28	4	13	B	N	N	N	L	N	G	W	D	.	W
30	1	28	4	14	N	3	N	N	N	N	G	W	D	N	W
31	1	28	4	15	B	N	N	N	N	N	G	W	D	N	W
32	1	28	4	16	N	N	N	N	N	N	G	W	D	N	W
33	1	28	4	17	B	N	.	.	.	.	.	.	.	.	.
34	1	28	4	18	B	N	.	.	.	.	.	.	.	.	.
35	1	28	5	1	B	N	N	N	L	N	G	W	D	N	W
36	1	28	5	2	B	N	N	N	N	N	G	W	D	N	W
37	1	28	5	3	B	N	N	N	L	N	M	W	D	N	W
38	1	28	5	4	B	N	N	N	L	N	M	W	D	N	W
39	1	28	5	5	B	N	N	N	N	N	G	W	D	N	W
40	1	28	5	6	B	N	N	N	N	N	G	W	D	N	W
41	1	28	5	7	B	N	N	N	L	N	M	W	D	N	W
42	1	28	5	8	B	N	N	N	N	N	G	W	D	N	W
43	1	28	5	9	B	N	N	N	L	N	G	W	D	N	W
44	1	28	6	1	B	N	N	N	N	N	G	W	D	N	S
45	1	28	6	2	B	N	N	N	N	D	G	W	D	U	S
46	1	28	6	3	N	N	N	N	N	D	G	W	D	U	S
47	1	28	6	4	B	N	N	N	N	D	G	W	D	U	S
48	1	28	6	5	N	N	N	N	N	D	G	W	D	U	S
49	1	28	6	6	N	N	N	N	N	D	G	W	D	U	S
50	1	28	6	7	N	N	N	N	N	D	G	W	D	U	S
51	1	28	6	8	N	N	N	N	N	N	G	W	D	N	W
52	1	28	6	9	N	N	N	N	N	N	G	W	D	N	W
53	1	28	1	1	B	N	N	N	L	N	M	W	D	N	W
54	1	28	1	2	B	N	N	N	L	N	M	W	D	N	W
55	1	28	1	3	B	N	N	N	L	N	M	W	D	N	W
56	1	28	1	4	B	N	N	N	L	N	M	W	D	N	W
57	1	28	1	5	B	N	N	N	L	N	M	W	D	N	W
58	1	28	2	1	N	N	N	N	N	D	G	W	S	U	S
59	1	28	2	2	N	N	N	N	N	D	G	W	S	U	S
60	1	28	2	3	N	N	N	N	N	D	G	W	S	U	S
61	1	28	2	4	N	N	N	N	N	D	G	W	S	U	S
62	1	28	2	5	N	N	N	N	N	D	G	W	S	U	S
63	1	28	3	1	B	N	N	N	N	N	G	W	D	N	W
64	1	28	3	2	B	N	N	N	N	N	G	W	D	N	W
65	1	28	3	3	B	N	N	N	N	N	G	W	D	N	W
66	1	28	3	4	B	N	N	N	N	N	G	W	D	N	W
67	1	28	3	5	B	N	N	N	N	N	G	W	D	N	W
68	1	28	3	6	B	N	N	N	N	N	G	W	D	N	W
.															
.															
.															
469	5	30	1	1	N	R	N	S	N	N	M	B	D	U	W
470	5	30	1	2	N	R	N	S	N	N	M	B	D	N	W
471	5	30	1	3	N	R	N	S	N	N	M	B	D	N	W
472	5	30	1	4	N	R	N	S	N	N	G	B	D	N	W
473	5	30	1	5	N	R	N	S	N	N	G	B	D	U	W
474	5	30	2	1	N	N	N	N	N	N	G	W	S	U	S
475	5	30	2	2	N	N	N	N	N	D	G	W	S	U	W
476	5	30	2	3	N	N	N	N	N	D	G	W	S	U	S
477	5	30	2	4	N	N	N	N	N	N	G	W	S	U	S
478	5	30	2	5	N	N	N	N	N	D	G	W	S	U	S
479	5	30	3	1	B	N	N	N	N	D	M	B	D	N	W
480	5	30	3	2	B	N	N	N	N	N	G	B	D	N	W
481	5	30	3	3	B	N	N	N	N	D	G	B	D	N	W

```
482  5  30  3  4   B N N N N D G B D N W
483  5  30  3  5   B N N N N N G B D N W
484  5  30  3  6   B N N N N D G B D N W
485  5  30  4  1   B N N N N N G B D N W
486  5  30  4  2   B N N N N N G W D U S
487  5  30  4  3   N R N S N N G W D U W
488  5  30  4  4   B N N N N D G B S U W
489  5  30  4  5   B N N N N D G B S U W
490  5  30  4  6   B N N N N N G B D N W
491  5  30  4  7   B N N N N N G W D U S
492  5  30  4  8   B N N N N D G B D N W
493  5  30  4  9   N N N N N N G W D N S
494  5  30  4  10  N N N N N D G B D N S
495  5  30  4  11  B N N N N N G B D N W
496  5  30  4  12  B N N N N D G B D U S
497  5  30  4  13  B N N N N D G W S U W
498  5  30  4  14  B N N N N D G B D N W
499  5  30  4  15  N N N N N N G B D N W
500  5  30  4  16  N N N N N N G W D U S
501  5  30  4  17  N N N N N N G B D N W
502  5  30  4  18  N N N N N N G B D N W
503  5  30  5  1   B N N N N D G B D N W
504  5  30  5  2   N R N N S N G B D U W
505  5  30  5  3   B N N N N D N B D U W
506  5  30  5  4   N R N S N N M . . . .
507  5  30  5  5   B N N N N N G B D N W
508  5  30  5  6   B N N N N D G B D N W
509  5  30  5  7   B N N N N N G B D N W
510  5  30  5  8   B R N N S N N M B D N W
511  5  30  5  9   N R N S S N N M . . . .
512  5  30  6  1   B N N N N N D G B D N W
513  5  30  6  2   N N N N N N D G B D U S
514  5  30  6  3   B N N N N N D G B D U W
515  5  30  6  4   B N N N N N D G W D U S
516  5  30  6  5   N N N N N N D G B D N W
517  5  30  6  6   B N N N N N D G W D N W
518  5  30  6  7   N N N N N N D G W D U W
519  5  30  6  8   N N N N N N D M W D U S
520  5  30  6  9   B N N N N D G W D N S
```

.
.
.

SAS Program (Five Files)

File 1: readme.txt

```
SASGENE 1.1
Program for Analysis of
Gene Segregation and Linkage
November 5, 1997
```

Instructions for Running SASGENE Macros

The SASGENE program for gene segregation and linkage analysis is written in SAS macro language. There are four SAS files. Three are macro files and one is an example. The first macro, SGENE, is for single-gene goodness-of-fit tests. The second macro, LINKAGE, is for analysis of gene linkage relationships. The third macro, CONVERT, is optional and converts gene values to "D" for dominant and "R" for recessive. STARTUP.SAS illustrates how to use the macros. The STARTUP.SAS file can easily be modified for other experiments of interest to the user.

The macros are written for version six and later versions of SAS. The amount of disk space required increases as the number of genes for the linkage analysis increases.

To use the macros, the user must create an input data file that will record data for the following fields: plot number, replication number, plant number, family number, generation number, and gene (or trait) names. Note that plot number, replication number, and plant number are used only for collecting data and are not used by the program for computing statistics. The user may specify any value for the family variable, but the macro requires values of 1, 2, 3, 4, 5, or 6 for the GNR (generation) variable (1 for P1, 2 for P2, 3 for F1, 4 for F2, 5 for BC1P1, 6 for BC1P2). Valid SAS variable names are used for the gene names. The genes (or traits) are variables (columns) and their values are observations (rows). Family and generation are identification variables. In the data file, the values of P1, P2, and F1 should not be omitted or the results may be incorrect.

The SGENE and LINKAGE macros require gene values to be coded as "D" for dominant, "R" for recessive, and "." or blank for a missing value. An optional macro, CONVERT, converts the original gene values to "D," "R," or missing. For each gene and family, the most frequent value for F1 is the dominant gene. Any other nonmissing values are treated as recessive, and any missing values are counted as missing.

An example of a SAS data set follows:

```
data orig;
   input PLOT REP FAMILY GNR BI $ RC $ DV $ SP $
   LL $ DF $ F $ B $ D $ U $ TU $;
   cards;
   1   1  20   1   N  R  N  S  N  N  M  B  D  N  W
   2   1  20   1   N  R  N  S  N  N  M  B  D  N  W
   3   1  20   1   N  R  N  S  N  N  M  B  D  N  W
   4   1  20   1   N  R  N  S  N  N  M  B  D  N  W
   .
   .
   run;
```

Either the macro code or a %INCLUDE (also known as %INC) statement is needed to define the macro to the SAS system. The user may include the macro into the program editor or use a %INC statement, such as %inc 'sgene.sas'. The %INC statement specifies the physical name of the external file where the macro is stored. The physical name is the name by which the host system recognizes the file. Depending on the host system and location of the file, the entire file name may need to be specified.

Examples:

```
%inc 'c:\mysas\sgene.sas';
%inc '~/sasmacro/sgene.sas';
```

The file, SGENE.SAS, contains the SAS macro, SGENE. File names, such as SGENE.SAS, usually carry the *sas* extension if the file is a SAS program or a SAS macro.

Once the macro is defined to SAS, the macro can be invoked. To invoke the macro, specify the %, the macro name (either SGENE, LINKAGE, or CONVERT), and the required parameters in parenthesis.

The SGENE macro has three parameters:

DS—name of the SAS data set to analyze
GENES—gene names from the SAS data set
P1—critical value for about half of the frequency of one parent to determine the expected segregation ratio (1:1 or 1:0) in BC1 generation

Example:

```
%sgene (ds=new,
        genes=BI RC DV SP LL DF F  B D U TU,
        p1=9);
```

The linkage macro has four parameters:

DS—name of the SAS data set to analyze
GENES—gene names from the SAS data set
P1, P2—critical value for about half of the frequency of the parents to determine if the phase is coupling or repulsion

Example:

```
%linkage (ds=new,
          genes=BI RC DV SP LL DF F  B D U TU,
          p1=9,
          p2=9);
```

The convert macro has three parameters:

DS—name of the SAS data set to convert
GENES—list of the desired gene names from the SAS data set
DSOUT—name of the SAS data set after conversion

Example

```
%convert (ds=orig,
          genes=BI RC DV SP LL DF F  B D U TU,
          dsout=new);
```

Several additional files are stored in the same location as the introduction:

STARTUP.SAS—example that illustrates how to use the macros
ORIG.DAT—sample data for the startup.sas file
CONVERT.SAS—file that contains the SAS macro convert
SGENE.SAS—file that contains the SAS macro sgene
LINKAGE.SAS—file that contains the SAS macro linkage

File 2: STARTUP.SAS

```
*************************************************************************
*                                                                       *
* SASGENE 1.1                                                           *
* Program for Analysis of                                               *
* Gene Segregation and Linkage                                          *
* November 5, 1997                                                      *
*                                                                       *
* Example of Invoking SASGENE macros                                    *
*                                                                       *
*************************************************************************;

*************************************************************************
*                                                                       *
* Specify file names and include macros.                                *
*                                                                       *
* 1.  Specify the name of the file where the data are stored.           *
*     The name is enclosed in single quotes.                            *
```

```
*     example: filename in 'orig.dat' ;                                    *
*                                                                          *
* 2.  Include the macros with the %INCLUDE (%inc) statement.               *
*     Specify the physical name of the external file where the macro       *
*     is stored.  The physical name is enclosed in single quotes.          *
*     example: %inc 'convert.sas';                                         *
*              %inc 'sgene.sas';                                           *
*              %inc 'linkage.sas';                                         *
*                                                                          *
*     Summary:                                                             *
*     The user only needs to change the information inside the             *
*     quotes on the FILENAME and %INCLUDE statements below.                *
*     The information inside the quotes specifies the name of the          *
*     external file where the data or macros are stored.  It may be        *
*     necessary to specify the entire file name inside the quotes.         *
*     example: %inc 'c:\sasmacro\convert.sas';                             *
***************************************************************************;

filename in 'example.dat';  /* name and location of data file     */
%inc 'convert.sas';    /* name and location of SAS macro CONVERT  */
%inc 'sgene.sas';      /* name and location of SAS macro SGENE    */
%inc 'linkage.sas';    /* name and location of SAS macro LINKAGE  */

***************************************************************************
*   include any desired titles and options                                *
***************************************************************************;
title 'Cucumber Gene Linkage Example';
options nodate pageno=1;
options linesize=80 pagesize=500;

***************************************************************************
*   Create SAS dataset                                                    *
*     The user will need to modify the INPUT statement to specify         *
*     the gene names from their experiment.  If list input is used,       *
*     then missing values should be coded with a "."                      *
*                                                                         *
*     Macros are expecting the following variable names:                  *
*     family = family code                                                *
*     gnr    = generation code                                            *
*                                                                         *
*     Macros are expecting the following values for GNR variable:         *
*      1 for P1                                                           *
*      2 for P2                                                           *
*      3 for F1                                                           *
*      4 for F2                                                           *
*      5 for BC1P1                                                        *
*      6 for BC1P2                                                        *
*                                                                         *
*     P1, P2 and F1 generations must be included                          *
*     for program to run (1 plant each is sufficient)                     *
***************************************************************************;
data original;
   infile in missover pad; /* MISSOVER & PAD are options on INFILE */
   input plot rep family gnr plnt bi $ rc $ dv $ sp $ ll $ df $
         f $ b $ d $ u $ tu $ ;
   run;
```

```
*************************************************************************
*   Invoke the CONVERT macro if the user needs to convert the gene      *
*   values to "D" or "R".  Otherwise delete the %convert statement.     *
*   The SGENE and LINKAGE macros are expecting the following gene       *
*   values:                                                             *
*    D for Dominant,                                                    *
*    R for Recessive,                                                   *
*    . or blank for missing value.                                      *
*                                                                       *
*   Specify the following parameters:                                   *
*      DS    - SAS dataset to convert                                   *
*      GENES - gene names from the SAS dataset                          *
*      DSOUT - output SAS dataset that has been converted               *
*************************************************************************;
%convert(ds=original,
         genes=BI RC DV SP LL DF F B D U TU,
         dsout=new);

*************************************************************************
*   Invoke the SGENE macro.                                             *
*   Modify the following parameters for your experiment:                *
*      DS    - SAS dataset to analyze (possibly the output dataset      *
*              from the CONVERT macro.                                  *
*      GENES - gene names from the SAS dataset                          *
*      P1    - critical value for about half of the frequency of one    *
*              parent to determine the expected segregation ratio       *
*              (1:1 or 1:0) in BC1 generation.                          *
*                                                                       *
*              Indicates the number of plants of parent 1               *
*              that you feel must have the trait                        *
*              before you accept it as uniform                          *
*              (for example, 15 plants of P1 measured;                  *
*              critical value set at 10,                                *
*              allowing 5 misclassifications)                           *
*************************************************************************;
%sgene(ds=new,
       genes=BI RC DV SP LL DF F B D U TU,
       p1=9);

*************************************************************************
*   Invoke the LINKAGE macro.                                           *
*   Modify the following parameters for your experiment:                *
*      DS    - SAS dataset to analyze (possibly the output dataset      *
*              from the CONVERT macro).                                 *
*      GENES - gene names from the SAS dataset                          *
*      P1, P2- critical value for about half of the frequency of        *
*              the parents to determine if the phase is coupling or     *
*              repulsion.                                               *
*                                                                       *
*              Indicates the number of plants of parent 1               *
*              that you feel must have the trait                        *
*              before you accept it as uniform                          *
*              (for example, 15 plants of P1 measured;                  *
*              critical value set at 10,                                *
*              allowing 5 misclassifications)                           *
*************************************************************************;
%linkage(ds=new,
```

```
      genes=BI RC DV SP LL DF F B D U TU,
      p1=9,
      p2=9);
```

File 3: CONVERT.SAS

```
*************************************************************************
*                                                                       *
*     SASGENE 1.1                                                       *
*     Program for Analysis of                                           *
*     Gene Segregation and Linkage                                      *
*     November 5, 1997                                                  *
*                                                                       *
*************************************************************************;

%macro convert
       (ds=_last_, /* SAS dataset to analyze(default:uses last one)*/
        genes=,    /* gene variable names                          */
        dsout=     /* name of new SAS dataset after conversion     */
        );

*************************************************************************
* Name:     CONVERT                                                     *
*                                                                       *
* Purpose:  Converts gene values to Dominant or Recessive               *
*                                                                       *
* Written:  09/14/95                                                    *
*                                                                       *
* Modified: 10/02/95                                                    *
*           03/05/97                                                    *
*                                                                       *
* Products: Base SAS                                                    *
*                                                                       *
* Example:  %convert(ds=save.orig,                                      *
*           genes=BI RC  DV SP LL DF F B D U TU SS NS,                  *
*           dsout=new);                                                 *
*************************************************************************;
proc format;
   value _gnrx
     1='P1'
     2='P2'
     3='F1'
     4='F2'
     5='BC1P1'
     6='BC1P2'
     ;
     run;
title2 'Gene Segregation and Linkage Analysis';
%local nogene word geneid i;

  /*  create nogenes macro variable                                    */
  /*  nogenes is the number of genes listed in &genes                  */
  %let nogenes=0;
  %if &genes ne %then %do;
      %let word=%scan(&genes,1);
```

```
      %do %while (&word ne );
          %let nogenes=%eval(&nogenes+1);
          %let word=%scan(&genes,&nogenes+1);
          %end;
      %end;
   /*  create geneid macro variable                                    */
   /*  geneid is the names of the genes in quotes                      */
   /*         used in array for identification in output               */
   %let word=%scan(&genes,1);
   %let geneid=%str(%'&word%');
   %do i=2 %to &nogenes;
       %let word=%scan(&genes,&i);
       %let geneid=%str(&geneid,%'&word%');
       %end;

proc sort data=&ds   out=_orig; by family; run;
data _generat; set _orig;
   length id 3;
   array y{*}    &genes;
   array yc{*} $ n1-n&nogenes (%unquote(&geneid));
   id=0;
   do _i_=1 to dim(y);
      id+1;
      code= y{_i_};
      gene=yc{_i_};
      output;
      end;
   keep family id gene gnr code;
   run;
proc sort data=_generat; by family id; run;

proc freq noprint;
   by family id gene;
   where code not=' ';
   tables code / out=_count;
   run;
proc means noprint; by family id ;
   var count;
   output out=_nocode n=n;
   run;
data _look; merge _count _nocode;  by family id;
   if n>2;
   run;
proc print label;
title3 'Observed frequencies for each gene locus and allele code';
title4 'These genes in this table have more than 2 codes:';
title5 '      some codes may have been misentered        ';
title6 'WARNING!!! Program will convert to 2 codes (D and R)
     ';
title7 '           Dominant will be assigned,
     ';
title8 '           other non-missing codes will be set to Recessive
     ';
   var family gene code count;
   label count='FREQUENCY';
   run;
```

```
/*  delete gene-family ids that do not make sense for analysis    */
/*  delete when the phenotype of P1 is the same as the            */
/*                  phenotype of P2                               */
title3 ' ';
proc freq     data=_generat noprint;
   by family id gene;
   tables code*gnr / out=_gnrcode(drop=percent) ;
   run;
data _gnrcode; set _gnrcode;
   if code=' ' then delete;
proc sort data=_gnrcode; by family id gene gnr descending count;
   run;
data _delete(keep=family id gene); set _gnrcode;
   by family id gene gnr;
   retain d1;
   if first.id then do;
      d1='        ';
      d2='        ';
      end;
   if first.gnr then do;
      if gnr=1 then d1=code;
         else if gnr=2 then do;
                            d2=code;
                            if d1=d2 then output _delete;
                            end;

      end;
   run;
proc print data=_delete(drop=id);
   title3 'These gene-family combinations will be deleted ';
   title4 'since the phenotype for P1 and P2 are the same ';
   title5 'and do not fit the assumptions of the analysis.';
   run;
data _generat _look; merge _generat _delete(in=yes);
   by family id gene;
   if yes then output _look;
      else output _generat;
   run;
proc freq     data=_look;
   by family id gene;
   tables code*gnr / missprint nocum nopercent norow nocol;
   label gnr='GENERATION';
   format gnr _gnrx.;
   run;

  /*   find the dominant gene by looking at generation 3 (F1)      */
title3 ' ';
proc freq noprint data=_generat;
   by family id gene;
   where gnr=3;
   tables code / out=_count;
   run;
proc sort; by family id count; run;

data _dom; set _count;
   by family id;
```

```
   array c  $ c1-c&nogenes;
   retain c1-c&nogenes;
   length c1-c&nogenes  $8;
   if first.family  then do;
      do _i_=1 to &nogenes;
         c{_i_}=' ';
         end;
      end;

   if last.id then c{id}=code;
   if last.family then output;
   keep family c1-c&nogenes;
   run;
data &dsout; merge _orig _dom;
   by family;
   array genes{*}     &genes;
   array dom{*}   $ c1-c&nogenes;

   do _i_=1 to dim(genes);
      if dom{_i_}=' ' then genes{_i_}=' ';/*useless data- no domi-
     nant*/
         else do;
         if genes{_i_}=dom{_i_}     then genes{_i_}='D';
            else if genes{_i_}=' ' then genes{_i_}=' ';
            else genes{_i_}='R';
         end;
      end;
   drop c1-c&nogenes _i_;
   run;
data _check; merge _orig _dom;
   by family;
   array genes     &genes;
   array dom $ c1-c&nogenes;
   array yc{*} $ n1-n&nogenes (%unquote(&geneid));
   id=0;
   do _i_=1 to dim(genes);
      id+1;
      gene=yc{_i_};
      old_code=genes{_i_};
      if dom{_i_}=' ' then new_code=' '; /*useless data- no dominant
     */
         else do;
         if genes{_i_}=dom{_i_}     then new_code='D';
            else if genes{_i_}=' ' then new_code=' ';
            else new_code='R';
         end;
      output;
      end;
   drop c1-c&nogenes n1-n&nogenes &genes;
   run;

title4 "Conversion to 'D' or 'R' for each gene and family";
proc freq;
   tables id*gene*family*new_code*old_code/list nopercent nocum
     nofreq;
```

```
   run;
proc datasets library=work memtype=data nolist;
   delete _check _count _dom _generat _look _nocode _orig
          _delete _gnrcode;
   quit;
%mend convert;
```

File 4: SGENE.SAS

```
*************************************************************************
*                                                                       *
*     SASGENE 1.1                                                       *
*     Program for Analysis of                                           *
*     Gene Segregation and Linkage                                      *
*     November 5, 1997                                                  *
*                                                                       *
*************************************************************************;

%macro sgene
        (ds=_last_, /* SAS dataset to analyze(default:uses last one)*/
         genes=,    /* gene variable names                          */
         p1=        /* freq of parent(P1) to determine Dom. or Rec. */
         );

*************************************************************************
* Name:     SGENE                                                       *
*                                                                       *
* Purpose:  Single Locus Goodness of Fit Test                           *
*                                                                       *
* Written:  06/22/95                                                    *
*                                                                       *
* Modified: 10/03/95                                                    *
*           03/05/97                                                    *
*                                                                       *
* Example:  %sgene(ds=dst,                                              *
*           genes=BI RC DV SP LL DF F B  D  U  TU ,                     *
*           p1=9);                                                      *
*************************************************************************;

%local nogene word geneid i;
title2 'Gene Segregation and Linkage Analysis';
title3 'Single Locus Goodness of Fit Test';
title4 'Probability >.05 is accepted as Single Locus';
options missing=' ';
proc format;
   picture _prob
       low-0.05   ='9.999*'
       0.05<-<0.06='9.999 '
       0.06-high  ='9.99  '
                  ='      '
      .
      ;
   value _gnrx
     1='P1'
     2='P2'
     3='F1'
```

```
      4='F2'
      5='BC1P1'
      6='BC1P2'
      ;
      run;

   /*  create nogenes macro variable                                    */
   /*  nogenes is the number of genes listed in &genes                  */
   %let nogenes=0;
   %if &genes ne %then %do;
       %let word=%scan(&genes,1);
       %do %while (&word ne );
           %let nogenes=%eval(&nogenes+1);
           %let word=%scan(&genes,&nogenes+1);
           %end;
       %end;

   /*  create geneid macro variable                                     */
   /*  geneid is the names of the genes in quotes                       */
   /*         used in array for identification in output                */
   %let word=%scan(&genes,1);
   %let geneid=%str(%'&word%');
   %do i=2 %to &nogenes;
       %let word=%scan(&genes,&i);
       %let geneid=%str(&geneid,%'&word%');
       %end;

data _gent(keep=id family gnr a gene aa bb ee)
     _look(keep=obs family gnr &genes);
    set &ds;
    length id aa bb ee 3
           obs 4
           a $ 1;
    array  y{*}     &genes;
    array yc{*} $  n1-n&nogenes  (%unquote(&geneid)) ;

 /* create an obs. for each gene                                       */
 /* a    will be the response variable for phenotype of individual     */
 /*      of each gene                                                  */
 /* gene will be the character id of each gene name                    */
 /* id   will be the numeric   id of the gene -used for sorting        */

   obs+1;
   id=0;
   do _i_=1 to dim(y);
         id+1;
         a=y{_i_}; gene=yc{_i_};
         a=upcase(a);
 /* ensure all values are in upper case                                */
 /* if the phenotype is dominant,  then aa=1                           */
 /* if the phenotype is recessive, then bb=1                           */
         aa=0; bb=0; ee=0;
         if      a ='D'  then aa=1;
         else if a ='R'  then bb=1;
         else if a =' '  then ee=1;
         else do;
```

```
                put '******* ERROR ******* '
                    'Invalid value for gene ' gene
                    ' (' gene '=a ')  at obs=' obs;
                output _look;
                end;
            output _gent;
            end;
    run;
    /* print any invalid data values for gene to notify user   */
title4 'Invalid data value for at least one gene'
       ' (value is not D, R, or missing)';
proc print data=_look;
    id obs;

    run;
proc datasets library=work nolist;
    delete _look;
    run;

title4 'Probability >.05 is accepted as Single Locus';
    /* compute the sums for number of dominant and recessive    */
    /* individuals in 6 generations                             */
proc means data= _gent noprint nway;
     class id family gnr;
     id gene;
     var aa bb ee;
     output out =_sum sum=d r missing;
     run;

    /* compute chi square and probability     */
data _chisq; set _sum;
    by id family;
    retain g1 omit;
    if first.family then do;
                         g1='   ';
                         omit='no ';
                         end;
    g1text='   ';

/* determine if the genotype of recurrent parent is dominant or   */
/* recessive;  this information is needed to choose               */
/* expected 1:1 or 1:0 for chisq in BC1 and BC2                    */
if gnr =1 then do;
    t=sum(d,r);
    if t<=0 then omit='yes';
    if 0<d<&p1 then g1='REC';
       else if d>=&p1 then g1='DOM';
       else g1=' ';
    end;

if omit='yes' then delete;

if gnr >3 then do;
    t=d+r;
    chisq=0; df=0;
```

```
/* expected is 3:1 for chisq in F2 */
if gnr=4 then do;
       gltext='3:1';
       /*chisq for 3:1      */
       chisq=(d-t*0.75)**2/ (t*0.75) + (r-t*0.25)**2/(t*0.25);
       end;

   /* choose expected 1:1 or 1:0 for chisq in BC1 and BC2 */
   /* according to dominant or recessive recurrent parent */
   else if gnr =5 then do;
        if g1='DOM' then do;
            gltext='1:0';
            chisq=((d-t)**2)/ t ;
            end;
        else if g1='REC' then do;
            gltext='1:1';
            chisq=(d-t*0.5)**2/(t*0.5) +
                  (r-t*0.5)**2/(t*0.5);
            end;
        end;
   else if gnr =6 then do;
        if g1='DOM' then do;
            gltext='1:1';
            chisq=(d-t*0.5)**2/(t*0.5) + (r-t*0.5)**2/(t*0.5);
            end;
        if g1='REC' then do;
            gltext='1:0';
            chisq=((d-t)**2)/ t ;
            end;
        end;
   df=1;
   prob=probchi(chisq,df);
   prob=1-prob;
   end;
 drop omit;
 drop id _type_ t g1;
 run;

proc datasets library=work nolist;
   delete _gent _sum;
   run;

proc print noobs label uniform data=_chisq;
   by notsorted gene family;
   pageby gene;
   format chisq 8.2  prob _prob.  gnr _gnrx.;
   label gnr='GENERATION';
   label d='DOMINANT';
   label r='RECESSIVE';
   label gltext='EXPECTED';
   label _freq_='N';
   run;
%mend sgene;
```

File 5: LINKAGE.SAS

```
*************************************************************************
*                                                                       *
*     SASGENE 1.2                                                       *
*     Program for Analysis of                                           *
*     Gene Segregation and Linkage                                      *
*     March 2, 1999                                                     *
*                                                                       *
*                                                                       *
*     linkage.sas of SASGENE 1.2 differs from SASGENE 1.1               *
*     because there was an error in the calculation of                  *
*     the SE for the F2 (coupling) as follows:                          *
*     in SASGENE 1.1 the formula was:                                   *
*                  se=( (1-p*p)*(1+p*p)/(2*t*(1+2*p*p)) )**0.5;         *
*     in SASGENE 1.2 the formula is now:                                *
*                  se=( (1-p*p)*(2+p*p)/(2*t*(1+2*p*p)) )**0.5;         *
*                                                                       *
*************************************************************************;

%macro linkage
       (ds=_last_, /* SAS dataset to analyze(default:uses last one) */
        genes=,    /* gene variable names                           */
        p1=,       /* freq of P1 to determine Coupling or Repulsion */
        p2=        /* freq of P2 to determine Coupling or Repulsion */
        );

*************************************************************************
* Name:     LINKAGE                                                     *
*                                                                       *
* Purpose:  Linkage Analysis for                                        *
*           Recombination Frequency Data in F2, BC1P1 & BC2P2 Pop.      *
*                                                                       *
* Written:  06/22/95                                                    *
*                                                                       *
* Modified: 10/03/95                                                    *
*           03/05/97                                                    *
*                                                                       *
* Example:  %linkage(ds=dst,                                            *
*           genes=BI RC DV SP LL DF F B  D  U  TU ,                     *
*           p1=9                                                        *
*           p2=9                                                        *
*           );                                                          *
*                                                                       *
* Note:     The number of genes listed affects the amount of            *
*           time the program takes to execute.  The resources for       *
*           your platform will determine the number of genes you        *
*           can use.  Increasing the number of genes increases          *
*           the work space that is needed.                              *
*************************************************************************;

%local nogenes word geneid i;
title2 'Gene Segregation and Linkage Analysis';
title3 'Recombination Frequency Data in F2, BC1P1 & BC1P2 Population';
title4 'Prob with * indicates gene pair might be linked';
options missing=' ';
```

```
proc format;
   picture _prob
      low-0.05   ='9.999*'
      0.05<-<0.06='9.999 '
      0.06-high  ='9.99  '
      .          ='      '
      ;
   value _gnrx
     4='F2'
     5='BC1P1'
     6='BC1P2';
   run;
  /*  create nogenes macro variable                                    */
  /*  nogenes is the number of genes listed in &genes                  */
  %let nogenes=0;
  %if &genes ne %then %do;
      %let word=%scan(&genes,1);
      %do %while (&word ne );
          %let nogenes=%eval(&nogenes+1);
          %let word=%scan(&genes,&nogenes+1);
          %end;
      %end;

  /*  create geneid macro variable                                     */
  /*  geneid is the names of the genes in quotes                       */
  /*         used in array for identification of output                */
  %let word=%scan(&genes,1);
  %let geneid=%str(%'&word%');
  %do i=2 %to &nogenes;
      %let word=%scan(&genes,&i);
      %let geneid=%str(&geneid,%'&word%');
      %end;

data _gent;
   set &ds;
   length id aa bb cc dd ee  3
          m n                $ 1;
   array  y{*}     &genes;
   array yc{*} $  n1-n&nogenes  (%unquote(&geneid)) ;

/* create an obs. for each pair of genes                               */
/* m   will be the response variable for gene1                         */
/* n   will be the response variable for gene2                         */
/* id  will be the numeric id of the (i,j)th combination of gene pair*/
/* gene1         character id of the i-th part of (i,j) pair           */
/* gene2         character id of the j-th part of (i,j) pair           */

 obs+1;
 id=0;
  do _i_=1 to dim(y)-1;
     do _j_=_i_+1 to dim(y);
        id+1;
        m=y{_i_};   n=y{_j_};
        m=upcase(m);
        n=upcase(n);
        gene1=yc{_i_}; gene2=yc{_j_};
        aa=0; bb=0; cc=0; dd=0; ee=0;
```

```
            if     m ='D' and n ='D' then aa=1;
           else if m ='D' and n ='R' then bb=1;
           else if m ='R' and n ='D' then cc=1;
           else if m ='R' and n ='R' then dd=1;
           else if m =' ' or  n =' ' then ee=1;
           else put '*******ERROR******* '
                    'Invalid data value on obs=' obs ' for '
                      yc{_i_}'=' m ' or '  yc{_j_}'=' n ;
        output;
        end;
      end;
   keep id family gnr m n gene1 gene2 aa bb cc dd ee;
   run;

  /* compute the sums of dominant and recessive individuals  */
  /* a=AABB b=AAbb c=aaBB d=aabb    */
proc means data= _gent noprint nway;
    class id family gnr;
    id gene1 gene2;
    var aa bb cc dd ee;
    output out =_sum(drop=_type_ ) sum=a b c d missing ;
    run;


data _P12; set _sum;
   by id family;
   retain phase
          p1dd p1dr p1rd p1rr
          p2dd p2dr p2rd p2rr
          omit;
   if first.family then do;
      p1dd=.; p1dr=.; p1rd=.; p1rr=.;
      p2dd=.; p2dr=.; p2rd=.; p2rr=.;
      phase=' ';
      omit='no ';
      end;

 if gnr=1 then do;
      t=sum(a,b,c,d);
      if t<=0 then omit='yes';
      if a=>&p1 then p1dd=1;
      if b=>&p1 then p1dr=1;
      if c=>&p1 then p1rd=1;
      if d=>&p1 then p1rr=1;
      end;
   else if gnr=2 then do;
      if a=>&p2 then p2dd=1;
      if b=>&p2 then p2dr=1;
      if c=>&p2 then p2rd=1;
      if d=>&p2 then p2rr=1;

   /* determine if phase= "C"(coupling),         */
   /*                    "R"(repulsion), or      */
   /*                    " "(useless phase).     */

      if         p1dd=1 and p2rr=1 then phase='C';
         else if p1rr=1 and p2dd=1 then phase='C';
```

```
        else if p1dr=1 and p2rd=1 then phase='R';
        else if p1rd=1 and p2dr=1 then phase='R';
      end;

   if omit='yes' then delete;

/* compute the chisq, probability, recombination frequencies (rf) */
/* and standard error (se).                                       */
if phase not=' ' and gnr>3 then do;
   t=sum(a,b,c,d);
   chisq=0; df=0;
   if gnr=4 then do;
      chisq=( a**2)/(t*9/16)+( b**2)/(t*3/16)+( c**2)/(t*3/16)
                +( d**2) /(t*1/16) -t ;

      div=b*c-a*d;
      if div ne 0 then do;
         p=( (-(b*c+a*d)+((b*c+a*d)**2+a*d*(b*c-a*d))**0.5 )/div)
           **0.5;
         se=( (1-p*p)*(2+p*p)/(2*t*(1+2*p*p)) )**0.5;
         end;
      if phase='C' then rf=1-p;
         else if phase='R' then rf=p;
      end;

    else if 5<=gnr <=6 then do;
      chisq=  (a-t*0.25)**2/(t*0.25)+(b-t*0.25)**2/(t*0.25)
            + (c-t*0.25)**2/(t*0.25)+ (d-t*0.25)**2/(t*0.25);
      rf= (b+c)/t;
      se=(rf*(1-rf)/t)**0.5;
      end;
    df=3;
    prob=probchi(chisq,df);
    prob=1-prob;
    end;
drop omit;
drop p1dd p1dr p1rd p1rr p2dd p2dr p2rd p2rr   div p;
run;

/*  only print for good phases and generations 4, 5, and 6  */
data _gnr4to6;
   set _p12;
   if phase=' ' then delete;
   if gnr >3;
   run;
proc sort; by gnr phase id family; run;

proc print noobs label split='*' uniform data=_gnr4to6;
   by gnr;
   pageby gnr;
   var gene1 gene2  family phase _freq_ a b c d missing chisq df prob
     rf se;
   format chisq 5.1    rf se 6.3    prob _prob. gnr _gnrx.;
   label gnr='GENERATION';
   label family='FAM';
   label _freq_='N';
   label missing='MISS*-ING';
```

```
   label se='STD *ERROR';
   run;
%mend linkage;
```

SAS Output: Single-Gene Goodness-of-Fit

Cucumber Gene Linkage Example
Single Locus Goodness of Fit Test
Probability >.05 is accepted as Single Locus
GENE=SS FAMILY=44

GENERATION	N	DOMINANT	RECESSIVE	MISSING	EXPECTED	CHISQ	DF	PROB
P1	45	45	0	0				
P2	45	1	40	4				
F1	54	49	5	0				
F2	162	103	55	4	3:1	8.11	1	0.004*
BC1P1	81	78	3	0	1:0	0.11	1	0.73
BC1P2	81	38	42	1	1:1	0.20	1	0.65

SAS Output: Linkage Analysis

Cucumber Gene Linkage Example
Recombination Frequency (RF) Data in F2, & BC1 Population
Prob with * indicates gene pair might be linked
GENERATION=F2

GENE1	GENE2	FAM	PHASE	N	A	B	C	D	MISS-ING	CHISQ	DF	PROB	RF	STD ERROR
U	SS	30	C	162	69	27	24	36	6	75.8	3	0.000*	0.323	0.036
U	SS	44	C	162	77	27	26	28	4	35.5	3	0.000*	0.350	0.038
U	NS	28	C	162	89	16	18	21	16	24.3	3	0.000*	0.265	0.034
U	NS	30	C	162	74	22	33	27	6	35.0	3	0.000*	0.364	0.038
RC	NS	30	R	162	83	34	24	15	6	4.8	3	0.18	0.559	0.042