

FAMD (Fingerprint Analysis with Missing Data) 1.02 Help

GENERAL	2
Purpose	2
Author	2
Citation	2
Declaration	2
System requirements	2
Source Code and compilation	2
Known issues	3
HOW TO USE	4
DEFAULTS	5
Default settings	5
Default input format	5
PROGRAM FUNCTIONS	6
File Menu	6
Load	7
Save DataMatrix	7
Select Analysis Output File	7
Select Tree Output File	7
Exit	8
Datamatrix Menu	9
Matrix Statistics	9
Restore Original Matrix	9
Frequency Statistics	9
Missing Data Statistics	9
Missing Data Replacement Settings	9
Replace Missing Data	10
Remove Individuals	10
With missing data...	10
Remove Loci	10
With missing data...	10
Monomorphic...	10
Analysis Menu	11
Jaccard (standard)	11
Minimum Jaccard	11
Maximum Jaccard	11
Average Jaccard	12
Shannon's index	12
Shannon Scaling...	13
Replicate analysis	14
Replicate analysis settings	14
Bootstrap Shannon's variance (SH+RMD)	14
Bootstrap Std. Jaccard Tree (BS+RMD)	15
Multiple Avg. Jacc. Trees (TR+JC)	16
Bootstrap Avg. Jacc. Trees (BS+JC)	16
Trees Menu	17
UPGMA (multifurcating)	18
Help Menu	18
About	18
Help	18

GENERAL

Purpose

This program was written for analysis of RAPD, AFLP or other fingerprint data, especially for data sets that contain ambiguous or missing data, so that the impact that missing data may have on the analysis can be better evaluated. The second intention was to provide a means to easily calculate the variance associated with Shannon's index.

Author

Philipp M Schlüter

Department of Higher Plant Systematics and Evolution
Institute of Botany
University of Vienna
Rennweg 14
A-1030 Vienna
Austria

Telephone: +43-1-4277-54149

E-mail: <philipp.maria.schlueter@univie.ac.at>

Citation

It is planned to submit a paper (Schlüter P.M. & Harris S. A.) dealing with this program to *Molecular Ecology Notes*, so you may check this journal in the near future.

Declaration

Although it is not likely and certainly not intended, this program might in some situation unforeseen by the programmer cause damage to a computer system. Therefore, you are reminded that the author accepts no responsibility for such an event and that you are using this program at your own risk. This software may be distributed freely provided that the copyright notice is not removed. If it is to be included in any commercially distributed package, the author should first be contacted. If you wish to modify the source code and rebuild this application so that it fits your own need, you may do so provided you do not remove the original copyright notice and provided you do not intend to distribute it commercially.

System requirements

FAMD was developed for 32-bit Windows operating systems and was tested on Windows XP.

A CPU with FPU (floating point unit) co-processor is required. The program was developed and tested on an Intel Pentium M system.

Source Code and compilation

Source code is available upon request from the author. FAMD was written in Borland Delphi 7. Some routines are based on older ones (Turbo Pascal and Microsoft Macro Assembler). I cannot guarantee that the code will be easy to follow for anybody but you are free to give it a try.

Known issues

Opening the input file

In some cases, it may be difficult to read an input file generated by e.g. the standard Windows Editor, because it may hide some additional characters in the file that you don't see in the Editor. In such a case, opening the input file fails.

Also, opening an input file may fail if the file contains certain combinations of characters that are per se ignored (such as space, tab and character 00h).

The inclusion of remark characters in certain places in the input file, especially if you try to delete a data row by bracketing it with remark characters, may cause the loading of the input file to fail.

Data removal and labelling

In some instances, when you are removing from the data set and you did not provide data labels with your data (so that the program uses internal labelling), the labelling can get confused. Although there was an attempt to fix this bug and it did no longer occur in all tested cases, it may still be the case that there exists a combination of parameters and actions that result in a confusion of labelling.

HOW TO USE

To use FAMD:

-) generate an input file that can be read by FAMD
-) load the input file into FAMD
-) specify the file you want your analysis results to be saved to
-) specify the file you want your trees to be saved to
-) if desired, modify your data matrix (`DataMatrix` menu)
-) carry out analysis (simple analysis or replicate analyses)
-) if you carried out a simple analysis that generates a similarity matrix, you may use the `Trees` menu to generate a tree

DEFAULTS

Default settings

Individual names given in input file:	yes
Locus names given in input file:	no
Input file has individuals in:	columns
End of data character:	*
Open remark character:	{
Close remark character:	}
Presence:	1
Ambiguity/missing data point:	?
Absence:	0
Characters ignored in input file:	Blank (00h), whitespace, tab, CR, LF
JC:	100
TR:	50
BS:	1000
SH:	300000
MDR:	yes
Missing data replacement by presences:	50%
Data removal threshold (individuals):	18%
Data removal threshold (loci):	25%
Shannon Log Base:	2
Frequencies = presences/locus per:	total presences
Analysis output file name:	analysis.txt
Tree output file name:	outtree.ph

Default input format

Although FAMd is quite flexible as regards input files, it sometimes crashes if there is a strange combination of blanks and tabs in the file. This can be overcome by replacing it all with blanks or all with tabs. The characters '{' and '}' can be used to exclude parts of a line from consideration as data, but I cannot guarantee that this works. Putting a complete line in {brackets} certainly does not work (FAMd crashes).

Please avoid blank lines in the input file.

You can include individual names and/or locus/marker lanes or neither.

The default format is like this:

```
      [Ind1 Ind2 Ind3 ... IndN]
[Loc1]  1    0    1    ...  0
[Loc2]  ?    0    ?    ...  0
      ..
[LocM]  1    0    0    ...  1
*
```

Items in [] can be left out

The asterisk (EndOfData character) at the end of the data block is compulsory. Blanks or tabs can delimit the entries; the program is fairly flexible regarding that. You can also swap individuals and loci (i.e. rows and columns) - you just need to set the input file parameters accordingly.

You can also choose the characters you use to represent presences, absences and missing data and you can re-save modified data matrices, changing any of those parameters.

PROGRAM FUNCTIONS

File Menu

Load

Displays a file-open dialogue box which lets you select a file to open. The File Parameter Selection box is displayed which lets you specify which characters are used for which purpose in the file:

```
Individuals in...
    columns      Data points in the file are read such that each line represents one locus.
    rows         Data points in the file are read such that each line represents one individual.

Header presence for
    individuals   If this option is checked, individual labels are assumed to be present. If this is
                  not set, labels will be given as Ind01, Ind02, etc.
    loci          If this option is checked, locus labels are assumed to be present. If this is not
                  set, labels will be given as Loc001, Loc002, etc.
```

Characters:

These need to be single characters. Characters '<' and '>' should not be used because they are used internally in the program.

```
Presence:      Default value='1'.
                Specifies a band presence.
Absence:       Default value='0'.
                Specifies a band absence.
Ambiguity:     Default value='?'.
                Specifies an unclear band; missing data point.
OpenRemark:    Default value='{'.
                After this character, text or numbers will be treated as a remark.
CloseRemark:   Default value='}'.
                Defines the end of a remark.
EndOfData:     Default value='*'.
                This character represents the end of data in the file. Any data after
                this character will be ignored.
```

Note: Remarks may not work properly under some circumstances.

Save DataMatrix

This lets you re-save your original data matrix. In the process, you can change the characters used to specify e.g. band absences and presences, or you can swap data rows and columns (individuals and loci). Remarks that might have been present in the original data file will be lost, as will be all data after the `EndOfData` character. If you have removed individuals or loci from the data set, you can use this option to save a sub-set of your original data set.

Use the File Parameter Selection box to specify which characters are used for which purpose in the file:

Individuals in...
 columns Data points in the file are read such that each line represents one locus.
 rows Data points in the file are read such that each line represents one individual.

Header presence for
 individuals If this option is checked, individual labels are assumed to be present. If this is not set, labels will be given as Ind01, Ind02, etc.
 loci If this option is checked, locus labels are assumed to be present. If this is not set, labels will be given as Loc001, Loc002, etc.

Characters:

These need to be single characters. Characters '<' and '>' should not be used because they are used internally in the program.

Presence: Default value='1'.
 Specifies a band presence.
Absence: Default value='0'.
 Specifies a band absence.
Ambiguity: Default value='?'.
 Specifies an unclear band; missing data point.
OpenRemark: Default value='{'.
 After this character, text or numbers will be treated as a remark.
CloseRemark: Default value='}'.
 Defines the end of a remark.
EndOfData: Default value='*'.
 This character represents the end of data in the file. Any data after this character will be ignored.

Note: Remarks may not work properly under some circumstances.

Select Analysis Output File

This lets you select a file to which numerical results can be saved. Note that by selecting a file you do not actually save to this file. If you do not select a file here, FAMD will use the default file name `analysis.txt` in the current directory.

Select Tree Output File

This lets you select a file to which trees generated during the analysis will be saved. Note that by selecting a file you do not actually write to this file. If you do not select a file name here, FAMD will use the default file name `outtree.ph` in the current directory. Trees will be saved in PHYLIP format. Tree names will be stored as root labels. Note that not every program may be able to interpret this correctly. You can use programs such as TreeView to open the tree file and convert trees to another format if desired, and in doing so choose to remove branch labels (i.e. tree names).

Exit

Quits the program. FAMD will not ask you to save data prior to quitting.

DataMatrix Menu

The DataMatrix menu lets you modify different aspects of the data. Generally, results are displayed on the screen and you are asked whether you would like to save them to the analysis file. If this file already exists you are asked whether data should be appended to it or whether the file should be overwritten.

Matrix Statistics

This option tells you how many individuals and loci were detected in your input file. It is often useful to use this option to check that the program loads exactly the data set you wanted it to load, or that the data set was loaded completely. In addition to the number of individuals and loci, also the size of the data matrix (i.e. number of data points, given by #individuals × #loci) and the amount of missing data in the data set are displayed.

Restore Original Matrix

This option lets you restore the original data matrix as first loaded when an input file was opened. It is useful, e.g. after data points have been removed from or replaced in the data matrix.

Frequency Statistics

This will save the frequency $p(i)$ of band presences in each locus i to the analysis output file. You are first asked how you want frequencies to be defined. The first option is:

$$p(i) = \frac{N_1(i)}{n} \quad \text{where: } N_1(i) \text{ is the number of band presences (1) in locus } i$$

n is the number of individuals

This is intuitive. The second option is:

$$p(i) = \frac{N_1(i)}{\sum_{k=1}^s N_1(k)}$$

where: $N_1(i)$ is the number of band presences (1) in locus i

s is the number of loci in the data set

$\sum N_1(i)$ is the total number of presences in the data set

The reason for giving this option of calculating frequencies lies in the formula by Bowman *et al.* (1969) for calculating the variance associated with Shannon's index.

Bowman, K. O., Hutcheson, K., Odum, E. P. and Shenton, L. R., 1969, Comments on the distribution of indices of diversity, *Proc. Intl. Symp. Stat. Ecol.* **3**: 315-359.

Missing Data Statistics

This will write information about missing data to your analysis output file. The output values are the percentage of missing data points in each individual and in each locus.

Missing Data Replacement Settings

This displays a dialogue box that lets you select how missing data should be treated by those routines that deal with missing data. No action is performed on the data matrix. The options are to replace all missing data points by presences, by absences or to randomly replace missing data by x % of presences and $(100-x)$ % of absences, where x is the value entered by you. The default is missing data replacement by 50 % presences.

Replace Missing Data

Selecting this option replaces missing data in your data set according to the settings specified in the menu *Missing Data Replacement Settings*. By doing so, the data matrix is modified and missing data in it replaced by discrete characters. Therefore, if you wish to proceed with the analysis using the original data matrix, you must first restore it.

Remove Individuals

With missing data...

This option removes individuals from the data matrix that have a percentage of missing data that is greater than the specified threshold percentage. If you wish to undo such a data removal to continue with the original data matrix, you must restore it.

Bug: Please note that in some circumstances the internal labelling of individuals can get confused (i.e., if you have not used individual names so that the program automatically generates some for you). However, this only concerns data labels, not data themselves.

Remove Loci

This option removes loci from the data matrix according to your choice of options. If you wish to undo such a data removal to continue with the original data matrix, you must restore it.

Bug: Please note that in some circumstances the internal labelling of loci can get confused (i.e., if you have not used individual names so that the program automatically generates some for you). However, this only concerns data labels, not data themselves.

With missing data...

Removes loci that have a percentage of missing data that is greater than the specified threshold percentage.

Monomorphic...

Removes monomorphic loci from the data set. This should be done for instance prior to calculation of Shannon's index (non-bootstrapping version). A locus with a monomorphic presence or absence is defined as follows:

Monomorphic presence: IF $(m \leq o)$ AND $(z=0)$.

Monomorphic absence: IF $(m \leq z)$ AND $(o=0)$

where $m = N_z(i)$, the number of missing/ambiguous data in locus i
 $o = N_0(i)$, the number of band absences in locus i
 $z = N_1(i)$, the number of band presences in locus i

All other bands are considered polymorphic.

Analysis Menu

The **Analysis** menu lets you perform different calculations on your current data set, such as different versions of Jaccard's coefficient. Similarity matrices produced in that way can then be subjected to UPGMA tree reconstruction in the **Trees** → UPGMA menu. Items in the **Replicate Analyses** submenu produce many data set replicates from your current data set and work with these and output results automatically. For work with many replicate data sets, you should first set your parameters in the respective settings dialogue box.

Jaccard (standard)

This calculates a similarity matrix based on Jaccard's coefficient of similarity from your current data matrix. Jaccard's coefficient for a pair of individuals i and j is defined as:

$$S_{ij,Jaccard} = \frac{N_{11}}{N_{11} + N_{01} + N_{10}}$$

where N_{xy} is the number of characters that have state x in individual i and state y in individual j . Possible character states are band presence (1), band absence (0) and missing data (?).

The similarity matrix will be written to your analysis output file.

Minimum Jaccard

This calculates a similarity matrix based on the minimum Jaccard's coefficient from your current data matrix. The minimum Jaccard's coefficient, taking into account missing data, for a pair of individuals i and j is defined as:

$$S_{ij,min} = \frac{N_{11}}{N_{11} + N_{01} + N_{10} + N_{?1} + N_{1?} + N_{?0} + N_{0?}}$$

where N_{xy} is the number of characters that have state x in individual i and state y in individual j . Possible character states are band presence (1), band absence (0) and missing data (?).

The similarity matrix will be written to your analysis output file.

If the data set does not contain missing data, the minimum Jaccard value will be identical to the standard Jaccard's coefficient.

Maximum Jaccard

This calculates a similarity matrix based on the minimum Jaccard's coefficient from your current data matrix. The minimum Jaccard's coefficient, taking into account missing data, for a pair of individuals i and j is defined as:

$$S_{ij,max} = \frac{N_{11} + N_{?1} + N_{1?} + N_{??}}{N_{11} + N_{01} + N_{10} + N_{?1} + N_{1?} + N_{??}}$$

where N_{xy} is the number of characters that have state x in individual i and state y in individual j . Possible character states are band presence (1), band absence (0) and missing data (?).

The similarity matrix will be written to your analysis output file.

If the data set does not contain missing data, the maximum Jaccard value will be identical to the standard Jaccard's coefficient.

Average Jaccard

This function calculates a similarity matrix based on the average Jaccard's coefficient from your current data matrix. For this, minimum ($s_{ij,min}$) and maximum ($s_{ij,max}$) Jaccard's coefficient are calculated. The average Jaccard coefficient is defined as the arithmetic average of values drawn randomly (uniformly) from the interval $[s_{ij,min}; s_{ij,max}]$. The number of random draws, JC can be defined in the Replicate Analysis Settings dialogue box. In addition to the average Jaccard value, the associated variance and standard deviation (square root of variance) is calculated. If the data set does not contain missing data, the average Jaccard value will be identical to the standard Jaccard's coefficient and variance and standard deviation will be zero.

The similarity matrix will be written to your analysis output file, as well as a matrix of corresponding variances and a matrix of corresponding standard deviations.

Shannon's index

This function calculates Shannon's index and its variance from your current data set. You should always first remove monomorphic loci manually from the data matrix, using the respective function in the DataMatrix menu. This is because depending on your definition of band frequencies, monomorphic loci may still contribute to the sum calculated (although they aren't supposed to), which will artificially change Shannon's index. Shannon's index is defined as:

$$I \approx -\sum_{i=1}^s p_i \log_2 p_i$$

where I is Shannon's index

p_i is the frequency of band presences in locus i , as defined as in the Shannon Scaling... dialogue box

s is the number of loci

$\log_2 x = \lg x$ is the logarithm to base 2.

The associated variance is calculated using the formula of Bowman *et al.* (1969):

$$\text{var}(I) \approx \frac{\sum_{i=1}^s p_i \log_2^2 p_i - (\sum_{i=1}^s p_i \log_2 p_i)^2}{n} + \frac{s-1}{2n^2} = \frac{\sum_{i=1}^s p_i \log_2^2 p_i - I^2}{n} + \frac{s-1}{2n^2}$$

where I is Shannon's index

p_i is the frequency of band presences in locus i , as defined as in the Shannon Scaling... dialogue box

s is the number of loci

n is the number of individuals

$\log_2 x = \lg x$ is the logarithm to base 2.

Bowman, K. O., Hutcheson, K., Odum, E. P. and Shenton, L. R., 1969, Comments on the distribution of indices of diversity, *Proc. Intl. Symp. Stat. Ecol.* **3**: 315-359.

The calculation of this variance will only work if p_i is defined as band presences in a locus relative to all band presences in the data set. Otherwise, the doing the calculation may result in a negative value. The standard deviation is calculated as the square root of the variance.

Shannon Scaling...

Here you can define which frequency definition and which logarithm base should be used for calculation of Shannon's index.

The frequency of band presences $p(i)=p_i$ can be defined:

- relative to all presences (frequency per data set):

$$p(i) = p_i = \frac{N_1(i)}{\sum_{k=1}^s N_1(k)}$$

where $p(i)$ is the band frequency in locus i
 $N_1(i)$ is the number of band presences in locus i
 s is the number of loci in the data set
 $\sum N_1$ is the total number of band presences in the data set.

- divided by the number of individuals (frequency per locus):

$$p(i) = p_i = \frac{N_1(i)}{n}$$

where $p(i)$ is the band frequency in locus i
 $N_1(i)$ is the number of band presences in locus i
 n is the number of individuals in the data set

The will program will always calculate Shannon's index using $\log_2 x = \lg x$ (the dual/binary logarithm), because this is A) Shannon is refers to binary data and B) it is the native logarithm for the computer's processor:

$$I_2 = I \approx -\sum_{i=1}^s p_i \log_2 p_i$$

where I_2 represents the fact that logarithm with base 2 was used

However, since in principle, any logarithm can be used, the program can re-scale Shannon's index accordingly by multiplying with a correction factor:

$$I_A = I_2 \log_A 2$$

where I_2 represents Shannon's index based on $\log_2 x$
 I_A represents Shannon's index based on $\log_A x$

You can select the following options:

- use $\log_2 x = \lg x$
- use $\log_e x = \ln x$
- use $\log_{10} x = \lg x$
- use $\log_A x$, where A is user-defined

Replicate analyses

Unlike the rest of options in the `Analysis` menu, the functions of the `Replicate Analyses` submenu work with replicates of your current data set and modify these replicate data sets automatically. Before carrying out any replicate analysis, you should define the parameters for analysis using the `Replicate analysis settings` dialogue. The parameters used by the individual functions are indicated in brackets after their name.

Replicate analysis settings

Here, you can define parameters for different analysis functions that operate on multiple data set replicates. You should do this before you start your analyses. The available parameters are:

- SH: Resampled data matrix replicates for Shannon
This number defines how many replicates of your current data set should be generated for estimating Shannon's index by data resampling (i.e. the number of bootstrap replicates).
- RMD: Replace missing data as (set in respective dialogue)
If checked, missing data in the data matrix will be replaced according to the parameters defined in the `Missing Data Replacement` dialogue box.
- BS: Resampled data matrix replicates for Jaccard
Defines from how many data set replicates UPGMA trees should be generated.
- JC: Average Jaccard from how many random draws
Defines the number of random draws from the interval $[s_{ij,min}; s_{ij,max}]$ that is used for calculating an average Jaccard value and its variance.
- TR: Number of average Jaccard trees to generate
Defines how many UPGMA trees based upon average Jaccard-derived distance matrices should be generated.

Bootstrap Shannon's variance (SH+RMD)

This option estimates Shannon's index and its variance by data resampling. Unlike the function `Shannon's Index` that operates on your current data set and for which you should first remove monomorphic loci from the data set, this option generates resampled data sets from your current data set and automatically removes monomorphic loci from it before calculating Shannon's index. Since every resampled data set may contain a different configuration of loci from the original data matrix and since missing data may be replaced randomly, it may be that loci that are treated as monomorphic are not monomorphic in another data set replicate. Therefore, you **SHOULD NOT** remove monomorphic bands manually from your data set (or replace missing data manually) for carrying out this function, since thereby you will limit the variation generated during bootstrapping.

This function uses the parameters `SH` and `RMD` set in the `Replicate Analysis Settings` dialogue box. It is implemented as follows:

Repeat the following `SH` times:

-) Generate a resampled data set by randomly choosing `s` loci from your current data set (every locus can be picked in every random draw), where `s` is the number of loci present in your current data set.
-) If `RMD` is set replace missing data so that monomorphic loci are removed from the newly generated replicate data set (according to the definitions given under `Remove Monomorphic...`)
-) From this data matrix, Shannon's index is calculated as described in the respective section

The series of values for Shannon's index are averaged and the variance calculated.

Please be aware that for high `SH` values, this procedure may take a considerable time.

Bootstrap Std. Jaccard Tree (BS+RMD)

This function can be used to generate multiple UPGMA trees based upon the standard Jaccard's coefficient from resampled data matrices and stores them in the tree output file which can then be imported into other programs and prepared for further analysis. For example, you may want to open the output file with TreeView, convert it into nexus format and use COMPONENT to generate a consensus tree.

This function uses the parameters `BS` and `RMD` set in the `Replicate Analysis Settings` dialogue box. It is implemented as follows:

Repeat the following `BS` times:

-) Generate a resampled data set by randomly choosing `s` loci from your current data set (every locus can be picked in every random draw), where `s` is the number of loci present in your current data set.
-) If `RMD` is set replace missing data so that monomorphic loci are removed from the newly generated replicate data set (according to the definitions given under `Remove Monomorphic...`)
-) From this data matrix, calculate similarity matrix based upon Jaccard's coefficient of similarity (standard definition)
-) Perform the UPGMA clustering algorithm on the distance (1 - similarity) matrix
-) Write a tree to the tree output file.

Multiple Avg. Jacc. Trees (TR+JC)

This function can be used to generate multiple UPGMA trees based upon the average Jaccard's coefficient and stores them in the tree output file which can then be imported into other programs and prepared for further analysis. For example, you may want to open the output file with TreeView, convert it into nexus format and use COMPONENT to generate a consensus tree.

This function uses the parameters `TR` and `JC` set in the `Replicate Analysis Settings` dialogue box. It is implemented as follows:

Repeat the following `TR` times:

-) Randomly draw `JC` values from the interval [$s_{ij,min}$; $s_{ij,max}$] (minimum to maximum possible Jaccard value) and calculate average Jaccard values from this series of numbers.
-) Generate a similarity matrix from the average Jaccard values
-) Perform the UPGMA clustering algorithm on the distance (1 - similarity) matrix
-) Write a tree to the tree output file

Bootstrap Avg. Jacc. Trees (BS+JC)

This function can be used to generate multiple UPGMA trees based upon the average Jaccard's coefficient from resampled data matrices and stores them in the tree output file which can then be imported into other programs and prepared for further analysis. For example, you may want to open the output file with TreeView, convert it into nexus format and use COMPONENT to generate a consensus tree.

This function combines the data resampling of loci used for bootstrapping values and the calculation of average Jaccard coefficients by sampling from the interval of possible values.

This function uses the parameters BS and JC set in the Replicate Analysis Settings dialogue box. It is implemented as follows:

Repeat the following BS times:

-) Generate a resampled data set by randomly choosing s loci from your current data set (every locus can be picked in every random draw), where s is the number of loci present in your current data set.
-) Randomly draw JC values from the interval $[S_{ij,min}; S_{ij,max}]$ (minimum to maximum possible Jaccard value) and calculate average Jaccard values from this series of numbers.
-) Generate a similarity matrix from the average Jaccard values
-) Perform the UPGMA clustering algorithm on the distance (1 - similarity) matrix
-) Write a tree to the tree output file

Trees Menu

UPGMA (multifurcating)

This generates a UPGMA (unweighted pair group method using arithmetic averages) tree from a distance [i.e., 1-similarity(Jaccard)] matrix that you have generated previously and stores the tree in the tree output file you have defined (or else to the default tree output file).

The algorithm implemented is a modified UPGMA algorithm that can generate multifurcating trees, in contrast to a strictly bifurcating implementation. This means that if there are two or more equally good choices for clustering groups of individuals, this will be realised as a multifurcation in the tree, rather than randomly choosing one of the possible choices to generate a strictly bifurcating tree. However, the chance that such a situation will occur in a real data set is very low.

Help Menu

About

Displays a message box with copyright notice, a short summary of the purpose of FAMD and the like. It also contains a legal disclaimer, which is repeated below:

FAMD (c) Copyright 2002-2005 Philipp M Schlüter.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, whether expressed or implied; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. The author (PMS) will not be liable for any special, incidental, consequential, indirect or similar damages due to loss of data or any other reason, even if he or an agent of his has been advised of the possibility of such damages. In no event shall the author be liable for any damages, regardless of the form of the claim. The person using the software bears all risk as to the quality and performance of the software.

Help

Displays this help file (famdhhelp.pdf).