Manual for KINGROUP v2.08+
Chapter 1

dmitry.konovalov@jcu.edu.au

Introduction

Please note that starting from the 2nd version, the KINGROUP program is aiming to replicate the Mac based KINSHIP in its entirety. If you discover functionality of KINSHIP that is missing in KINGROUP, please let me know.

Some parts of this manual are based on, and/or extensively adopted from, the KINSHIP manual . Such sections are highlighted and referenced to the original manual of Keith Goodnight. Heavy reuse of the KINSHIP manual should help those users who have had some experience with the KINSHIP program already. For those users who wish to learn more about KINSHIP, the original KINSHIP manual is available from the KINGROUP website (for those without a Mac) and the KINSHIP website together with KINSHIP program.

Pairwise Likelihood Calculations

See and [a pdf is available from www.kingroup.org].

Perform Significance Test

The following highlighted text is adopted from the Kinship Manual :

Confidence levels for the pairwise likelihood values are estimated empirically through simulations for a given hypothesis $\{R_m, R_p\}$. Following Goodnight & Queller (1999), a simulated pair is generated as follows:

1. alleles of X are randomly set based on the allele frequencies $\{P_X\}$;
2. maternal allele Y_m is then set to X_m with probability of R_m;
3. with probability $1 - R_m$, the allele Y_m is randomly set based on the population allele frequencies;
4. steps 2 and 3 are repeated for paternal alleles.

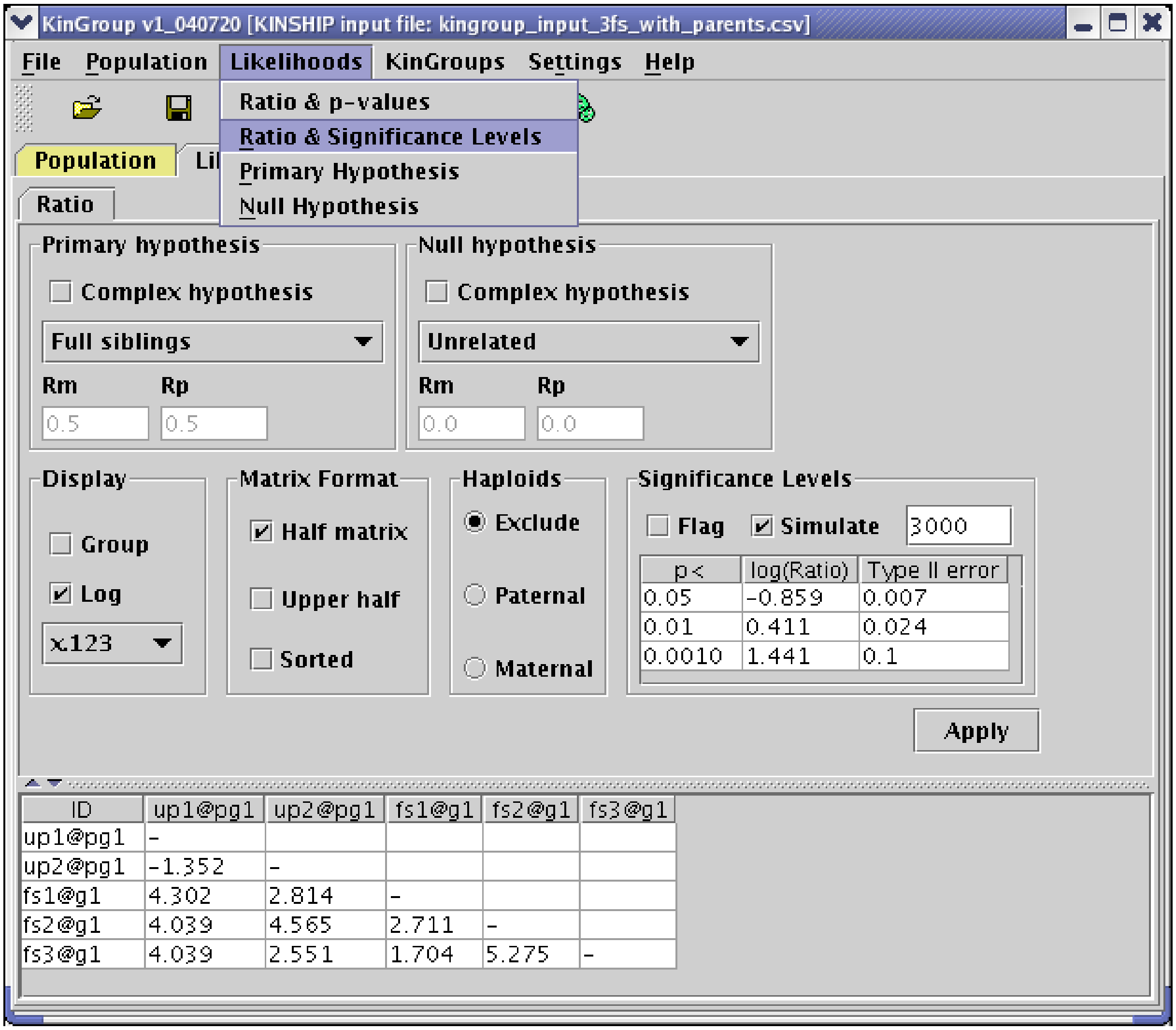
Given the probability of rejection of the null hypothesis when it is true (the type I error rate p_α), *KinGroup* can find a significance value, r_α , from the simulations as follows:

1. A number of simulated pairs, specified by the user, are generated according to the null hypothesis $H_0 : \nu = \{R_m^\nu, R_p^\nu\}$;
2. *KinGroup* then calculates the ratio of the pairwise likelihood for the primary hypothesis over the null hypotheses for each pair;
3. Based on the resulting distribution of ratios $\{r_x^\nu, x = 1,...M\}$, *KinGroup* finds the significance value r_α corresponding to the p_α -value. For example, for $p_\alpha = 0.05$, r_α is approximated by one of r_x^ν that is greater or equal then 95% of all $\{r_x^\nu\}$.

Three preset choices for p_α are 0.05, 0.01, and 0.001.

The type II error rate is calculated by repeating the above process, but generating pairs according to the primary hypothesis $H_1 : \pi = \{R_m^\pi, R_p^\pi\}$. The type II error rate (p_β), the probability of not rejecting the null hypothesis when it is false, is calculated as the percentage of ratio values $\{r_x^\pi\}$ that are smaller then the significance value r_α .

Example



Primary hypothesis

☐ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☐ Complex hypothesis

Unrelated

Rm 0.0 Rp 0.0

Display

☐ Group

☑ Log

x.123

Matrix Format

☑ Half matrix

☐ Upper half

☐ Sorted

Haploids

● Exclude

○ Paternal

○ Maternal

Significance Levels

☐ Flag ☑ Simulate 3000

p<	log(Ratio)	Type II error
0.05	-0.859	0.007
0.01	0.411	0.024
0.0010	1.441	0.1

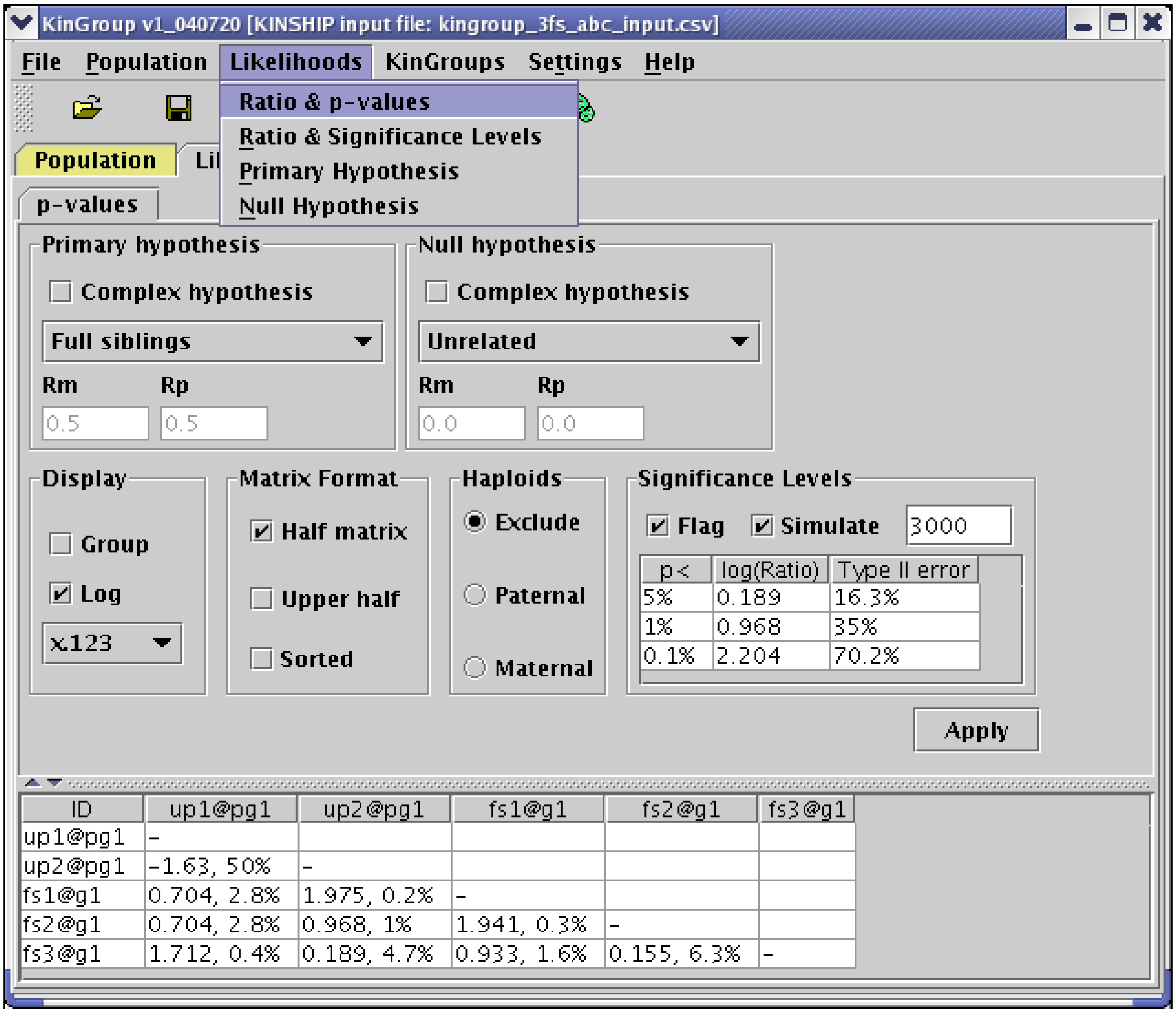
ID	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
up1@pg1	-				
up2@pg1	-1.352	-			
fs1@g1	4.302	2.814	-		
fs2@g1	4.039	4.565	2.711	-	
fs3@g1	4.039	2.551	1.704	5.275	-

Figure 1 Significance test for examples/kinship_complex/kingroup_input_3fs_with_parents.csv

Matrix of p-values

More detailed information could be generated by selecting Menu | Likelihoods | Ratio & p-values (Figure 2). In the case of significance levels (see previous subsection) the ratios may be flagged using 4 markers ("ns", "*", "**" and "***") in relation to the levels generated for each of the p-values (0.05, 0.01, and 0.001). This option allows flagging each of the pairwise ratios with the corresponding p-values.

In order to find a p-value for a given ratio, the requested number of pairs is randomly generated based on the null hypothesis. All resulting ratios are then sorted and the p-value is calculated as the percentage of all simulated ratios that are greater or equal to the given ratio value.



Primary hypothesis

☐ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☐ Complex hypothesis

Unrelated

Rm 0.0 Rp 0.0

Display

☐ Group

☑ Log

x.123

Matrix Format

☑ Half matrix

☐ Upper half

☐ Sorted

Haploids

● Exclude

○ Paternal

○ Maternal

Significance Levels

☑ Flag ☑ Simulate 3000

p<	log(Ratio)	Type II error
5%	0.189	16.3%
1%	0.968	35%
0.1%	2.204	70.2%

ID	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
up1@pg1	-				
up2@pg1	-1.63, 50%	-			
fs1@g1	0.704, 2.8%	1.975, 0.2%	-		
fs2@g1	0.704, 2.8%	0.968, 1%	1.941, 0.3%	-	
fs3@g1	1.712, 0.4%	0.189, 4.7%	0.933, 1.6%	0.155, 6.3%	-

Figure 2 Matrix of p-values for examples/kinship_fs/kingroup_3fs_abc_input.csv loaded with the first 4 (four) loci.

Complex Hypothesis

"…*Complex hypotheses:* A complex hypothesis is one in which the user specifies a range of r values instead of a single set. When a complex hypothesis is entered, *Kinship/KinGroup* will find the likelihoods for all r values within the range (the user can set how finely it divides the search space) and use the highest one that it finds in calculating the final ratio. A complex hypothesis represents a hypothesis that a given pair is related in any one of several possibilities. Its most common use will be in the null hypothesis, for example to test whether pairs are full siblings as opposed to half-siblings, cousins, or unrelated. (A simple null hypothesis could only test for full-siblings vs. a single one of the possibilities at a time.)

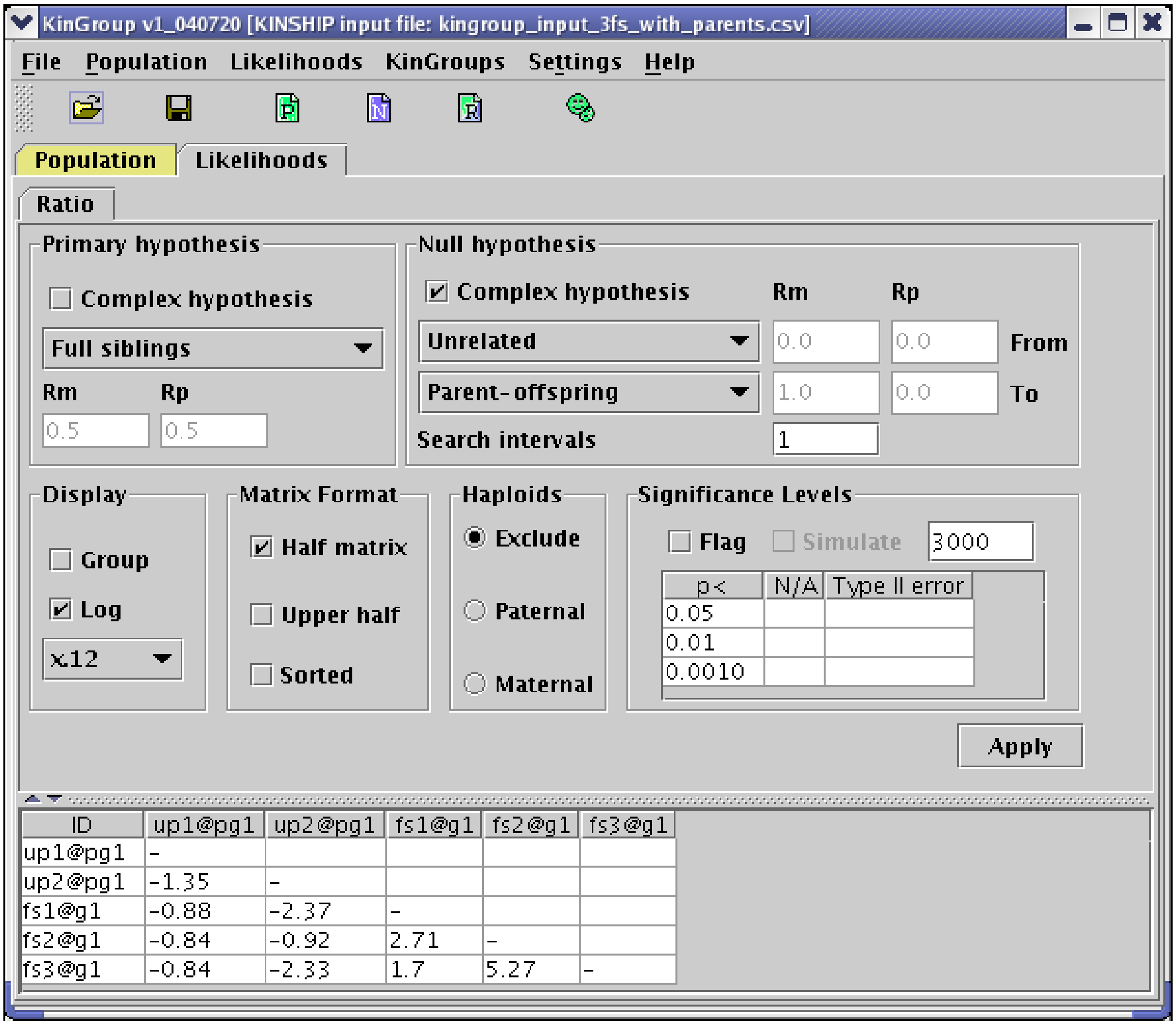
Caution should be used in choosing the complex hypothesis option. *Kinship/KinGroup* does not confine its tests to particular significant values of r_p or r_m but checks the full range of possibilities. If your range is too broad, then the meaning of the results is less clear. In addition, *Kinship/KinGroup* is unable to perform significance tests when either hypothesis is complex…"- *adopted from the Kinship Manual* .

Both primary and null hypotheses may be set as complex. In the example below, see Figure 3, only one interval is requested resulting in two values of Rm: 0 and 1. If one of the descent identity values stays the same in the "To:"-option, KinGroup scans the one that varies the number of times equal to the intervals plus one. If, however, both are different, a two dimensional grid of all possible combinations of Rm and Rp is calculated.

Example

Load "Kinship Complex" sample input file, see "Sample Input File" section on how to load this file. Figure 3 and Figure 4 show the resulting ratios for a complex and simple null hypothesis correspondingly.

The true relationship is known between "up" - unrelated parents and "fs" - full siblings since the sample file was randomly generated to conform to the Mendellian inheritance rules. Simple null hypothesis of "unrelated" individuals is not sufficient to distinguish between parents and their siblings, see Figure 4. The only ratio that is significantly lower than the rest is the one between the two unrelated parents, "up1" and "up2", which relationship corresponds exactly to the "unrelated" null hypothesis.



Primary hypothesis

□ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☑ Complex hypothesis

	Rm	Rp	
Unrelated	0.0	0.0	From
Parent-offspring	1.0	0.0	To

Search intervals 1

Display

□ Group

☑ Log

x.12

Matrix Format

☑ Half matrix

□ Upper half

□ Sorted

Haploids

● Exclude

○ Paternal

○ Maternal

Significance Levels

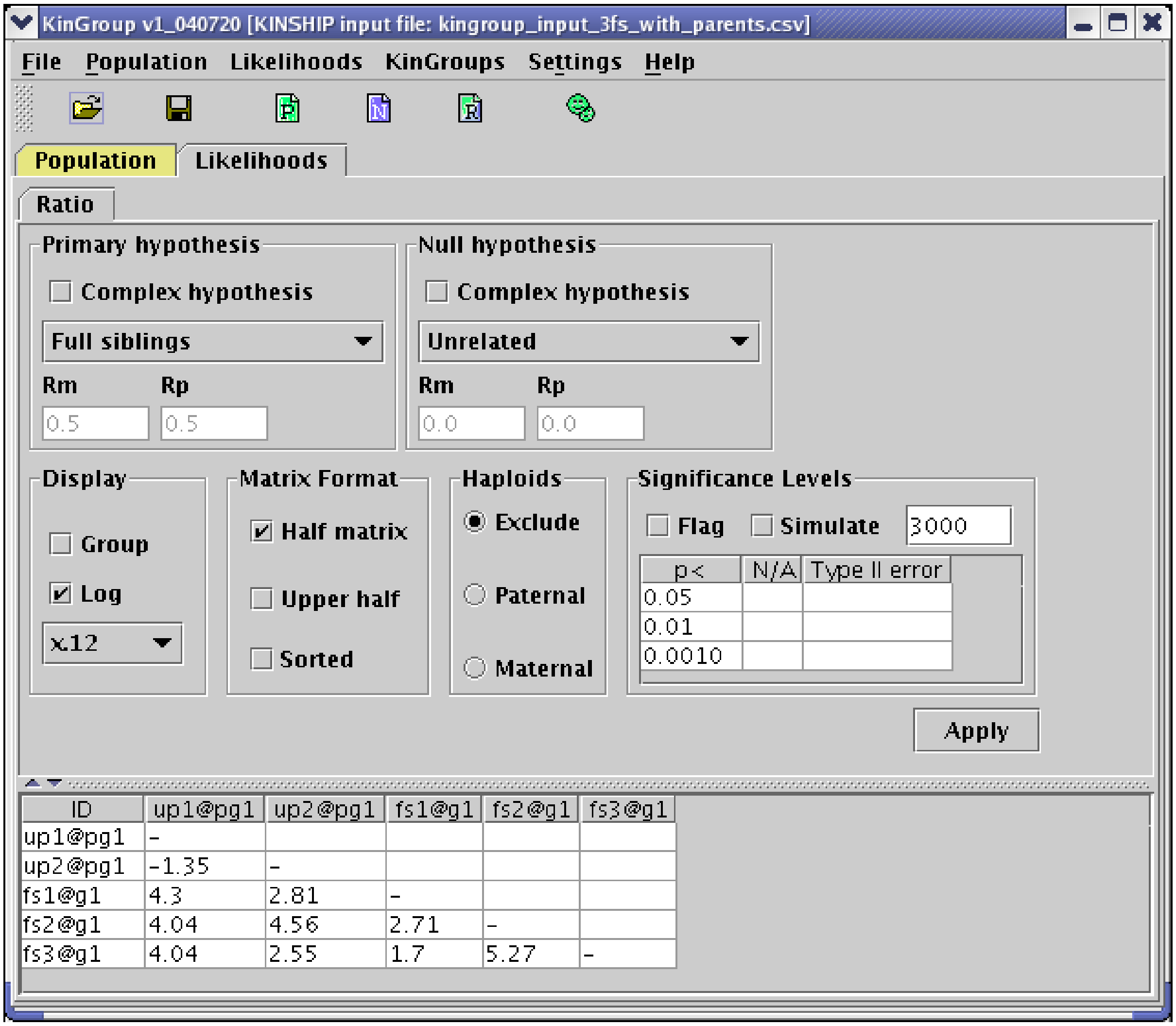
□ Flag □ Simulate 3000

p<	N/A	Type II error
0.05		
0.01		
0.0010		

ID	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
up1@pg1	-				
up2@pg1	-1.35	-			
fs1@g1	-0.88	-2.37	-		
fs2@g1	-0.84	-0.92	2.71	-	
fs3@g1	-0.84	-2.33	1.7	5.27	-

Figure 3 Ratios of primary to null likelihoods using a complex hypothesis for examples/kinship_complex/kingroup_input_3fs_with_parents.csv.

Once the “parent-offspring” relationship is included in the null hypothesis resulting parent-to-siblings pairwise likelihood ratios drop significantly, see Figure 3. These are the values between “up1, up2” and “fs1, fs2, fs3” in Figure 3.



Primary hypothesis

□ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

□ Complex hypothesis

Unrelated

Rm 0.0 Rp 0.0

Display

□ Group

☑ Log

x12

Matrix Format

☑ Half matrix

□ Upper half

□ Sorted

Haploids

● Exclude

○ Paternal

○ Maternal

Significance Levels

□ Flag □ Simulate 3000

p<	N/A	Type II error
0.05		
0.01		
0.0010		

ID	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
up1@pg1	-				
up2@pg1	-1.35	-			
fs1@g1	4.3	2.81	-		
fs2@g1	4.04	4.56	2.71	-	
fs3@g1	4.04	2.55	1.7	5.27	-

Figure 4 Null hypothesis is a simple hypothesis for examples/kinship_complex/kingroup_input_3fs_with_parents.csv.

Kin Group Reconstruction

Pairwise likelihoods provide with valuable information about a population especially when combined with the simulation to determine confidence levels. However quite often the given set of individuals must be partitioned into subgroups based on some relationship hypothesis, for example when all full siblings must be extracted for further analysis. Various approaches exist to perform such partitioning, for example and references therein.

KinGroup uses a partitioning approach in which individuals are added to partitions one at a time and all possible new partitions are created. The overall likelihood of each partition is then calculated and only the one with highest value is retained for the next iteration . Two kin group reconstruction algorithms have been implemented in this version of KinGroup, namely *Descending Ratio* and *Exhaustive Descent*.

Method

Using notation introduced in the Section "Pairwise Likelihood Calculations", a primary hypothesis $\pi = \{R_m^\pi, R_p^\pi\}$ is tested against null hypothesis$\nu = \{R_m^\nu, R_p^\nu\}$, where superscripts π and ν denote primary and null hypotheses$\{R^\pi\}$ and$\{R^\nu\}$. Pairwise likelihoods for a pair of individuals are denoted as

$$L_{ii'}(R_m^\pi, R_p^\pi) = L_{ii'}^\pi, \qquad L_{ii'}(R_m^\nu, R_p^\nu) = L_{ii'}^\nu. \qquad \textbf{(Eq. 0)}$$

Overall likelihood of a partition is defined as a product of pairwise likelihoods over all unique pairs in the sample

$$L = \prod_{i<i'} L_{ii'}^\alpha, \qquad \textbf{(Eq. 0)}$$

where $\alpha = \pi$ when i and i' belong to the same subgroup S, and where $\alpha = \nu$ when i and i' belong to different subgroups. In other words, the primary hypothesis is used within a subgroup and the null hypothesis between subgroups.

Structure of the KinGroup search algorithms

The algorithms are run in cycles starting with all individuals belonging to the pool of unassigned individuals. At each algorithm cycle one individual is assigned into kin groups. The cycles continue while the unassigned pool is not empty.

Each cycle consists of a number of steps:

1. Select one individual from the pool of unassigned individuals.
2. Build candidate partitions. Each partition is a collection of kin groups were each individual is only assigned to one group.
3. Select which partitions to keep for the next cycle.

Descending Ratio

The *Descending Ratio* search algorithm is described in .

Exhaustive Descent

The *Exhaustive Descent* search algorithm builds every possible combination of partitions by taking individuals in the order determined by the *Descending Ratio* search algorithm. However, all resulting combinations are kept for the next step and not just the combination with the best likelihood as in the *Descending Ratio*. The practical usefulness of such exhaustive search is limited to small data sets (order of tens) as the number of possible combinations rapidly approaches an incomputable level.

However the *Exhaustive Descent* algorithm is invaluable for testing the *Descending Ratio* algorithm, as it is impossible to determine if the *Descending Ratio* achieved the best partition without building all possible combinations.

Reconstruction Options

Examples

File naming abbreviations	**Description**
FS, HS	Full or half siblings
Input/results	Input or results file. For example, kingroup_2fs_input.csv
ED, DR	Exhaustive Descent or Descending Ratio algorithm results, e.g. kingroup_2fs_results_ed_us.csv
U,Q,S	Unique, sequence, sorted options, e.g. the

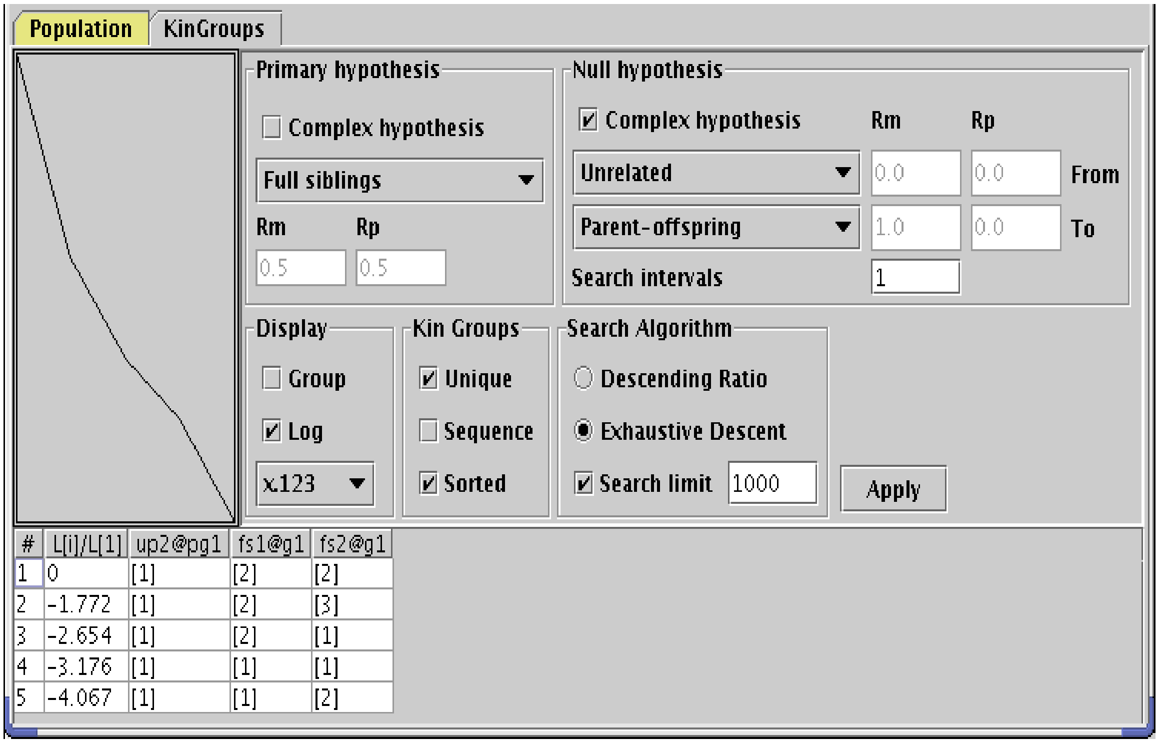
	file kingroup_2fs_results_ed_us.csv was generated with "unique" and "sorted" options switched on.

Two Full-Siblings

Location: www.kingroup.org/examples/kingroup_fs/kingroup_2fs_input.csv. This file contains one parent and its two full-sibling children. The second parent is also present in the file for verification but is commented out for this example.

This example is one of the simplest to consider and it is used to illustrate all KinGroup partitioning options. In this example we will focus on the following question: **"Given a population, which individuals are full-siblings?"**

Results for an Exhaustive Descent search are shown in Figure 5. For an explanation on loading this file see the "Directory: kingroup_fs" section on page 28 and for help with the complex hypothesis settings see the "Complex Hypothesis" subsection on page 7.



Primary hypothesis

□ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☑ Complex hypothesis Rm Rp

Unrelated 0.0 0.0 From

Parent-offspring 1.0 0.0 To

Search intervals 1

Display

□ Group

☑ Log

x.123

Kin Groups

☑ Unique

□ Sequence

☑ Sorted

Search Algorithm

○ Descending Ratio

● Exhaustive Descent

☑ Search limit 1000

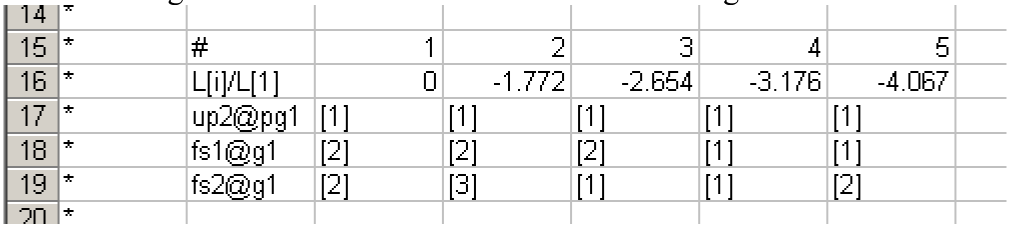
#	L[i]/L[1]	up2@pg1	fs1@g1	fs2@g1
1	0	[1]	[2]	[2]
2	-1.772	[1]	[2]	[3]
3	-2.654	[1]	[2]	[1]
4	-3.176	[1]	[1]	[1]
5	-4.067	[1]	[1]	[2]

Figure 5 Exhaustive Descent results for examples/kingroup_fs/kingroup_2fs_input.csv

The Results were saved into a file: kingroup_2fs_results_ed_us.csv, with Figure 6 showing an extract from this file. Because the "Sorted" option was selected, the top most partition in Figure 5 and the left most partition in Figure 6 is the partition with the highest overall likelihood. This partition has "up2"- unrelated parent in group [1] and two of its full-sibling children "fs1" and "fs2" in group [2]. The numbering of the kin groups is arbitrary in this version of KinGroup since the emphasis is on partitioning of a population into kin groups. Hence no special meaning is attached to the full-siblings being placed into group two and not into group one.

Please note that if the "by group" option is used and user saves the results, only the selected group will be processed and saved using Exhaustive Descent (ED). The rest of the groups will be processed using Descending Ratio (DR).

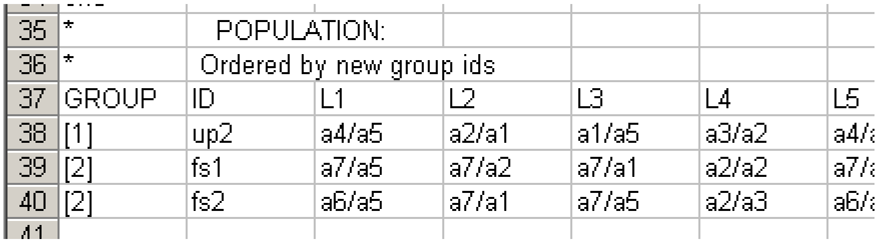
To allow for additional comparison between different partitions, the overall likelihoods are also displayed for each partition. These are displayed as a graph and the second column in Figure 5 and as a row in the results file see Figure 6.



14	*						
15	*	#	1	2	3	4	5
16	*	L[i]/L[1]	0	-1.772	-2.654	-3.176	-4.067
17	*	up2@pg1	[1]	[1]	[1]	[1]	[1]
18	*	fs1@g1	[2]	[2]	[2]	[1]	[1]
19	*	fs2@g1	[2]	[3]	[1]	[1]	[2]
20	*						

Figure 6 Possible partitions from kingroup_2fs_results_ed_us.csv.

The results file also contains the frequency block that was used plus the new set of kin groups according to the first partition (Figure 7). When the "**Sorted**" option is selected, the first partition will be the one with the highest overall likelihood. As always, the results file is saved in the KINSHIP format so it could be reloaded into KinGroup with the new kin groups for further analysis.



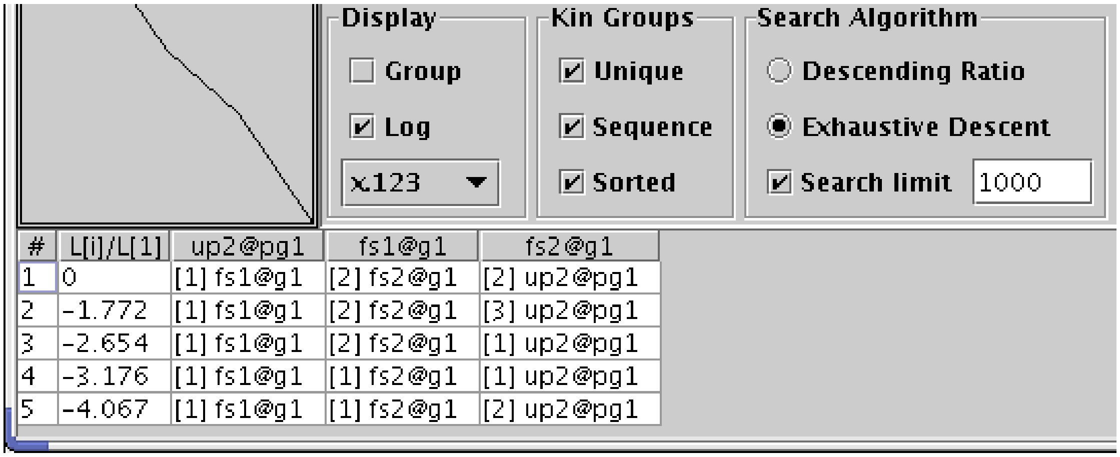
35	*	POPULATION:					
36	*	Ordered by new group ids					
37	GROUP	ID	L1	L2	L3	L4	L5
38	[1]	up2	a4/a5	a2/a1	a1/a5	a3/a2	a4/a
39	[2]	fs1	a7/a5	a7/a2	a7/a1	a2/a2	a7/a
40	[2]	fs2	a6/a5	a7/a1	a7/a5	a2/a3	a6/a
41							

Figure 7 Resulting kin groups from kingroup_2fs_results_ed_us.csv.

The "**Unique**" option does not contribute in the case of Exhaustive Descent since only unique partitions are built exhaustively until the "**Search limit**" is hit or your computer runs out of memory. The search limit is set to 1,000 in the example shown in Figure 5.

The "**Sequence**" option (Figure 8) shows in what order individuals were taken for group assignment. In the case of Exhaustive Descent all possible partitions are made first for the full-siblings "fs1" and "fs2". By running the algorithm a few times the only difference will be that the first individual may vary between "fs1" and "fs2". That is because their pair-wise primary to null hypothesis ratio is the highest amongst all thee possible ratios:

fs1-fs2, fs1-up2, and fs2-up2. The choice of which individual from the pair is taken first is random. However the next individual is always the remaining from the pair.



Display

□ Group

☑ Log

x.123

Kin Groups

☑ Unique

☑ Sequence

☑ Sorted

Search Algorithm

○ Descending Ratio

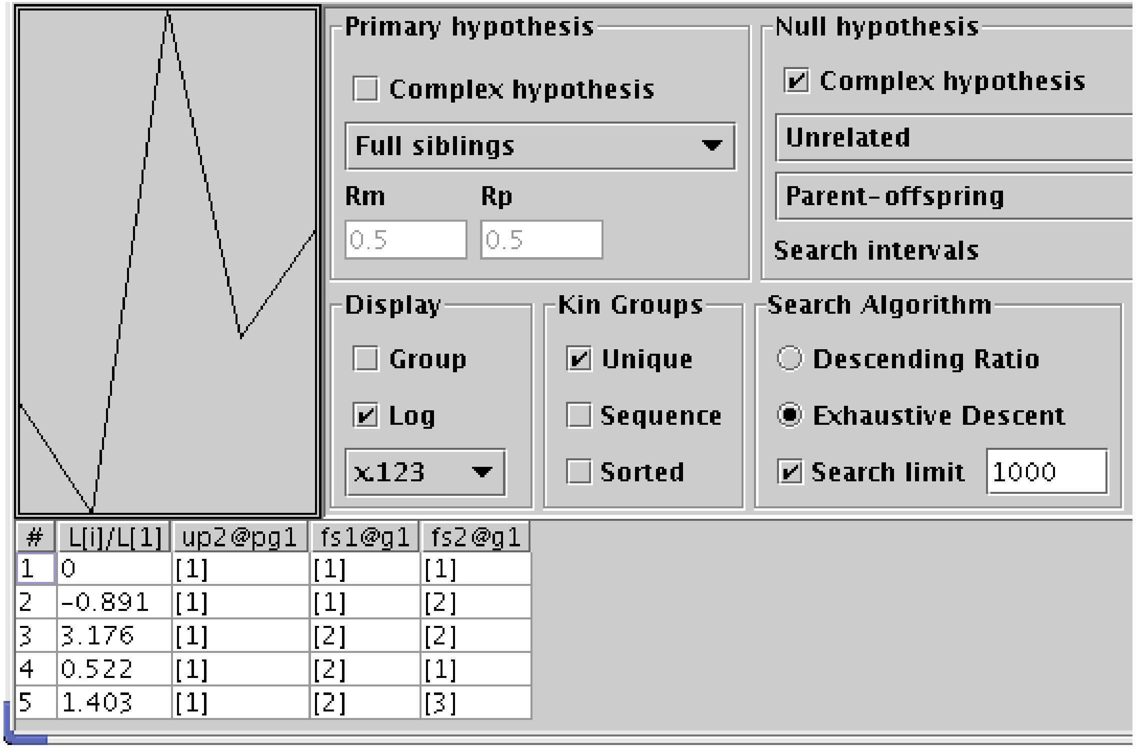
● Exhaustive Descent

☑ Search limit 1000

#	L[i]/L[1]	up2@pg1	fs1@g1	fs2@g1
1	0	[1] fs1@g1	[2] fs2@g1	[2] up2@pg1
2	-1.772	[1] fs1@g1	[2] fs2@g1	[3] up2@pg1
3	-2.654	[1] fs1@g1	[2] fs2@g1	[1] up2@pg1
4	-3.176	[1] fs1@g1	[1] fs2@g1	[1] up2@pg1
5	-4.067	[1] fs1@g1	[1] fs2@g1	[2] up2@pg1

Figure 8 Exhaustive Descent results for kingroup_2fs_input.csv with sequence.

The "**Sorted**" option is always expected to be switched on. The unsorted choice (Figure 9) will report partitions in the order they are stored internally in KinGroup, which could be quite random.



Primary hypothesis

□ Complex hypothesis

Full siblings

Rm 0.5

Rp 0.5

Null hypothesis

☑ Complex hypothesis

Unrelated

Parent-offspring

Search intervals

Display

□ Group

☑ Log

x.123

Kin Groups

☑ Unique

□ Sequence

□ Sorted

Search Algorithm

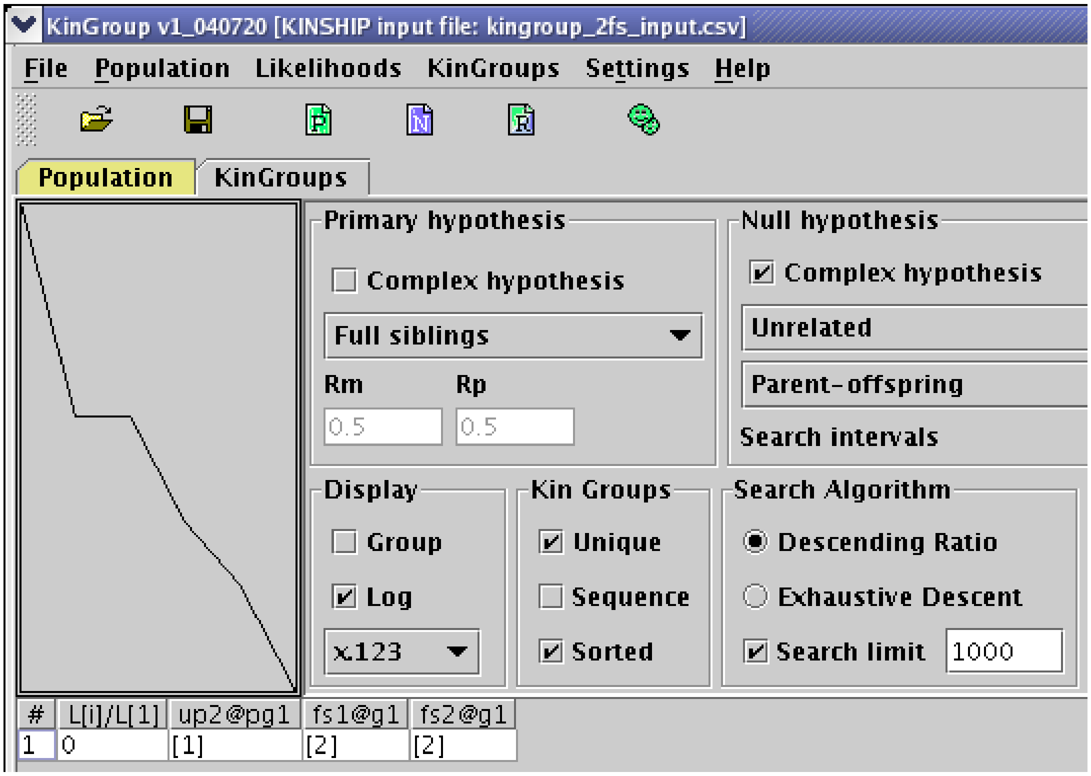
○ Descending Ratio

● Exhaustive Descent

☑ Search limit 1000

#	L[i]/L[1]	up2@pg1	fs1@g1	fs2@g1
1	0	[1]	[1]	[1]
2	-0.891	[1]	[1]	[2]
3	3.176	[1]	[2]	[2]
4	0.522	[1]	[2]	[1]
5	1.403	[1]	[2]	[3]

Figure 9 Unsorted Exhaustive Descent results for kingroup_2fs_input.csv.



Primary hypothesis

☐ Complex hypothesis

Full siblings

Rm 0.5

Rp 0.5

Null hypothesis

☑ Complex hypothesis

Unrelated

Parent-offspring

Search intervals

Display

☐ Group

☑ Log

x.123

Kin Groups

☑ Unique

☐ Sequence

☑ Sorted

Search Algorithm

● Descending Ratio

○ Exhaustive Descent

☑ Search limit 1000

#	L[i]/L[1]	up2@pg1	fs1@g1	fs2@g1
1	0	[1]	[2]	[2]

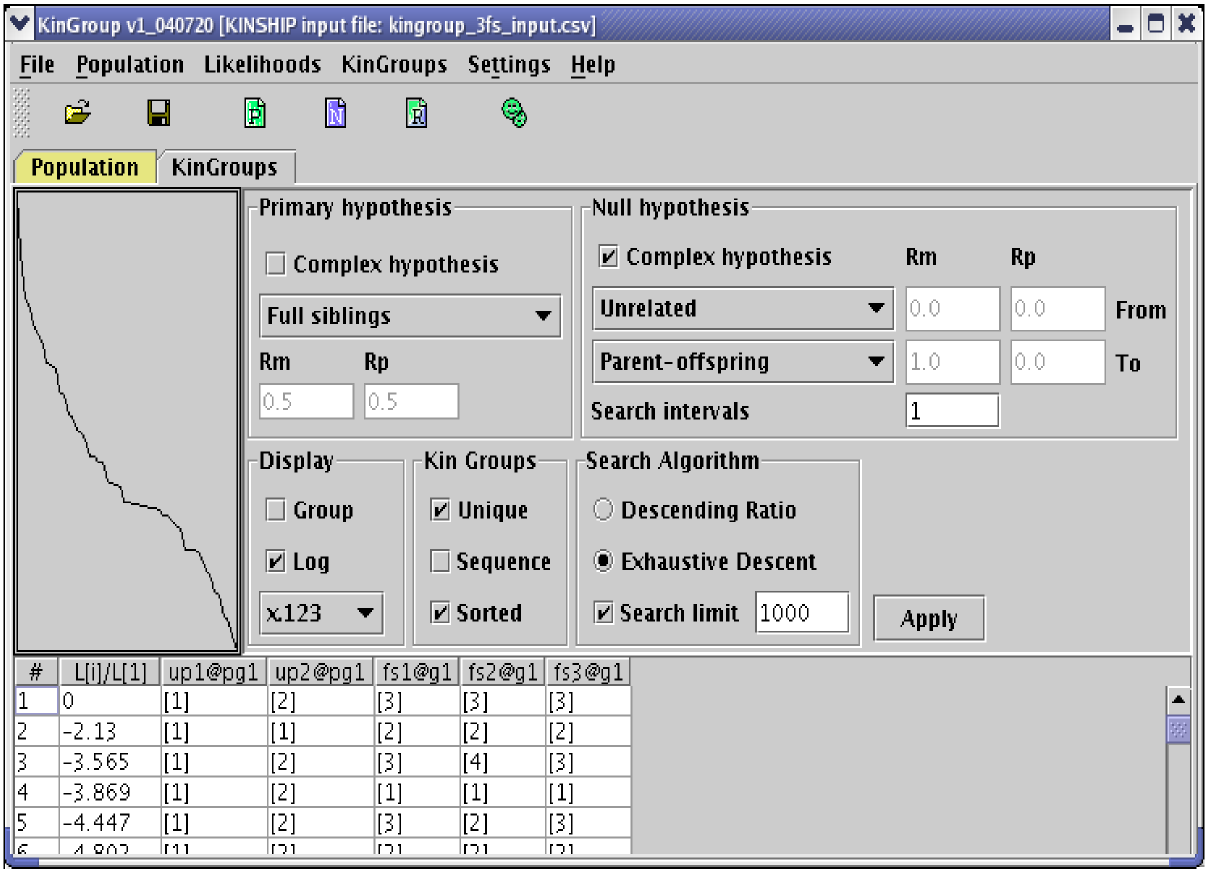
Figure 10 Descending Ratio results for kingroup_2fs_input.csv

The results of Descending Ratio algorithm for the same population are shown in Figure 10. Both algorithms have found the same kin groups correctly. The effect of various options is left to other more complicated examples.

Three Full-Siblings

Location: www.kingroup.org/examples/kingroup_fs/kingroup_3fs_input.csv. This file contains two parents (namely “up1” and “up2” – unrelated parents in the “pg1” – parental group one) and their three “fsX” - full-sibling children. All up this population contains five individuals.

This is a slightly more complicated example compared to the Two Full-Siblings example. Firstly, the Exhaustive Descent is run to obtain the theoretically correct answer within the limits of the overall likelihood approach of the Descending Ratio algorithm. Please note that just five individuals required 52 different partitions, see Figure 11 and Figure 12.


KinGroup v1_040720 [KINSHIP input file: kingroup_3fs_input.csv]
File Population Likelihoods KinGroups Settings Help
Population
KinGroups
Apply

Primary hypothesis: □ Complex hypothesis; Full siblings; Rm 0.5; Rp 0.5

Null hypothesis: ☑ Complex hypothesis; Unrelated, Rm 0.0, Rp 0.0 From; Parent-offspring, Rm 1.0, Rp 0.0 To; Search intervals 1

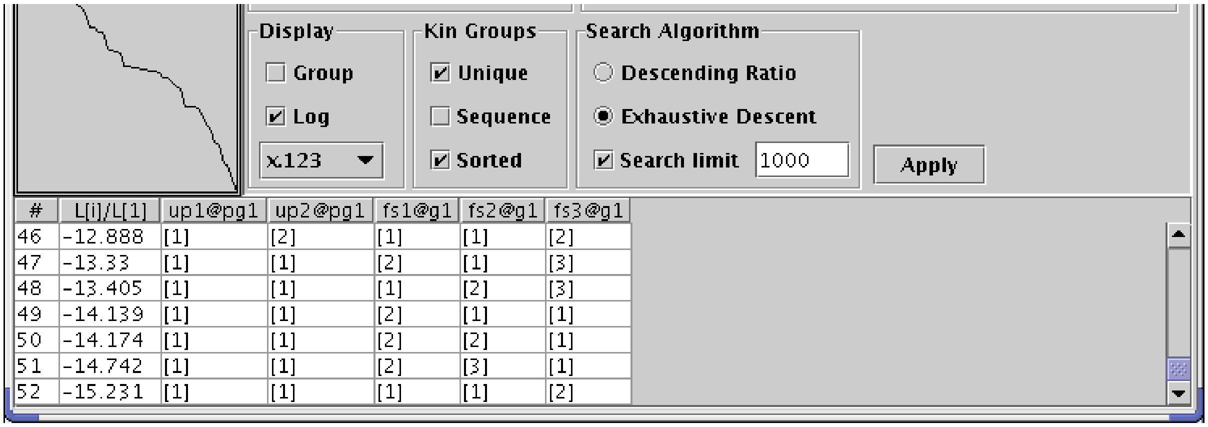
Display: □ Group; ☑ Log; x.123

Kin Groups: ☑ Unique; □ Sequence; ☑ Sorted

Search Algorithm: ○ Descending Ratio; ● Exhaustive Descent; ☑ Search limit 1000

#	L[i]/L[1]	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
1	0	[1]	[2]	[3]	[3]	[3]
2	-2.13	[1]	[1]	[2]	[2]	[2]
3	-3.565	[1]	[2]	[3]	[4]	[3]
4	-3.869	[1]	[2]	[1]	[1]	[1]
5	-4.447	[1]	[2]	[3]	[2]	[3]

Figure 11 Exhaustive Descent results for kingroup_3fs_input.csv



Display: □ Group; ☑ Log; x.123

Kin Groups: ☑ Unique; □ Sequence; ☑ Sorted

Search Algorithm: ○ Descending Ratio; ● Exhaustive Descent; ☑ Search limit 1000

#	L[i]/L[1]	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
46	-12.888	[1]	[2]	[1]	[1]	[2]
47	-13.33	[1]	[1]	[2]	[1]	[3]
48	-13.405	[1]	[1]	[1]	[2]	[3]
49	-14.139	[1]	[1]	[2]	[1]	[1]
50	-14.174	[1]	[1]	[2]	[2]	[1]
51	-14.742	[1]	[1]	[2]	[3]	[1]
52	-15.231	[1]	[1]	[1]	[1]	[2]

Figure 12 Exhaustive Descent results for kingroup_3fs_input.csv with 52 partitions.

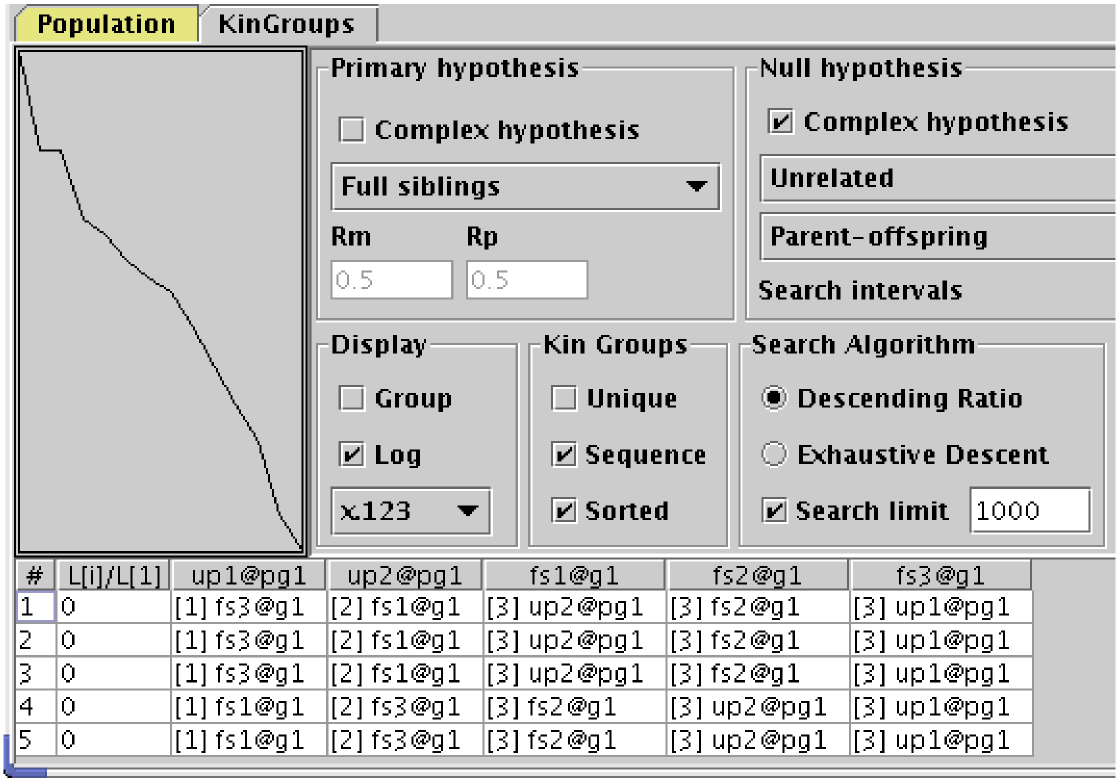
The options that can be used with the Descending Ratio algorithm are illustrated in Figure 13.

<u>The “**Unique**” option:</u> When the highest pairwise ratio is found amongst unassigned individuals the order in which one or the other pair member is taken is random. That variance may yield a different partition path and result in a number of different partitions.

However, such cases mainly occur with populations that do not have sufficient allelic and/or loci information.

The number of attempts is limited to the population size if it is less than ten or ten otherwise. The practical usefulness of a larger number of attempts is questionable so KINGROUP limits it to a maximum of ten. A typical example of when all attempts yield exactly the same kin groups is shown in Figure 13, even though
attempt #1 used the order: “fs1”, “fs3”, “up2”, “fs2”, and “up1”,
while for example
attempt #3 used: “fs3”, “fs1”, “fs2”, “up2”, and “up1”.

By not selecting the “**Unique**” option (and selecting the “**Sequence**” option) it is possible to see which individual is added at each cycle. This combination of options as shown in Figure 13 is useful for testing and experimenting with the algorithm.



Primary hypothesis

□ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☑ Complex hypothesis

Unrelated

Parent-offspring

Search intervals

Display

□ Group

☑ Log

x123

Kin Groups

□ Unique

☑ Sequence

☑ Sorted

Search Algorithm

● Descending Ratio

○ Exhaustive Descent

☑ Search limit 1000

#	L[i]/L[1]	up1@pg1	up2@pg1	fs1@g1	fs2@g1	fs3@g1
1	0	[1] fs3@g1	[2] fs1@g1	[3] up2@pg1	[3] fs2@g1	[3] up1@pg1
2	0	[1] fs3@g1	[2] fs1@g1	[3] up2@pg1	[3] fs2@g1	[3] up1@pg1
3	0	[1] fs3@g1	[2] fs1@g1	[3] up2@pg1	[3] fs2@g1	[3] up1@pg1
4	0	[1] fs1@g1	[2] fs3@g1	[3] fs2@g1	[3] up2@pg1	[3] up1@pg1
5	0	[1] fs1@g1	[2] fs3@g1	[3] fs2@g1	[3] up2@pg1	[3] up1@pg1

Figure 13 Descending Ratio non-unique results for kingroup_3fs_input.csv

The “**Search limit**” option has a slightly different meaning for this algorithm. If selected, KinGroup will try to build alternative partitions by moving each individual into a different group. At this stage it is not clear how to calculate confidence levels for the final partition. The search “around” the local area provides some feel if the found solution stands out amongst possible alternatives or is it only marginally better.

Full Siblings Kin Group Reconstruction

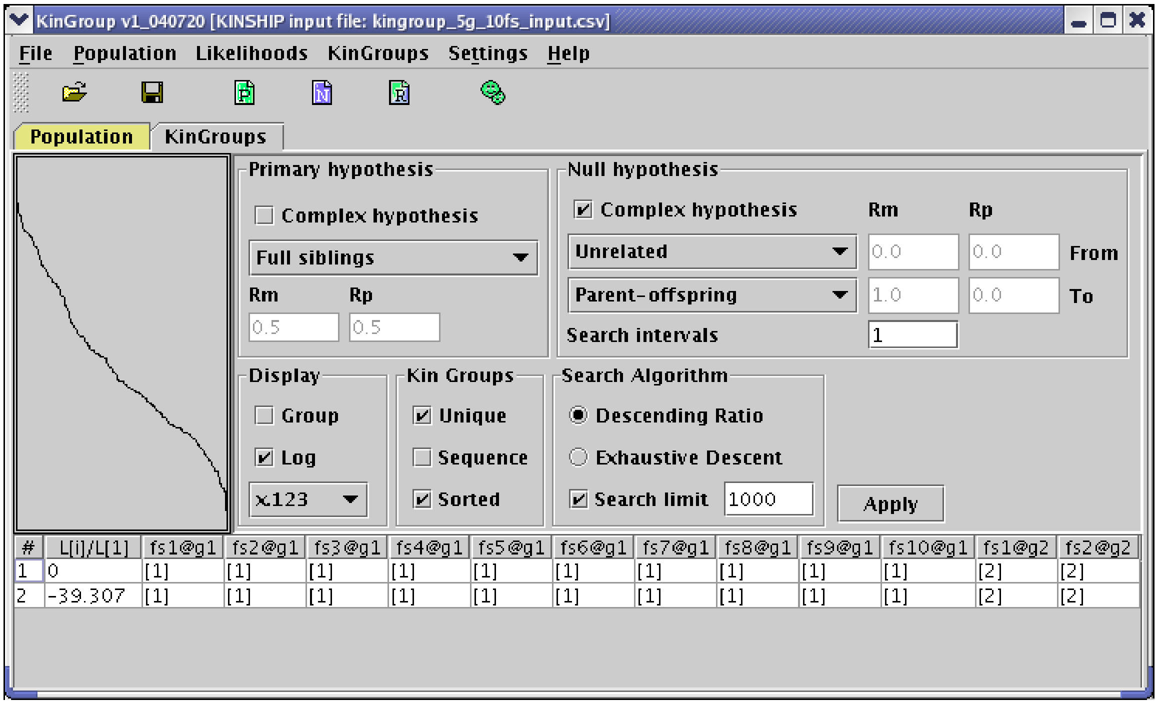
Table 1 summarizes kin group reconstruction results for full-siblings. All results files were generated with the Descending Ration algorithm and the files contain correctly reconstructed kin groups. All files can be found at www.kingroup.org/examples/kingroup_fs/.

Input and Results file	**Description**	**Population size**	**Alleles**	**Loci**
kingroup_2g_3fs_input.csv kingroup_2g_3fs_results.csv	Two groups of 3 full-siblings plus their parents. Ten loci with 10 alleles in each locus. **(E)** - denotes equally frequent alleles.	10	10 **(E)**	10
kingroup_2g_8fs_input.csv kingroup_2g_8fs_results.csv	Two groups of 8 full-siblings plus their parents.	20	10 (E)	10
kingroup_5g_10fs_input.csv kingroup_5g_10fs_results.csv	Five groups of 10 full-siblings, without parents (Figure 14) Because the parents are absent, the correct kin groups could be reconstructed even using the simplest “unrelated” null hypothesis (Figure 15).	50	10 (E)	10
kingroup_10g_10fs_input.csv kingroup_10g_10fs_results.csv	Ten groups of 10 full-siblings plus their parents.	120	20 (E)	20
kingroup_10g_8fs_input.csv kingroup_10g_8fs_results.csv	Ten groups of 8 full-siblings plus their parents. **(R)** - denotes random allelic frequencies.	100	50 **(R)**	20
kingroup_10g_18fs_input.csv kingroup_10g_18fs_results.csv	Ten groups of 18 full-siblings plus their parents.	200	50 (E)	20
kingroup_20g_48fs_input.csv	Twenty groups of 48 full-siblings plus their parents.	1,000	50 (E)	20

Table 1 Examples of input and results for full-siblings reconstruction.

As discussed earlier the Descending Ratio may produce a number of partitions, see Figure 14 and Figure 15. This is a good indication that the discriminative power of the used set of alleles and loci is reaching its limit. This is could be verified by reducing the number of loci to 5 (five) when kingroup_5g_10fs_input.csv is imported. With only 5 loci, the Descending Ratio algorithm makes a number of wrong group assignments.

The current limit on the population size is between 200 and 1,000 individuals depending on user’s computer configuration.


KinGroup v1_040720 [KINSHIP input file: kingroup_5g_10fs_input.csv]
File Population Likelihoods KinGroups Settings Help
Population
KinGroups
Apply

Primary hypothesis
□ Complex hypothesis
Full siblings
Rm 0.5 Rp 0.5

Null hypothesis
☑ Complex hypothesis

	Rm	Rp	
Unrelated	0.0	0.0	From
Parent-offspring	1.0	0.0	To

Search intervals 1

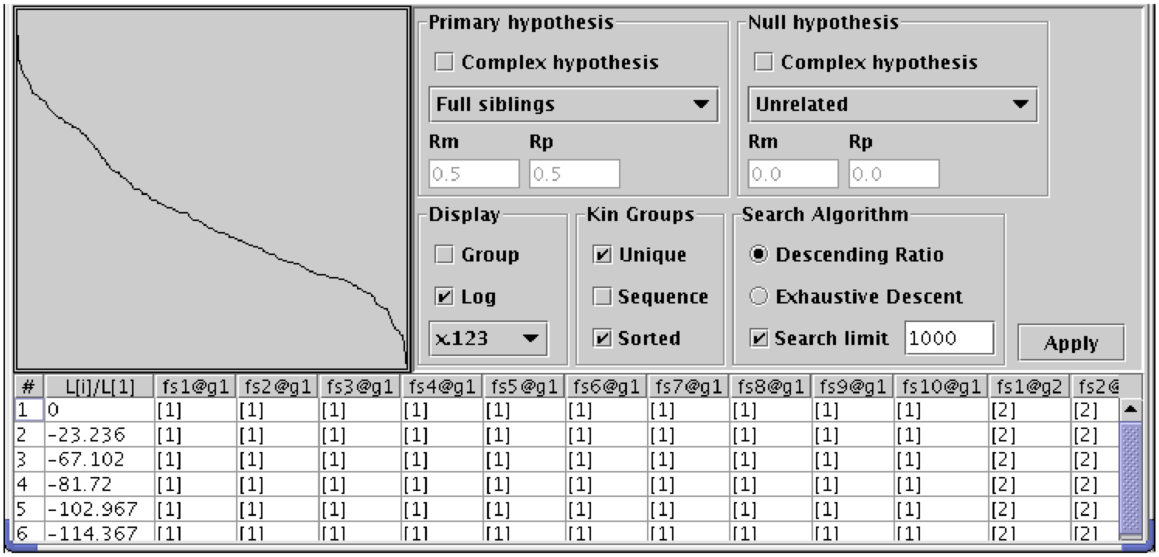
Display
□ Group
☑ Log
x.123

Kin Groups
☑ Unique
□ Sequence
☑ Sorted

Search Algorithm
● Descending Ratio
○ Exhaustive Descent
☑ Search limit 1000

#	L[i]/L[1]	fs1@g1	fs2@g1	fs3@g1	fs4@g1	fs5@g1	fs6@g1	fs7@g1	fs8@g1	fs9@g1	fs10@g1	fs1@g2	fs2@g2
1	0	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
2	-39.307	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]

Figure 14 Descending Ratio results for kingroup_5g_10fs_input.csv



Primary hypothesis
□ Complex hypothesis
Full siblings
Rm 0.5 Rp 0.5

Null hypothesis
□ Complex hypothesis
Unrelated
Rm 0.0 Rp 0.0

Display
□ Group
☑ Log
x.123

Kin Groups
☑ Unique
□ Sequence
☑ Sorted

Search Algorithm
● Descending Ratio
○ Exhaustive Descent
☑ Search limit 1000

#	L[i]/L[1]	fs1@g1	fs2@g1	fs3@g1	fs4@g1	fs5@g1	fs6@g1	fs7@g1	fs8@g1	fs9@g1	fs10@g1	fs1@g2	fs2@
1	0	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
2	-23.236	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
3	-67.102	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
4	-81.72	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
5	-102.967	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]
6	-114.367	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[2]

Figure 15 Descending Ratio with simple null hypothesis for kingroup_5g_10fs_input.csv

Full and Half-Siblings

All files for this section can be found at www.kingroup.org/examples/kingroup_hs/.

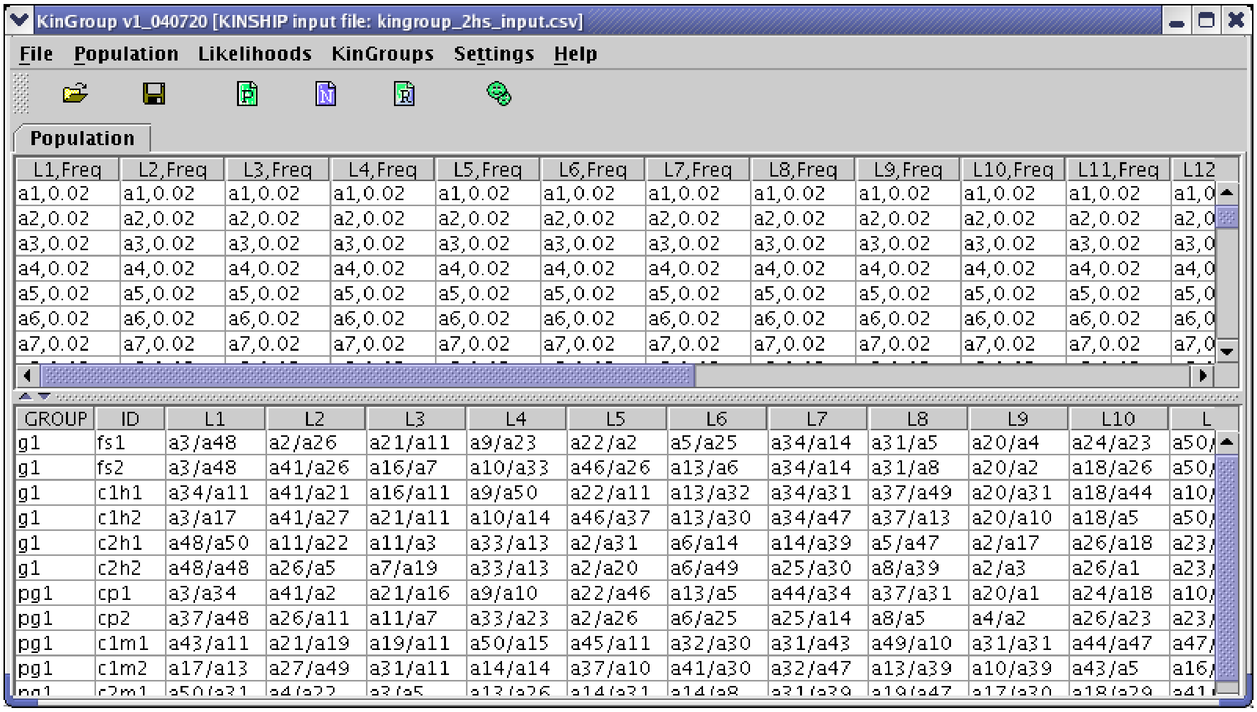
The first example is a population with two full-siblings and 2 half siblings from each of the common parents, see Figure 16.

Abbreviations	Description
_g1	Group #1
_pg1	Parental group #1
_fs#	Full siblings from common parents “cp1” and “cp2”
_cp#	Common parents # 1 and 2.
_c1m#	Mating partners of the common parent “cp1”.
_c1h#	Half-siblings to the “fs” full-siblings. These are the off-springs of common parent “cp1” and the corresponding mate “c1m”. For example, c1h1 is the off-spring of cp1 and c1m1.

Table 2 Abbreviations used in kingroup_2hs_input.csv

Figure 17 and Figure 18 show the results of Descending Ratio with incorrect null hypothesis. The null hypothesis was left the same as in the previous examples dealing only with unrelated and parent-offspring relationships. In this example there are 4 (four) half-siblings (two from each common parent), while the null hypothesis only assumes two possible values: Rm=Rp=0 [unrelated] and Rm=1 Rp=0 [parent -offspring].

The correct null hypothesis for this example should include the half-sibling relationship, which is in terms of Rm and Rp could be expressed as: Rm=0.5, Rp= 0. By setting the number of intervals to 2 (two), see Figure 19, all three relationships are included in the null hypothesis. This version of KinGroup does not distinguish paternal and maternal alleles and therefore Rm and Rp yield the same likelihoods if their values are swapped, e.g. Rm=0.5, Rp=0 is exactly the same as Rm=0, Rp=0.5.

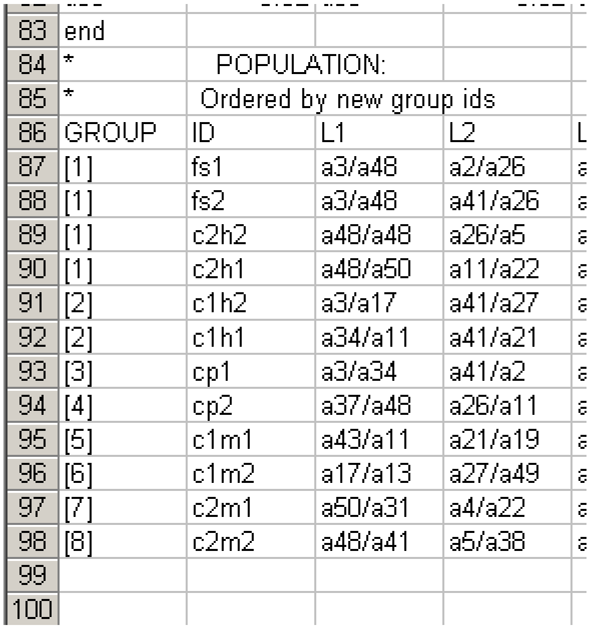


KinGroup v1_040720 [KINSHIP input file: kingroup_2hs_input.csv]
File Population Likelihoods KinGroups Settings Help
Population

L1,Freq	L2,Freq	L3,Freq	L4,Freq	L5,Freq	L6,Freq	L7,Freq	L8,Freq	L9,Freq	L10,Freq	L11,Freq	L12
a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0.02	a1,0
a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0.02	a2,0
a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0.02	a3,0
a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0.02	a4,0
a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0.02	a5,0
a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0.02	a6,0
a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0.02	a7,0

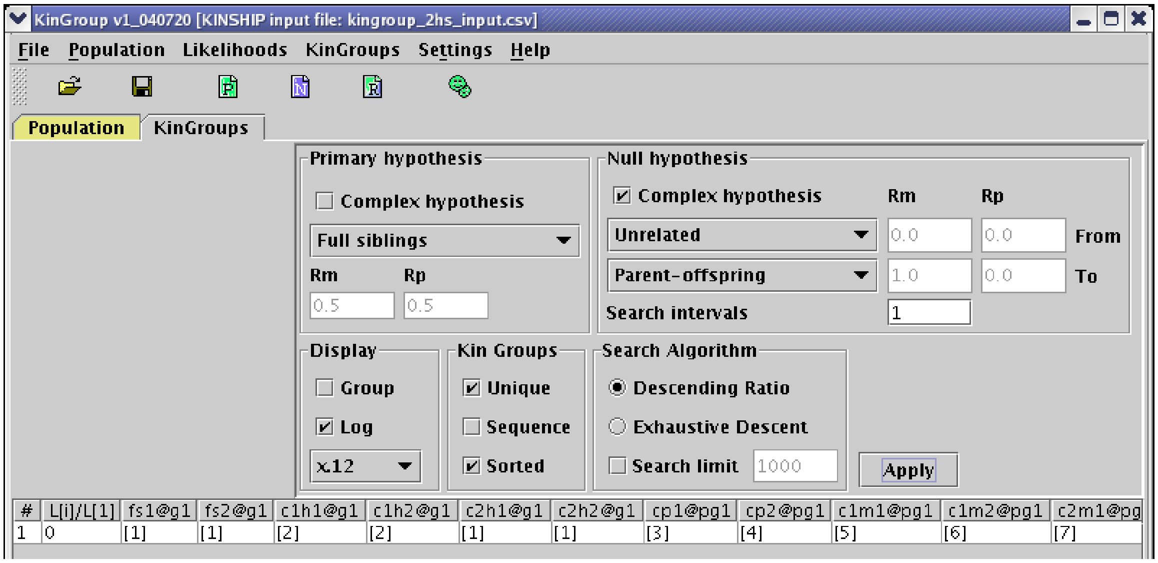
GROUP	ID	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L
g1	fs1	a3/a48	a2/a26	a21/a11	a9/a23	a22/a2	a5/a25	a34/a14	a31/a5	a20/a4	a24/a23	a50/
g1	fs2	a3/a48	a41/a26	a16/a7	a10/a33	a46/a26	a13/a6	a34/a14	a31/a8	a20/a2	a18/a26	a50/
g1	c1h1	a34/a11	a41/a21	a16/a11	a9/a50	a22/a11	a13/a32	a34/a31	a37/a49	a20/a31	a18/a44	a10/
g1	c1h2	a3/a17	a41/a27	a21/a11	a10/a14	a46/a37	a13/a30	a34/a47	a37/a13	a20/a10	a18/a5	a50/
g1	c2h1	a48/a50	a11/a22	a11/a3	a33/a13	a2/a31	a6/a14	a14/a39	a5/a47	a2/a17	a26/a18	a23/
g1	c2h2	a48/a48	a26/a5	a7/a19	a33/a13	a2/a20	a6/a49	a25/a30	a8/a39	a2/a3	a26/a1	a23/
pg1	cp1	a3/a34	a41/a2	a21/a16	a9/a10	a22/a46	a13/a5	a44/a34	a37/a31	a20/a1	a24/a18	a10/
pg1	cp2	a37/a48	a26/a11	a11/a7	a33/a23	a2/a26	a6/a25	a25/a14	a8/a5	a4/a2	a26/a23	a23/
pg1	c1m1	a43/a11	a21/a19	a19/a11	a50/a15	a45/a11	a32/a30	a31/a43	a49/a10	a31/a31	a44/a47	a47/
pg1	c1m2	a17/a13	a27/a49	a31/a11	a14/a14	a37/a10	a41/a30	a32/a47	a13/a39	a10/a39	a43/a5	a16/
pg1	c2m1	a50/a31	a4/a22	a3/a5	a13/a26	a14/a31	a14/a8	a31/a39	a19/a47	a17/a30	a18/a29	a41

Figure 16 Population loaded from kingroup_2hs_input.csv



83	end				
84	*	POPULATION:			
85	*	Ordered by new group ids			
86	GROUP	ID	L1	L2	L
87	[1]	fs1	a3/a48	a2/a26	a
88	[1]	fs2	a3/a48	a41/a26	a
89	[1]	c2h2	a48/a48	a26/a5	a
90	[1]	c2h1	a48/a50	a11/a22	a
91	[2]	c1h2	a3/a17	a41/a27	a
92	[2]	c1h1	a34/a11	a41/a21	a
93	[3]	cp1	a3/a34	a41/a2	a
94	[4]	cp2	a37/a48	a26/a11	a
95	[5]	c1m1	a43/a11	a21/a19	a
96	[6]	c1m2	a17/a13	a27/a49	a
97	[7]	c2m1	a50/a31	a4/a22	a
98	[8]	c2m2	a48/a41	a5/a38	a
99					
100					

Figure 17 Results for kingroup_2hs_input.csv with incorrect null hypothesis.



Primary hypothesis

☐ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☑ Complex hypothesis

	Rm	Rp	
Unrelated	0.0	0.0	From
Parent-offspring	1.0	0.0	To

Search intervals 1

Display

☐ Group

☑ Log

x.12

Kin Groups

☑ Unique

☐ Sequence

☑ Sorted

Search Algorithm

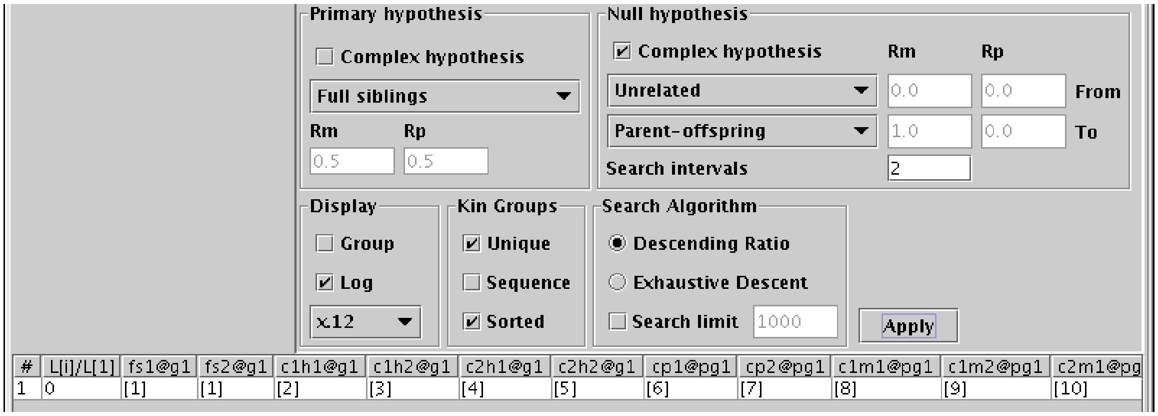
● Descending Ratio

○ Exhaustive Descent

☐ Search limit 1000

#	L[i]/L[1]	fs1@g1	fs2@g1	c1h1@g1	c1h2@g1	c2h1@g1	c2h2@g1	cp1@pg1	cp2@pg1	c1m1@pg1	c1m2@pg1	c2m1@pg
1	0	[1]	[1]	[2]	[2]	[1]	[1]	[3]	[4]	[5]	[6]	[7]

Figure 18 Descending Ratio for kingroup_2hs_input.csv with incorrect null hypothesis.



Primary hypothesis

☐ Complex hypothesis

Full siblings

Rm 0.5 Rp 0.5

Null hypothesis

☑ Complex hypothesis

	Rm	Rp	
Unrelated	0.0	0.0	From
Parent-offspring	1.0	0.0	To

Search intervals 2

Display

☐ Group

☑ Log

x.12

Kin Groups

☑ Unique

☐ Sequence

☑ Sorted

Search Algorithm

● Descending Ratio

○ Exhaustive Descent

☐ Search limit 1000

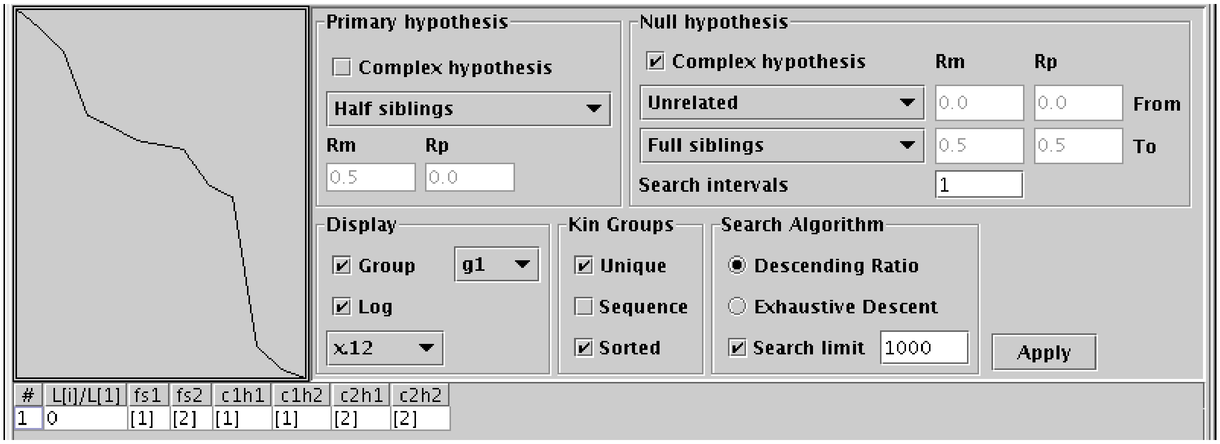
#	L[i]/L[1]	fs1@g1	fs2@g1	c1h1@g1	c1h2@g1	c2h1@g1	c2h2@g1	cp1@pg1	cp2@pg1	c1m1@pg1	c1m2@pg1	c2m1@pg
1	0	[1]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Figure 19 Descending Ratio for kingroup_2hs_input.csv.

Kin Groups

This section illustrates the convenience of using KinGroup for the reconstruction of a wide variety of kin groups using the example of half siblings in Figure 16.

The first question to consider is "**Given a population, which individuals are half-siblings?**"



Primary hypothesis

□ Complex hypothesis

Half siblings ▼

Rm 0.5 Rp 0.0

Null hypothesis

☑ Complex hypothesis

	Rm	Rp	
Unrelated ▼	0.0	0.0	From
Full siblings ▼	0.5	0.5	To

Search intervals 1

Display

☑ Group g1 ▼

☑ Log

x12 ▼

Kin Groups

☑ Unique

□ Sequence

☑ Sorted

Search Algorithm

● Descending Ratio

○ Exhaustive Descent

☑ Search limit 1000

Apply

#	L[i]/L[1]	fs1	fs2	c1h1	c1h2	c2h1	c2h2
1	0	[1]	[2]	[1]	[1]	[2]	[2]

Figure 20 Half-sibling kin groups in group g1 of kingroup_2hs_input.csv

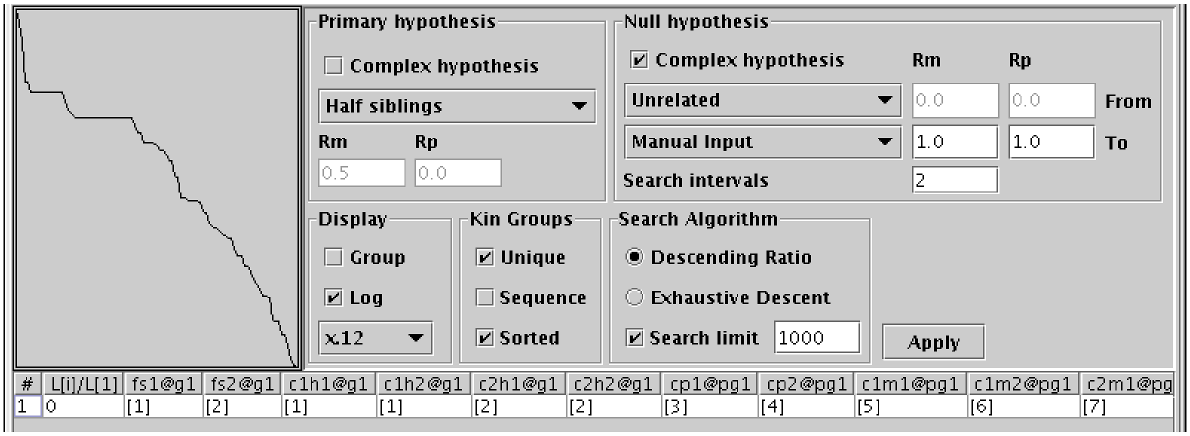
The half-sibling kin groups within the g1-group are shown in Figure 20. This is a simplified example in which we know that the g1-group contains only half-siblings and full-siblings. The null hypothesis was set to cover only “unrelated” and “full-siblings”. Note that the “half-siblings” hypothesis is also a part of the null hypothesis but KinGroup excludes it from the choices given by the null hypothesis. This is very convenient since otherwise it could be impossible to devise a suitable null hypothesis (using the current options) without overlapping with the primary values. In this version, KinGroup only excludes the simple component of the primary hypothesis from the choices generated by a complex **null** hypothesis.

Mates of cp1				Mates of cp2
	_cp1		_cp2	
_c1m1	_c1h1	_fs1	_c2h1	_c2m1
_c1m2	_c1h2	_fs2	_c2h2	_c2m2

Table 3 Half-siblings kin groups in kingroup_2hs_input.csv

Table 3 illustrates the correct half-siblings kin groups found in Figure 20. In particular, “c1h1”, “c1h2” and “fs1” have the same common parent “cp1” but a different second parent.

The correctly identified half-sibling kin groups are shown in Figure 21 with Figure 22 demonstrating the importance of specifying the null hypothesis. Not including the “full-siblings” relationship in the null hypothesis yields incorrect kin groups (Figure 22).



Primary hypothesis
□ Complex hypothesis
Half siblings
Rm 0.5 Rp 0.0

Null hypothesis
☑ Complex hypothesis
Unrelated Rm 0.0 Rp 0.0 From
Manual Input Rm 1.0 Rp 1.0 To
Search intervals 2

Display
□ Group
☑ Log
x.12

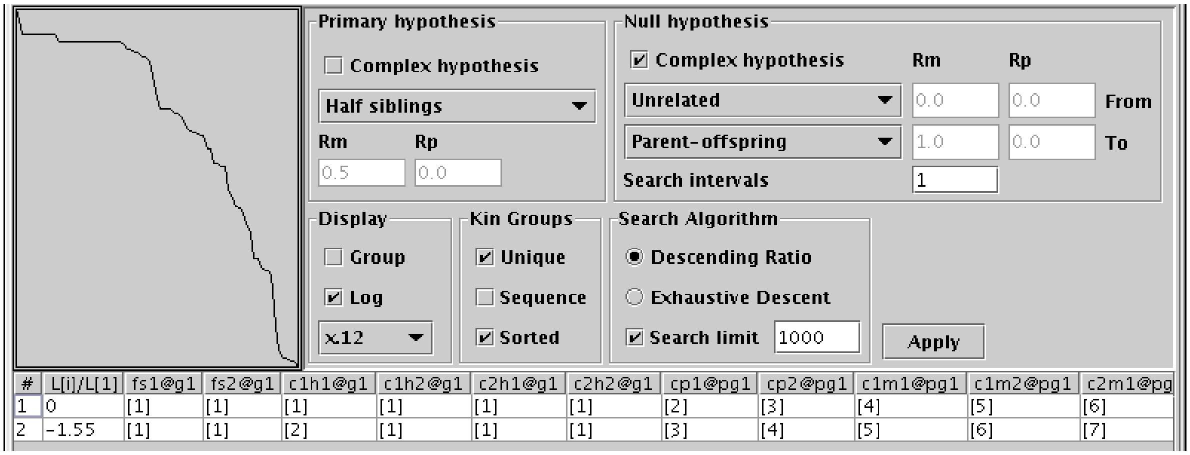
Kin Groups
☑ Unique
□ Sequence
☑ Sorted

Search Algorithm
● Descending Ratio
○ Exhaustive Descent
☑ Search limit 1000

Apply

#	L[i]/L[1]	fs1@g1	fs2@g1	c1h1@g1	c1h2@g1	c2h1@g1	c2h2@g1	cp1@pg1	cp2@pg1	c1m1@pg1	c1m2@pg1	c2m1@pg
1	0	[1]	[2]	[1]	[1]	[2]	[2]	[3]	[4]	[5]	[6]	[7]

Figure 21 Half-siblings kin groups in kingroup_2hs_input.csv



Primary hypothesis
□ Complex hypothesis
Half siblings
Rm 0.5 Rp 0.0

Null hypothesis
☑ Complex hypothesis
Unrelated Rm 0.0 Rp 0.0 From
Parent-offspring Rm 1.0 Rp 0.0 To
Search intervals 1

Display
□ Group
☑ Log
x.12

Kin Groups
☑ Unique
□ Sequence
☑ Sorted

Search Algorithm
● Descending Ratio
○ Exhaustive Descent
☑ Search limit 1000

Apply

#	L[i]/L[1]	fs1@g1	fs2@g1	c1h1@g1	c1h2@g1	c2h1@g1	c2h2@g1	cp1@pg1	cp2@pg1	c1m1@pg1	c1m2@pg1	c2m1@pg
1	0	[1]	[1]	[1]	[1]	[1]	[1]	[2]	[3]	[4]	[5]	[6]
2	-1.55	[1]	[1]	[2]	[1]	[1]	[1]	[3]	[4]	[5]	[6]	[7]

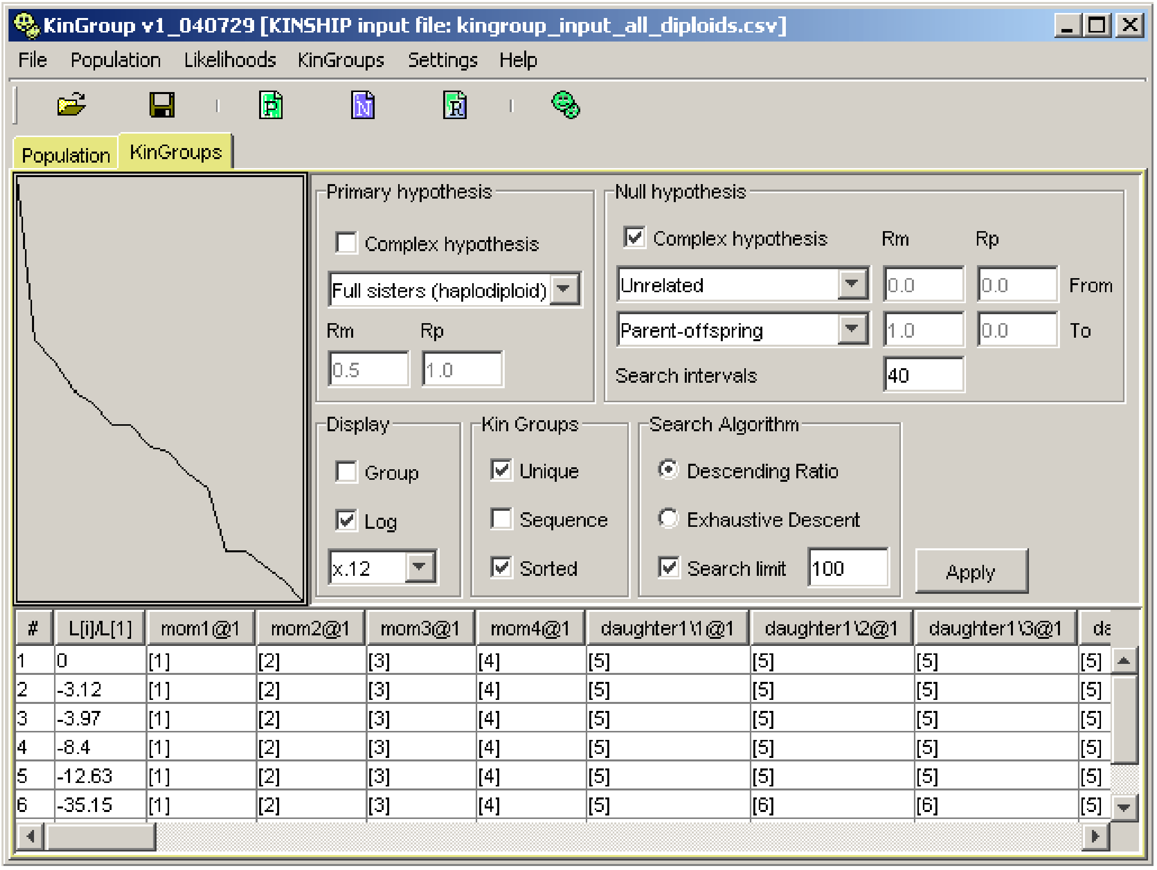
Figure 22 Incorrect null hypothesis for half-siblings kin groups

Haplodiploids

Input file examples/kingroup_hd/kingroup_input.csv contains 4 unrelated mothers (mom1 to mom4), their mates (dad1 to dad4), 8 daughters from each mother (e.g., daughter1/1 to daughter1/8 are the offspring of mom1 + dad1), 3 males mated to 3 of the daughters of the first mother (mate1 to 3), and 3 sets of 8 daughters from these second matings (e.g., daughter1/1/1 to daughter 1/1/8 are the offspring of daughter1/1 + mate1).

The original input file is used to construct an "easy" data set where there are 4 sets of unrelated daughters and their mothers (file kingroup_input_fs_with_parents.csv). A "harder" data set (with 3 sets of related sister groups and their mothers) is stored in kingroup_input_all_diploids.csv.

In order to partition the "harder" population, null hypothesis was chosen (Figure 23) to cover entire spectrum of relationships by setting the "Search intervals" to 40 (with the corresponding step 0.025). Such choice of null hypotheses includes haplodiploid cousins (Rm=0.375, Rp=0). The Descending Ratio was able to partition the population of 60 individuals into the correct set of kin groups, see kingroup_results_all_diploids.csv.



KinGroup v1_040729 [KINSHIP input file: kingroup_input_all_diploids.csv]

Primary hypothesis
- □ Complex hypothesis
- Full sisters (haplodiploid)
- Rm 0.5
- Rp 1.0

Null hypothesis
- ☑ Complex hypothesis
- From: Unrelated, Rm 0.0, Rp 0.0
- To: Parent-offspring, Rm 1.0, Rp 0.0
- Search intervals 40

Display
- □ Group
- ☑ Log
- x.12

Kin Groups
- ☑ Unique
- □ Sequence
- ☑ Sorted

Search Algorithm
- ● Descending Ratio
- ○ Exhaustive Descent
- ☑ Search limit 100

#	L[i]/L[1]	mom1@1	mom2@1	mom3@1	mom4@1	daughter1\1@1	daughter1\2@1	daughter1\3@1	da
1	0	[1]	[2]	[3]	[4]	[5]	[5]	[5]	[5]
2	-3.12	[1]	[2]	[3]	[4]	[5]	[5]	[5]	[5]
3	-3.97	[1]	[2]	[3]	[4]	[5]	[5]	[5]	[5]
4	-8.4	[1]	[2]	[3]	[4]	[5]	[5]	[5]	[5]
5	-12.63	[1]	[2]	[3]	[4]	[5]	[5]	[5]	[5]
6	-35.15	[1]	[2]	[3]	[4]	[5]	[6]	[6]	[5]

Figure 23 Correct partitioning achieved by the Descending Ratio algorithm on the population from examples/kingroup_hd/kingroup_input_all_diploids.csv

Population Generation

Sample Input Files

All sample input files are located at www.kingroup.org/examples or in the [location on your computer where you installed KinGroup]/kingroup/examples - directory. This version of KinGroup can only read files in the KINSHIP format.

To import a kinship file, select “Menu | Population | Import Kinship File …”, see Figure 24.

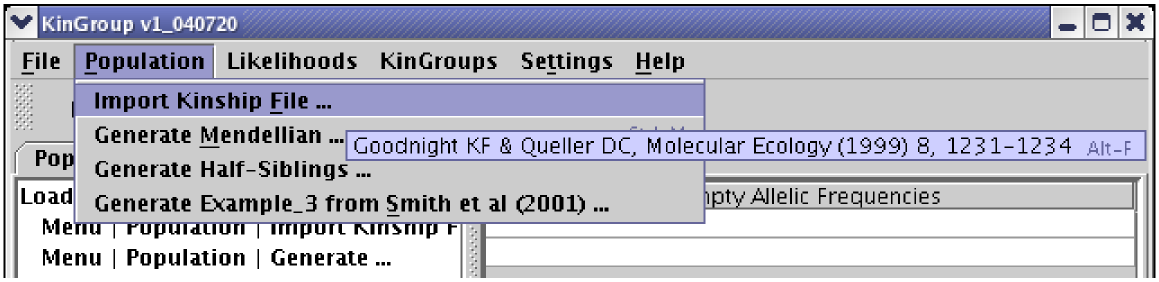
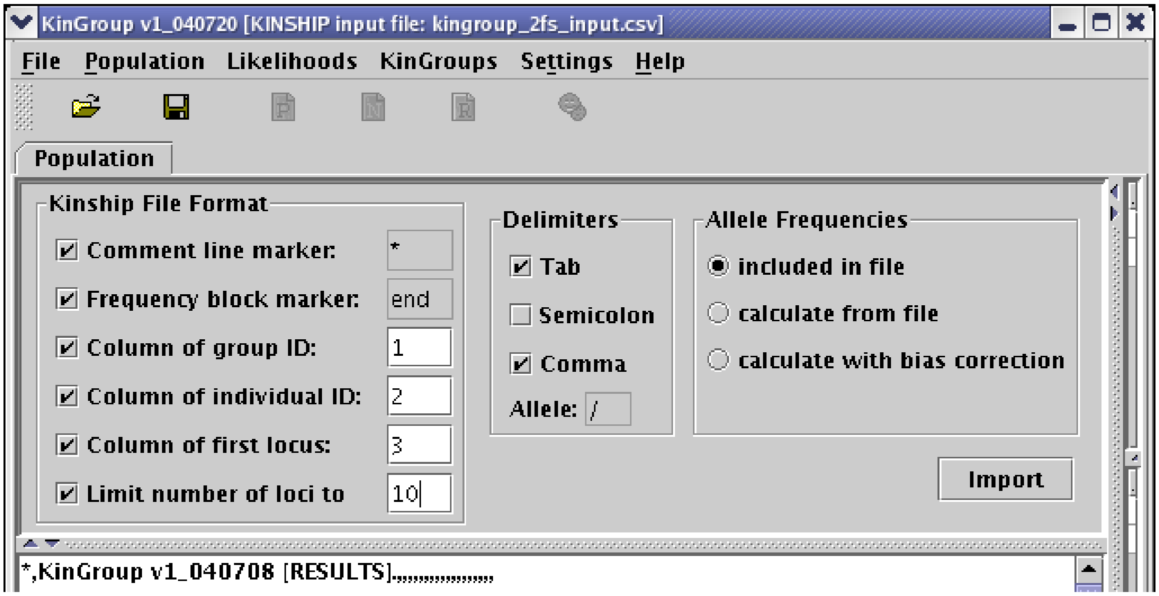

KinGroup v1_040720
File
Population
Likelihoods
KinGroups
Settings
Help
Import Kinship File ...
Generate Mendellian ...
Generate Half-Siblings ...
Generate Example_3 from Smith et al (2001) ...
Goodnight KF & Queller DC, Molecular Ecology (1999) 8, 1231-1234 Alt-F
Pop
Load
Menu | Population | Import Kinship F
Menu | Population | Generate ...
npty Allelic Frequencies

Figure 24 How to import a KINSHIP file.

Directory: kingroup_fs

Location: www.kingroup.org/examples/kingroup_fs/. This directory contains a number of sample populations containing full siblings. Each of the populations were generated using KinGroup via “Menu | Population | Generate Mendellian …”, see the “Population Generation” section on page 26 for more details.


KinGroup v1_040720 [KINSHIP input file: kingroup_2fs_input.csv]
File Population Likelihoods KinGroups Settings Help
Population
Import

Kinship File Format

- ☑ Comment line marker: *
- ☑ Frequency block marker: end
- ☑ Column of group ID: 1
- ☑ Column of individual ID: 2
- ☑ Column of first locus: 3
- ☑ Limit number of loci to 10

Delimiters

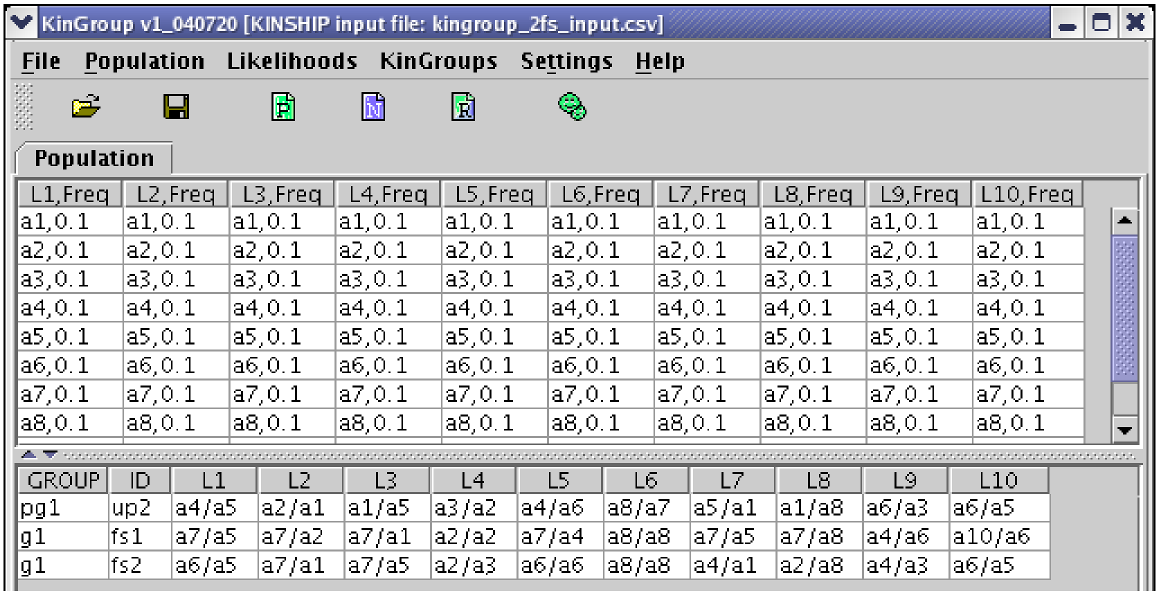
- ☑ Tab
- □ Semicolon
- ☑ Comma
- Allele: /

Allele Frequencies

- ● included in file
- ○ calculate from file
- ○ calculate with bias correction

*,KinGroup v1_040708 [RESULTS],,,,,,,,,,,,,,,,,,,

Figure 25 How to load kingroup_2fs_input.csv


KinGroup v1_040720 [KINSHIP input file: kingroup_2fs_input.csv]
File Population Likelihoods KinGroups Settings Help
Population

L1,Freq	L2,Freq	L3,Freq	L4,Freq	L5,Freq	L6,Freq	L7,Freq	L8,Freq	L9,Freq	L10,Freq
a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1
a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1
a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1
a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1
a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1
a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1
a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1
a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1

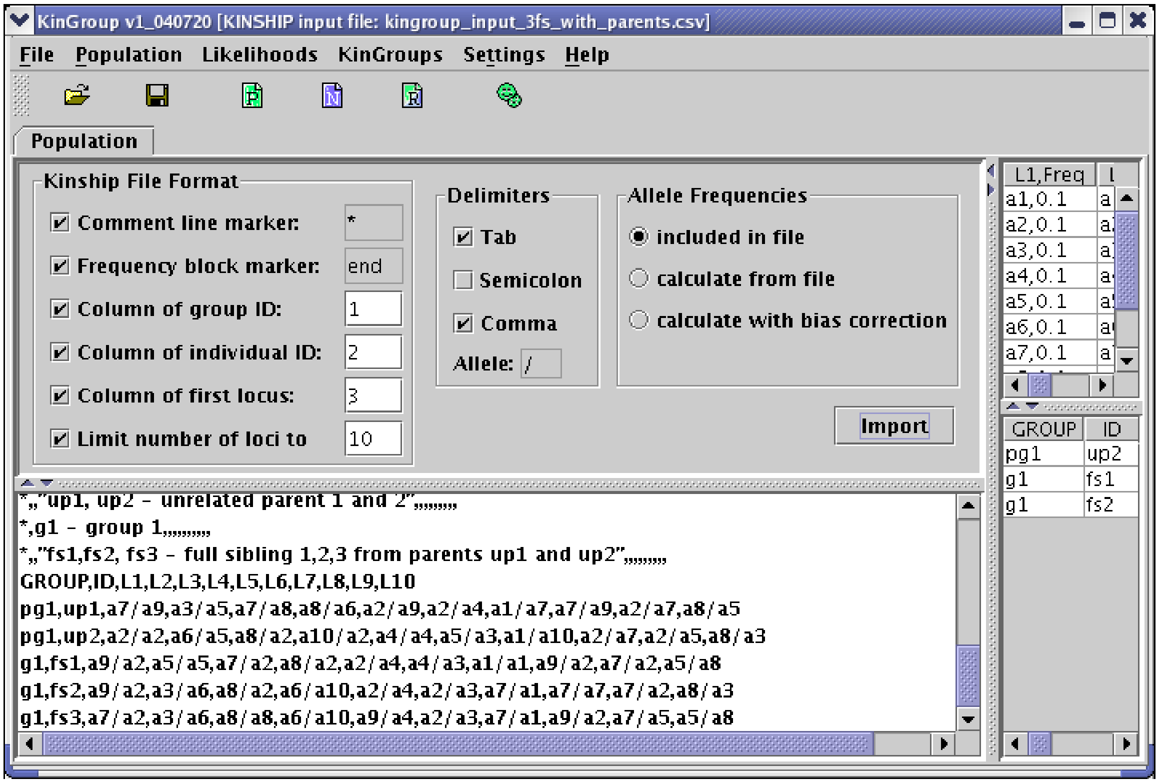
GROUP	ID	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
pg1	up2	a4/a5	a2/a1	a1/a5	a3/a2	a4/a6	a8/a7	a5/a1	a1/a8	a6/a3	a6/a5
g1	fs1	a7/a5	a7/a2	a7/a1	a2/a2	a7/a4	a8/a8	a7/a5	a7/a8	a4/a6	a10/a6
g1	fs2	a6/a5	a7/a1	a7/a5	a2/a3	a6/a6	a8/a8	a4/a1	a2/a8	a4/a3	a6/a5

Figure 26 Imported kingroup_2fs_input.csv

Kinship Complex

File location: www.kingroup.org/examples/kinship_complex/input_3fs_with_parents.csv

This input file contains three full siblings (hence the abbreviation _*3fs*_) plus two unrelated parents. Figure 27 shows the choice of file format options that are required to import this sample file successfully. Figure 28 shows the result of successful import.


KinGroup v1_040720 [KINSHIP input file: kingroup_input_3fs_with_parents.csv]
File
Population
Likelihoods
KinGroups
Settings
Help
Population
Import

Kinship File Format

☑ Comment line marker: *

☑ Frequency block marker: end

☑ Column of group ID: 1

☑ Column of individual ID: 2

☑ Column of first locus: 3

☑ Limit number of loci to 10

Delimiters

☑ Tab

□ Semicolon

☑ Comma

Allele: /

Allele Frequencies

● included in file

○ calculate from file

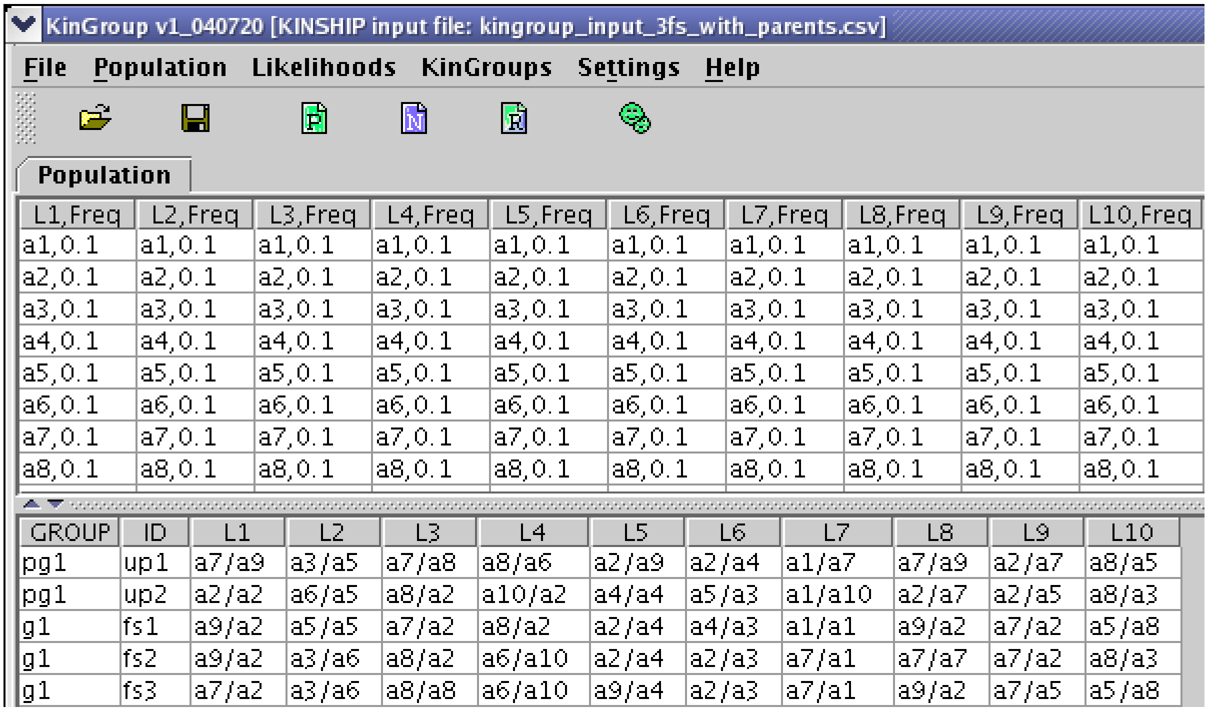
○ calculate with bias correction

```
*,,"up1, up2 - unrelated parent 1 and 2",,,,,,,,
*,g1 - group 1,,,,,,,,,
*,,"fs1,fs2, fs3 - full sibling 1,2,3 from parents up1 and up2",,,,,,,,
GROUP,ID,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10
pg1,up1,a7/a9,a3/a5,a7/a8,a8/a6,a2/a9,a2/a4,a1/a7,a7/a9,a2/a7,a8/a5
pg1,up2,a2/a2,a6/a5,a8/a2,a10/a2,a4/a4,a5/a3,a1/a10,a2/a7,a2/a5,a8/a3
g1,fs1,a9/a2,a5/a5,a7/a2,a8/a2,a2/a4,a4/a3,a1/a1,a9/a2,a7/a2,a5/a8
g1,fs2,a9/a2,a3/a6,a8/a2,a6/a10,a2/a4,a2/a3,a7/a1,a7/a7,a7/a2,a8/a3
g1,fs3,a7/a2,a3/a6,a8/a8,a6/a10,a9/a4,a2/a3,a7/a1,a9/a2,a7/a5,a5/a8
```

L1,Freq	L
a1,0.1	a
a2,0.1	a
a3,0.1	a
a4,0.1	a
a5,0.1	a
a6,0.1	a
a7,0.1	a

GROUP	ID
pg1	up2
g1	fs1
g1	fs2

Figure 27 File format options for input_3fs_with_parents.csv


KinGroup v1_040720 [KINSHIP input file: kingroup_input_3fs_with_parents.csv]
File
Population
Likelihoods
KinGroups
Settings
Help
P
N
R
Population

L1,Freq	L2,Freq	L3,Freq	L4,Freq	L5,Freq	L6,Freq	L7,Freq	L8,Freq	L9,Freq	L10,Freq
a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1	a1,0.1
a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1	a2,0.1
a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1	a3,0.1
a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1	a4,0.1
a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1	a5,0.1
a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1	a6,0.1
a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1	a7,0.1
a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1	a8,0.1

GROUP	ID	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
pg1	up1	a7/a9	a3/a5	a7/a8	a8/a6	a2/a9	a2/a4	a1/a7	a7/a9	a2/a7	a8/a5
pg1	up2	a2/a2	a6/a5	a8/a2	a10/a2	a4/a4	a5/a3	a1/a10	a2/a7	a2/a5	a8/a3
g1	fs1	a9/a2	a5/a5	a7/a2	a8/a2	a2/a4	a4/a3	a1/a1	a9/a2	a7/a2	a5/a8
g1	fs2	a9/a2	a3/a6	a8/a2	a6/a10	a2/a4	a2/a3	a7/a1	a7/a7	a7/a2	a8/a3
g1	fs3	a7/a2	a3/a6	a8/a8	a6/a10	a9/a4	a2/a3	a7/a1	a9/a2	a7/a5	a5/a8

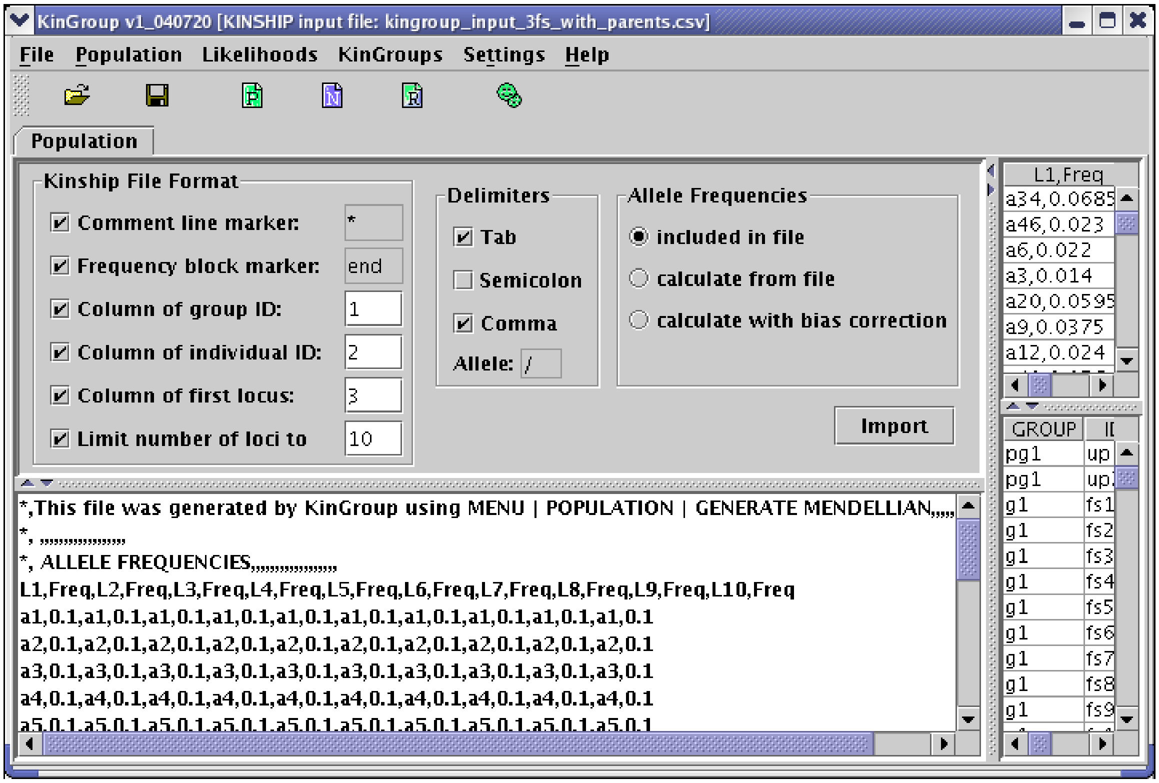
Figure 28 Loaded input_3fs_with_parents.csv

KINSHIP File Format

It is recommended to present your input files in comma-separated values (.csv) format. Such format is recognized by Excel and it is easy to work with.

The rest of this section is based on, and extensively adopted from, the KINSHIP manual :

> "… The basic organization of the Kinship files is that (in spreadsheet format) each individual occupies one row and each variable one column. The file can contain comment lines which begin with the character * (asterisk); KinGroup will ignore such lines completely. In addition, missing information can be simply left blank in the TEXT file.
>
> A data file consists of two main sections. Both of them are optional (although at least one of the two must be present or else the file is simply empty):
>
> **Allele frequency block:** When an allele frequency block is present in the data file (Figure 29), the user may select to use these frequencies exclusively for performing all calculations. When no allele frequency block is present, the user must select how the allele frequencies should be calculated based on the individual data in the file. The first available option in such case is just to calculate the frequencies each allele occurs in a given locus. The second option is to use bias correction, see more on that later. …"- *adopted from the Kinship Manual* .



KinGroup v1_040720 [KINSHIP input file: kingroup_input_3fs_with_parents.csv]
File
Population
Likelihoods
KinGroups
Settings
Help
Population
Import

Kinship File Format

- ☑ Comment line marker: *
- ☑ Frequency block marker: end
- ☑ Column of group ID: 1
- ☑ Column of individual ID: 2
- ☑ Column of first locus: 3
- ☑ Limit number of loci to 10

Delimiters

- ☑ Tab
- □ Semicolon
- ☑ Comma
- Allele: /

Allele Frequencies

- ● included in file
- ○ calculate from file
- ○ calculate with bias correction

```
*,This file was generated by KinGroup using MENU | POPULATION | GENERATE MENDELLIAN,,,,,
*, ,,,,,,,,,,,,,,,,,
*, ALLELE FREQUENCIES,,,,,,,,,,,,,,,,,
L1,Freq,L2,Freq,L3,Freq,L4,Freq,L5,Freq,L6,Freq,L7,Freq,L8,Freq,L9,Freq,L10,Freq
a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1,a1,0.1
a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1,a2,0.1
a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1,a3,0.1
a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1,a4,0.1
a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1,a5,0.1
```

L1,Freq
a34,0.0685
a46,0.023
a6,0.022
a3,0.014
a20,0.0595
a9,0.0375
a12,0.024

GROUP	ID
pg1	up
pg1	up
g1	fs1
g1	fs2
g1	fs3
g1	fs4
g1	fs5
g1	fs6
g1	fs7
g1	fs8
g1	fs9

Figure 29 Example of importing a KINSHIP file.

Since the program does not require it, you will want to include a frequency block primarily when you have additional information on allele frequencies beyond that available in the current data set (for example, if the file is only one of several data sets which have been genotyped in the population of interest). You may also want to include a frequency block if you want to include a group ID variable but override the bias-correction to allele frequencies (see "Allele frequencies and group ID" below).
NOTE: When both and allele frequency block and an individual genotype block are present in a file, the allele frequency block must come first.

The data must be arranged with each locus occupying two columns: the first for the names of the alleles and the second for the frequency of that allele. The first locus must occupy columns 1 and 2, the second 3 and 4, and so on, see an example in Figure 30. The loci should read across, right to left, in the same order as they will appear in the individual genotype data.

The first non-comment line of the block must be a list of locus names, in the allele-names column for each locus. Starting with the second line, the information for allele frequencies begins.
The allele frequency block continues for as many lines as there are alleles in the most poly-allelic locus. Because the program has no *a priori* way of knowing how many alleles it will

find, you must tell it when the allele frequency block is ending by including the word “**end**” on a line by itself at the end of the block.
The number of loci, and the number of alleles at each locus, is limited only by the available memory of the computer.

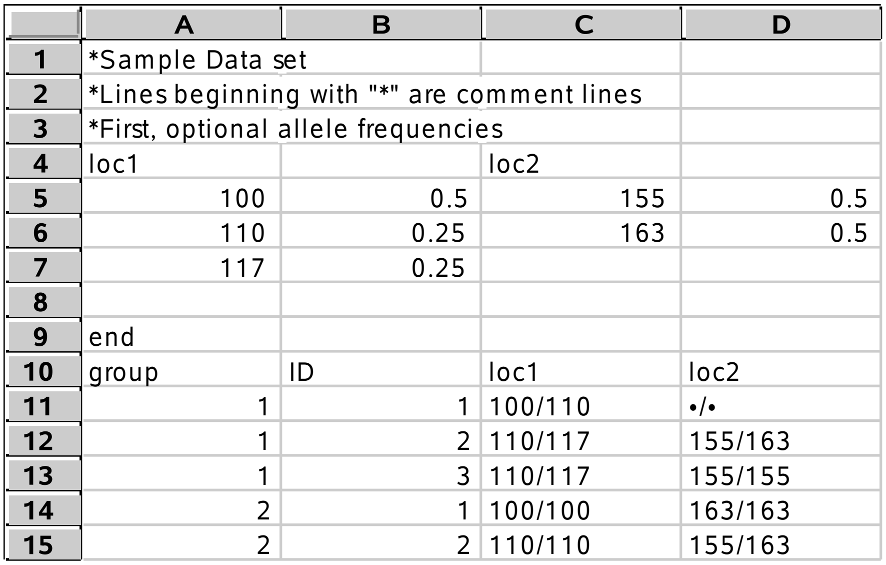
Individual genotypes block: This version of KinGroup always requires this section. If however all you interested in is simulation based on the allele frequencies, you need to include a couple of individuals so that KinGroup has something to load (see the “Simulations” section below for information about the simulation function).
NOTE: When both an allele frequency block and an individual genotype block are present in the file, the individual genotype block must come second, see example in Figure 30.

The first non-comment line of the individual genotype block must give the variable names of each column.
KinGroup reads the following variables: group ID, individual ID, and genotype. **Please note that this version of KinGroup does not have a facility to read Mother’s ID and Father’s ID.** Additional variables may be present in the file, although the program will not read them. These variables can occupy any column in any order, with the only restrictions being: 1) That loci occupy a set of consecutive columns, and that they read left to right in the same order as they appear in the allele frequency block (if one is present).
Missing information for any variable can be indicated with the • character (option-8) on the Macintosh keyboard, or may simply be left blank. The • character cannot be used as an actual value for any variable: the program will always treat it as indicating missing information. All other characters both alphabetic and numeric can be used. For alphabetic characters, case is significant (i.e. upper and lower case are treated as different).
Each locus should occupy one column. For diploid genotypes, both allele names are placed in the same column, separated by a delimiter character (which the user can specify).

EXAMPLE:
If the delimiter character is “/” and two alleles are named “100” and “110”, then valid diploid genotype entries could be:

```
100/110           100/100           110/100           etc.
A haploid genotype could be written in any of the following ways:
100/•       100/        100         •/100 etc.
Likewise, a missing genotype could be any of the following:
•/•         /           <blank>
```

The end of the individual genotype block is marked by the end of the file; no special designator is required for it.
Figure 30 shows a portion of a Kinship data file as displayed in Microsoft Excel. The KinGroup distribution package also contains a variety of sample data files, which you can examine to see the file format.



	A	B	C	D
1	*Sample Data set			
2	*Lines beginning with "*" are comment lines			
3	*First, optional allele frequencies			
4	loc1		loc2	
5	100	0.5	155	0.5
6	110	0.25	163	0.5
7	117	0.25		
8				
9	end			
10	group	ID	loc1	loc2
11	1	1	100/110	•/•
12	1	2	110/117	155/163
13	1	3	110/117	155/155
14	2	1	100/100	163/163
15	2	2	110/110	155/163

Figure 30 Part of a Kinship set, showing both allele frequency and individual genotype blocks

KINSHIP File Format options in Figure 29 are (from top to bottom):

Comment line marker: This version of Kingroup does not allow changing this marker, which is preset to "*" – asterisk. Any line starting with this marker is ignored during the import of a KINSHIP file. User may use such lines for commenting of the data. Note that this option cannot be unchecked.

Frequency block marker: This version of Kingroup does not allow changing this marker, which is preset to "end". During the import of a KINSHIP file the program checks for this marker. If encountered, the program tries to proceed assuming that the information above that marker was in fact the allele frequency block. Note that this option cannot be unchecked.

Column of group ID: The primary function of the group ID, if present, is to specify sets of putative relatives for use in applying a bias correction to the allele frequency calculations (see below). When present, this variable also allows you to split the data file into groups and optionally to perform the likelihood comparisons only among pairs in the same group. If there is no group variable, the program will always perform comparisons among all pairs in the file. (This remains an option even when Group ID is present.)

Column of individual ID: When present, this variable allows the program to label the rows and columns of the output matrix with the ID of the individual each row and column represents. Without this variable, the program will label them to some defaults. While this might be acceptable if the only question is (for example) whether all individuals in a group are full siblings of one another, normally you will want to use this variable.

Column of first locus: This is the first in the set of consecutive columns that give the genotypes at each locus. Remember that all loci must occupy a consecutive set of columns and must read from left-to-right in the same order as they appear in the allele frequencies block, if present, although the set may begin with any column. Note that this option cannot be unchecked.

Limit number of loci to: This box indicates how many columns of locus information the program will read. It need not be the same as the full number present in the file: you can instruct the program, for example, to read only the first 3 out of 6 loci listed in the file.

Delimiters options in Figure 29 are (from top to bottom):

Tab, Semicolon, Comma: These are recognizable column delimiters. Please note that KinGroup will replace tab delimiters with commas before showing the content of the opened file, see the pane under the formatting options in Figure 29.

Allele delimiter character: The character in this box is the one the program will use to separate the alleles in individual genotypes. The default is "/".

Allele Frequencies options in Figure 29 are (from top to bottom):

Included in file: Check this box if your data file contains an allele frequencies block and you wish to use the frequencies. KinGroup also supports the option of ignoring supplied frequencies if one of the calculating options is chosen.

Calculate from file: Regardless of frequency block presence, the program will calculate the frequencies based on how often a particular allele occurs in a locus across entire loaded population.

Calculate with bias-correction: Regardless of frequency block presence, allele frequencies are calculated for a group in such a way that only allele occurrence outside the group contributes to the corresponding frequency.

During the bias-correction procedure, some allelic frequencies could become zero for a particular group. That could happen when corresponding alleles only occur in that group but not in the rest of the population. Such cases are deemed to be extreme and in this version **KinGroup does not make any further adjustments to such frequencies leaving them as zeros**.

As discussed in , when performing relatedness calculations using population allele frequencies obtained from the same data set as the individuals being measured, a bias correction must be applied to those frequencies. The same consideration applies to the likelihood calculations performed by KinGroup. Basically, any individuals who by hypothesis are relatives of the individual(s) under consideration will be expected to have allele frequencies closer to those individuals than the true population mean. Their inclusion in the limited sample of a data set thus biases its measure of population frequencies in that direction.
The solution is to exclude from background frequency calculations all individuals who might be relatives of the current individual.

KinGroup uses a "Group ID" variable to identify sets of such individuals. As it performs calculations on a given pair of individuals, the program will use allele frequencies obtained from the data set excluding the group to which the current pair belongs.

Please note that this version of KinGroup only supports bias-corrected allele frequencies for a population with multiple groups and only for pair-wise calculations within groups.

Index

References