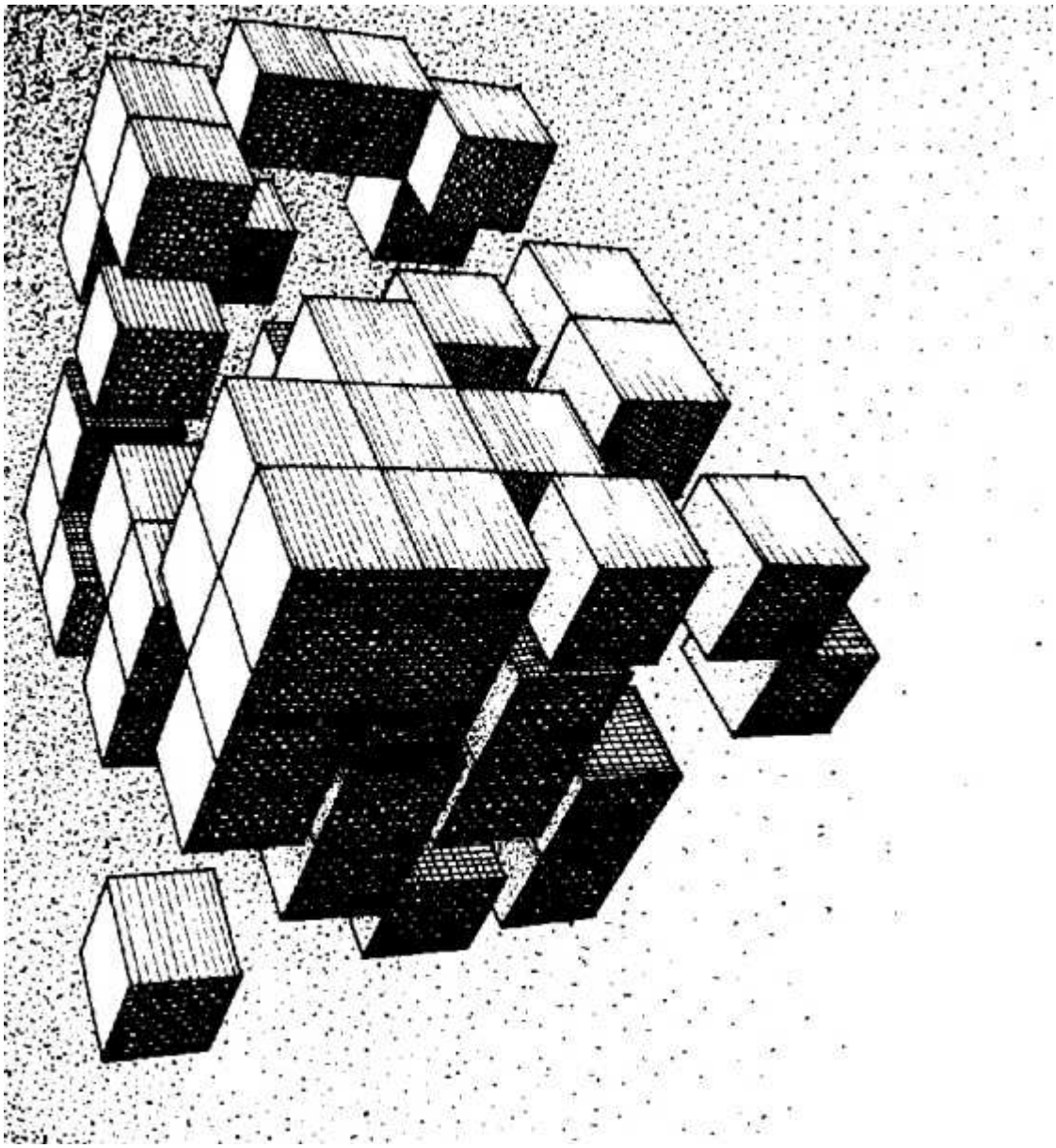
3WayPack

A program suite for three-way analysis

P.M. Kroonenberg & Y. de Roo



The Three-Mode Company, Leiden University

3WayPack

A program suite for three-way analysis

by Pieter M. Kroonenberg and Yves de Roo

3WayPack is an integrated suite of programs to analyse three-way data. All facets of analysing such data are included, such preprocessing, analysing the data and postprocessing the output. Moreover facilities are included to allow for easy navigation of the output and make publication-quality graphics based on the output.

3WayPack

Printed: September 2010 in The Netherlands

Publisher

The Three-Mode Company
Leiden University

Technical Editors

Pieter M. Kroonenberg

Progamming

Yves de Roo
Pieter M. Kroonenberg

Cover Designer

Siep Kroonenberg

Production

Pieter M. Kroonenberg
Yves de Roo

Special thanks to:

All the people who contributed to this document directly and indirectly.
Thanks go to Wiley-Blackwell who published Applied Multiway Data Analysis which contains the theory behind these programs as well as many examples of this type of analysis. The Department of Child and Family is thanked for providing me with the opportunity to work on this manual.
Last not least, we want to thank EC Software who wrote this great help tool called HELP & MANUAL which printed this document.

Table of Contents

1 Introducing 3WayPack

Some history
3WayPack is an integrated suite of programs for three-way data analysis. The elementary analysis programs for the basic three-mode models, i.e. the Tucker2 and Tucker3 have a long history going back to 1975 when I (Kroonenberg) developed the Tucker2 program in connection with my master thesis under supervision of Jan de Leeuw. The development of a program for the Tucker3 model followed later and the mathematical background was published in *Psychometrika* in 1980.

Initially programming consisted of working with IBM 80-column punch cards and the idea of an integrated environment for three-mode data analysis did not yet exist. With the advent of large main frames this possibility became theoretically possible but only when personal computers became common in scientific environments did the opportunity for designing menu systems really arise.

Our first attempt was a DOS-based interactive environment called Interface3 (or IF3) programmed in Pascal by Piet Brouwer. This interface is still available from The Three-Mode Company and it contains a number of programs not yet available in the present version of 3WayPack. This interface was entirely keyboard driven which seems now out-of-date but in reality is very efficient. A more serious drawback is that the program can only access RAM memory, thereby seriously limiting the size of the data sets that can be analysed. In addition due to an early decision about the inner workings of the interface it needs the whole data set in memory twice which limits the size of the data sets even more.

The desire to have a truly Windowed program was a long standing one and different versions were developed by different persons among them Gert van Donselaer. The present version is entirely due to the efforts of Yves de Roo who also introduced me to Delphi Pascal so that I could adjust the program where necessary. The design however is a communal effort from both of us, be it that the implementation is entirely due to Yves de Roo.

The present version (October 2010) seems sufficiently stabilised to distribute to other researchers. We hope that it will contribute to their research efforts. The programs are distributed by The Three-Mode Company which is entirely a figment of my imagination without any legal basis or status. As such it is a part of the Leiden University and its website is hosted on the university servers: http:\\three-mode.leidenuniv.nl

---

About the authors

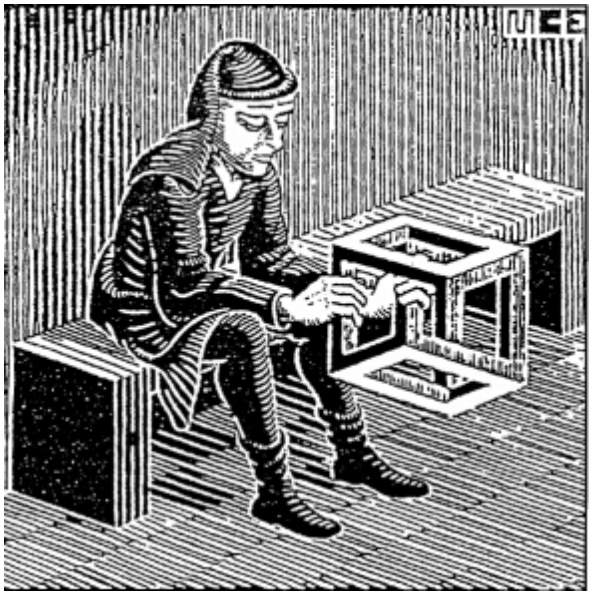
Dr. Pieter M. Kroonenberg is a professor in the area of multivariate analysis in particular of three-way data at the department of Child and Family Studies, Leiden University. Address: Wassenaarseweg 52, 2333 AK Leiden, The Netherlands; Tel. +31-71-5273446/3434; e-mail: kroonenb@fsw.leidenuniv.nl. At present his university home page is http://www.socialsciences.leiden.edu/educationandchildstudies/childandfamilystudies/organisation/staffcfs/kroonenberg.html, while his professional home page is http://three-mode.leidenuniv.nl/ (The Three-Mode Company), but do not expect these addresses to last until eternity .

Yves de Roo is an independent consultant and scientific programmer who before retirement headed the ICT group of the *Netherlands Institute for Advanced Study* in the Humanities and Social

Sciences (NIAS) in Wassenaar, The Netherlands.

1.1 The Three-Mode Data Analyst

The Three-Mode Data Analyst



This *Three-Mode Data Analyst* is officially called *Man with Cuboid* and is displayed here by courtesy of The M.C. Escher Company. Please respect the wishes of those who support M.C. Escher's work by not republishing this image without their consent. All M.C. Escher works (©) 2005 The M.C. Escher Company - Baarn, The Netherlands. All rights reserved. Used by permission.

1.2 Overview 3WayPack

Introduction
The program suite 3WAYPACK is designed to analyse three-way data using *analysis programs* each of which estimate the parameters of one or more three-way models.The suite itself is referred to as 3WAYPACK while the user interface is called TWPACK.

Basic structure
The basic structure of the suite consists of an interface Windows program (written in Borland Delphi Pascal) called TWPACK which is used to guide the flow of setting up the analyses, the data manipulation and the output processing. The actual work is done by the preprocessing program, analyses programs and postprocessing or output manipulation programs which are written in Fortran95.

Job file
The basic memory of work carried out with the programs is located in a job file, which contains the parameters of the particular analysis carried on a specific data. Thus this file is both a record of which analysis has been carried out and it can be used to do the same type of analysis again. As it contains a list of all the files used by the analysis, it is sensitive to moving files to other directories

including itself. Messing with the memory of a system is seldom an easy and probably not a clever thing to do. Full details on the job file can be found in the job file section in the *Setting-up the analysis* part.

Work flow
The work flow of an analysis is that the parameter specification is done in the interface TWPACK. The results from the specification are stored in command files. Then the interface calls an external program which uses the command file for its execution. When the external program has done its work, it will have produced a number of output files (text files, plot files, and html files) which can be inspected via the interface and which can be further processed via the programs available through the postprocessing section of the interface. The effect of this is that the whole suite will produce a fair number of output files which will reside in the directory of the data file (see File name section). Further details can be found in the appropriate sections.

Programs included in 3WayPack
The suite contains programs for three-way analysis, utilities necessary to handle three-way data, and programs to further process the output files to enhance interpretation. The main stay are the analysis programs to estimate the parameters of methods/models, such as three-mode principal component analysis in several forms (TUCKALS2 and TUCKALS3 programs), parallel factor analysis (TRILIN), exploratory three-way analysis of variance with a single observation per cell (PREPROC3), exploratory two-way multivariate analysis with a single observation per cell (PREPROC3) and three-mode mixture method of clustering for continuous data (MIXCLUS3). With the models included in the package also two-mode (or standard) principal component analysis, two-mode (or standard) singular value decomposition (TUCKALS2 and TUCKALS3), replicated principal component analysis (TUCKALS3), and two-mode mixture method of clustering (MIXCLUS3) can be solved.

Programs not yet included in 3WayPack
Programs which are available from The Three-Mode Company but not yet included in the suite are Generalised Procrustes Analysis (PROCRUS), three-mode correspondence analysis (ANACOR3), three-mode PCA for multimode covariance matrices (T3COVAR) and simultaneous component analysis (SIMULCMP). A further program for the Tucker3 model based on regression algorithms rather than eigenvalue algorithms (T3GEPCAM) is available as well.

Preprocessing programs
In order to be able to carry out meaningful analyses raw three-way data often have to undergo some editing and preprocessing, such as subtracting means and equalising variance, swapping of modes or filling in starting values for the missing data. All these preprocessing options are available in the program PREPROC3.

Postprocessing or interpretation programs
Generally the output of an analysis program needs to be further processed for interpretation. In the program suite it is possible to carry out a number of such postprocessing such as rotations of components and core arrays (T3ROTATE and ROCOCO), inspection of residuals (RESIDUAL), sophisticated plotting of the results via joint biplots and nested-mode biplots (JOINTPLT).

Utilities
The suite also contains a number of utilities which make life easier for the analyst.
- *Case-to-Slice*. Data conversion from case-mode to slice-mode. All analysis programs expect the

three-way data to be in slice-mode, while common statistical packages tend to supply the data in case-mode form.

- *Find Missing*. Create a missing-data locatoin file. The analysis programs need to have information about the locations of the missing data in the data file. FINDMISSING will create such as a file with the locations in the data array of all missing values.
- *Data Selector*. Removing rows or slices from a data set can be a hassle but this program will take care of this.
- *Rotate 3D*. Three-dimensional rotation program for visualising three-dimensional component spaces [to become available soon].

Language (especially in the Index)

The manual has been written by a nonnative speaker as much as possible in British English rather than American English. Virtually all endings of verbs use the *-ise* form rather than *-ize*.The letter *u* is always present in words like *colour* and *favourite*. Furthermore the process of subtracting means is indicated as *centring* rather than *centering*. These practices exert of course their influence on the ordering of words in the index, so beware.

Redundancies

In many places it looks in the printed version of this Manual as if the information is redundant in the sense that the same information is presented twice, sometimes even on the same page. This is a consequences of having a help file system that produces both the printed manual and the on-line help file with context-sensitive help from a single master file. An additional annoying feature is that headings may dangle at the bottom of the page but this is caused by the same set-up.

References

The primary reference for most of the procedures implemented in the software is the book Kroonenberg (2008). This does not imply that this book is the only source for the techniques and procedures implemented in 3WayPack but most examples in the book have been carried with this program suite which was extended during the writing of the book.

The other general source for multiway analysis is the book by Smilde, Bro, & Geladi (2004). It contains far more mathematical information and the emphasis is on chemical applications.

The references from those book as well as the bibliography on the website of The Three-Mode Company should give access to all the literature in the field.

1.2.1 3WayPack work flow

Introduction

The work flow of an analysis is that the specification is done in the interface TWPACK. The results from the specification are stored in command files. Then the interface calls an external program which uses the command file for its execution. When the external program has done its work, it will have produced a number of output files (text files, plot files, and html files) which can be inspected via the interface and which can be further processed via the programs available through the postprocessing section of the interface. The effect of this is that the whole suite will produce a fair number of output files which will reside in the directory of the data file. Further details can be found in the appropriate sections.

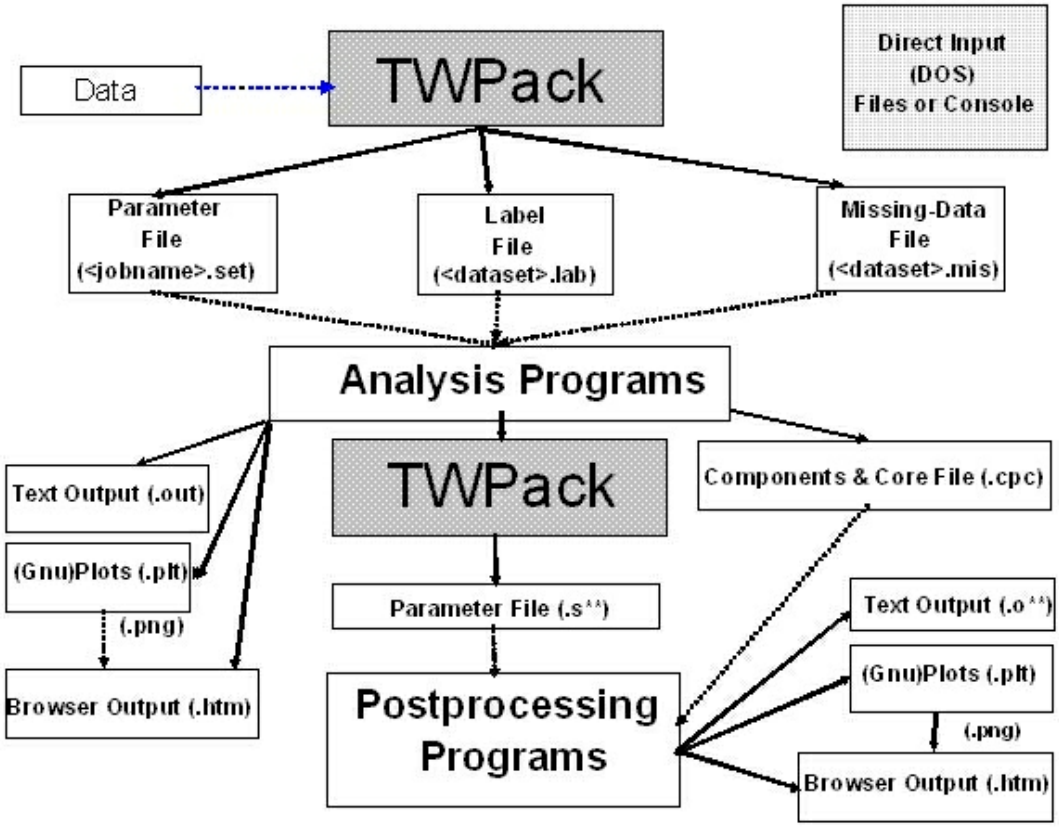


Fig. General overview of the work flow of 3WayPack

1.2.2 TWPack structure

Introduction

The graph below sets out the basic structure of the TWPACK interface of 3WAYPACK.

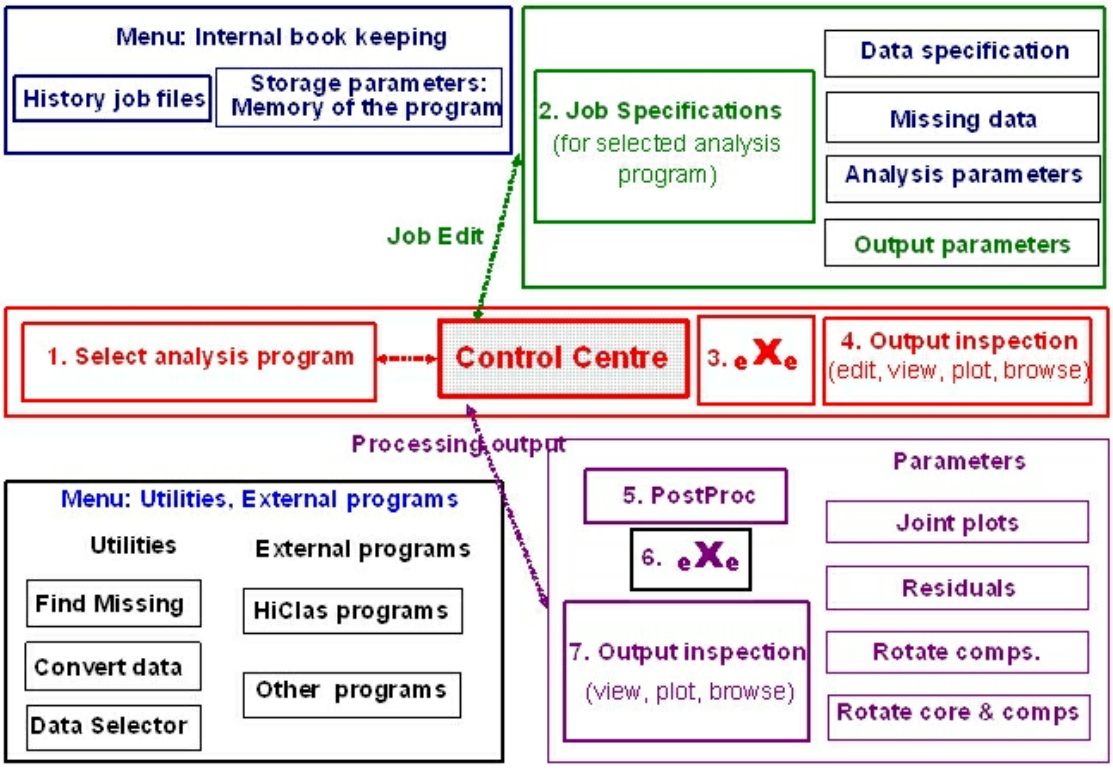


Fig. Overall structure of TWPack (X = Execute)

1.2.3 File naming conventions

Introduction

Because both the interface TWPACK and the other programs produce and need a large number of files, a strict naming convention is necessary to keep especially the data directory in some order. Therefore all files related to the data have the same name as the data but different extensions, while all files generated by the programs have the name of the job file with appropriate additional characters and extensions to keep the file names unique.

Type of files

The files relevant in 3WAYPACK consist of five groups

1. System files
2. Job file
3. Data-related files
4. Preprocessing files
5. Analysis-program files
6. Postprocessing files

System files

The system files are files that are used by the interface but they are rather diverse. They need to reside in the program or system directory for the program to work properly. They consist of executables, graphics files, help files and the like.

- exe files: TWPACK.EXE - the interface; other exe files contain the preprocessing, analysis, postprocessing, plot and other programs.
- ico, jpg, bmp, gif files: graphics for the programs
- chm: 3WayPack.chm; browser-help file (also works stand-alone)
- pdf: 3WayPack.pdf; help file as pdf.
- ini: 3WayPack.ini: file with the location of your preferred editor (default: NOTEPAD.EXE)

Job file

- *job file*: <jobname>.3wp; memory of the job also containing all parameters related to the specific job including analysis parameters and postprocessing parameters. The jobname is the basis for naming all output files.

Data-related files

The data-related files are files which describe the data in some way such as a label file, a file indicating at which locations there are missing values, etc. Such files should reside in the data directory.

- *data set*: <datasetname>.dat
- *label file*: <datasetname>.lab; contains the labels of the levels of the modes
- *missing-data file*: <datasetname>.mis; contains the locations of the missing data
- external-configuration file; <datasetname>.ext

Preprocessing files

The preprocessing program PREPROC3 needs a couple of input file and produces itself a text output file and optionally a new data file.

Input files, system related

- *program info file:* <jobname>.fil; file with names of directories and the name of the data set

- *input parameter file*: <jobname>.set; file with input parameters for the preprocessing program.

Output files
- *text output file*: <jobname>.out; the basic output file; it can be inspected with an external editor or the internal viewer
- *new data file*: <jobname>.pp3; optionally produced after swapping, preprocessing, etc.
- *split-half data files*: <jobnamexxx>.dat, where xxx is D1A, D1B, D2A, D2B; produced at request after split-half spceification.

Analysis-program files
Input files, system related
- *program info file:* <jobname>.fil; file with names of directories and the name of the data set.
- *input parameter file*: <jobname>.set; file with input parameters for the analysis programs.

Output files, system related
- *cpc-file*: <jobname>.cpc: file containing components and core array for further use by postprocessing programs. The number of decimals is considerably larger than in the basic output files.
- *warning/error file*: <jobname>.err; file containing warning and/or error messages from the analysis and postprocessing programs. Only present if errors or warnings have been issued. If it is available a button will become available so it can be inspected.

Output files analysis programs

<jobname>.out	the basic output file; can be inspected with an external editor or the internal viewer
<jobname>.plt	the plot command file to be used by GNUPLOT, the plotting program provided in 3WAYPACK
,jobname>xx.png	files containing bitmap plots for the browser; xx can be anything to identify the plot to the browser; which needs unique names for displaying them
<jobname>.htm	the general browser command file containing html instructions for the frame in which the output is displayed
<jobname>Br.htm	the left-hand navigation browser file containing html instructions for the navigating
<jobname>Bd.htm	the right-hand content browser file containing html instructions for displaying the output

Postprocessing files
Input or set-up files with parameters

• <jobname>.sro	the input file of T3ROTATE
• <jobname>.src	the input file of ROCOCO
• <jobname>.sre	the input file of RESIDUAL
• <jobname>.sjp	the input file of JOINTPLT

Output files

• <jobname>.oro	the basic output file of T3ROTATE
• <jobname>.orc	the basic output file of ROCOCO
• <jobname>.ore	the basic output file of RESIDUAL
• <jobname>.ojp	the basic output file of JOINTPLT
• <jobname>.res	file with residuals produced by RESIDUAL

- <jobname>.fit file with fitted data produced by RESIDUAL
- <jobname>XX.plt the plot command file to be used by GNUPLOT
- <jobname>XX.htm general browser command file
- <jobname>XXBr.htm left-hand navigation browser file
- <jobname>XXBd.htm right-hand content browser file
- <jobname>.rpc: file containing components and core array after rotation for further use for JOINTPLT (extension should be renamed to cpc before usage; after rename the original cpc file first).

where XX = JP, RE, RO, RC for JOINTPLT, RESIDUAL, T3ROTATE or ROCOCO, respectively.

Note: At present the naming for T3ROTATE and ROCOCO are identical using RO, rather than RO and RC, respectively. This will be corrected as soon as possible.

1.2.4 Terminology

Common terms used in this manual

There are a number of terms which are not necessarily very common in standard two-mode statistics. To help understanding the options offered by the programs contained in this program suite an abbreviated version of the glossary contained in Kroonenberg (2008) is included in this manual.

1.3 Overview preprocessing programs

Introduction

The program suite 3WAYPACK contains at present one preprocessing program that can be assessed from the screen - PREPROC3. A number of other programs or utilities can be accessed from the Main Menu bar.

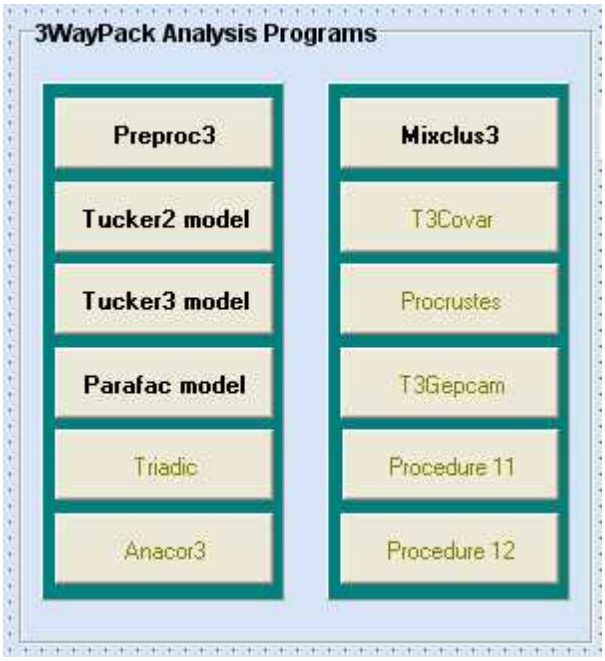


Fig. Program listing

The PREPROC3 program listed above is one of the core programs listed in bold and black.

Utility programs

In addition, the suite contains a few utility programs which also do a certain amount of preprocessing. They are available from the Utilities entry on the Main Menu bar and are explained in the Utility section of the manual.

- *Case-to-slice*: Change the input data from case-mode to frontal-slice mode.
- *FindMissing*: A utility to create a missing-data file with locations of the missing data.
- *DataSelector*: A utility to select subsets from a three-way array by eliminating or selecting designated slices from the data set.

1.3.1 Preproc3 - Preprocessing three-way data

Program description

PREPROC3 is a program for preprocessing three-way data. It can also produce two types of analyses of variance with a single observation per cell.

The program can perform the following functions
1. *Swapping* the dimensions of a three-way array;
2. *Estimating missing data* according to several ANOVA models;
3. *Centring* three-way arrays in all relevant manners;
4. *Normalising* three-way arrays in all relevant manners;
5. *Printing* all means from a three-way ANOVA model before and after preprocessing;
6. Computing and printing three-way *ANOVA summary tables* before and after preprocessing;
7. Computing and printing two-way *MANOVA summary tables* before and after preprocessing;
8. Creating two sets of two orthogonal *split-half data sets*.

All (M)ANOVA models mentioned have only a single observation per cell.

PREPROC3 will produce a data set <datasetname>.pp3 on which the requested operations specified in options 1, 2, 3 or 4 have been carried out. Option 8 will produce four data sets: <datasetnamexxx>.dat where xxx is D1A, D1B, D2A, D2B, respectively.

Reference:
Kroonenberg (2008), Chapter 6.

1.4 Overview analysis programs

Introduction
The program suite 3WAYPACK contains a number of programs for carrying our three-way analysis.

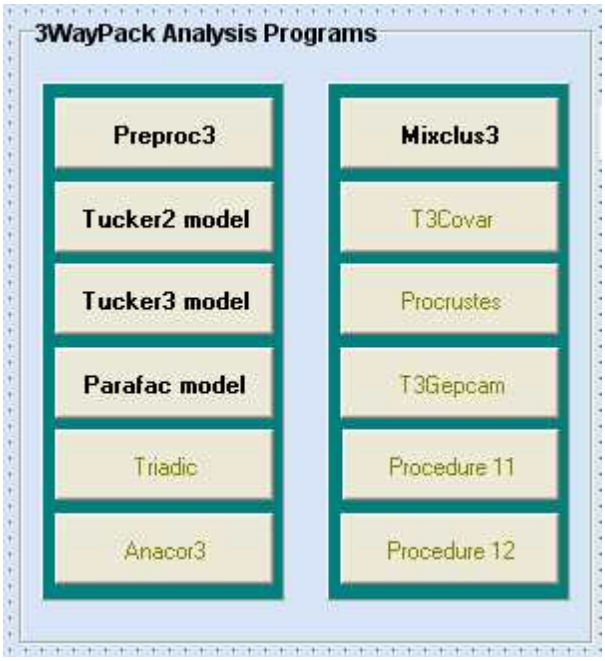


Figure: Program listing

The programs listed in the Figure **Program Listing** are the core programs of which the operational programs are listed in bold and black. The other programs indicated are available but at present only run under an older (DOS-type) interface IF3, which can be made available on request. They will be supplied for free with a purchase of the present suite or for a small fee if requested separately.

1.4.1 Tuckals2 - Three-mode principal component analysis

Program description

TUCKALS2 is a program for fitting the Tucker2 model, i.e. a three-mode principal component model with dimension reductions along the first two of the three ways and an extended core array, which has as its dimensions the number of components of way 1 (P) and way 2 (Q), and the number of levels of way 3 (i.e. $P \times Q \times K$).

The program can also be used for the IDIOSCAL model, if the input is a set of K symmetric matrices with either distances or squared distances. For a single slice ($K = 1$) the program performs a singular value decomposition and can thus be used for standard two-mode principal component analysis

References

Kroonenberg (2008), Chapter 4, Section 4.5.2 (Tucker 2 model); Chapter 2, Section 2.4.5

1.4.2 Tuckals3 - Three-mode principal component analysis

Program description

TUCKALS3 is a program for fitting the Tucker3 model, i.e. a three-mode principal component model with dimension reductions along all three ways and a full core array, which has as dimensions the

number of components of each of the three ways (i.e. $P \times Q \times R$).

The program can also be used for a two-mode singular value decomposition, a weighted (or replicated) principal component analysis, or simple two-mode PCA. Tucker's three-mode scaling (Tucker, 1972) will be performed if the input frontal slices are symmetric.

References
Kroonenberg (2008), Section 4.5, Section 4.5.3 (Tucker 3 model); Tucker (1972).

1.4.3 Trilin - Parallel factor analysis

Program description

TRILIN is a program for fitting the *Parafac model*, i.e. a three-mode component analysis with a super diagonal core array (Harshman, 1970). The Parafac model, also called the canonical decomposition model (*Candecomp model* - Carroll & Chang, 1970), is an extension of principal component analysis in which the components are the same for each level of the third way except for proportionality constants. This proportionality is also referred to as the *parallel proportional profiles* principle. In some publications the model is referred to as the *Candecomp-Parafac model* or *CP-model*. The symmetry of the model makes that the proportionality is true for each way.

It is both a Tucker2 model with diagonal core and a Tucker3 model with an equal number of components for each way and a superdiagonal core array.

The program can also be used for the *Indscal model*, if the input is a set of K symmetric matrices with Euclidean distances or squared Euclidean distances.

References:
Kroonenberg (2008), Chapter 4, section 4.6.1 (Parafac model); Chapter 2, Section 2.4.5 (Indscal model)

1.4.4 Mixclus3 - Three-mode mixture method of clustering

Program description

MIXCLUS3 is a program for estimating the parameters of the three-mode cluster analysis model which is based on the mixture method of clustering model developed by Kaye E. Basford, University of Queensland, Australia.

The method assumes that the subjects (objects, etc.) are members of the clusters with a specific probability.

For each cluster the variables are assumed to have a multivariate normal distribution with a covariance matrix which is either common to all groups or unique for each but constant across

conditions. The means of each group are free to vary across conditions.

The program can also be used for the two-mode mixture method of clustering (MIXCLUS2).

References
Kroonenberg (2008), Chapter 16.

1.4.5 Anacor3 - Three-mode correspondence analysis

Program description

ANACOR3 is a program to perform three-mode correspondence analysis. The technique generalises standard correspondence analysis to three-way contingency tables.

A three-mode correspondence analysis will results in displays showing all three modes simultaneously, so that the relationships between the three categorical variables making up the contingency table can be examined.

A complete analysis within 3WAYPACK is done in two parts. ANACOR3 is first run from the Main screen. The next step is to proceed to the postprocessing section. In the Postprocessing Screen again select ANACOR3 to properly scale the output of the first part and produce the relevant plots.

References
Kroonenberg (2008), Chapter 17.

This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.4.6 T3Covar - Three-mode principal component analysis

Program description

T3COVAR is a program for three-mode principal component analysis especially geared towards handling multimode covariance matrices (such as multitrait-multimethod matrices).

It will take both raw data and a multimode covariance matrix as its input.

It can be used as an exploratory form of *structural equation modelling* without restrictive assumptions.

References
Kroonenberg (2008). Chapter 3, Section 3.6.3; Chapter 5, Section 5.7.3

This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.4.7 T3Gepcam - Three-mode principal component analysis

Program description

T3GEPCAM is a program to fit the Tucker3 model using a regression based algorithm for estimating

the parameters of the model. It is especially geared towards handling missing data efficiently.

The procedure and program was devised by Weesie and Van Houwelingen (1983) and has been adapted and incorporated in 3WAYPACK with their permission.

References
Kroonenberg (2008), Chapter 5, Section 5.7.3

This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.4.8 Procrus - Generalised Procrustes analysis

Program description

PROCRUS is a program to perform Generalised Procrustes analysis. The technique computes the optimal consensus configuration for a set of configurations and will rotate all configurations optimally to this consensus configuration (or centroid).

The data for this program are assumed to be a set of configurations with the same number of columns, but the number of rows need not be the same and some of them may be missing.

References
Kroonenberg (2008), Chapter 3, Section 3.73 and Chapter 7, Section 7.8.4.

This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.4.9 Triadic - Parallel factor analysis

Program description

TRIADIC is a program for fitting the Parafac model, i.e. a three-mode component model with a superdiagonal core matrix. The program uses a non-standard algorithm using triadic fitting. It allows for easy inclusion of constraints on the components.

References
Kroonenberg (2008), Chapter 4, Section 4.10.7; Chapter 5, Section 5.6.3
.

This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.4.10 Simulcmp - Simultaneous component analysis

Program description

SIMULCMP is an analysis program to carry out simultaneous component analysis as developed by Kiers and Timmerman, University of Groningen.

The present program is an adaptation of the MATLAB program by Marieke Timmerman (see Timmerman (2001)).

References
Kroonenberg (2008), Chapter 4, Section 4.6.2; Chapter 5, Section 5.6.3
.
This program is not yet available in this version of 3WayPack. Please contact The Three-Mode Company for a copy of this program.

1.5 Overview postprocessing programs

Introduction
The postprocessing programs process the output of a three-mode program in order to further enhance and facilitate interpretation. This can take the form of making special plots for the component spaces, rotating component matrices and/or core arrays for easier interpretation, or simply calculating residuals.

Available programs
The major programs which operate on the output are
- JOINTPLT - the creation of a joint biplot of two modes given component of the third mode plus nested-mode biplots.
- RESIDUAL - the calculation of residuals to assess the overall fit of the model with respect to the data. Model-based estimates for the data values are compared with the observed values.
- T3ROTATE - rotating components with counter-rotations of the core array. Options: varimax, oblimin and constant first component.
- ROCOCO - varimax rotation of components and core array jointly in order to get optimal simplicity in all matrices or in those especially selected for rotation.

1.5.1 JointPlt - Joint biplots & Nested-mode biplots

Program description

JOINTPLT constructs joint biplots for two modes (the *display modes*) given a component of the third mode (the *reference mode*). In addition, the program can construct nested-mode biplots which show all first-third combination components in the space of the second mode. For some data sets the coordinates of such plots can be considered component scores. Especially ordered modes can be used to display their trajectories in the space of the second mode.
No joint plots are available for TRILIN as they is incompatible with the Parafac model fitted by that program. Please use per-component plots instead.

References
Kroonenberg (2008), Chapter 11, Section 11.5.3.

1.5.2 Residual - Residuals after three-mode analyses

Program description
This program will calculate standardized residuals and plot them against the standardised fitted data. Plotting can be done per frontal slice or for the entire data set. This program also provides an option to write either the residuals and/or the fitted data to a file called <jobname>. res and <jobname>. fit respectively. Note that the fitted data are such that they perfectly fit the model, so that they can be used for simulation etc.

References
Kroonenberg (2008), Chapter 11, Section 11.3-11.7.

1.5.3 T3Rotate - Rotating components

Program description
This program will rotate the components of any or all of the three ways.

The rotations/transformations available are
1. *Varimax rotation* on orthonormal components (equivalent to the Harris-Kaiser cluster rotation);
2. *Oblimin transformation* of orthonormal components;
3. *Optimally constant first component*. This option can sometimes be used to facilitate interpretation, especially of reference modes for joint plots or for time modes.

References
Kroonenberg (2008), Chapter 10, Section 10.3.

1.5.4 Rococo - Rotating components and/or core array

Program description
ROCOCO (Rotating core and components) is a program to rotate the component matrices and the core array simultaneously. i.e.any combination of component matrices and core array can be rotated simultaneously with the varimax procedure.

The program in 3WAYPACK is a Fortran version of a MATLAB program developed by Henk A.L. Kiers, University of Groningen.

References
Kroonenberg (2008), Chapter 10, Section 10.4.

2 Setting-up the analyses

Introduction

The work flow of an analysis is that the specification is done in the interface TWPACK. The results from the specification are stored in command files. Then the interface calls an external program which uses the command file for its execution. When the external program has done its work, it will have produced a number of output files (text files, plot files, and browser files) which can be inspected via the interface and which can be further processed via the programs available through the postprocessing section of the interface. The effect of this is that the whole suite will produce a fair number of output files which will reside in the directory of the data file. Further details can be found in the appropriate sections.

The Figure **Work flow program suite** gives a schematic overview of the links between the parts of the program.suite. The last figure shows the structure of the interface TWPACK itself.

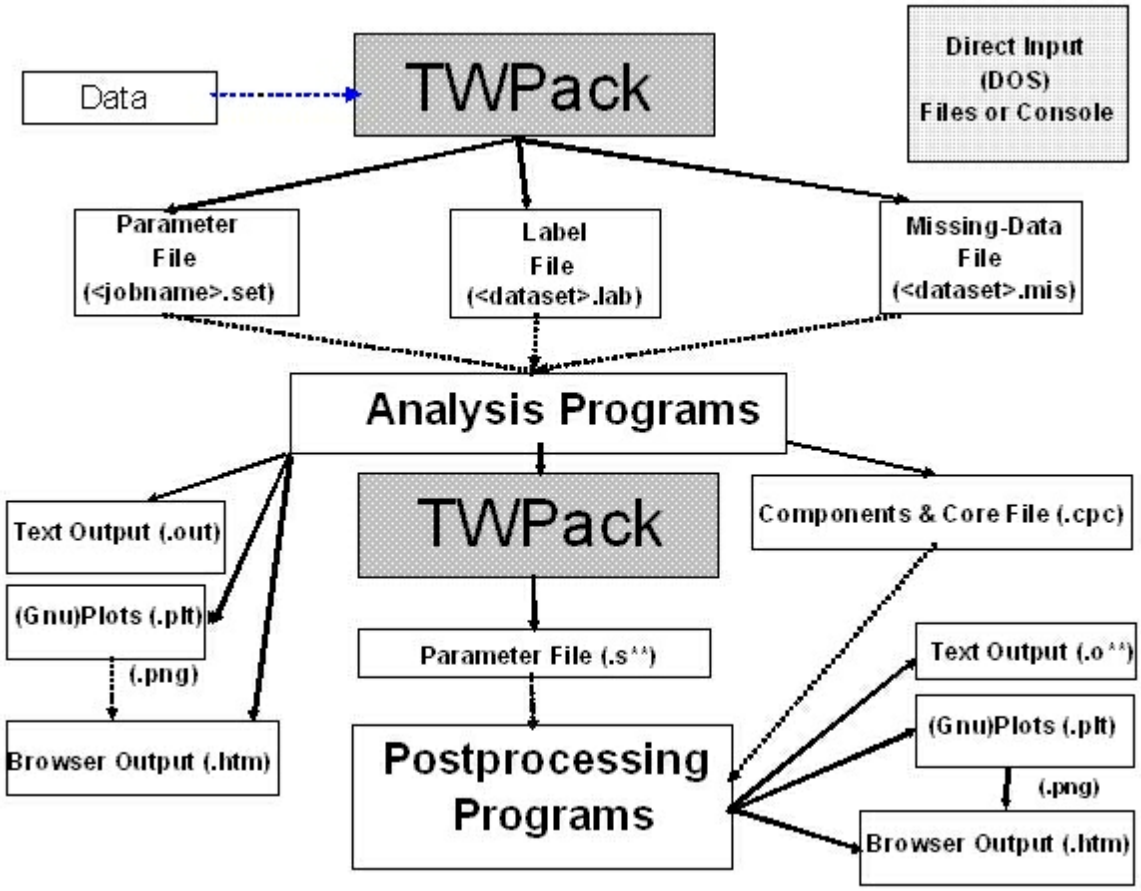


Figure: Work flow program suite

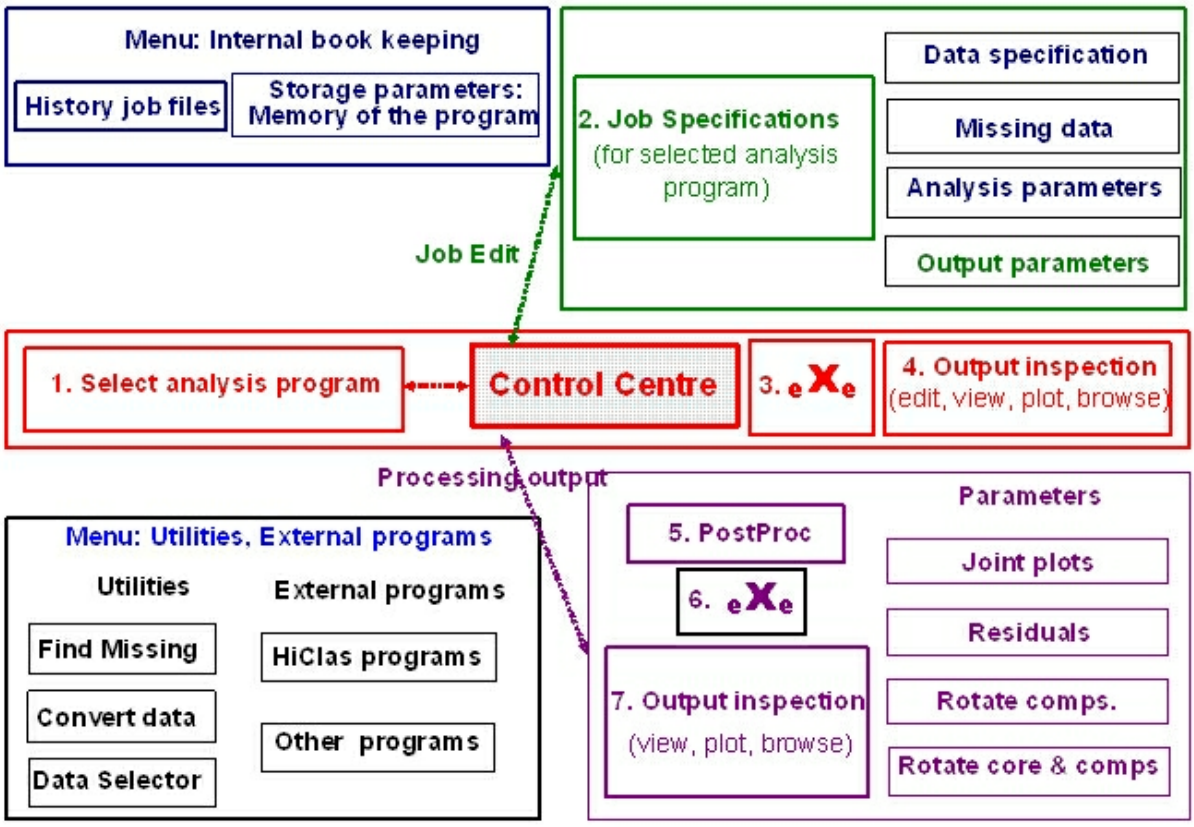
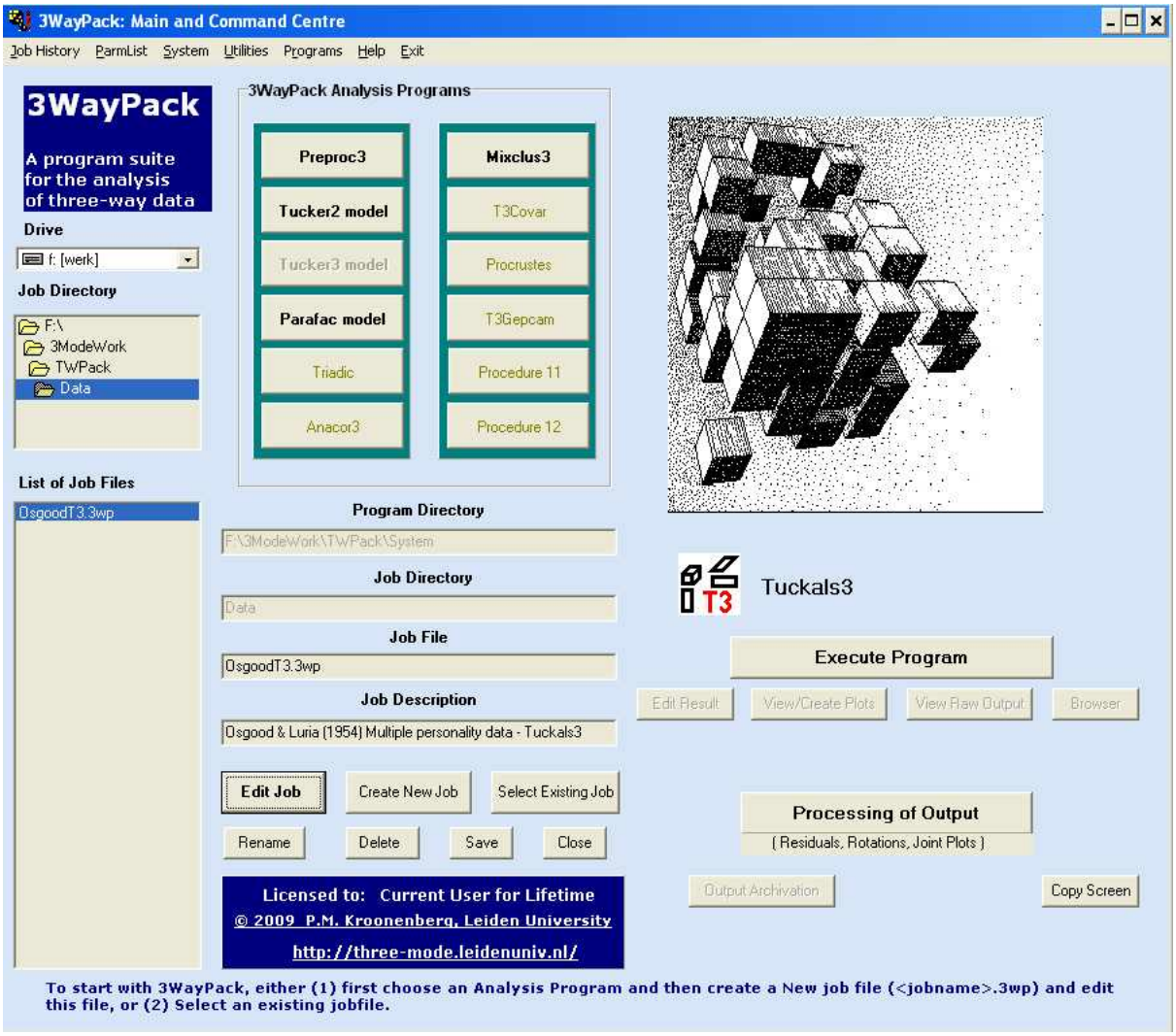


Figure: Structure of the interface TWPack - $_eX_e$ = Execute

2.1 Main screen

Introduction

The Main Screen of the program suite is the command centre for carrying out three-way data analysis. All analyses start and end here. After selecting an analysis method and specifying parameters for the method selected, one returns here to actually execute the program, to inspect, print and plot the output. Further processing of the output starts here as well. Several other programs and utilities can be accessed from Main screen. In other words, it is the alpha and omega of a three-way analysis.



Main tasks

The main tasks to be carried out for most three-mode analyses are the following.

Preliminaries

- Select the directory of the data set to be analysed

Preprocessing (if applicable)

- Create a job file for preprocessing the data set
- Edit the preprocessing job file and execute the analysis (details see below).
- If you have created a new data set, inspect it, and it is suggested to change its extension from `pp3` to `dat`

Main analysis

- Choose an analysis method
- Create a job file which will contain the parameters of the analysis. It is analogous to a syntax file or command file of other programs.
- Edit the job file, i.e. specify the data set and parameters for the analysis. Each analysis method has its own job file.
- Execute the analysis
- Inspect, print and plot the output.
- Archive output files to prevent overwriting with the next analysis

Postprocessing

- Calculate residuals, make interpretative plots, rotate the coefficients (if appropriate).

2.1.1 Main menu bar

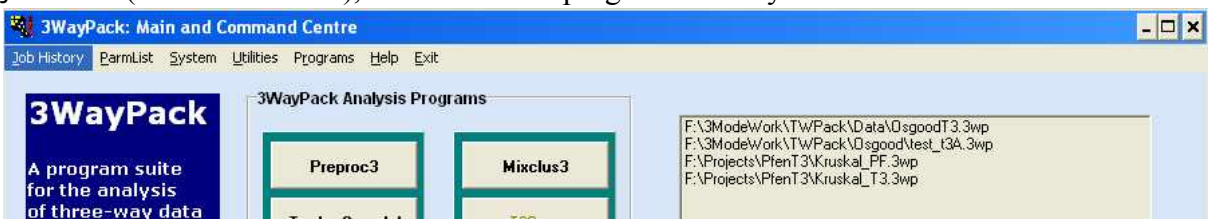
Introduction

While the Main Screen is the command centre for TWPACK several additional facilities are provided via the Main menu bar.

- Job history
- ParmList
- System
- Utilities
- Programs
- Help
- Exit

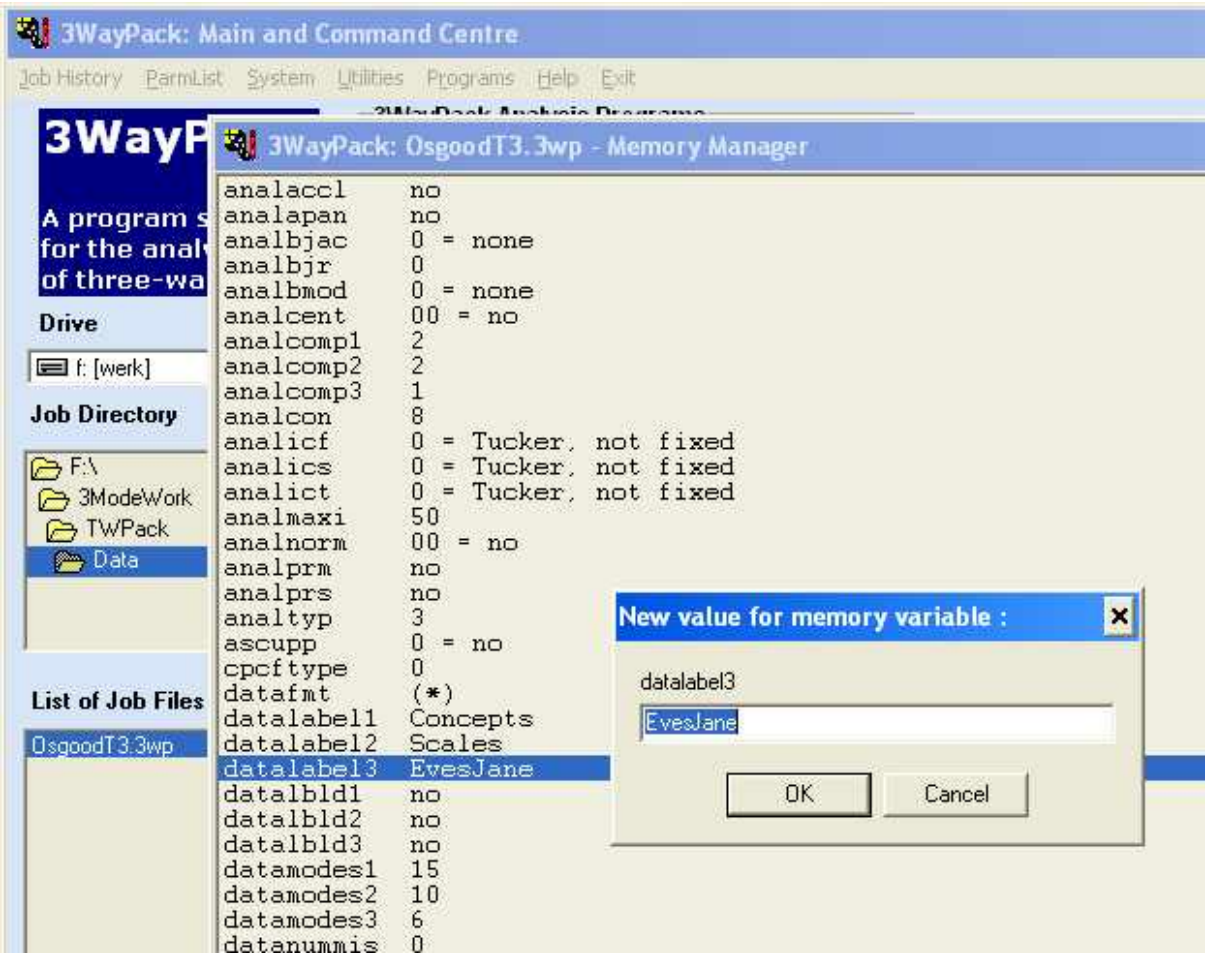
Job History

The program suite 3WAYPACK keeps track of the job files you have made and saved. By clicking on Job History in the main menu bar a list of previous job appears in place of the logo of *The Three-Mode Company*. On clicking the desired job, it will be loaded into memory ready for further work. Mostly you will then go to the **Edit Job** button to adjust the parameters. The job history file, joblistfile (without extension), is located in the program directory.



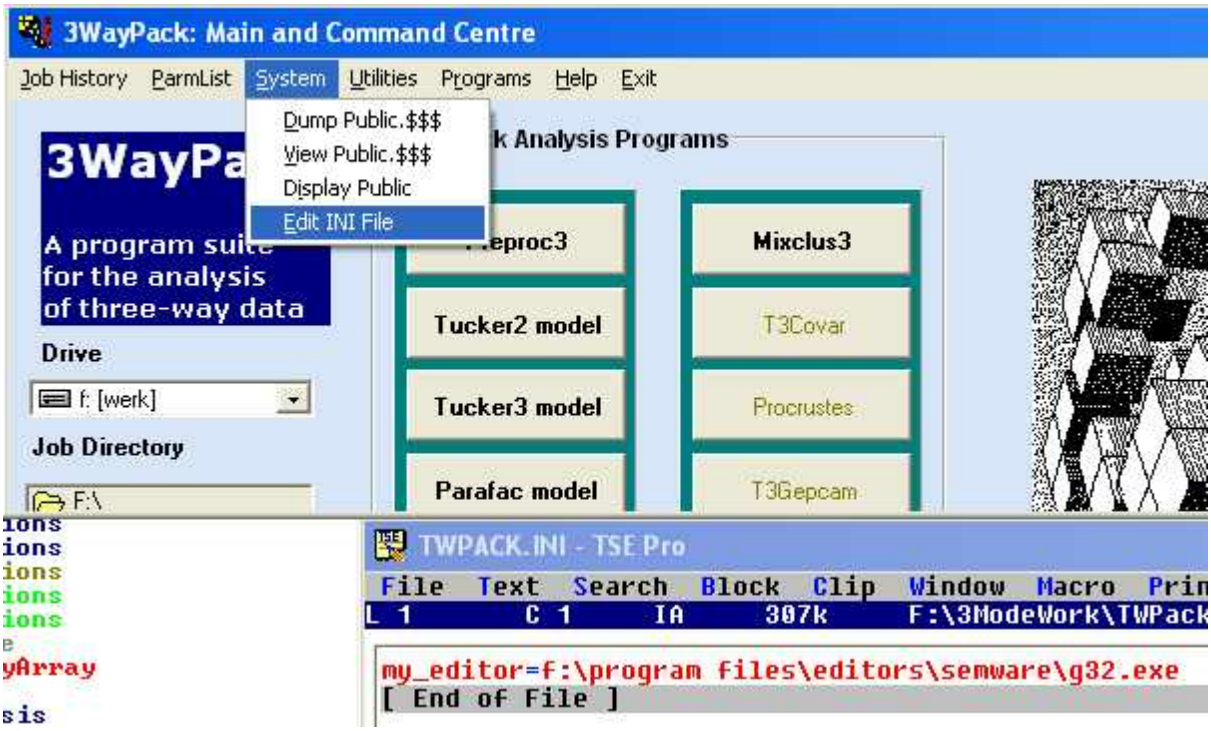
ParmList

As can be seen below, after pressing ParmList on the main menu bar, the Memory Manager for the current job OsgoodT3.3wp will appear. In the example on the screen, the user seems to want to change the label for the third mode from 'EvesJane" into something else.



System

The first options in the menu item are for development purposes only. The first relevant entry is the Edit INI File one. Clicking on this entry will open the ini file of the program (TWPack.ini) which allows you to specify you favourite editor for examining and modifying your output. NOTEPAD is the default editor but it is inadequate for serious work. If you are using NOTEPAD, please set its font at *Terminal*, its font style at *Regular* and its font size at *9* or *12*; unfortunately no 10point font is available.



Utilities

At present there are three utility programs.

CASE-TO-SLICE for changing three-way data from a case-mode format to a slice-mode formatand

FIND MISSING for locating missing data and creating a file with the locations of the missing data.

DATA SELECTOR for selecting subsets from the data array.

Other similar utilities will be included later.

Programs

External programs, not necessarily developed by *The Three-Mode Company* can be made available from the Programs entry of the main menu bar. The two programs listed dealing with hierarchical binary clustering methods are not distributed by *The Three-Mode Company* but can be obtained from Eva Ceulemans, Andreas Vesaliusstraat 2, 3000 Leuven, Belgium (eva.ceulemans@ped.kuleuven.be). After adding them to the program directory, the programs will be recognised by TWPACK.

Help
Clicking this option will bring you to this help file.

Exit
Is just that.

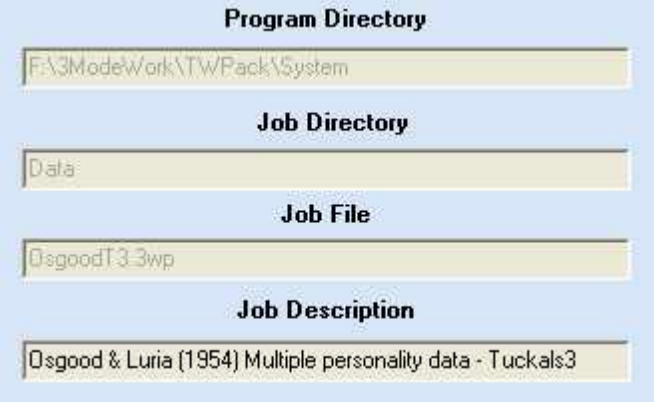
2.1.2 Drives & Directories

Introduction
There are three *directories* which are central to the operation of 3WAYPACK.

- *Program directory*, i.e. the directory of the interface TWPACK, the executables of the programs and other program-related files.
- *Job directory*, i.e. the directory in which the job file of the current job is located. One can decide to store all job files in a common directory or store each job file with the data set it refers to.
- *Data directory*, i.e. the directory in which data reside as well as its label file, missing-data file, and external configuration file. This directory will also be used for storing all files generated by the programs pertaining to the job at hand.

Drive and directories
On the left-hand side of the Main Screen, the current drive and job directory are listed and all existing job files in this directory. Job files have the extension <.3wp>.

To start an analysis first go to the directory in which an existing job file is located or a new one should be located. Usually this will be chosen to be the data directory, the more so because each new type of analysis, even with the same data, requires a new job file. Another reasonable alternative is to store all job files in a special directory, e.g. a subdirectory of the program directory.



Directories
On the Main Screen we also find a listed of all current directories including the program directory (which cannot be changed), the job directory, and the name of the current job file (if available).

Job directory
The contents of the job directory box cannot be changed manually, other then by choosing another directory.

Job file
If a job file has been chosen it will be listed here. The job name cannot be changed here. Please use one of the job file options.

Job description
The job description is optional, but filling this out might help in more easily identifying the content of the job file later. This title is for documentation in TWPACK only.

Starting an analysis
To start an analysis one either has to define a new job file or use an existing job file (see Job files) .

An existing job file can be selected by (1) going to its directory in the drive and directories panel or (2) via the Select Existing Job button (see also Job files) . A directories will give a listing of all files with the name <jobname>.3wp. The extension <.3wp> is obligatory for job files, and it is the only one TWPack recognises.

2.1.3 Job files

Introduction
A job file contains the memory of an analyses. It is what a syntax file is to program packages like SPSS. The details of an analysis as well as of the data set analysed are stored in the job file so that if the analysis needs to be done again or in a slightly different form at a later stage, all the details are present in the job file. For an example of a job file see below.

If the name of the job file is osgood.3wp the first part or JobName is the basic identifier of an analysis. All files created by 3WAYPACK will have the file name <jobname><xx>.<id>, where <id> identifies the content of the file, and <xx>, if present, is generally a further identifier of the postprocessing program used. Please choose your job name with care to avoid conflicts with

already existing file names and preferably give it a name which indicates the analysis used. For instance, a good choice would be OsgoodT3 to indicate that a Tucker3 analysis has been performed on the Osgood data. It is important to realise that a job file links a data set and a particular analysis. Once chosen, it cannot be changed. However, a data set can be linked to any number of job files. This is done to keep all output files of an analysis together.

Job files may be edited by hand outside the program but please do this with care because incorrect spellings or options can wreak havoc with its interpretability by the interface. From the main menu bar using the option ParmList, a job file can also be opened and edited. The interface has a special program for this (see the edit job file option).

Operations on and with job files

Edit job (file)
Edit the current job file specified in the Job File box. This is the button that will lead you to the specifications of the entire analysis.

Create new job (file)
After first having chosen an analysis program you can create a new job file for a new analysis, either with the same or different data. If another job file is still active you will be asked whether you want to save it.

If your data are not in the current directory, first go to the data directory, and only then press New. After you have saved the job file (<jobname>.3wp) in the data directory, the Job specifications screen will automatically appear to let you specify the details of your job, the data set, and the analysis.

Select existing job (file)
Clicking on the Select Existing Job button will allow you to select an already existing job file. You can also select an existing job from the List of Job Files panel on the left-hand side of the screen, or click on the Job History entry in the main menu bar.

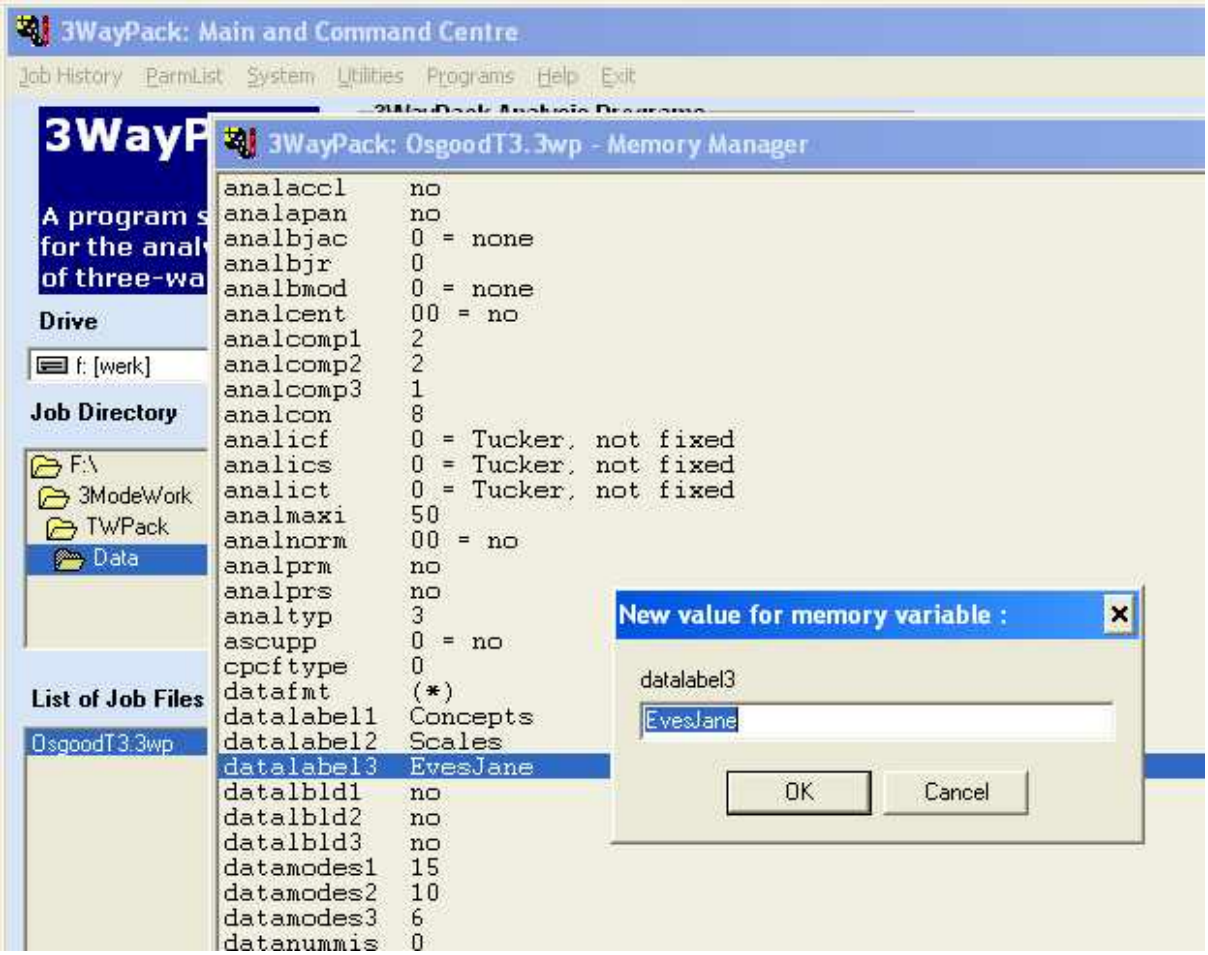
Rename (job file)
Rename an existing job file. Nothing else will happen.

Delete (job file)
Delete an old job file. You will be asked to confirm the deletion.

Save (job file)
Save the present job file. During the operations the program will save your job file whenever necessary, but if you change jobs it is wise to do it yourself as well.

Close (job file)
Save and close the present job file to select to another job file.

Editing the job file from within the program
As can be seen below, after pressing ParmList on the main menu bar, the Memory Manager for the current job OsgoodT3.3wp will appear. In the example on the screen, the user seems to want to change the label for the third mode from EVESJANE into something else.



2.1.4 Execute procedure

Introduction
After choosing a Job file and editing it for the desired analysis, pressing the Execute Program button will start execution of the current analysis program.

Figure: Execute Program

During execution a so-called 'Dos box' appears which shows the progress of the analysis program. Especially with more time-consuming analyses this box will be visible for a while. It will generally display the progress of the iteration procedure which is working on estimating the parameters of the model analysed.

After the analysis program is finished this Dos box will disappear and the interface will take over again. Depending on the options chosen two or four of the buttons related to the output will be enabled (see Figure **Execute program**).

If the text output file and the plotting file for the present job already exist the old output files will be overwritten. If this is undesirable, the existing output files should be renamed beforehand using the Archivation Output button.

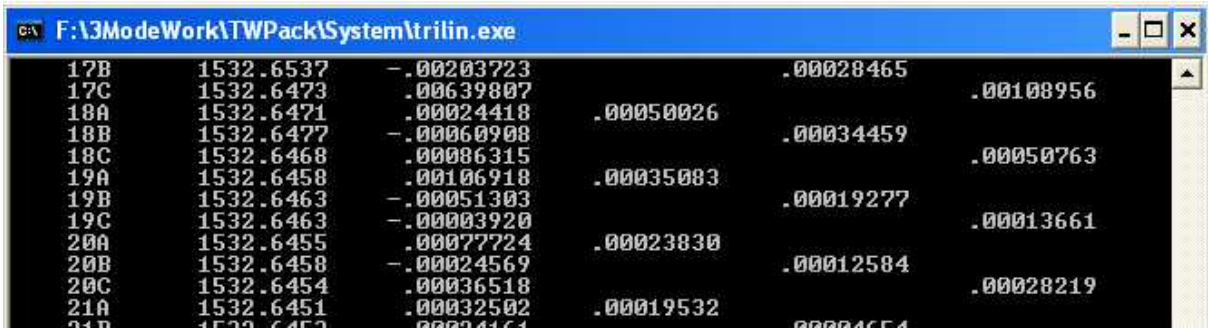


Figure: Dos Box

2.1.5 Output and plot files

Introduction

The analysis programs can produce three kinds of output files: (1) Basic text files with text-based plots; (2) Plot files and (3) HTML files. To view these files TWPACK calls external programs. The text output file can also be viewed but not edited with the internal viewer (View Raw Output). During the viewing or editing of an output file TWPACK itself is not accessible. It only becomes accessible again after the viewing program has been closed. If it is desired to continue with the analysis while inspecting output, the output should be opened outside TWPACK, for instance via the Windows EXPLORER or any other file management program or editor.

Text output file

The text output of the programs has the extension .out and its full name is <jobname>.out. Any previous file with the same name will be overwritten is lost unless it was archived earlier under a different name (see also the Output Archivation button).

Viewing text output file
The simplest way to inspect the output file is via the View Raw Output button. This will only allow for inspection, no changes are possible.

Editing text output file
The <jobname>.out can be edited with the default editor of your system (generally NOTEPAD) or with a user-specified editor (see the main menu item System for specifying your own favourite editor). TWPACK calls this editor with the name of the output file as its sole parameter. You may also use the editor to rename output files but the archivation facility in the program can do this as well.

Browser output
Because the output of three-way programs can be quite voluminous and therefore difficult to negotiate, the program suite has the option to produce (HTML) output which can be viewed in a browser. The structure of the HTML files is such that there is a navigation bar on the left and a display part on the right, similar to such programs as, for instance, SPSS AND SAS.

In the Print/Plot Options screens of the analysis programs and the postprocessing programs, the HTML code for viewing the output in a browser can be requested.

3WAYPACK has its own internal browser, but outside TWPACK the browser output can be inspected using any browser.

Plots
If 'real' plots (i.e. publication-quality plots) are requested the analysis programs produce a file with plotting instruction for an external program GNUPLOT. The programs produce *either* a plotting file with vector coordinates (graphics type.wmf; .emf) *or* a plotting file with bitmap instructions (graphics type: .png).

The first type of plotting file is meant for creating plots which can be viewed and stored on the clipboard for further processing in graphical and presentation software, such as Adobe ILLUSTRATOR, Microsoft POWERPOINT and Lotus FREELANCE. The second type of plotting file produces a bitmap or binary plotting file (extension .png) which cannot be edited but which are used in a browser.

Which type of plotting file is produced depends on whether a browser-readable output has been requested. It is thus not possible to get both types of plot files at the same time. If both are desired, the analysis program should be run twice with renaming the first plotting file (via the archive facility) to prevent it from being overwritten. Specifying HTML output is done generally on the Print/Plot Options screens of the analysis programs and the Options screen of the postprocessing programs.

View and store plots (vector graphics; .wmf; .emf)
If publication-quality editable plots have been requested and browser output has *not* been requested, editable plots will be created by the program GNUPLOT after pressing the View/Create Plots button. The analysis programs will have generated the command language for GNUPLOT which is stored in the file <jobname>.plt. By pressing the button the actual plots are made and they are shown one by one. How the plots can be stored (as .emf or .wmf files) is explained in the GNUPLOT entry of this help file.

Creating plots for the browser
Plots to be displayed in a browser are generally bitmaps or binary files which cannot be further manipulated other than resized. In 3WAYPACK these plots will have the name <jobname><xyz>.png, where <xyz> depends on the type of plot that is produced. Please do not change the <xyz>

specification manually because the browser file needs these names to properly display the graphs. For the plots to be shown in the browser they should be made before the browser is called via the Browser button.

2.1.6 Warnings-and-errors file

Introduction

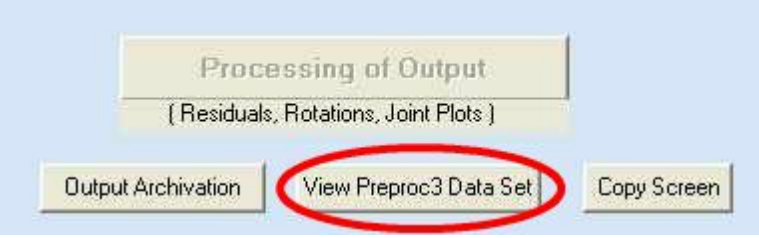
The analysis programs have an extensive warning and error checking system. Once a warning has been issued or an error has been detected, the programs will open an error file (<jobname>.err) to list the problem in question. The same message is also printed in the output file (<jobname>.out). When the error file is created a button will become available for inspection of this error file. Note that it may occur that only warnings are issued but that the analysis itself does not contain any errors.

Should you find errors due to an analysis which did not generate an error file, please relay this to *The Three-Mode Company*.

2.1.7 View preprocessed data

Introduction

The program PREPROC3 will create new data sets depending on the options chosen. In particular, if PREPROC3 modifies the data in any way it creates a new data file in the same directory where the original data file resides. The name of this data set is <datasetname>.pp3.



Viewing the Preproc3 data set

This new data set can be viewed using the View Preproc3 Data Set button on the Main Screen. You may consider to rename the extension of the data set from .pp3 to dat, but this is not necessary. The renaming should be done outside 3WAYPACK or you could open.the output file with the editor and change both the name and its extension from within the editor.

Once the data set has been viewed **the button will be no longer available**.

2.1.8 Postprocessing of output

Introduction

The output of the three-way analysis programs can be analysed further via a series of postprocessing programs. In particular, residuals can be calculated, core arrays and components can be rotated, joint biplots and nested-mode plots can be made.

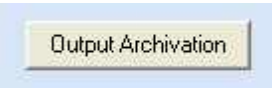
Details
The postprocessing programs use special files written by the analyses programs, so-called CPC-files (<jobname>.CPC), which should not be deleted from the data directory. These programs are available from the the PostProcessing screen which can be accessed by clicking on the Processing of Output button.

Content CPC-files
The CPC-files contain the components and core matrix (TUCKALS) or component weights (TRILIN) as well as the overall sum of squares and fitted sum of squares. This information forms the basis for all postprocessing programs.

Postprocessing programs
- JOINTPLT: Joint biplots of components of two ways given a component of the third mode and nested-mode biplots;
- RESIDUAL: Computing and displaying residuals and residual versus fitted data plots, output fitted data and residuals;
- T3ROTATE: Rotation and/or transformation of component matrices;
- ROCOCO: Rotation of core array and components.

2.1.9 Archive/Rename output files

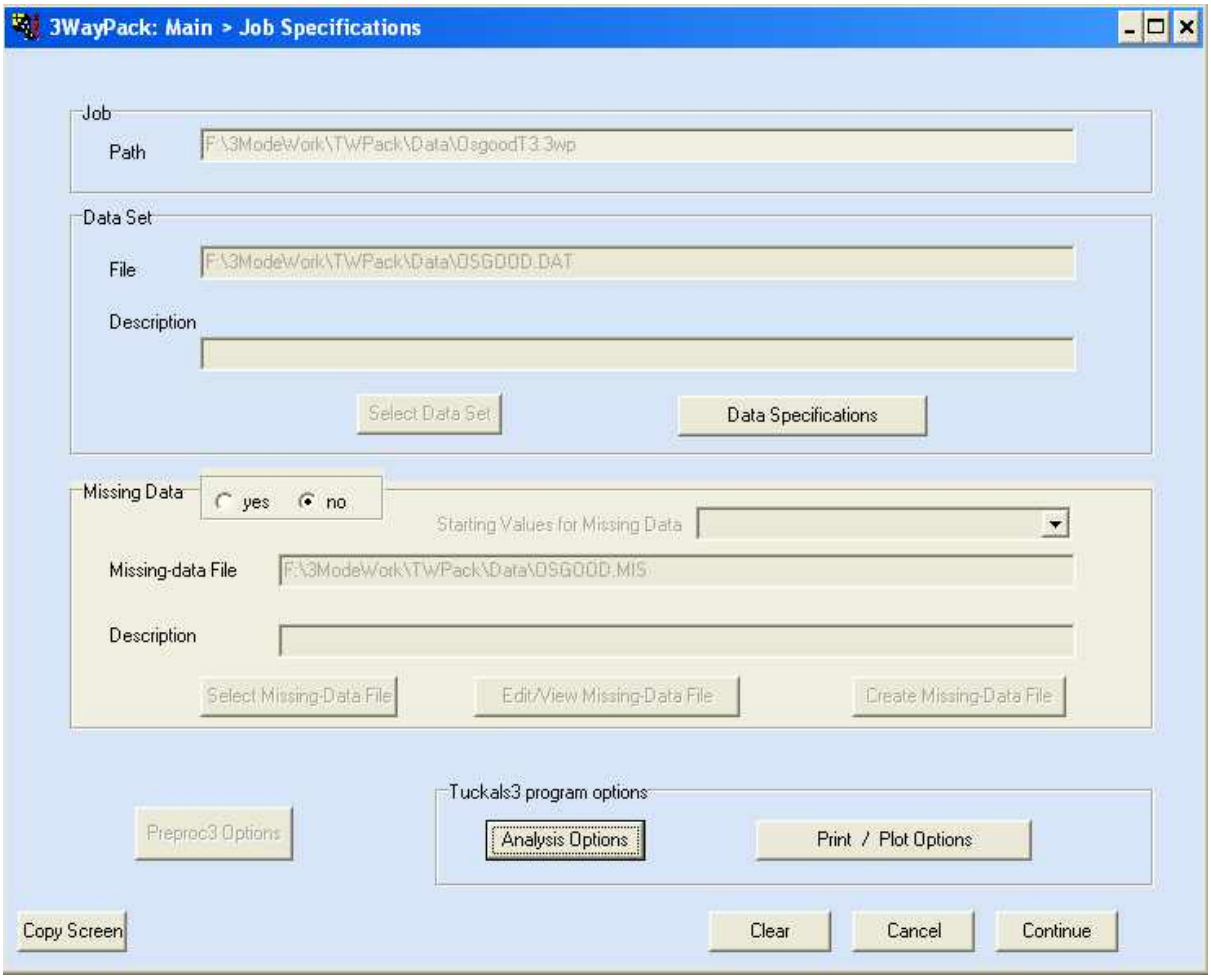
Introduction
Output files in 3WAYPACK are named on the basis of the name of the job file, e.g. <jobfile>.out. Therefore each time an analysis is run with the same job file all existing output files with the same job name are overwritten. The Output Archivation option allows renaming the following output files.



Output file names overwritten in 3WayPack via the Main screen
- <jobname>.out the basic output file
- <jobname>.plt the plot command file
- <jobname>.htm the general browser command file
- <jobname>Br.htm the left-hand navigation browser file
- <jobname>Bd.htm the right-hand content browser file

2.2 Job specifications screen

Introduction
The Job specifications screen is the pivot screen from which the specifications of a job at hand are made. The first task is to select the data set of the job and provide its specifications. If present, missing data are then detailed, followed by the setting the analysis parameters and the output parameters related to printing and plotting. When all this is done one can return to Main Screen to execute the program and after that inspect and postprocess the output.



2.2.1 Listing of job file

Introduction

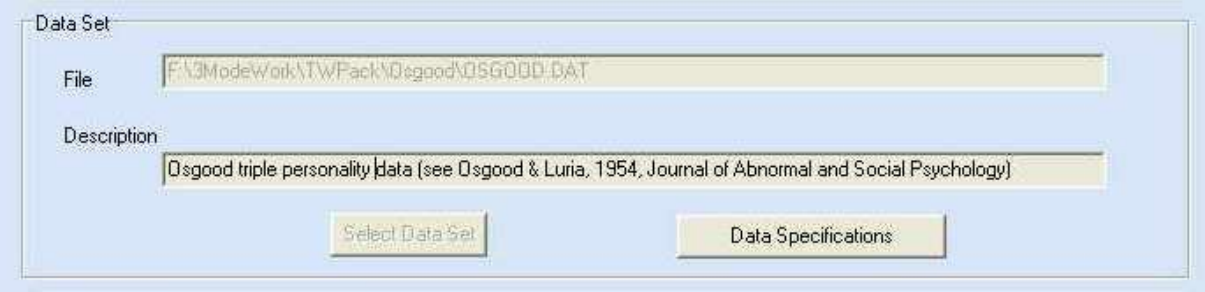
This field indicates the name and location of the current job file. It cannot be edited in this screen.

If you want another job, press Cancel and go back to the Main Screen.

2.2.2 Data set specifications

Introduction

This part of the screen allows the specification of the location of the data file and that of the data themselves.



Location
The location of the file is listed automatically if the information is already available from the current job file. If the information is available it means that you are working with a previously created job file. In that case you cannot change the data set. Once a job file and data set are linked the same job name cannot be associated with another data set. This prevents all kinds of errors and keeps the outcomes of the analyses done with this job file together.

Specifications
Select data set
When no data set is yet associated with a job name go to the `Select Data Set` button for selecting the required data set. Note that the desired data set must already be available in the directory for selection. 3WAYPACK itself has no facilities to type in data.

Data specifications
After selecting the data set proceed to the `Data Specifications` to specify the size of the data, the input format and the labels for the three modes..

Description
The `Description` field may be used for specifying a characterisation of the data set. However, this information is not printed in the output and is for reference on this screen only. The title for the job used for labelling the output can be specified on the `Analysis screen` for each program: Tucker2 model; Tucker3 model; Parafac model; Mixclus3.

Manual input data
There are no provisions in 3WAYPACK for typing in data. The best advice is to create a plain ASCII or text file either via a spreadsheet, a statistical package, or by directly typing them in an editor. Please note that Word files are not appropriate, they first have to be converted to plain text before they can be used. To be consistent it is may best to only use the extension `.dat` for data files.

2.2.3 Analysis options

Introduction
Given that the data have been properly specified the next task for the user is to provide the details for the analysis, such as the number of components, the number of analyses, execution of an external analysis, bootstrapping, jack-knifing, specification of centring, normalisation, type of initial configurations, algorithm specifics, etc.

Analysis options button
This button will lead you to the Analysis Options Screen for the current analysis program. On the <ProgramName> analysis screen you can specify the necessary details of your analysis. The button is sensitive to the program you have chosen so that it will bring up the appropriate screen for the analysis.

2.2.4 Print and plot options

Introduction
Given that the data and the analysis options have been properly specified, print and plot options can be detailed such as printing of components, iteration process, detailed output, plotting of components (line plotting or real graphics), printing output in browser format, etc.

Print/plot options button
This button will lead you to the Print/Plot Options Screen for the current analysis program. The button is sensitive to the program you have chosen so that it will bring up the appropriate screen for the analysis.

Specifications
On the <ProgramName> Print/plot options screen you can specify which and how much output you would like to have. The output can be quite voluminous for large data sets so choose your printed output and plots with care.

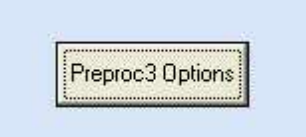
Basic output
The basic output is plain text output including the requested plots in a simple line format.

Publication-quality plots
In addition you can request that the same plots are made by a special plotting program GNUPLOT which can produce publication-quality plots. Often such plots will have to be further processed to make them completely satisfactory.

Browser output
The program will produce at request HTML-output suitable to be displayed in a browser like Firefox, Safari, Internet Explorer, Opera, etc. The program suite also has a built-in simple browser. The browser output allows for easy navigation by using frames like several other major statistical software packages.

2.2.5 Preproc3 options

Introduction
PREPROC3 only has a single options screen, Preproc3 Options Screen. Therefore it has a special button for specifying all aspects of its analysis. The button only becomes enabled if the PREPROC3 program has been specified for the job.



2.2.6 Missing-data specification

Introduction
Virtually any data set has a number of missing values about which decisions have to be made. The analysis programs have facilities for handling them during the iterative processes to estimate the parameters. In essence the model parameters are estimated at each iteration and using the newly estimated parameters in that iteration estimates for the missing data are calculated. This process is single imputation, i.e. after convergence of the algorithm one single estimate is generated based on the model. This in contrast with multiple imputations in which several estimates are produce allowing an assessment of the sensitivity of the solution to the missing data.

3WAYPACK has a slightly unusual way of working with missing data. Two aspects are relevant for properly handling missing data:
1. *Locations* of missing data in the data array;
2. *Starting values* for the missing data to start the iterative procedure in an analysis program.

At present there are several ways to construct the file containing the locations of the missing data in the data array. In a later version an attempt will be made to make this process more transparent and robust.

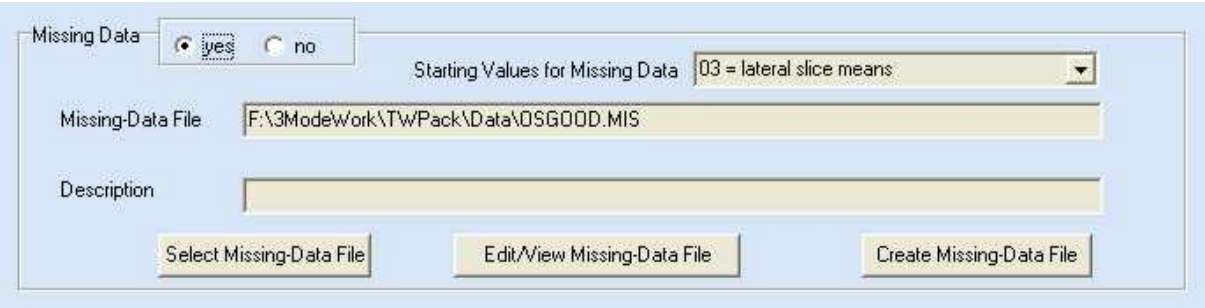


Figure: The missing-data panel

Ways of specifying starting values
The programs use or create starting values for imputing the missing values. During the model estimation the missing data are estimated as well (effectively using an EM-procedure). But for the estimation processes to start, initial values or starting values for the missing data are necessary. Different starting values may have different effects on the outcomes of the analysis.

If a data set contains missing data there are several options to specify starting values for filling up the holes in the data set.

1. In simple cases use the Missing-data panel.Click the Yes radio button to activate the missing data section
2. For more complex cases use the PREPROC3 program from the Main Screen.

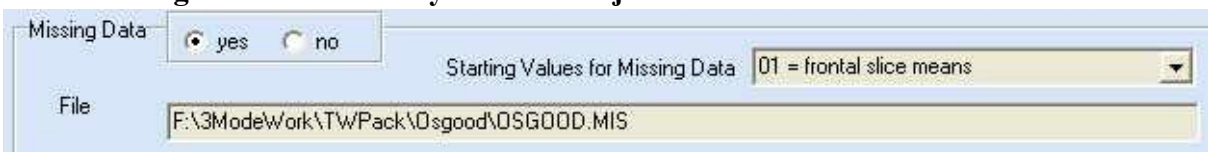
Missing-data file
The analysis programs except MIXCLUS3 need a special file containing in free format the coordinates (i,j,k) in the data array where the missing data are located. A missing-data file with the name <datasetname>.mis has to be created in which these locations of the missing values in the data set are stored. For details see the Missing values screen.

Description of missing values
The Description field may be used to provide information about the missing data. This information will, however, not appear in the output, and is for your reference in TWPACK only.

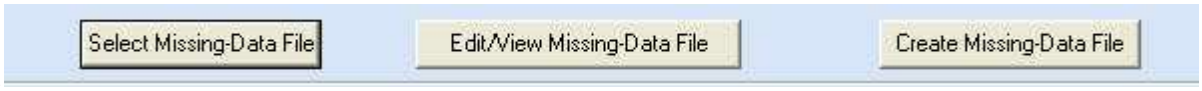
Locations of missing data

Introduction
In order to get the starting values for missing data and in order to be able to estimate them the analysis programs need a file with the *locations* in the data set of these missing data.

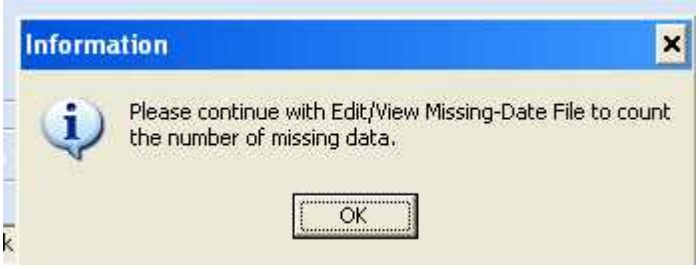
Name missing-data file is already available in job file.



The field Missing-Data File indicates the name and location of the missing-data file, which is a file containing the locations of the missing data in the data file. It is listed automatically if the information is already available from the current job file. Note that the only acceptable name for a missing-data file is <datasetname>.mis preceded by its complete path. The missing-data file can be edited or viewed after clicking the Edit/View Missing-Data File button.The number of missing data should also be available in the job file. If in doubt, go to Edit/View Missing-Data File and the number of missing values will be counted.



Missing data file <datasetname>.mis is present on disk but not present in job file
If a missing-date file is available but has not yet been used or linked with the present job, click the Select Missing-Data File button to select the file. After that you have to proceed to Edit/View Missing-Data File to let the program count the number of missing data.
You will be given a warning to that effect:



Please note that you may get errors when you already have user-defined estimates for the missing data in your data, but have not specified them in the Starting Values for Missing Data combobox (See Figure **Starting values**).

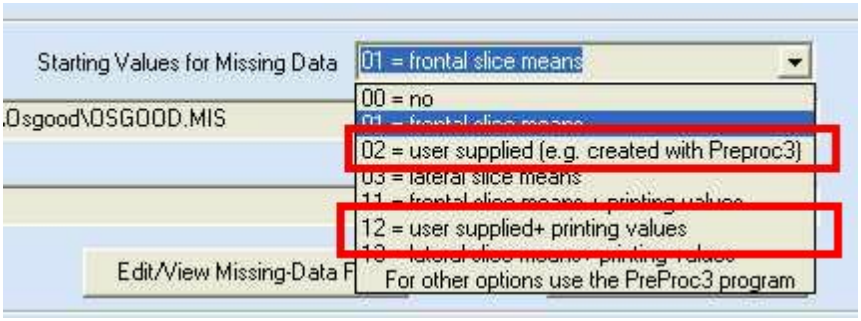


Figure: Starting values

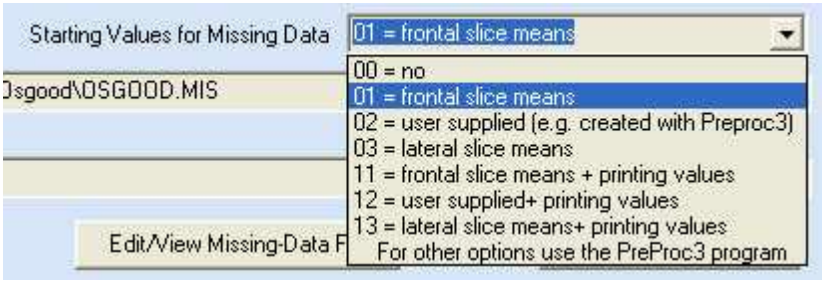
Missing data file does not exist

If this file does not exist click the Create Missing-Data File button and the Missing Values Screen will appear to let you make the specifications. In particular the codes for the missing values have to be specified. Their locations in the data set will be written to the Missing-data file (<datasetname>.mis) and they will be counted. The file will be placed or should reside in the data directory. Note that if an already existing missing-data file is selected it will be emptied first. Cancelling does not undo this.

Starting values for missing data

Introduction

All three-mode component analysis programs need starting values for missing data. Once these are available the model is estimated as if all values are valid. After each cycle of estimation of all parameters, the missing values are recalculated to conform with the model. This approach is generally referred to as an EM-algorithm (Expectation-Maximisation).



Options for starting values

- Lateral slice means (Default)
- Frontal slice means
- User supplied values. In this case, it is assumed that the starting values are already present in the data set at the locations of the missing data.The idea behind having an option User supplied starting values is that they can be generated elsewhere. For instance by PREPROC3 (see Main Screen) is especially designed to do this. This program is also capable of calculating starting values for a large variety of three-way ANOVA models.

For all three cases you can request that the initial starting values are printed in the output.

2.3 Data specifcations screen

Introduction

This screen should be used to specify the particulars of the data to be analysed, such as the dimensions of the data array, labels for the modes, and the data input format.

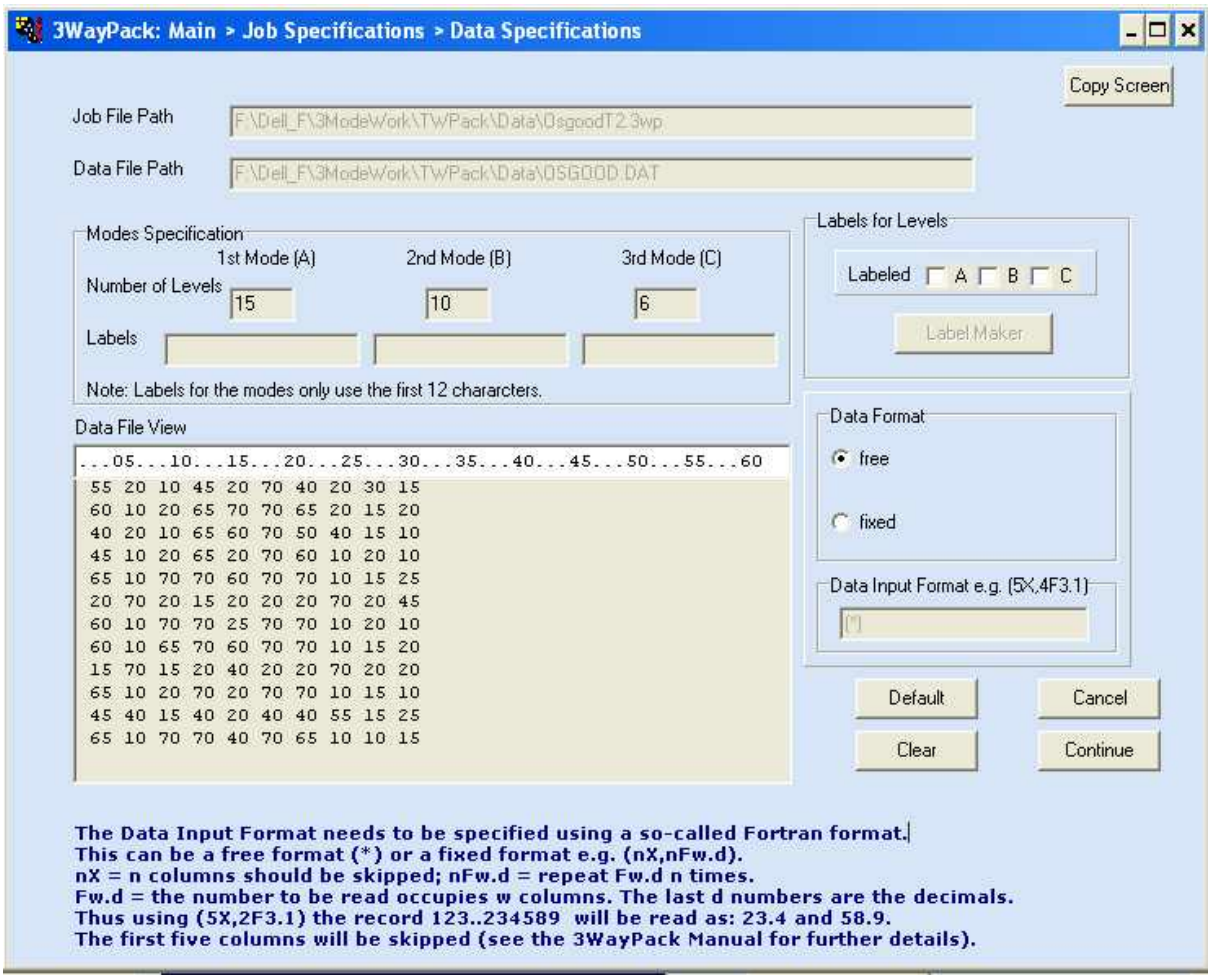


Figure: Data specifications screen

Data arrangement

The data should be stored in frontal-slice mode, i.e. as K frontal slices of size $I \times J$ arranged as a *tall matrix* with KI rows and J columns (see Figure **Slice-mode arrangement**). However, often data are in case-mode, i.e. the frontal slices of size $I \times J$ are positioned next to each other, thus arranged as a *wide matrix* with I rows and KJ columns (see Figure **Case-mode arrangment**). Typically this is the arrangement used in standard software packages like SPSS and SAS. To convert case-mode data into frontal-slice mode data please use the program CASE-TO-SLICE from the Utilities entry in the Menu Bar on the Main Screen.

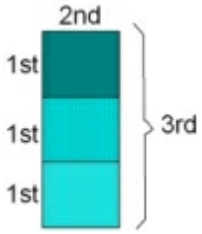


Figure: Slice-mode arrangement

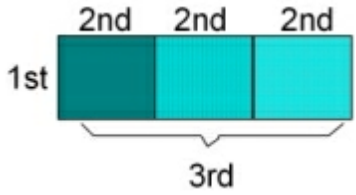


Figure: Case-mode arrangement

Data input format

The analysis programs were designed in an era in which it was common to specify the data input format analogously to the way it was specified in Fortran, the most common computer language at the time. This type of specification is called a *Fortran input format*. This format can have two forms: a free format and a fixed format.

- *Free format*. The numbers on the record are separate by spaces (*not* comma's!), and all data for a single case need to be on the same record. Furthermore, the numbers will be read from the beginning of the record and no numbers will be skipped. There may be more numbers at the end of a record than will be read in. Thus one could use the ends of records for descriptions of the data.
- *Fixed format*. A fixed format requires that all numbers on subsequent records are nicely lined up (easy for visual inspection!). The format provides a precise description and instruction which numbers are to be read in which manner. Typical examples are (4F3.1), which means that four numbers are being read in each takes up three places on a record. Of these numbers the last one is the decimal and the first two the numbers before the decimal. Thus 123 on a record means that the data value is 1.23.
  For the data shown in the Data File View the format is (10F3.1) or the 10 levels of Mode B (the second mode) each take up 3 places and the last digit is the decimal. Thus the first number will be read in as 5.5. For further details see the entry Data input format.

2.3.1 Job and data

Introduction

The Data Specification screen first provides information about the current job file and the chosen data file.

Job file

This text box indicates the name and location of the current job file. The job file is the memory of the analysis and should therefore reside where it was created. Moving a job file to another location requires editing all the paths contained in the file and is not recommended. The name of the job file cannot be edited in this screen. If you want another job please go back to the Main Screen

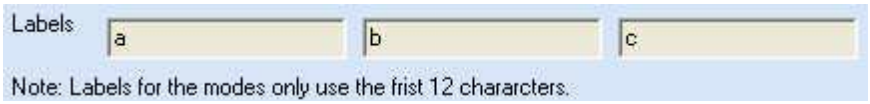
Data file

The path of the data file cannot be changed in this screen. The philosophy in making the program is that each job file is exclusively linked to a single data set. Analysing a different data set requires a

different job file.

2.3.2 Labels for modes

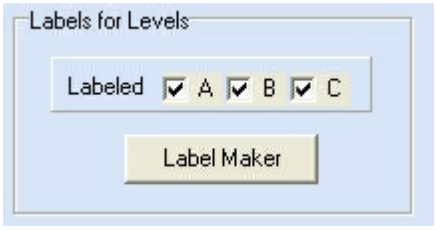
Introduction
In order to make sense of the output the user is urged to supply meaningful labels for each of the modes.



Specification
The labels may consist of up to 12 characters per mode. The labels will be used to identify the modes in the output, and the output is virtually unreadable without such labels. Thus labelling as above with simply a, b and c is *not* recommended. Only the first twelve characters will be used for labelling the output.

2.3.3 Labels for levels

Introduction
In order to facilitate reading the output, the levels for each of the three modes should be labelled. It is strongly recommended to do so.

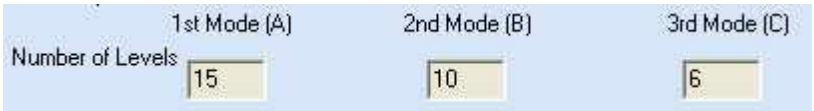


Specifications
Labels may be supplied for any or each of the modes by checking the appropriate box. The labels themselves should be specified via the Label Maker screen, which appears after pressing the appropriate button. After specification, the labels will be stored in a label file <dataname>.lab.

Please check the mode(s) which you want to label and then go to the Label Maker. The Label Maker can also read in existing label files. Note that when there are very many levels in a mode, specifying labels is not always useful. However, in that case the default numbering for the levels can be used.

2.3.4 Number of levels

Introduction
In order to relay the size of the data to the analysis programs the number of levels for each mode have to be specified correctly.



First mode
The number of levels in the first mode (also referred to as Mode A); indicated in the output and manual by *I*.
The default number of levels in this mode is 2.

Second mode
The number of levels in the second mode (also referred to as Mode B), indicated in the output and manual by *J*.
The default number of levels in this mode is 2.

Third mode
The number of levels in the third mode (also referred to as Mode C), indicated in the output and manual by *K*.
The default number of levels in this mode is 1.

The maximum number of levels that can be accommodated by the programs for anyone mode is 99999. If you have more levels please contact The Three-Mode Company.

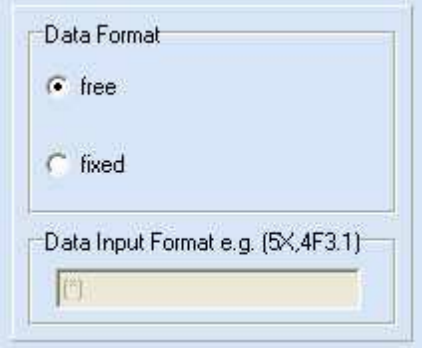
If only one level for the third mode is specified both the Tuckals2 and Tuckals3 will perform a singular value decomposition which can be interpreted as a standard (or two-mode) *principal component analysis* given the proper input scaling (see e.g. the normalisation section of PREPROC3). It is possible to specify a single level for one of the other modes, but this requires a highly unusual data arrangement. If there is one level in the first mode, a data matrix is seen as a single horizontal slice of *K* rows of *J* elements. If there is only one level in the second mode the data array should consist of a single column of length *KI*.

Data arrangement
The data should be stored in frontal-slice mode, i.e. as *K* frontal slices of size $I \times J$ arranged as a *tall matrix* with *KI* rows and *J* columns. However, often data are in case-mode, i.e. the frontal slices of size $I \times J$ are positioned next to each other, thus arranged as a *wide matrix* with *I* rows and *KJ* columns. Typically this is the arrangement used in standard software packages like SPSS and SAS. To convert case-mode data into frontal-slice mode data please use the CASE-TO-SLICE utility from the Utilities entry in the Menu Bar on the Main Screen.

2.3.5 Data input format

Introduction
In order for the data to be properly analysed their format needs to be relayed to the analysis programs. This requires careful attention and specification. The user is always advised to check in the output whether the data have been properly specified. By default the first five and the last five records of the input file are printed, except when a complete listing of the data in the output is requested.

There are two kinds of data formats, free format and fixed format. For free format no further specification is necessary, a fixed format needs a detailed specification of the Data Input Format. For further detailed discussion of formats, see the section in Input Format Specification. A condensed version is given below.



Free format

The data may be read in using *free format* where one or more spaces separate the numbers. Different records can thus have different lengths but should have an equal number of values. The number of levels specified in the second mode (Mode B) will determine how many values will be read in per case. Fewer values than the number of levels in mode B will lead the analysis programs to start reading on the next record. If there are more values on a record than there are levels in mode B, the extra values will not be read. One could use this property to put the level numbers for identification and other information at the end of the records. A record will be read from the beginning and its values analysed so no numbers can be skipped.

The alternative is to use a *fixed format* in which case the specific field for each data value on a record needs to be specified. Moreover in the latter case each row must have exactly the same format. Fixed format, however, allows for skipping certain values, for instance case numbers or identification tags at the beginning of a row.

Fixed format - Data input format

The Data Input Format needs to be specified using a so-called Fortran format.
This can be a free format (*) or a fixed format e.g. (nX,nFw.d).
nX = n columns should be skipped; nFw.d = repeat Fw.d n times.
Fw.d = the number to be read occupies w columns. The last d numbers are the decimals.
Thus using (5X,2F3.1) the record 123..234589 will be read as: 23.4 and 58.9.
The first five columns will be skipped (see the 3WayPack Manual for further details).

The analysis programs were designed in the era that it was common to specify the format analogous to the way it specified in Fortran, the most common computer language at the time. This type of specification is called a *Fortran input format*. Typical examples are (5X,4F3.1), which means that the first five columns are skipped and then four numbers are being read in, each taking three places on the record. The number after the F, here 3, is called the *field width (w)* and the last one is the number of decimal places or *decimal (d)*. The general specification is thus F*w.d* Of these numbers the last one is the decimal and the first two the numbers before the decimal point. Thus with a format specification of F3.1, the number 123 on a record means that the data value is 12.3. For the data shown in the Data File View (see Figure **Data File View**) the format is (10F3.1). The 10 levels of Mode B (the second mode) each take up 3 places and the last digit is the decimal. Thus the first number will be read in as 5.5.

A valid Fortran-style input format must be provided if you have indicated that the data have a fixed format.

Please check your format specification against the Data File View panel. Data formats which are

inconsistent with the data in the data file may lead to incorrect analyses, or untimely end of the program with a so-called run-time error. If there are more data in the file than specified, a warning will be issued by the analysis program and the analysis program will proceed. If there are fewer data than specified the analysis program will issue a warning and end its execution. Please correct the incorrect specifications before proceeding.

More complex formats

Skipping places (columns) at the beginning of a record/line of the data file

In Fortran more complex formats can be specified than are indicated here. For instance, one may want to skip the first 10 places of the data file because they indicate the level numbers for the 3rd and 1st mode, respectively. In such a case the format will look like (10X,4F3.1).

Skipping places (columns) in the middle of a record/line

Yet another possibility is the skip a particular variable, say the second one. Then the format reads (F3.1,3X,2F3.1). Thus of the four variables only the first, third and fourth are read into the program. To skip two or more consecutive values a specification must be given for each them, e.g. use the specification (F3.1,3X,3X,F3.1) to skip values 2 and 3 and only read values 1 and 4.

Variables with different numbers of places/columns

Often variables need to have different numbers of place/columns because their numbers are larger or smaller. For example (1X,F3.1,F5.2,3F3.0) indicates that the first number is to be skipped and that there are 8 variables: the first takes up three columns and has one decimal place, the second needs 5 columns and has 2 decimal places, followed by 3 variables taking up 3 columns each and their values have no places behind the decimal. Thus a line in the data file could look like (rulers line is given above the values):

```
12345678901234567890 (this is the ruler line)
 234 1045 23 45 67 Subject 1(this is a record in the data set)
```

With the format (1X,F3.1,F5.2,3F3.0) the data values read in are 23.4 10.45 23 45. The specification `Subject 1` will not be read and is used for identification purposes.

Values with decimal points

In most data sets the data values already include a decimal point in their specification, i.e. data are listed as 2.34 rather 234 if there are two decimals. In Fortran formats the decimal specification in the data file always takes precedence over the format specification. Thus if the format is specified as F5.2 and the data are 2.345, then the number used in the programs is 2.345. Note that the decimal point takes up one column so that it has to be included in the field width. Thus 2.345 takes up 5 columns and the Fortran specification should be for example be F5.1 or F5.2, and *not* F4.3.

2.3.6 Data file view

Introduction

When the `Data specification` button is pressed in the `Job specifications screen` the `Data specification screen` is called and immediately the data set is read in and displayed in the `Data File View` panel.

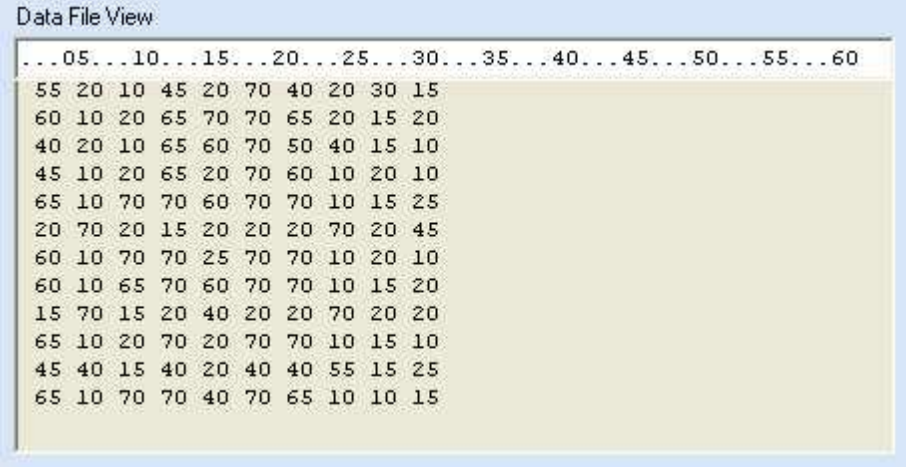


Figure: Data file view

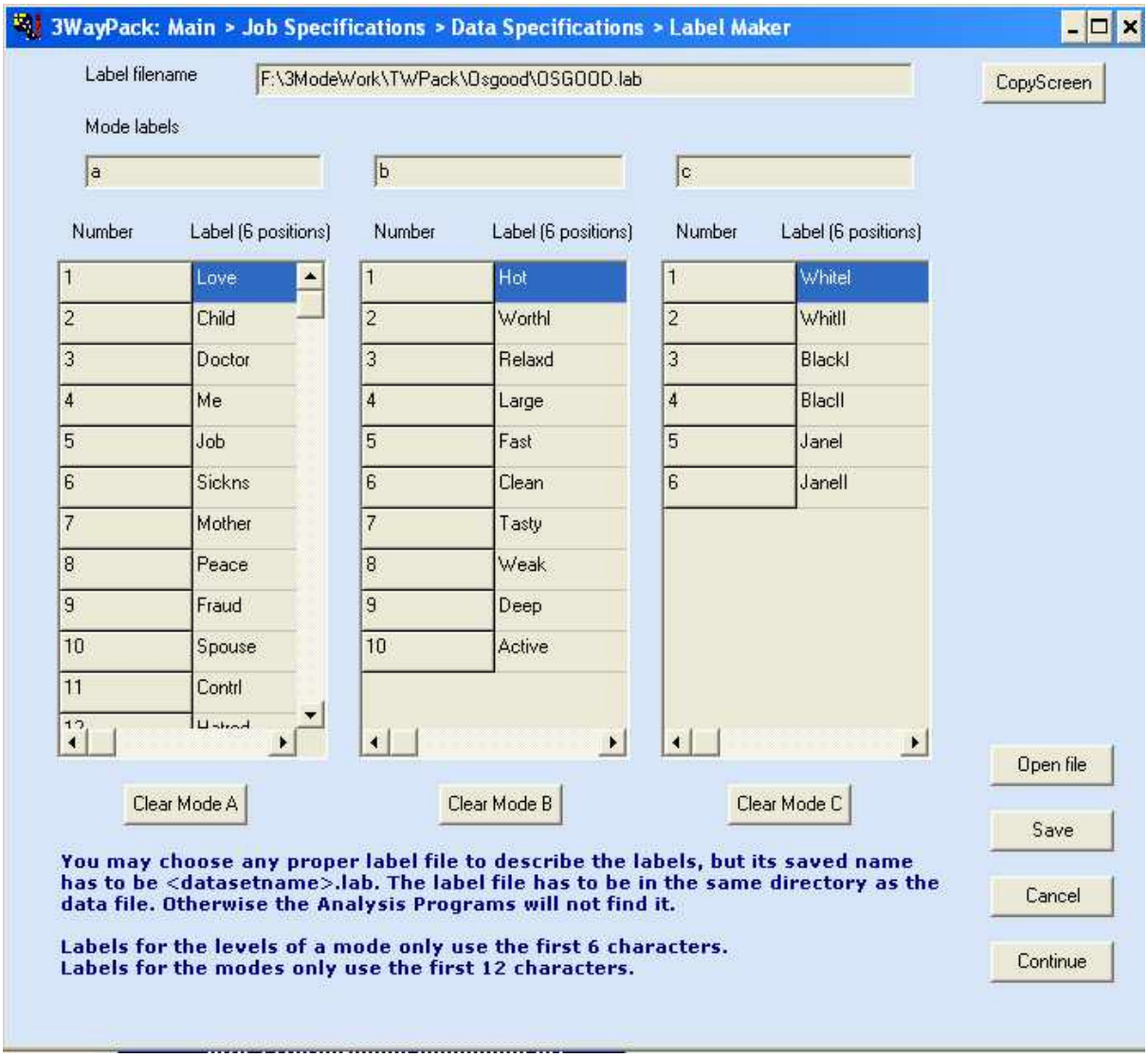
Content

The data file is displayed with a ruler on top to help specifying the input format. Note that placing the cursor in this field will allow scrolling with the arrow keys or the Page Up and Page Down keys. However, the wheel of the mouse does not work in this case.

2.4 Label maker screen

Introduction

This screen is entered via the Data specifications screen by clicking on the Label Maker button provided at least the box of one mode is checked . It can be used for specifying the labels for the analyses.



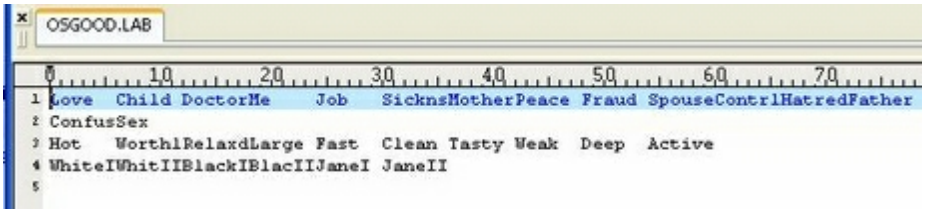
Types of labels

There are two kind of labels.

- *Mode labels*.
  These 12-character labels contain the names of the modes which can also be specified in the `Data Specifications Screen`. They may be respecified and/or changed on this screen.

- *Level labels and label file*
  If a label file is available with the same name as your data file, i.e. `<datafilename>.lab`, its contents will be read automatically and the labels are shown on the screen. If this file does not exist, this screen will let you specify the necessary level labels. However, the program does not cater for more than 200 level labels in any one mode.
  To *add or change* level labels simply click on the appropriate box and type in a new level label. Labels can be up to 6 characters long.
  When you are done, press the Save button and a file`<datasetname>.lab` will be saved. You will not have a choice of naming the file, because the analysis programs expect that the labels will reside in a file called.`<datasetname>.lab`.

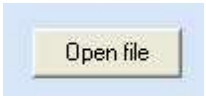
Example Label file



2.4.1 Label file

Introduction

Pressing the Open file button will bring up a dialogue box from which you can select an existing label file, which is expected to have the name <datasetname>.lab, but any other appropriate label file may be used as long as it is renamed to <datasetname>.lab and placed in the data directory upon leaving the Label Specification screen.



Name of label file

Even though the selected label file may have any name it should be saved as<datasetname>.lab in the data directory because the analysis programs expect that it is named that way.

Content label file

The label file contains a maximum of 13 labels of 6 characters each on a line. The labels for a mode always start on a new record. They are always arranged such that the first-mode labels precede the second-mode labels and they precede the third-mode labels. This order is maintained even if no labels are specified for a mode.

In Figure **Label file** the first record contains the first13 labels of the first mode, the second line the remaining two labels of the first mode, the third record contains the ten labels of the second mode, and the fourth record the six labels of the third mode.

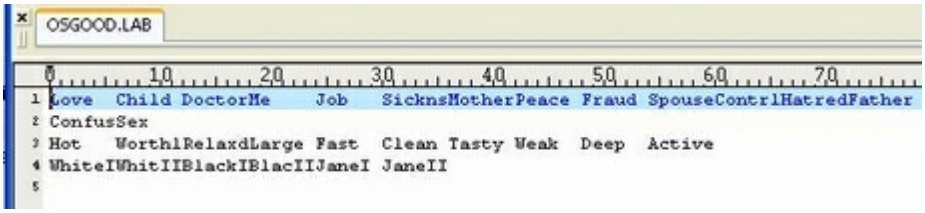
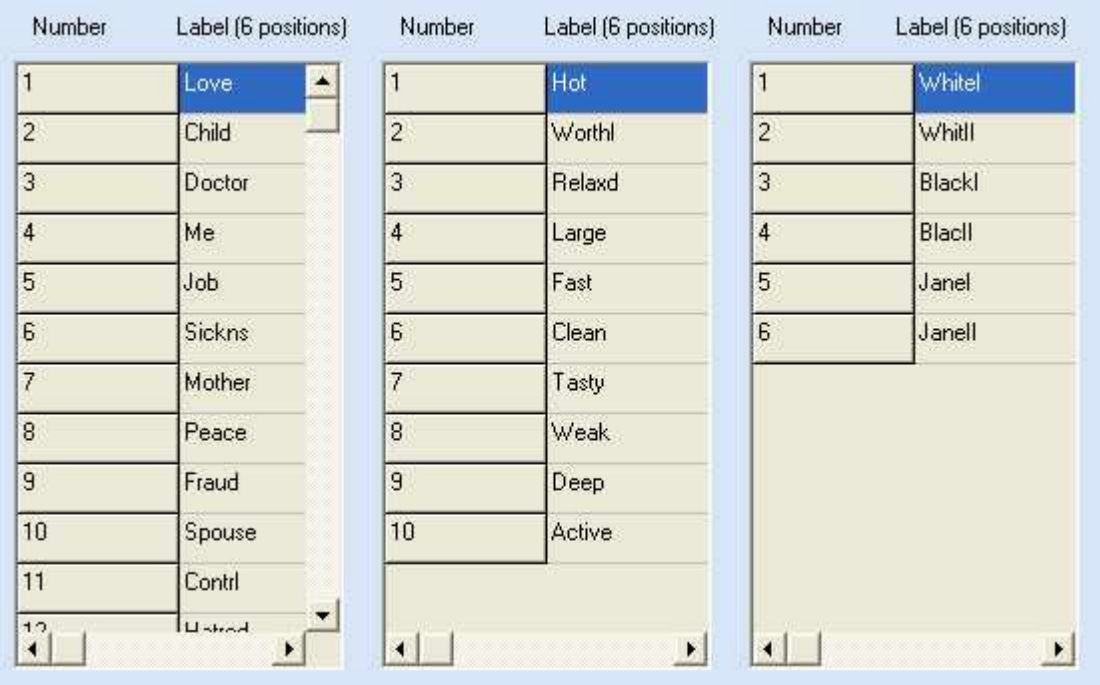


Figure: Label file

2.4.2 Level labels

Introduction

It is advisable to specify labels for the levels of each mode because it makes the output of a three-mode analysis readable. At present the labels can be at most 6 characters long. Please try to make the labels clearly different to improve legibility of the plots.



Specifying level labels

To *add or change* level labels simply click on the appropriate box and type in the new level label. Labels can be up to 6 characters long

When you are done, press the Save button and a file`<datasetname>.lab` will be saved in the data directory. You will not have a choice of naming the file, because the analysis programs expect that the labels will reside in a file called.`<datasetname>.lab`.

Label file

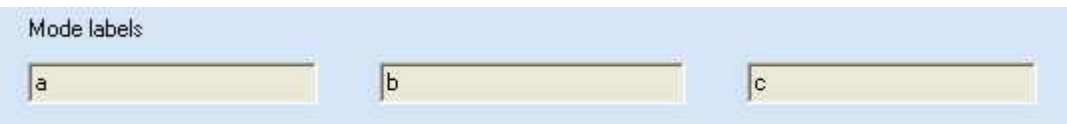
If a label file is available in the data directory with the same name as your data file, i.e. < `datafilename`>.`lab`, its contents will be read automatically and the labels are shown on the screen. If this file does not exist, this screen will let you specify the necessary level labels. However, the program does not cater for more than 200 level labels in any one mode.

.

2.4.3 Mode labels

Introduction

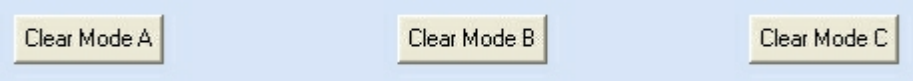
These 12-character labels contain the names of the modes which are usually already specified in the `Data specifications screen`. They may be, however, be respecified here.



2.4.4 Clear labels

Introduction

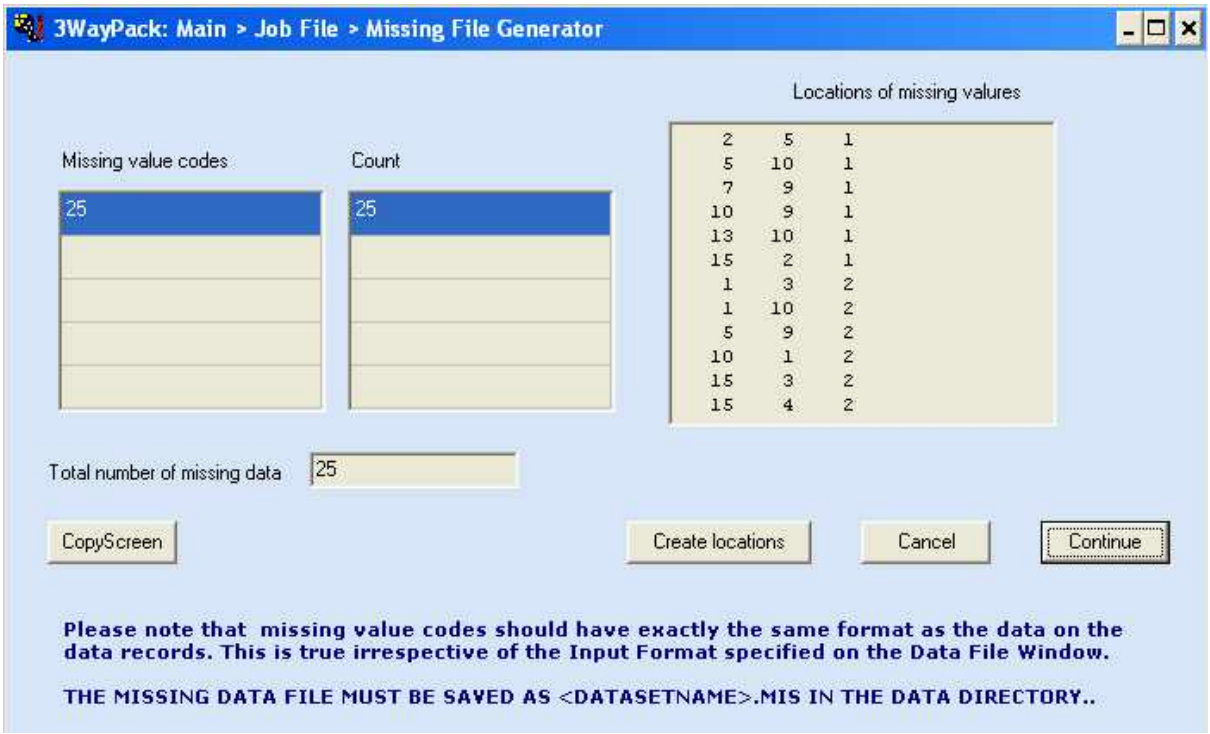
Rather than deleting the labels of a mode one by one the labels can be deleted per mode by pressing the appropriate button.



2.5 Missing values screen

Introduction

This screen is reached via the Job specifications screen either by pressing the button Edit/View Missing-Data File button or the Create Missing-Data File button. This screen can be used for specifying the missing-values codes for an analysis. Up to five different codes can be specified. The missing-value generator will search the data set for these codes and counts their numbers. It will also indicate in the right-most column the locations of the missing data. The location information is transferred to a special file <datasetname>.mis which is used by the analysis programs TRILIN, TUCKALS2 and TUCKALS3.



Missing-data file

If a missing-data file is available in the data directory with the same name as your data file, i.e < datasetname>.mis it will be read automatically. If not, this screen will let you specify the necessary missing values. However, the program does not cater for more than five different missing-value codes. Moreover, the missing-value codes should be the same for all variables. No variable should have a missing-value code which is a valid value for another variable.

To add or change a missing-value code simply click on the appropriate box and type in the required code. Then press the create locations button to search for the locations in data set and count them.

When you are done, press the Save button and a file <datasetname>.mis will be saved in the data directory. You will not have a choice of naming the file, because the analysis programs expect a file with this specific name.

Please note that missing-value codes should have exactly the same format as the data on the data

records. This is true irrespective of the Input Format specified on the Data specifications screen.

Example missing-data file

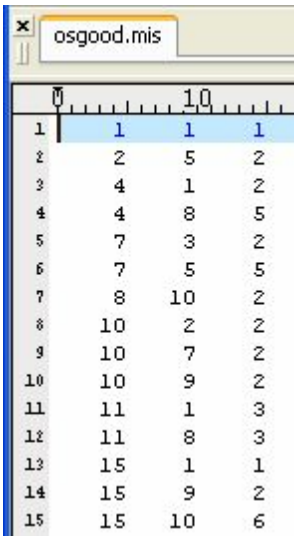


Figure: Missing-data file

2.5.1 Locations of missing values

In this panel the locations of the missing data in the data set are listed. The numbers (i,j,k) indicate the location of the missing data point in the data set. For instance, the first entry in the Figure **Location missing-values panel**. (2 5 4) indicates that x_{254} is missing. What the actual value in the data set is at that location is not listed. Generally it is one of the missing-value codes but if the missing-data file already exists, it could also be an initial estimate for that data point obtained via PREPROC3 or another program.

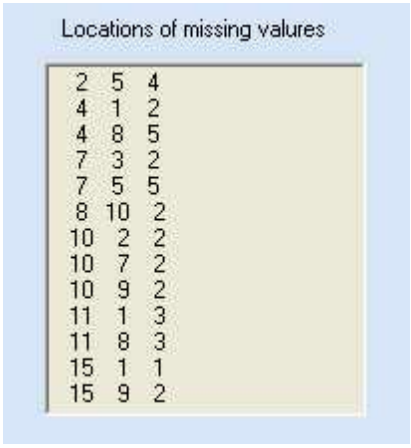
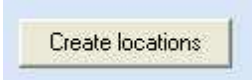


Figure: Location missing-values panel

2.5.2 Missing value codes

Introduction

The program allows for up to five missing-value codes. The data set is searched for these codes when the Create locations button is pressed.



The codes are counted and their frequencies are listed in the Missing-values panel (see Figure

Missing-values panel).

Format requirement

Please note that missing-value codes should have exactly the same format as the data on the data records. This is true irrespective of the Input Format specified on the Data specifications screen.

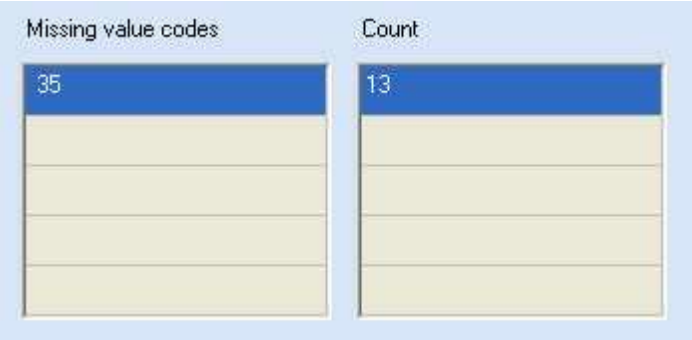


Figure: Missing-values panel

2.5.3 Number of missing values per code

Introduction

During the search of the data set for missing values the number of missing values for each code are counted. These numbers will be listed in the Missing-value codes panel (see Figure. **Missing-value codes panel**)

Thus there are 13 instances with the missing value code 35. Each missing value code and the number of its occurrence in the data set will be listed on a new line. For such an example see the complete screen.

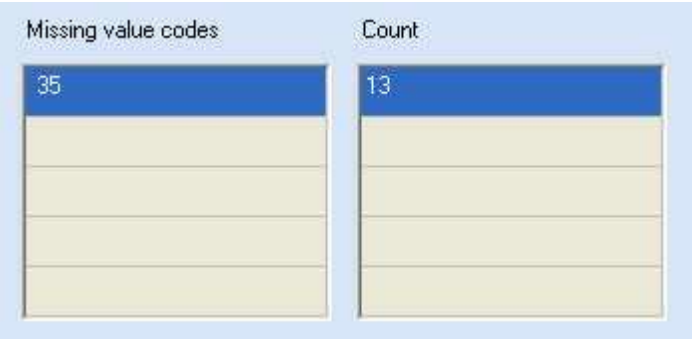


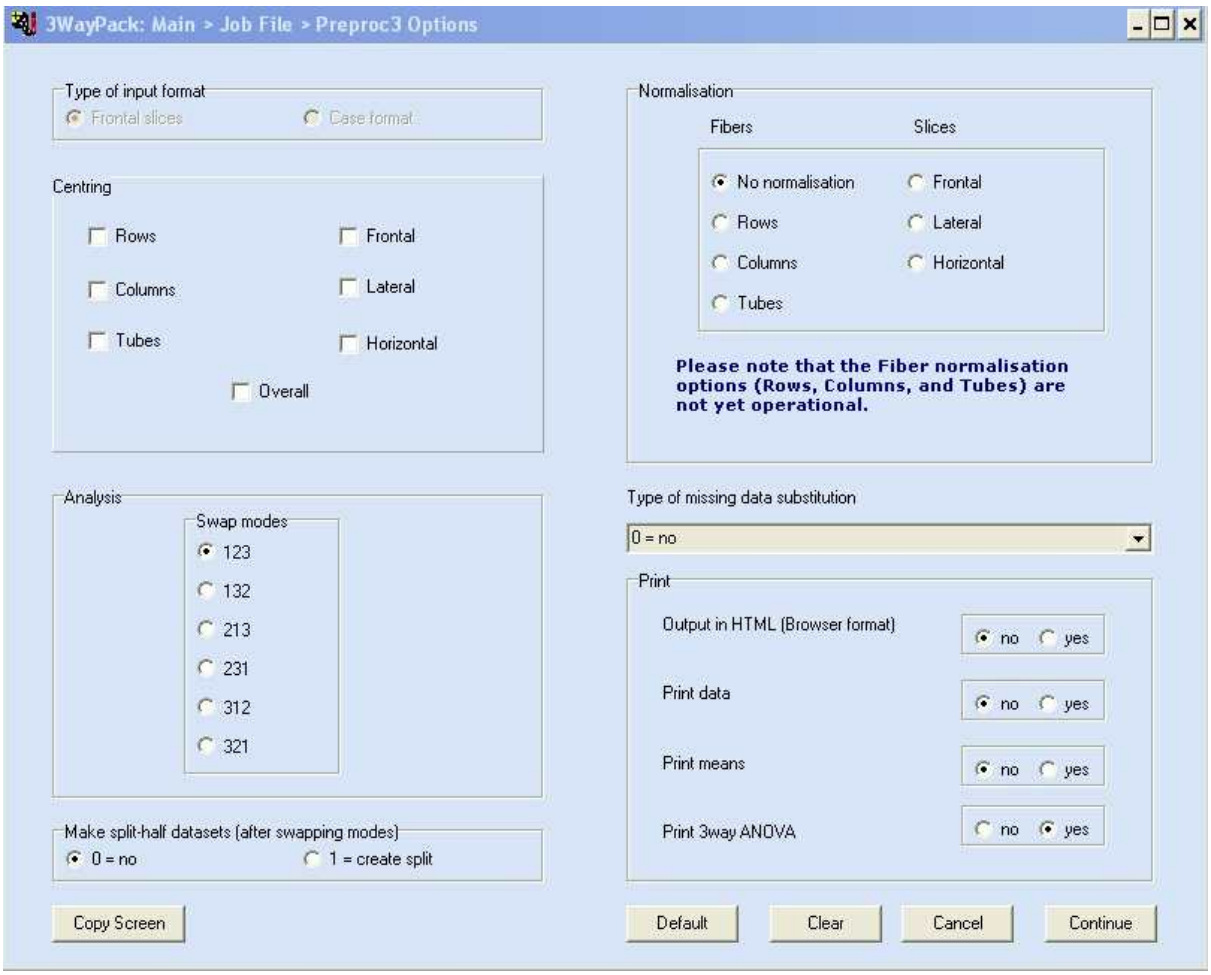
Figure: Missing-value codes panel

3 Preprocessing program

3.1 Preproc3 options

Introduction

PREPROC3 is a program for preprocessing three-way data, but it also fits two analysis-of-variance models with one observation per cell. A description of many details about the preprocessing contained in the program can be found in Kroonenberg (2008, Chapter 6)



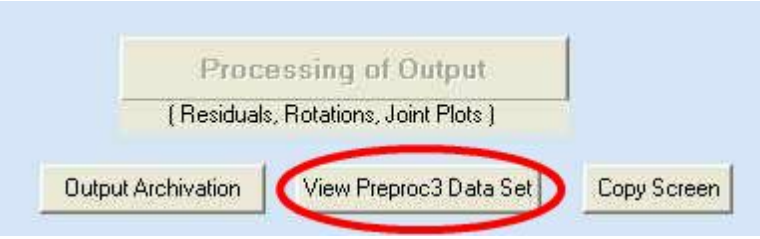
PREPROC3 options

1. *Swapping* the dimensions of a three-way array;
2. *Estimating missing* data according to ANOVA models;
3. *Centring* three-way arrays;
4. *Normalising* three-way arrays;
5. *Printing* all possible means;
6. Computing and printing *three-way* ANOVA summary tables;
7. Creating a *new data set* in accordance with the changes specified.
8. *Creating subsets* of the data, in particular orthogonal split-half files

Creation of a new data set

If PREPROC3 modifies the data in any way (i. e. if options 1, 2, 3, 4, 8 from the above list are

requested) it creates a new data file in the same directory where the original data file resides. The name of this data set is <datasetname>.pp3.



This new data set can be viewed using the View Preproc3 Data Set button on the Main Screen. You may consider to rename the extension of the data set from PP3 to DAT, but this is not necessary. The renaming can only be done outside 3WAYPACK.

Once the data set has been viewed the button will disappear.

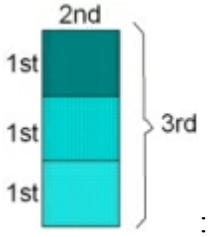
3.1.1 Data arrangement

Introduction

The standard data arrangement for all three-way analysis programs is *frontal-slice mode*, i.e. the data matrices of each level of the third mode are placed below each other in a so-called *tall matrix,* also referred to as a tall combination-mode matrix**.**

Tall matrix

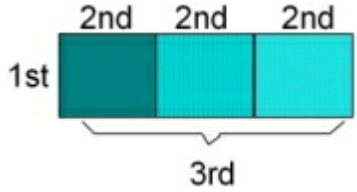
The format of this arrangement of the data array is referred to as the *frontal-slice mode* of the data.



:

Wide matrix

The data arrangement of most statistical packages, such as SPSS and SAS, is however case-mode, thus as a wide matrix.



Converting from case-mode to frontal-slice mode

Please use the CASE-TO-SLICE utility from the Main Menu bar to convert data from case-mode to frontal-slice mode.

3.1.2 Centring

In nearly all three-mode analysis it is advantageous to centre the data array in some way. In two-way there is generally only one option: Centre each variable by taking the mean across the subjects. With three-way data there are several ways to centre the data and the most common ones are included in PREPROC3.

Centring three-way data

There are three basic ways to centre three-way data.

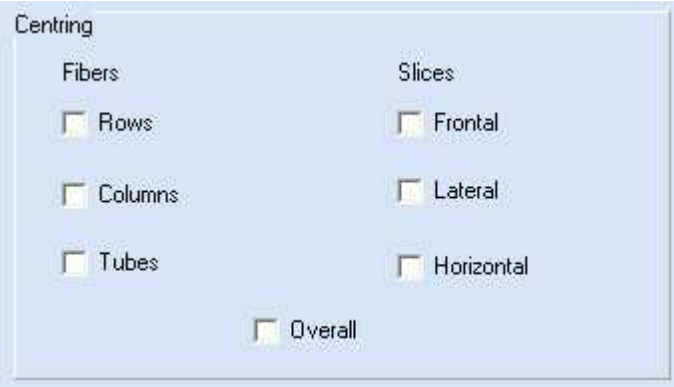
1. *Fibre centring*: each column/row/tubes is set in deviation of its mean by subtracting a fibre mean. This is generally the recommended option.
2. *Slice centring*: the data of a frontal/lateral/horizontal slice are set in deviation from the slice mean. This is relatively seldom applied.
3. *Overall centring*: the overall mean is subtracted from all data. In general this does not serve any purpose.

Centring & swapping

To prevent errors it was decided to define the centring with respect to the original arrangement of the data, i.e.before *swapping*.

Combining centrings

Different combinations of centring are possible. The program will correctly handle a situation where redundant centrings have been specified, for instance if both overall centring and centring per row has been specified.



Type of centrings

The means will be indicated with the letter *M*.

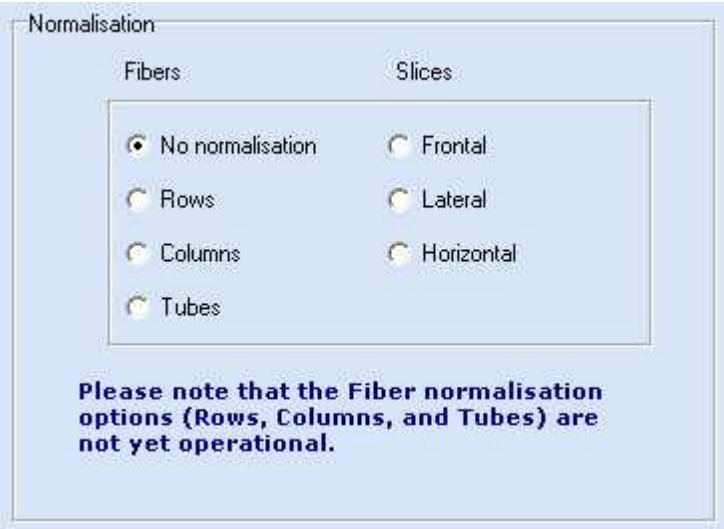
1. *Column centring:* Across Mode 1 (= Mode A): centring each column *jk* of the original array:
   Definition: $z_{ijk} = x_{ijk} - M_{.jk}$
2. *Row centring*: :Across Mode 2 (= Mode B): centring each row *ik* of the original array
   Definition: $z_{ijk} = x_{ijk} - M_{i.k}$
3. *Tube centring*: Across Mode 3 (= Mode C): centring each tube *ij* of the original array
   Definition: $z_{ijk} = x_{ijk} - M_{ij}$
4. *Horizontal slice centring*: Per 1st mode slice: centring per horizontal slice of the original array
   Definition: $z_{ijk} = x_{ijk} - M_{i.}$
5. *Lateral slice centring*: Per 2nd mode slice: centring per lateral slice of the original array
   Definition: $z_{ijk} = x_{ijk} - M_{.j.}$
6. *Frontal slice centring*: Per 3rd mode slice: centring per frontal slice of the original array
   Definition: $z_{ijk} = x_{ijk} - M_{.k}$
7. *Overall centring*: Overall: centre entire data array
   Definition: $z_{ijk} = x_{ijk} - M_{..}$

Centring, normalisation and standardisation
For centring in combination with normalisation or standardisation see Normalisation section.

3.1.3 Normalisation

Aspects of normalisation

- *Definition*. Normalisation is the rescaling of the original or previously centred data, such that the sum of the squared data elements in a submatrix is 1.
- *Centring*.
  The choice of type of normalisation can generally be made independent of the type of centring.
- *Standardisation*.
  Normalisation in itself will not produce *z*-scores (standard scores: mean 0, standard deviation 1) unless the data in the normalised submatrix have a zero mean.
- *Sum of squares*.
  Normalisation results in sums of squares equal to 1 rather than mean squares equal to 1. This is primarily done because sums of squares may be partitioned into independent sums of squares, while mean squares based on degrees-of-freedom cannot be partitioned in that way.
- *Perpendicular normalisations*.
  If normalisation is attempted perpendicular to a previously performed centring, the normalisation destroys the centring. Therefore such combinations are not allowed in the program. Only one type of normalisation can be selected at a time. Multiple normalisations require an iterative process, but this is not included in the program.



Types of normalisations

1. *Horizontal slice normalisation*: normalising each original horizontal slice *i*.
   Definition:: $z_{ijk} = x_{ijk}/s_{i..}$
2. *Lateral slice normalisation*: normalising each original lateral slice *j*
   Definition:: $z_{ijk} = x_{ijk}/s_{.j.}$
3. *Frontal slice normalisation*: normalising each original frontal slice *k*
   Definition:: $z_{ijk} = x_{ijk}/s_{..k}$
4. *Fibre normalisations* (rows, columns, tubes) have not yet been implemented.

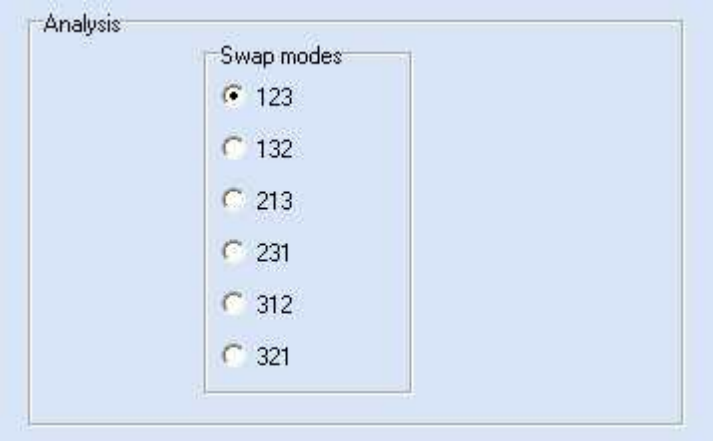
Standardisation
Standardisation is defined as centring followed by normalisation. Note that in three-way analysis centring and normalisation are not necessarily done in the same way. For instance, for three-way profile data, i.e. subjects by variables by conditions, column centring is usually combined with lateral slice normalisation. This type of standardisation makes is possible to interpret the principal coordinates of the variable coordinates as variable-component correlations or 'loadings'.

If only one component has been specified in the third mode, the above standardisation will produce the results of an standard principal component analysis.

3.1.4 Swapping data matrix

Definition of swapping
Transposition of one of modes is the operation to rearrange the data box such that the mode take a different place in the data box. PREPROC3 can swap a three-way array in any desired manner.

For example, a subjects (1st mode) by scales (2nd mode) by situations (3rd mode) data box can be transposed into a situations (1st mode) by scales (2nd mode) by subjects (3rd mode) data box. In which case the arrangement (1,2,3) is changed into (3,2,1) where the numbers refer to the original mode numbers.



Uses for swapping
In some applications it is desirable to have a specific mode

1. as the first mode; for instance, subjects for three-way profile data, or
2. as the third mode; for instance, in case of time points because it is desired not to have components for the time mode. This requires an application with Time as the third mode in a TUCKALS2 analysis
3. as the third mode; for instance, in case of three-way rating scales. This requires the subjects to be in the third way.

In such cases the modes have to be swapped or transposed. Such swapping is particularly relevant for analyses with the Tucker2 model because it is not symmetric in all modes. Moreover, judicious swapping allows TUCKALS3 to compute the correct latent covariances.

3.1.5 Missing-data substitution

Introduction
3WAYPACK has a slightly unusual way of working with missing data. Two aspects are relevant for properly handling missing data:

1. *Locations* of missing data in the data array;
2. *Starting values* for the missing data to start the iterative procedure in an analysis program.

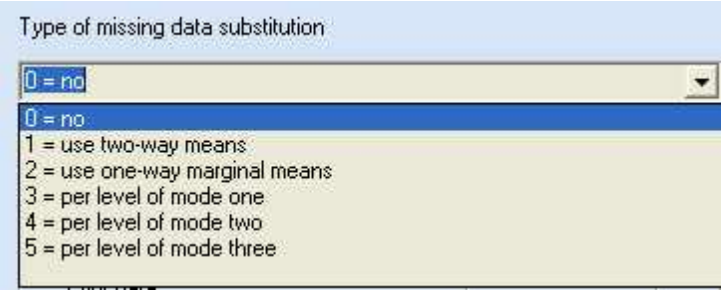
Locations

In order to get the starting values for missing data and in order to be able to estimate them the analysis programs need a file with the *locations* of these missing data in the data set. These locations can be created via the Job specification screen or the utility FindMissing.

Starting values

Analysis programs need starting values for these missing data so that the model can be estimated. During the iterations for estimation of the parameters, the missing data are re-estimated as well.

The analysis programs such as TUCKALS3 can calculate starting values themselves using the most common methods, but PREPROC3 has a comprehensive selection of analysis-of-variance models for estimating starting values for missing data.



Basic ANOVA models for estimating the missing values

1. *Three-way ANOVA model* with a single observation per cell (Options 1, 2)
2. *Multivariate two-way ANOVA model* with a single observation per cell (Options 3, 4, and 5).

The options provided specify how many and which terms of the (M)ANOVA model should be used for the estimation.

Types of ANOVA models for estimation

Notation: v_{ijk} = missing value at location (i,j,k); m = overall mean; a_i = row effect; b_j = column effect; c_k = tube effect; ab_{ij}, ac_{ik}, bc_{jk} = two-way interactions.

1. *Use two-way means*
   Estimates are based on the three-way analysis of variance model minus the three-way interaction term.
   Missing value v: $v_{ijk} = m+a_i+b_j+c_k+ab_{ij}+ac_{ik}+bc_{jk}$
2. *Use one-way marginal means*
   Estimates are based on the three-way main effects model.
   Missing value v: $v_{ijk} = m+a_i+b_j+c_k$
3. *Per level of Mode 1 (= Mode A)*
   Estimates to be used if the levels of the first mode have incompatible measurement levels.
   Missing value v: $v_{ijk} = m_i+ab_{ij}+ac_{ik}$
4. *Per level of Mode 2 (= Mode B)*
   Estimates to be used if the levels of the second mode have incompatible measurement levels.
   Missing value v: $v_{ijk} = m_{jk}+ab_{ij}+bc_{jk}$

5. *Per level of Mode 3 (= Mode C)*
   Estimates to be used if the levels of the third mode have incompatible measurement levels.
   Missing value v: $v_{ijk} = m_k + ac_{ik} + bc_{jk}$

The algorithms used to estimate missing data are convergent if the data matrix is not singular, that is, if there are no submatrices completely filled with missing data. If there are fibres which have no valid data the estimated values are not unique.

Note that if there are missing data but they are not estimated in PREPROC3, the program cannot perform centring or normalization.

3.1.6 Printing options

PREPROC3 will print means at various junctures of the program and will also print the results of a fixed effects analyses of variance with one observation per cell.

Type of printing

Output in HTML (Browser output)
To facilitate navigation html output files can be created in addition to the standard line print. The output is formatted as a frame with a navigation bar and a content field (see also File naming conventions).

Print data.
By default only the first five and the last five lines of the data are printed. This options allows a printing of the entire data array per frontal slice.

Print means
If requested and applicable, all means will be printed
(a) after swapping,
(b) before and after estimation of missing data,
(c) before and after centring, and
(d) after normalisation.

Print ANOVA: 3-way ANOVA and two-way MANOVA
If requested and applicable, a three-way ANOVA and a two-way multivariate ANOVA are printed. Which of the two is applicable is for the user to decide.
Printing is done
(a) before and after estimation of missing data, and
(b) before and after centring/normalization.

As there is no real error term, the three-way interaction is used as the within sum of squares in the calculation of the *F* statistic. However, in large data sets it is expected that a substantial part of the three-way interaction can be modelled with only a few multiplicative components and thus a limited number of degrees-of-freedom. The 'real' mean error sum of squares is generally expected to be smaller than the full three-way interaction mean square, so that the *F* values are only a lower bound.

3.1.7 Split-half data sets

Introduction
Preproc3 can create random split-half data sets which can be used for assessing the stability of a solution or assessing the appropriate numbers of components.

Which data sets are created.
By choosing to spilt-half the data set four different new data sets will be created. The complete data set is first split in four parts, A, B, C, D. Then A+B en C+D will constitute one pair of data sets and A+C and B+D will give the other pair.

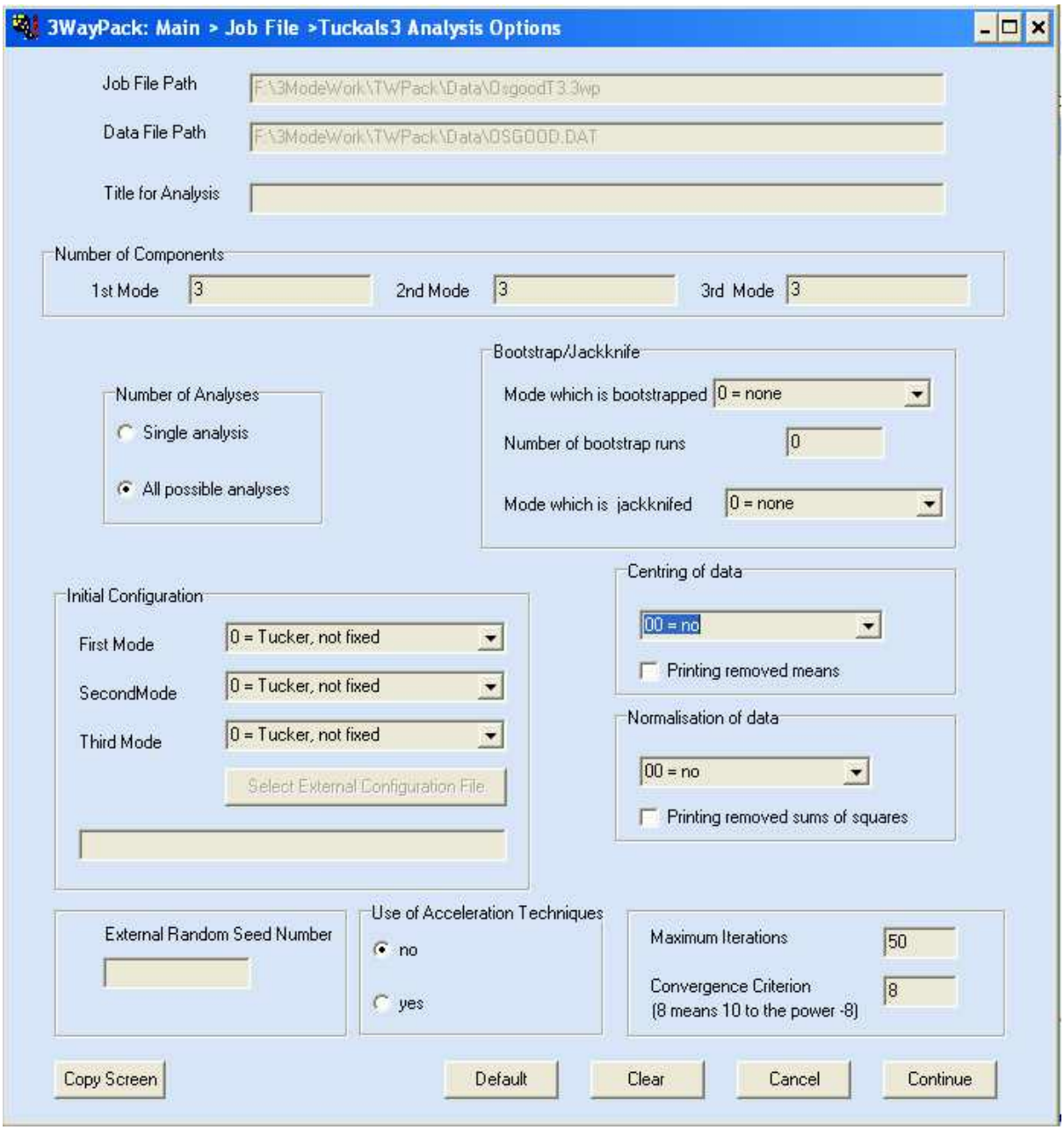
Data set names
The data sets will be labelled as <jobnameDXX>.dat with XX = 1A, 1B, 2A, 2B, respectively (see also File naming conventions).

4 Analysis programs

4.1 Tuckals3 analysis options

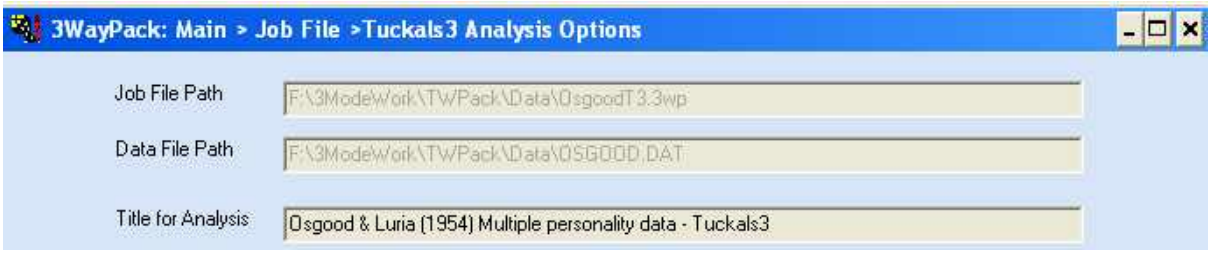
Introduction

The Tuckals3 Analysis Options screen is accessed via the Job Specification screen via the Analysis Options button. On the present screen the details of the analysis can be specified such as the number of components, the number of analyses, execution of an external analysis, bootstrapping, jack-knifing, specification of centring, normalisation, type of initial configurations, algorithm specifics, etc.



4.1.1 Job, data, and title

Introduction
The Tuckals3 Analysis Options screen first provides information about the current job file, the data file and gives the opportunity to specify a title for labelling the output.



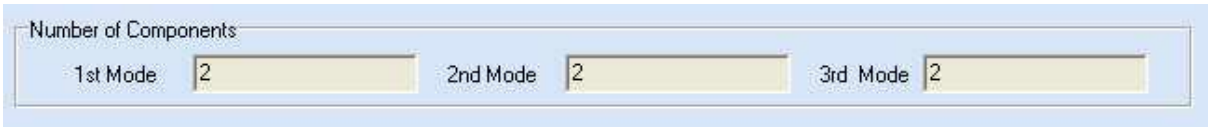
Job file
This text box indicates the name and location of the current job file. It cannot be edited in this screen. For another job or job name please go back to the Main Screen

Data file
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file.

Title for the analysis
The title specified in this text box will be used to label the output. It is suggested to specify here any specific details which cannot easily be read from the output, such as specifics about the data and their origin, etc.

4.1.2 Number of components

Introduction
By default the first two modes have two components and the third mode a single one.



Specifications
- The specification of the numbers of components without selecting any other option is sufficient for an analysis, be it a trivial one.
- The number of components must be less or equal to the number of levels.
- The number of components have to satisfy the minimum-product rule, i.e. the product of the number of components of two modes may not be smaller than the number of components of the remaining mode.

First mode
The number of components (P) for the first mode should be less than or equal to the number of levels (I). Moreover, $P \leq Q \times R$.

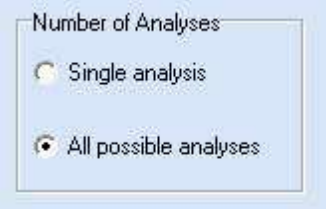
Second mode
The number of components (Q) for the second mode should be less than or equal to the number of

levels (J). Moreover, $Q \leq P \times R$.

Third mode
The number of components (R) for the third mode should be less than or equal to K, the number of levels. Moreover, $R \leq P \times Q$.

4.1.3 Number of analyses

Introduction
By default, TUCKALS3 will perform a single analysis based on the numbers of components supplied.



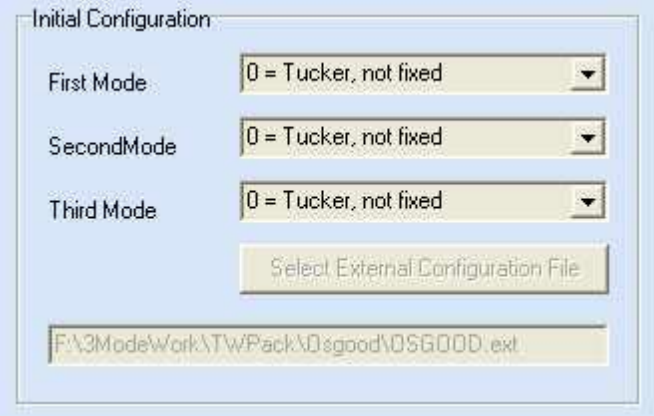
All possible analyses
In order to decide how many components the data support and to decide which solution is most appropriate, the program can calculate All possible analyses for the Tucker3 model with numbers of components less than or equal to the numbers of components indicated in the Number of components panel up to a 5×5×5-solution. The output per analysis is minimised to lower execution times. For full details of the All Possible analyses option and examples of the output see Kroonenberg (2008, Section 8.5).

Full output can only be obtained from a single analysis.

Because of the minimum-product rule, i.e. the product of the number of components of two modes may not be smaller than the number of components of the remaining mode, not all solutions are possible. For instance, a 3×2×1-solution violates the minimum-product rule because $2 \times 1 < 3$. A 3×2×1 specification for the components will provide a 2×2×1-solution.

4.1.4 Initial configurations

Introduction
The iterative procedure to solve the parameter estimation of the Tucker3 model, TUCKALS3, needs initial configurations for the component matrices **A**, **B**, and **C**. If there is no special reason to do otherwise, it is advised to stay with the default option as it will provide excellent starting values. In fact, if there is a perfect solution, it will be found by the default procedure. The initial configurations of the three modes have the same options but it is not necessary to use the same one for every mode. For full details see Kroonenberg (2008, Section 5.7.3).



Options

0 = Tucker, not fixed (default)

- The initial configuration is derived from the data.
- By default, the program computes an initial configuration for each mode by calculating ten steps towards a solution as described in Tucker (1966). This involves stacking the slices underneath each other into a tall matrix, e. g. $J \times K$ by I (a process called *matricisation*), and performing a PCA on this tall matrix.
- Generally the final TUCKALS3 solution will be close to the starting Tucker3 solution, especially with a limited number of components.

1 = External, fixed

- An external configuration is used and it is not modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, this configuration may be read in and kept fixed during the analysis. In other words, the rest of the solution is fitted around the external fixed configuration.
- Comparing the fits without and with the fixed solution will allow an assessment of how well the external configuration fits the data.
- The external solution is assumed to be orthonormal (orthogonal with unit-length components). If this is not the case the program will issue a warning, but will proceed as if the configuration is correct. In such a case the output will not necessarily be interpretable; especially the partitioning of the fit might no longer be valid.
- The external configuration should be in a file accessible to the program (preferably `<datasetname>.ext`) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.
- For full details see Kroonenberg (2008, Section 4.10.7).

2= External, not fixed.

- An external configuration is available and it is modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, you may request the program to read this configuration and use it as a starting configuration.
- The external configuration should be in a file accessible to the program (preferably `<datasetname>.ext`) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.

- For full details see Kroonenberg (2008, Section 4.10.7).

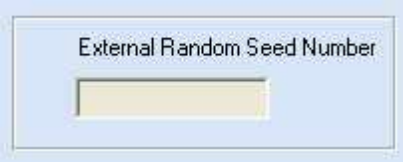
3 = Random, not fixed.

- A random initial (orthonormalised) configuration will be used.
- This option is primarily useful for research purposes.
- It could for instance be used to assess the sensitivity of the program to different starting solutions. (The answer is 'not sensitive'.)
- To create different starting configurations for each run, change the random seed.

External configurations

- External configurations for different modes should be together in one external configuration file in the order of the modes.Thus in the file a first-mode configuration should always precede a second-mode one and a second-mode configuration a third-mode one.
- Once an external configuration is specified for any mode, the program will not rest until you have supplied a file name.
- The numbers in the file should be in free format, i.e. there should be at least one blank between numbers. The more decimals the better.

External random seed number



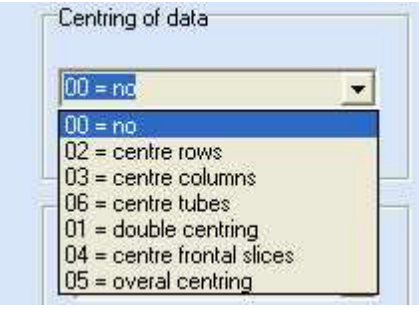
The algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. Normally a Tucker (1966) starting value is computed, but to search for different solutions for the parameters different random initial configurations can be specified via using different random seed numbers.

4.1.5 Centring

Introduction

Centring is the process of removing one or more means from the data. For three-way data there are several possibilities of which the more useful ones are included in the three-mode programs. Some additional options are available in PREPROC3.

Centring across a mode will cause the components of that mode to be centred. Note that centring fundamentally changes the data, so that it is virtually impossible to compare analyses of differently centred data. For further centring options see Preproc3 centring.



Options

None: no centring: Default

Row centring: Fibre centring across columns per row-tube combination

- Row centring indicates that for all $I \times K$ rows the means are calculated by summing over the index j and dividing by J. These means are subtracted from the data values, so that all rows are in deviation scores.
- As variables are mostly in the second mode, this is a unusual option for multiway profile data. It is more common but still rarely used for multiway monopolar rating scales. One should never centre across bipolar rating scales due to the ambiguity of their orientation.

Column centring: Fibre centring across rows per column-tube combination

- Column centring indicates that for all $K \times J$ columns the means are calculated by summing over the index i and dividing by I, and that these means are subtracted from the data values, so that all columns are in deviation scores. As consequence the row or first-mode components are centred as well.
- As variables are often in the second mode, this is the recommended common option for profile data.

Tube centring: Fibre centring across tubes per row-column combination

- Tube centring indicates that for all $I \times J$ tubes the means are calculated by summing over the index k and dividing by K. These means are subtracted from the data values, so that all tubes are in deviation scores.
- This option is especially useful if subjects are in the third mode which occurs for multiway rating scales or multiway scaling data (*individual differences scaling*).

Double centring: Centre both columns and rows

- Centring across the first and second mode, also called double centring, means that for both the $K \times I$ and the $J \times K$ means are calculated by summing over the index j plus dividing by J, and summing over i plus dividing by I, respectively. These means are subtracted from the data values (and the overall mean is added in again to maintain the overall size of the data), so that all rows and all columns are in deviation scores.
- This option is especially useful for symmetric frontal slices when one wants to transform (squared) distances or dissimilarities into scalar products, for instance if one wants to perform a three-mode scaling analysis or an individual differences scaling analysis.

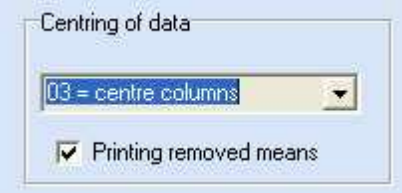
Frontal slice centring: per frontal slice

- For each frontal slice the mean is calculated by summing over all indices (i,j) and dividing by IJ . Within each slice, this mean is subtracted from the data values.

Overall centring

- The mean is calculated over all data values, and subsequently subtracted. This option is rarely used.

Printing removed means



The program offers the opportunity to print the removed means. Inspecting them can be a very useful exercise to get a feel of the variability in those means. An analysis of such means can be executed in PREPROC3 where an analysis of variance can be performed on the means.

4.1.6 Normalisation

Introduction

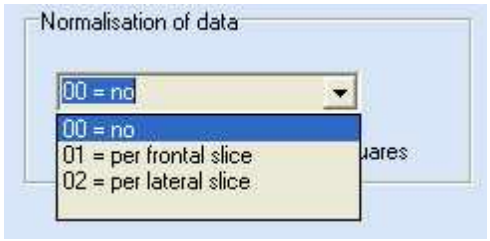
Normalisation is the process of equalising fibre or slice sums of squares. After normalisation either the fibres or slices have a total sum of squares of one.

Normalisation is entirely independent of the centring, so that the removed values do not generally represent variances.

No attempt has been made to divide the sum of squares by degrees of freedom, so that the removed values are not mean squares. This means, for instance, that the total sum of squares of the normalised three-way data is equal to *I* (frontal-slice normalisation) or equal to *J* (lateral-slice normalisation).

For horizontal-slice normalisation the ways should be swapped via PREPROC3, which program can also perform the required normalisations. When both centring and normalisation are selected, centring will always be carried out first.

Centring and normalisation should not be done orthogonally as the normalisation will destroy the centring.



Options

None: no normalisation. Default

Frontal slice normalisation

- Each frontal slice is normalised such that the sum of squares of its elements is equal to one.
- This option is generally used for symmetric frontal slices in multidimensional scaling designs, or when the variables, attributes, etc. are in the third mode.

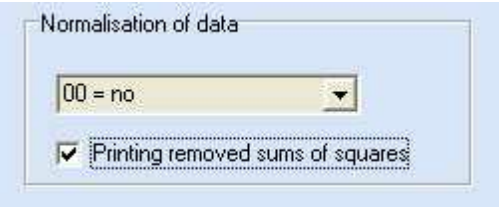
Lateral slice normalisation

- Each lateral slice is normalised such that the sum of squares of its elements is equal to one.
- This is the standard option for multiway profile data if the second modes are variables, attributes, etc.

Column fibre normalisation

- All $J \times K$ columns are normalised such that their sums of squares are equal to one.
- This option is not yet operational and is not generally recommended, but there are situations in which it might be useful.

Printing removed sums of squares



To check whether the normalisation was carried out properly the removed sums of squares may be listed.

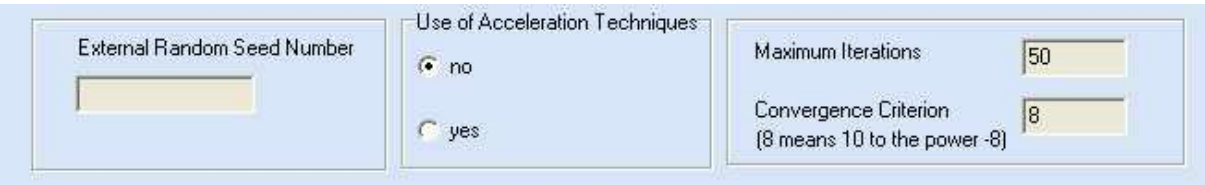
4.1.7 Iteration parameters

Introduction

The algorithm to estimate the parameters of the Tucker3 model is the heart of the program.

During the iterations the parameters of the model are improved step by step until no further improvement is possible according to the convergence criteria set by the user or until a preset maximum number of iterations is reached. The iterative algorithm needs a set of, preferably well-chosen, starting values.

In some cases the iterations can be speeded-up via a special procedure. In practice this is not often the case.



Convergence

The default for the convergence of the discrepancy or loss function is 10^{-8} (10 to the power -8). From this value the criterion for the norm of component matrices is derived.

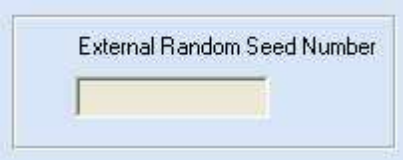
To change the convergence criterion specify the size of the exponent, e. g. 10 for 10^{-10}. On most machines the criterion should not be less than roughly 10^{-16}.

Maximum number of iterations

When the maximum number of iterations is reached, the program will finish as if it had converged and will print all relevant information.

The default is set at 50, the maximum value is 99999.

External random seed number



As mentioned above the algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. Normally a Tucker (1966) starting value is computed, but to search for different solutions for the parameters random initial configurations can be specified via using different random seed numbers.

This random seed specification is only necessary if a random initial configuration is requested. If the random seed is internally determined, the same random number will be used for each run of an analysis. Thus specifying the random seed allows changing the initial seed to create different initial configurations.

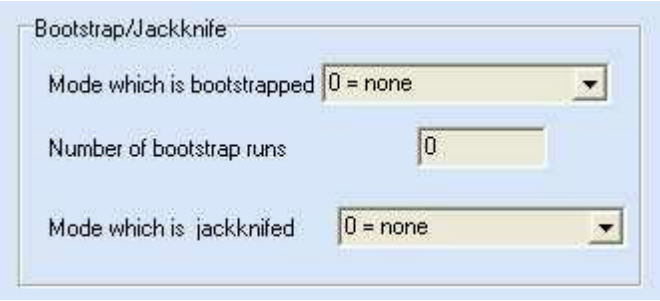
Use of acceleration technique

Even though the program is usually fast enough, some facilities have been built in to speed up convergence. One of those is the use of successive overrelaxations. The overrelaxation should converge, but in case of suspicious behaviour, such as negative differences of the discrepancy function, please rerun the analysis without the acceleration. Another method used to speed up the process is limited convergence checking and reduced output during iteration.

4.1.8 Bootstrap and jackknife analyses

Introduction

It is possible to perform bootstrap and jackknife analyses with the TUCKALS3 program, but these parts are still in its experimental stages. Users should first consult Kroonenberg (2008, p. 188, and the references mentioned there) before attempting to use these options.



Bootstrap

The bootstrap option performs the requested number of bootstrap analysis. It is always assumed that the variables or attributes are in the second mode, but the (stochastic) mode to be bootstrapped

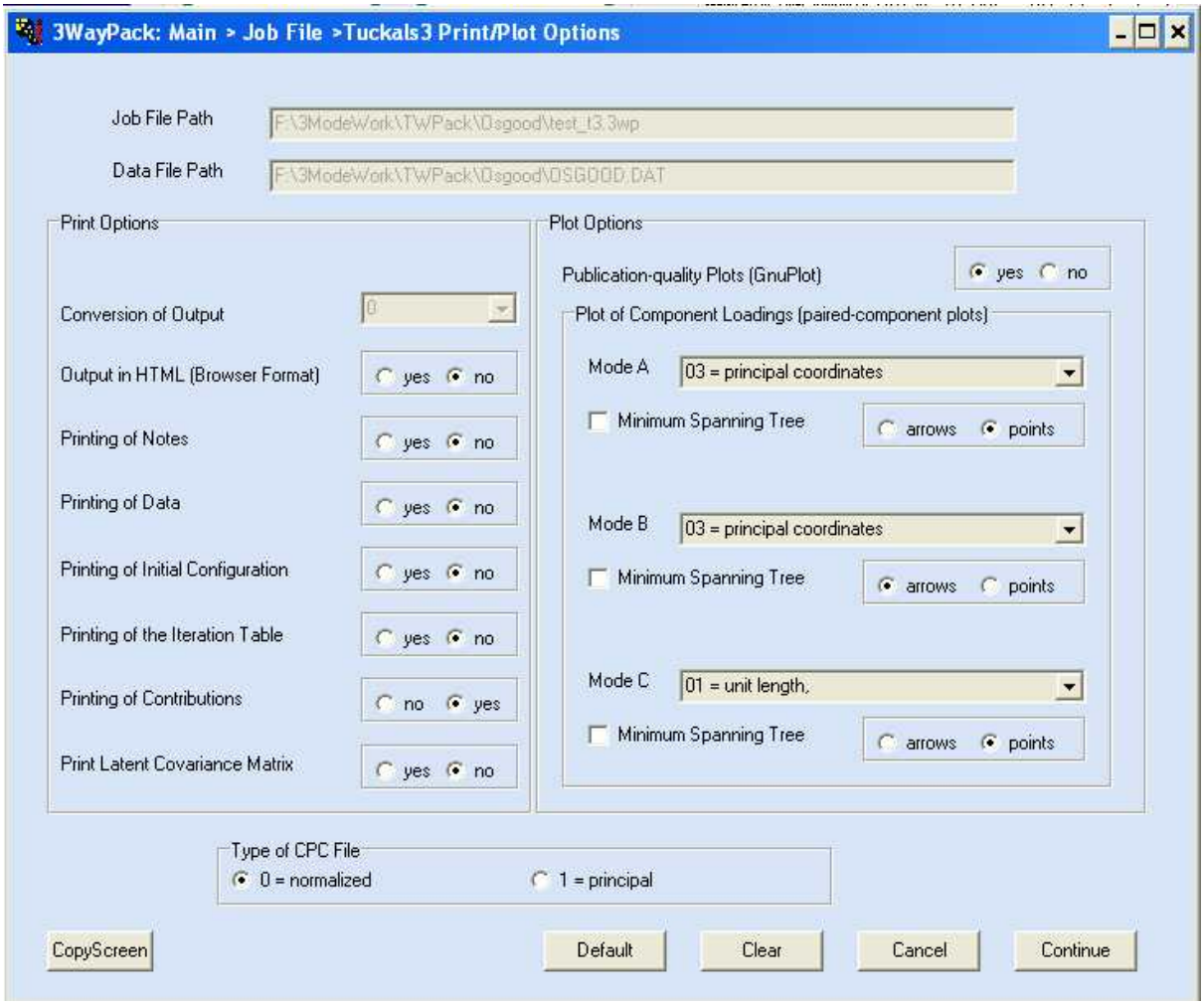
may be the first or the third mode. Be prepared for a serious wait if your problem is at all large. For full details see Kroonenberg (2008, Section 8.8.1).

Number of bootstrap runs
The number of draws with replacement from the requested mode can be specified. For each draw a complete analysis is performed, which can be time consuming.

Jackknife
The jackknife option allows for a jackknife procedure on the data to establish its statistical stability. For the requested mode a random half of each slice is left out of the analysis. Afterwards its values are estimated with the model derived. Therefore 2**I* analysis for a horizontal-slice jackknife or 2**K* analyses for a frontal-slice jackknife are performed. For full details see Kroonenberg (2008, Section 8.8.2) and his references.

4.2 Tuckals3 print/plot options

Introduction
The `Tuckals3 print/plot screen` allows you to specify the kind of print and plot output you want the program to produce. Selecting all options can produce quite a voluminous output..



Print options
The standard output is always written to a file <jobname>.out which can be viewed in the internal viewer or by an external editor. There is also an option to produce browser compatible and navigable output which can include plots.

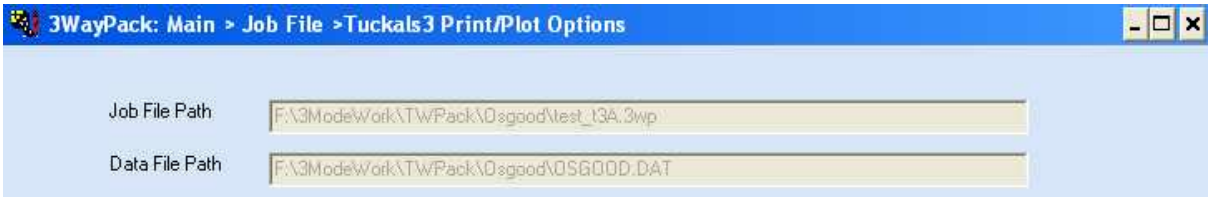
Plot options
Basic line plots, if requested, will always be printed in the standard output file, but publication-quality plots can only be obtained by choosing the GNUPLOT option. TUCKALS3 writes special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create button on the Main Screen the plots are created.

When also browser output has been requested .png-bitmap graphics but no editable are created which can be displayed in the browser.

Editable graphics in so-called vector format are produced when there is no browser output. For further details see the Gnuplot section.

4.2.1 Job and data files

Introduction
The TUCKALS3 print/plot options screen first provides information about the current job file and the data file.



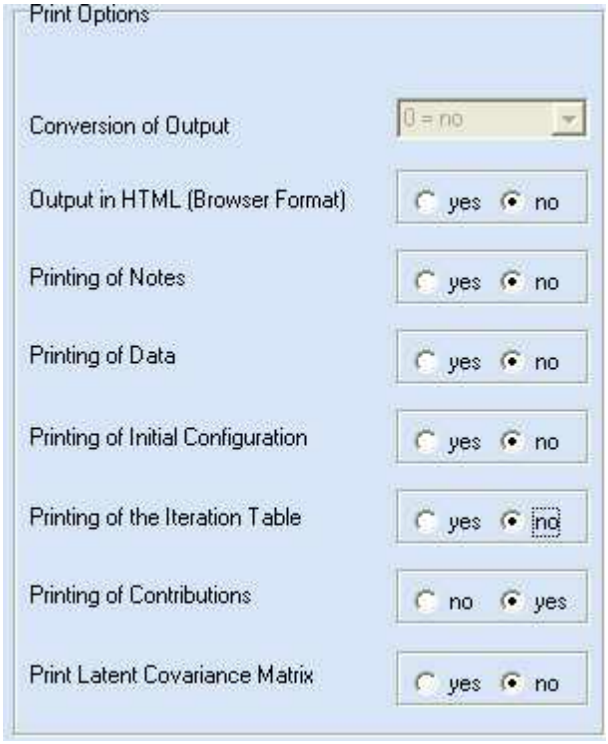
Job file
This text box indicates the name and location of the current job file. It cannot be edited in this screen. For another job or job name please go back to the Main Screen

Data file
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file

4.2.2 Print options

Introduction
The program can produce a large amount of output which is regulated via the options on the TUCKALS3 print/plot options screen. Even with the defaults, the output can be quite voluminous if there are many levels in one of the modes. The output increases even further when more than four components are specified for any one mode due to the way the component matrices are printed. The Conversion of Output option is disabled (see the general entry for details).

The option to produce browser output is designed to make navigation through the output easier. Some newer types of output may not yet be available in browser output.



Options

- Conversion of output (disabled but can be made available upon request)
- Output in browser format
- Printing of notes
- Printing of data
- Printing of initial configurations
- Printing of iteration table
- Printing of contributions to the fit
- Printing of latent covariance matrix

Navigable browser output

Introduction

When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser for the standard output.

The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.

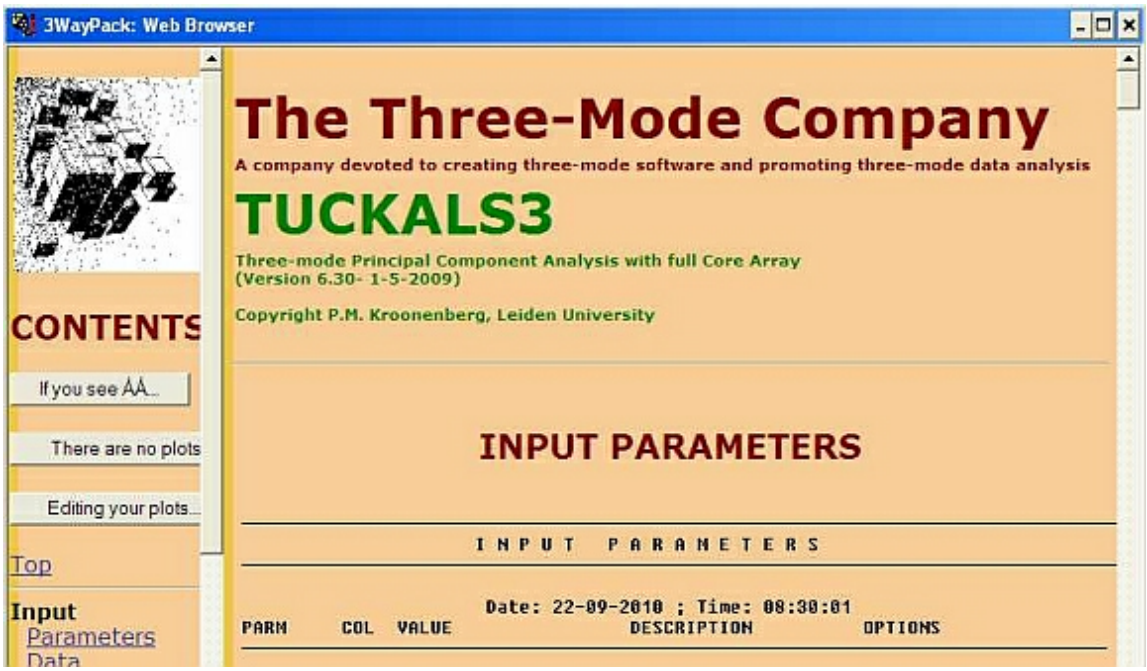
If plots have been requested and they have been created on the Main Screen via the View/ Create button, they will display in the browser as well.

The program will write three files with HTML code for displaying in a browser:

<jobname>.htm	When displaying the output outside 3WAYPACK, this file should be double-clicked, it contains the basic frame for displaying the HTML output.
<jobnameBr>.htm	This file is the side bar containing the navigations
<jobnameBd>.htm	This file contains the actual output.

If plots are available in png-format the browser will automatically insert them into the output. Some newer types of output may not yet be available in browser format.

To print options; *To plot options*



Print explanatory notes

Introduction

If this option is checked some basic interpretational comments with the respect to the output are printed in the output file.

Notes for some newer types of output may not yet be available.

To print options; *To plot options*

Print complete data set

Introduction

If checked the complete data file will be printed per frontal slice. This may generate quite a bit of output.

By default the first five and the last five lines/records of the data file are printed to allow checking whether the input data have been correctly read.

To print options; *To plot options*

Print initial configurations

Introduction

The algorithm for computing the parameters of the Tucker3 model needs starting values for the contents of the component matrices. The program will internally create such initial configurations or these configurations can be read into the program.

The default initial configuration is an approximation to an ordinary PCA on tall combination-mode matrices (e. g. $K \times I$ by J), and its results correspond to the output from the original algorithm by Tucker (1966).

The initial configuration, whether it is an internally determined or an external one, may be

inspected to investigate the (provisional) Tucker solution, or to ascertain the correct reading of an external configuration.

To print options; *To plot options*

Print iteration table

Introduction

Depending on your theoretical interest, you may want to see how quickly the program converges. As the progress of the iteration is not only written to the output file, but is also visible on screen in a DOS box, you have something to look at while the program is running.

When the acceleration technique is used please to check that the algorithm is converging correctly.

When there are missing data you may notice that the function is not converging monotonically (negative signs appear), but this is no reason for concern. This is due to an inaccurate, but efficient, calculation of the loss function during iteration; eventually the negative signs will disappear. If they do not you are in trouble, possibly due to too many missing data points or inadequate starting values.

If this option is not checked, a counter appears on the screen in a DOS box, and in the output file only the final results of the iterative process will be displayed but not the details of the iterations themselves

To print options; *To plot options*

Print fit contributions

Introduction

The program will print by default how well each subject, variable, condition has been fitted by the Tucker3 model specified. In addition, the fitted and residual sums of squares are plotted in a sums-of-squares plot.- SS(Res) versus SS(Fit); see for details Kroonenberg (2008, Section 12.6).

Sometimes you may not be interested in this information or the output becomes too voluminous. In such cases the printing of these fit measures may be suppressed.

Note that when external configurations are used the contributions will not be printed, because the total sums of squares are not necessarily partitioned into the fitted sums of squares and the residual sums of squares.

To print options; *To plot options*

Print latent covariance matrix

Introduction

The latent (or core) covariance matrix is an PQ by PQ matrix with mean squares for each combination of components p and q, summed over the components of the third mode (see Kroonenberg, 2008, Chapter 15, for a detailed explanation).

Checking this option will produce these latent or core covariances. The latent covariance matrix has not seen many practical uses, so far.

To print options; *To plot options*

4.2.3 Plot options

Introduction

The program has two options to produce plots for components:

Types of plots
- *paired-components plots*: pairs of components from one mode are plotting against each other
- *all-components plots*.: all components of a mode are plotted against the level labels or level numbers if no labels are present.

Scaling plots
There are also three ways to scale component plots:
- *normalised coordinates*: components have length one
- *unit-mean-square coordinates*: the mean-squared lengths of components have length one
- *principal coordinates*: components have lengths equal to the square roots of their eigenvalues.

Plots are included in the *standard output* but also *publication-quality plots* via the plotting program GNUPLOT can be produced.

Minimum spanning trees
The minimum-spanning tree can be calculated to determine whether points in the component space lie close together. At present only the information to draw these minimum-spanning trees are provided but they are not yet plotted by the program itself.

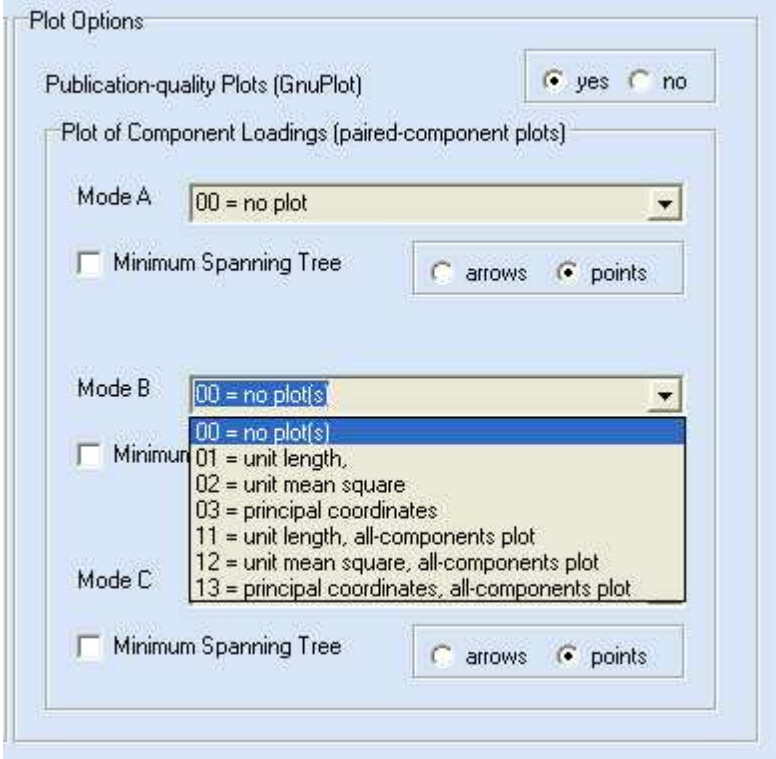
Arrows or points
One also has a choice to plot the levels of the mode either as points or as arrows emanating from the origin.

Component plots

Introduction
TUCKALS3 has various options to produce plots for components: paired-components plots and all-components plots. The components are printed using three different scalings but only one of them can be plotted in a single run. When P (or Q, or R) is less than or equal to 4, all components will be plotted against one another. Components 5 and higher will only be plotted against the first component.

GnuPlot
Publication-quality plots can be specified via the GnuPlot option. (see the Utilities section for more details on GNUPLOT).

Go to sections: Options; Arrows or points; Printing



Options

- *None*: no component plots; default.

- *Standard coordinates*: Unit lengths: lengths components = 1
  The lengths of all components are equal to one. Thus, the scatter in the plots is not adjusted to the explained variability (eigenvalues). This differs from the situation for the component loadings (column objects) in an ordinary two-mode PCA, but it is the standard way for component scores of the subjects (row objects).

- *Unit-mean-square coordinates*: lengths components = number of levels
  The lengths of all components are equal to the number of levels in a mode. The outward appearance of the plots is identical to that of the unit-length scaling above, due to the automatic adjustment of the scales of the plots. This scaling has the advantage that the values across modes are not influenced by different numbers of levels in the modes.

- *Principal coordinates*: lengths = square root of eigenvalues
  The lengths of the components are equal to the square root of standardised component weights. If the total sum of squares of the data is equal to the number of levels of a mode, then this scaling corresponds to the situation in regular two-mode PCA with standardised variables. However, the values are not necessarily comparable with regular 'loadings' (variable-component correlations), because this will depend on the centring and normalisation of the original variables. The effect of this scaling is that the scatter in the plot will depend on the differences in variability accounted for by the components.
  Principal coordinates have the advantage that distances between the levels in the space are correctly represented.

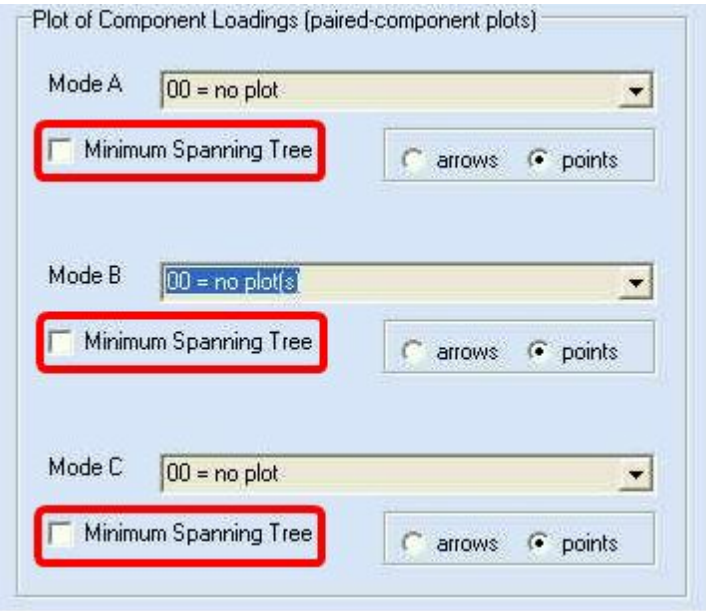
Arrows or points
For each of the modes the user can specify whether the levels of the modes should be represented by arrows or by points. Generally arrows are chosen if the components are in principal coordinates and points are selected in the other two cases, but there is no obligation to do so.

All-components plots
Each of the above options can be combined with an all-components plot. This can be done by choosing the same options but the value of the options is increased with 10.

Minimum-spanning tree

Introduction
The minimum-spanning tree is a path diagram which connects the points in a multidimensional space such that the sum of all distances between the points is as short as possible. Effectively this means that each point is connected with its nearest neighbour. With this information it is possible to assess in a lower dimensional subspace whether two points are close together in the higher-dimensional full space.

Thus one can use the minimum spanning tree to evaluate the closeness of two points in the two-dimensional plots which are produced by the program while the fitted component space is three-dimensional or more. This option is thus only useful for three-dimensional and higher dimensional spaces.

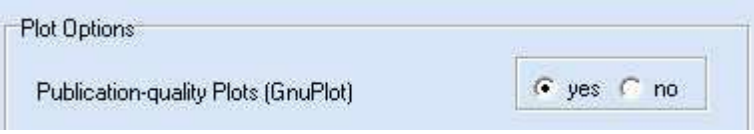


The program prints which points are connected at which distance based on the component space. At present there are not yet plots to display the minimum spanning tree in the component plots produced by the program.

Publication-quality plots

Introduction
When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create button on the Main Screen the plots are created. See the Utilities section for more details

on GNUPLOT.



When browser output has been selected only png-bitmap graphics are created which can be displayed in the browser.Editable graphics in so-called vector format are only produced when there is no browser output.

Please associate in Windows the extension .plt with the GNUPLOT program so that double clicking on the .plt file in the data directory the plots can be created outside 3WAYPACK.

For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

4.2.4 Type of CPC file

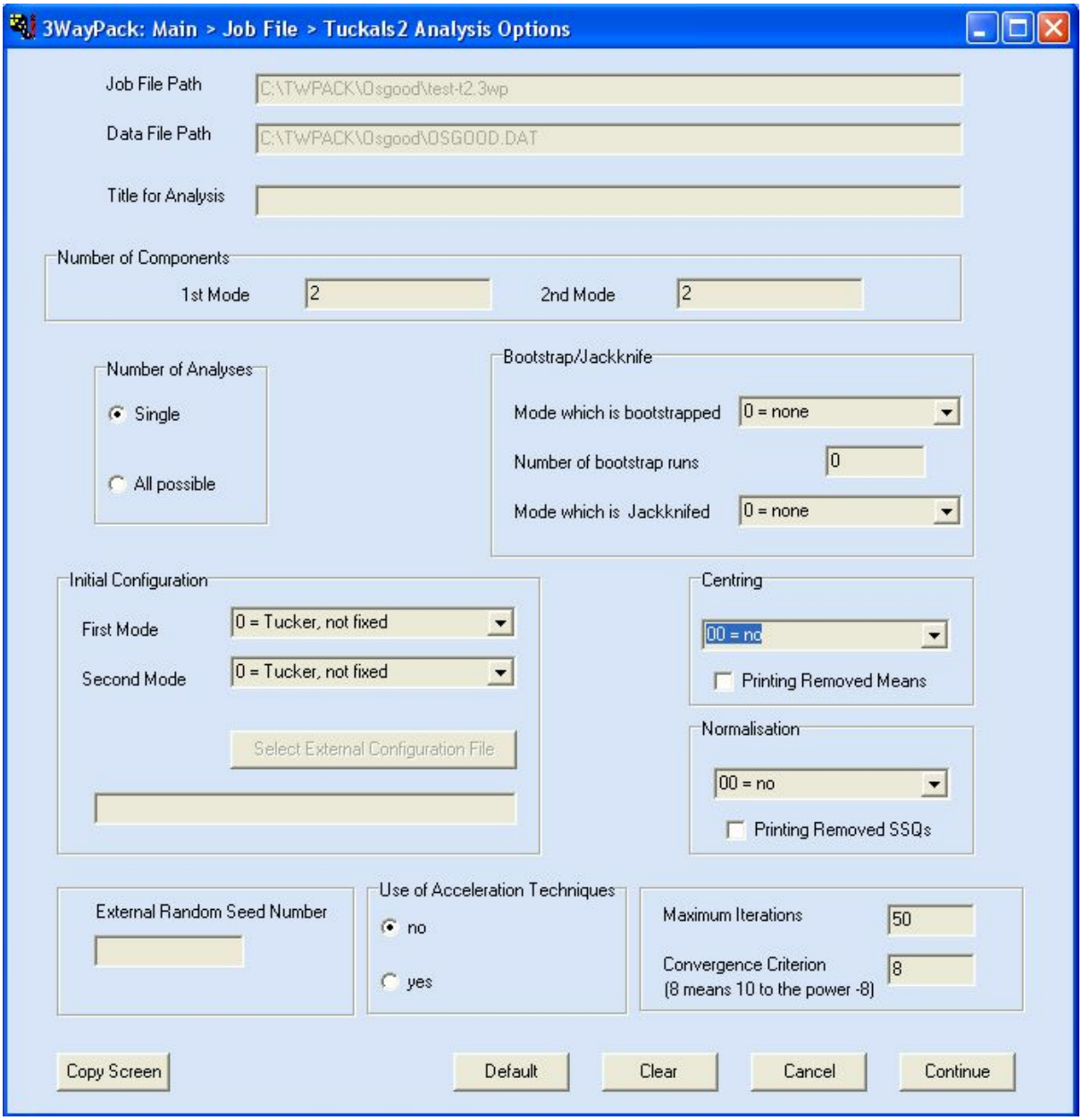
Introduction

The analysis programs write a CPC-file, which contains the coordinates of the components and the raw core array as well as the total and fitted sum of squares. This file stores this information for use in the postprocessing programs. By default the normalised coordinates should be used. For very specific core rotations the principal coordinates are required.

4.3 Tuckals2 analysis options

Introduction

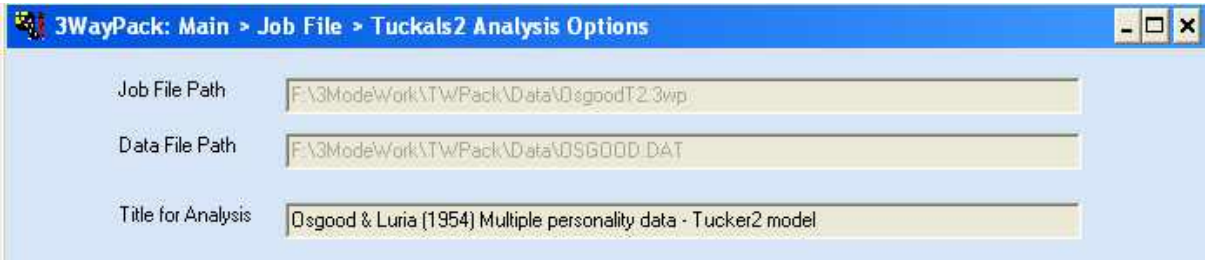
The Tuckals2 Analysis Options screen is accessed via the Job Specification screen via the Analysis Options button. On the present screen the details of the analysis can be specified such as the number of components, the number of analyses, execution of an external analysis, bootstrapping, jack-knifing, specification of centring, normalisation, type of initial configurations, algorithm specifics, etc.



4.3.1 Job, data, and title

Introduction

The Tuckals2 Analysis Options screen first provides information about the current job file, the data file and gives the opportunity to specify a title for labelling the output.



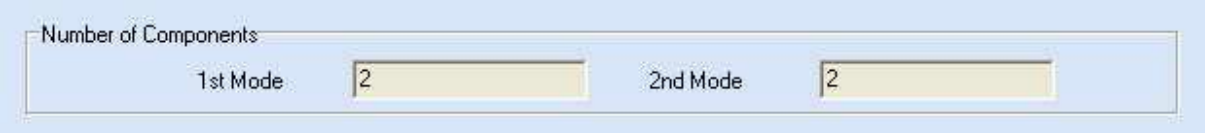
Job file
This text box indicates the name and location of the current job file. It cannot be edited in this screen. For another job or job name please go back to the Main Screen

Data file
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file.

Title for the analysis
The title specified in this text box will be used to label the output. It is suggested to specify here any specific details which cannot easily be read from the output, such as specifics about the data and their origin, etc.

4.3.2 Number of components

Introduction
By default the first two modes have two components and the third mode a single one.



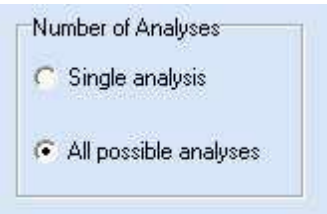
Specifications
- The specification of the numbers of components without selecting any other option is sufficient for an analysis, be it a trivial one.
- The number of components must be less or equal to the number of levels.
- The number of components have to satisfy theTucker2 variant of the minimum-product rule, i.e. the product of the number of components of one mode times the number of levels in the third mode may not be smaller than the number of components of the remaining mode

First mode
The number of components (P) for the first mode should be less than or equal to the number of levels (I). Moreover, $P \leq Q \times K$.

Second mode
The number of components (Q) for the second mode should be less than or equal to the number of levels (J). Moreover, $Q \leq P \times K$.

4.3.3 Number of analyses

Introduction
By default, TUCKALS2 will do a single analysis based on the numbers of components supplied.



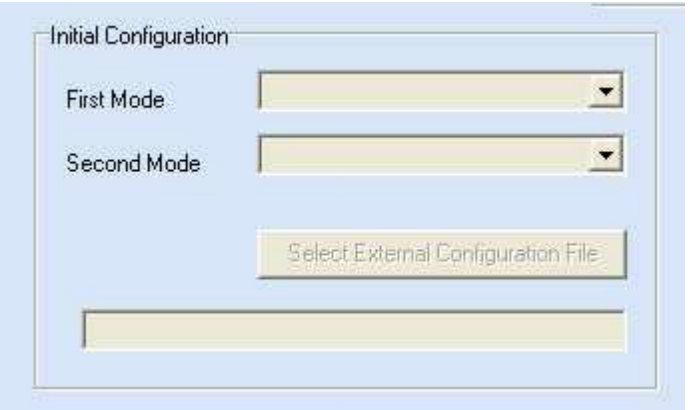
All possible analyses
In order to decide how many components the data support and to decide which solution is most appropriate, the program can calculate All possible analyses for the Tucker2 model with numbers of components equal to or less than the numbers of components indicated in the Number of components panel up to a 5×5-solution. The output per analysis is minimised to lower execution times. For full details of the All Possible analyses option and examples of the output see Kroonenberg (2008, Section 8.5).

Full output can only be obtained from a single analysis.

Because of the minimum-product rule, i.e. the product of the number of components of one mode times the number of levels in the third mode may not be smaller than the number of components of the remaining mode, not all solutions are possible. For instance, a 3×2×1-solution violates the minimum-product rule because 2×1 < 3. A 3×2×1 specification for the components will provide a 2×2×1-solution.

4.3.4 Initial configurations

Introduction
The iterative procedure to solve the parameter estimation of the Tucker2 model, TUCKALS2, needs initial configurations for the component matrices **A**, and **B**. If there is no special reason to do otherwise, it is advised to stay with the default option as it will provide excellent starting values. In fact, if there is a perfect solution, it will be found by the default procedure. The initial configurations of the three modes have the same options but it is not necessary to use the same one for every mode. For full details see Kroonenberg (2008, Section 5.7.3).



Options
0 = Tucker, not fixed (default)
- The initial configuration is derived from the data.
- By default, the program computes an initial configuration for two modes by calculating ten steps towards a solution as described in Tucker (1966). This involves stacking the slices underneath each other into a tall matrix, e. g. $J \times K$ by I (a process called *matricisation*), and performing a PCA on this tall matrix.
- Generally the final TUCKALS2 solution will be close to the starting Tucker2 solution, especially with a limited number of components.

1 = External, fixed

- An external configuration is used and it is not modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, this configuration may be read in and kept fixed during the analysis. In other words, the rest of the solution is fitted around the external fixed configuration.
- Comparing the fits without and with the fixed solution will allow an assessment of how well the external configuration fits the data.
- The external solution is assumed to be orthonormal (orthogonal with unit-length components). If this is not the case the program will issue a warning, but will proceed as if the configuration is correct. In such a case the output will not necessarily be interpretable; especially the partitioning of the fit might no longer be valid.
- The external configuration should be in a file accessible to the program (preferably `<datasetname>.ext`) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.

2= External, not fixed.

- An external configuration is available and it is modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, you may request the program to read this configuration and use it as a starting configuration.
- The external configuration should be in a file accessible to the program (preferably `<datasetname>.ext`) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.

- For full details see Kroonenberg (2008, Section 4.10.7).

3 = Random, not fixed.

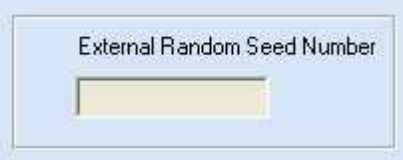
- A random initial (orthonormalised) configuration will be used.
- This option is primarily useful for research purposes.
- It could for instance be used to assess the sensitivity of the program to different starting solutions. (The answer is 'not sensitive'.)
- To create different starting configurations for each run, change the random seed.

- For full details see Kroonenberg (2008, Section 4.10.7).

External configurations

- External configurations for different modes should be together in one external configuration file in the order of the modes.Thus in the file a first-mode configuration should always precede a second-mode one and a second-mode configuration a third-mode one.
- Once an external configuration is specified for any mode, the program will not rest until you have supplied a file name.
- The numbers in the file should be in free format, i.e. there should be at least one blank between numbers. The more decimals the better.

External random seed number



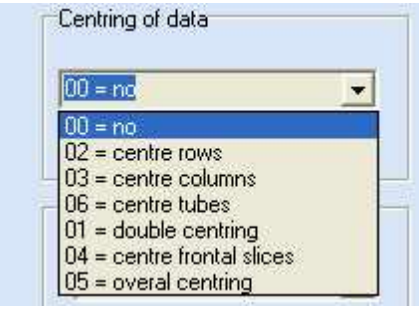
The algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. Normally a Tucker (1966) starting value is computed, but to search for different solutions for the parameters different random initial configurations can be specified via using different random seed numbers.

4.3.5 Centring

Introduction

Centring is the process of removing one or more means from the data. For three-way data there are several possibilities of which the more useful ones are included in the three-mode programs. Some additional options are available in PREPROC3.

Centring across a mode will cause the components of that mode to be centred. Note that centring fundamentally changes the data, so that it is virtually impossible to compare analyses of differently centred data. For further centring options see Preproc3 centring.



Options

None: no centring: Default

Row centring: Fibre centring across columns per row-tube combination

- Row centring indicates that for all $I \times K$ rows the means are calculated by summing over the index j and dividing by J. These means are subtracted from the data values, so that all rows are in deviation scores. As a consequence the row of first-mode components are centred as well.
- As variables are mostly in the second mode, this is a unusual option for multiway profile data. It is more common but still rarely used for multiway monopolar rating scales. One should never centre across bipolar rating scales due to the ambiguity of their orientation.

Column centring: Fibre centring across rows per column-tube combination

- Column centring indicates that for all $K \times J$ columns the means are calculated by summing over the index i and dividing by I, and that these means are subtracted from the data values, so that all columns are in deviation scores.
- As variables are often in the second mode, this is the recommended common option for profile data.

Tube centring: Fibre centring across tubes per row-column combination

- Tube centring indicates that for all $I \times J$ tubes the means are calculated by summing over the index k and dividing by K. These means are subtracted from the data values, so that all tubes are in deviation scores.
- This option is especially useful if subjects are in the third mode which occurs for multiway rating scales or multiway scaling data (individual differences scaling).

Double centring: Centre both columns and rows

- Centring across the first and second mode, also called double centring, means that for both the $K \times I$ and the $J \times K$ means are calculated by summing over the index j plus dividing by J, and summing over i plus dividing by I, respectively. These means are subtracted from the data values (and the overall mean is added in again to maintain the overall size of the data), so that all rows and all columns are in deviation scores.
- This option is especially useful for symmetric frontal slices when one wants to transform (squared) distances or dissimilarities into scalar products, for instance if one wants to perform an individual differences scaling analysis.

Frontal slice centring: per frontal slice

- For each frontal slice the mean is calculated by summing over all indices (i,j) and dividing by IJ. Within each slice, this mean is subtracted from the data values.

Overall centring

- The mean is calculated over all data values, and subsequently subtracted. This option is rarely used.

Printing removed means

The program offers the opportunity to print the removed means. Inspecting them can be a very useful exercise to get a feel of the variability in those means. An analysis of such means can be executed in PREPROC3 where an analysis of variance can be performed on the means.

4.3.6 Normalisation

Introduction

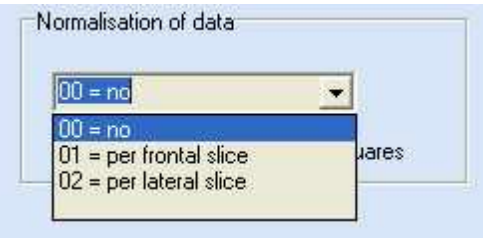
Normalisation is the process of equalising the fibre or slice sums of squares. After normalisation either the fibres or slices have a total sum of squares of one.

Normalisation is entirely independent of the centring, so that the removed values do not generally represent variances.

No attempt has been made to divide the sum of squares by degrees of freedom, so that the removed values are not mean squares. This means, for instance, that the total sum of squares of the normalised three-way data is equal to I (frontal-slice normalisation) or equal to J (lateral-slice normalisation).

For horizontal-slice normalisation the ways should be swapped via PREPROC3, which program can also perform the required normalisations. When both centring and normalisation are selected, centring will always be carried out first.

Centring and normalisation should not be done orthogonally as the normalisation will destroy the centring.



Options

None: no normalisation. Default

Frontal slice normalisation

- Each frontal slice is normalised such that the sum of squares of its elements is equal to one.
- This option is generally used for symmetric frontal slices in multidimensional scaling designs.

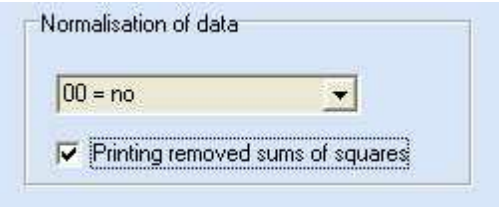
Lateral slice normalisation

- Each lateral slice is normalised such that the sum of squares of its elements is equal to one.
- This is the standard option for multiway profile data if the second modes are variables, attributes, etc.

Column fibre normalisation

- All $J \times K$ columns are normalised such that their sums of squares are equal to one.
- This option is not yet operational and is not generally recommended, but there are situations in which it might be useful.

Printing removed sums of squares



To check whether the normalisation was carried out properly the removed sums of squares may be listed.

4.3.7 Iteration parameters

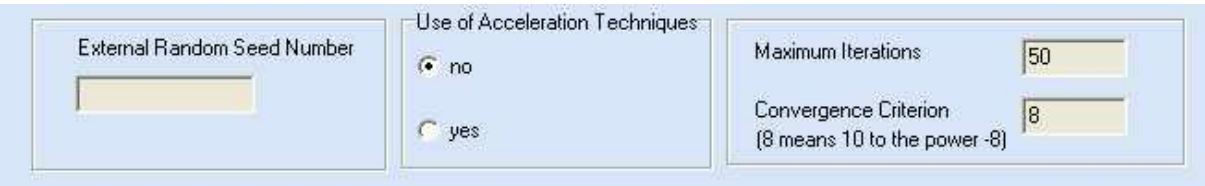
Introduction

The algorithm to estimate the parameters of the Tucker2 model is the heart of the program.

During the iterations the parameters of the model are improved step by step until no further improvement is possible according to the convergence criteria set by the user or until a preset maximum number of iterations is reached. The iterative algorithm needs a set of, preferably well-

chosen, starting values.

In some cases the iterations can be speeded-up via a special procedure. In practice this is not often the case.



Convergence

The default for the convergence of the discrepancy or loss function is 10^{-8} (10 to the power -8). From this value the criterion for the norm of component matrices is derived.

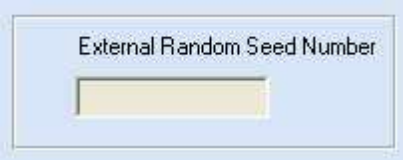
To change the convergence criterion specify the size of the exponent, e. g. 10 for 10^{-10}. On most machines the criterion should not be less than roughly 10^{-16}.

Maximum number of iterations

When the maximum number of iterations is reached, the program will finish as if it had converged and will print all relevant information.

The default is set at 50, the maximum value is 99999.

External random seed number



As mentioned above the algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. Normally a Tucker (1966) starting value is computed, but to search for different solutions for the parameters random initial configurations can be specified via using different random seed numbers.

This random seed specification is only necessary if a random initial configuration is requested. If the random seed is internally determined, the same random number will be used for each run of an analysis. Thus specifying the random seed allows changing the initial seed to create different initial configurations.

Use of acceleration technique

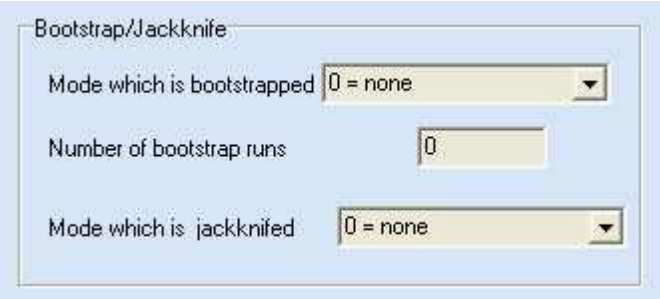
Even though the program is usually fast enough, some facilities have been built in to speed up

convergence. One of those is the use of successive overrelaxations. The overrelaxation should converge, but in case of suspicious behaviour, such as negative differences of the discrepancy function, please rerun the analysis without the acceleration. Another method used to speed up the process is limited convergence checking and reduced output during iteration.

4.3.8 Bootstrap and jackknife analyses

Introduction
It is possible to perform bootstrap and jackknife analyses with the TUCKALS2 program, but these parts are still in its experimental stages. Users should first consult Kroonenberg (2008, p. 188, and the references mentioned there) before attempting to use these options.

For the Tucker2 model these options have not yet been tested so that users are advised not to use them at present, except out of curiosity.



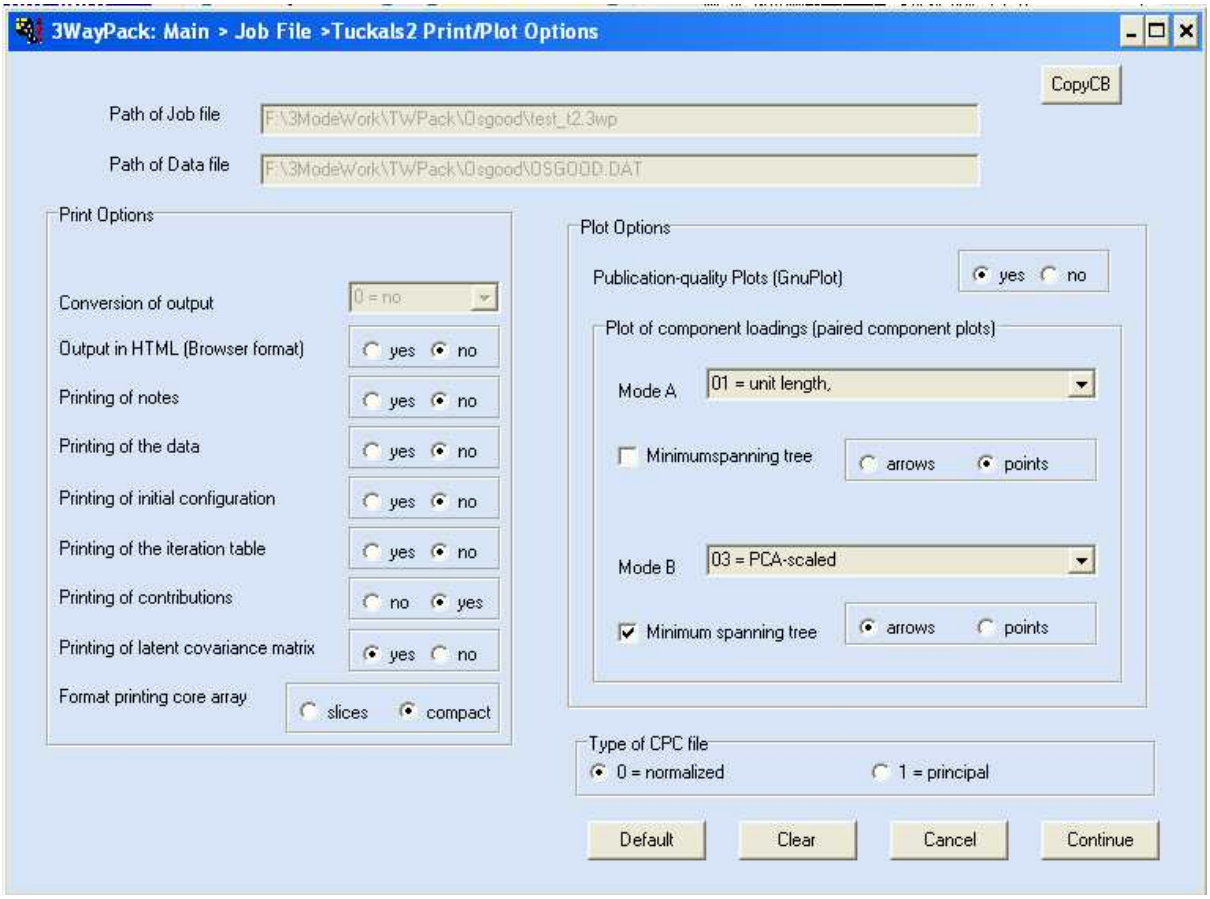
Bootstrap
The bootstrap option performs the requested number of bootstrap analysis. It is always assumed that the variables or attributes are in the second mode, but the (stochastic) mode to be bootstrapped may be the first or the third mode. Be prepared for a serious wait if your problem is at all large.

Number of bootstrap runs
The number of draws with replacement from the requested mode can be specified. For each draw a complete analysis is performed, which can be time consuming.

Jackknife
The jackknife option allows for a jackknife procedure on the data to establish its statistical stability. For the requested mode a random half of each slice is left out of the analysis. Afterwards its values are estimated with the model derived. Therefore 2**I* analysis for a horizontal-slice jackknife or 2**K* analyses for a frontal-slice jackknife are performed. For full details see Kroonenberg (2008, Section 8.8.2) and his references.

4.4 Tuckals2 print/plot options

Introduction
The Tuckals2 print/plot screen allows you to specify the kind of print and plot output you want the program to produce. Selecting all options can produce quite a voluminous output.



Print options

The standard output is always written to a file <jobname>.out which can be viewed in the internal viewer or by an external editor. There is also an option to produce browser compatible and navigable output which can include plots.

Plot options

Basic line plots, if requested, will always be printed in the standard output file, but publication-quality plots can only be obtained by choosing the GNUPLOT option. TUCKALS2 writes special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create button on the Main Screen the plots are created.

When also browser output has been requested .png-bitmap graphics but no editable graphs are created which can be displayed in the browser.

Editable graphics in so-called vector format are produced when there is no browser output. For further details see the Gnuplot section.

4.4.1 Job and data files

Introduction

The TUCKALS2 print/plot options screen first provides information about the current job file and the data file. .

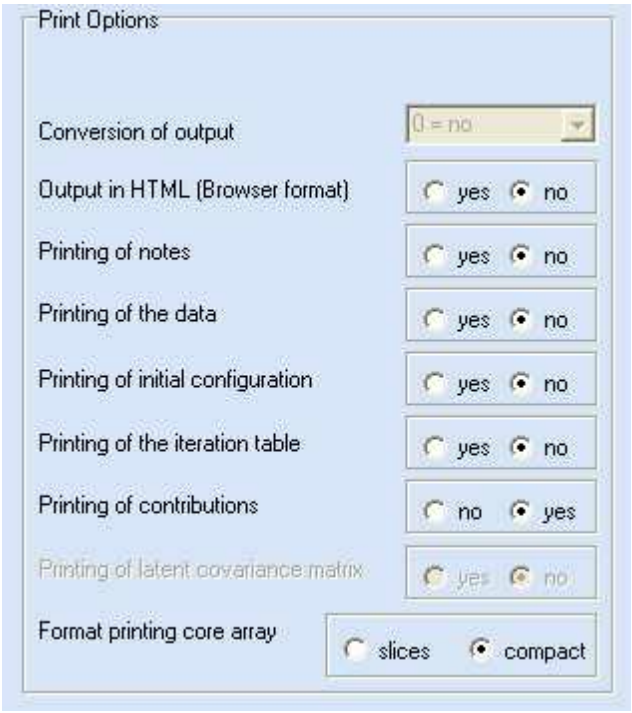
Job file
This text box indicates the name and location of the current job file. It cannot be edited in this screen. For another job or job name please go back to the Main Screen

Data file
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file.

4.4.2 Print options

Introduction
The program can produce a large amount of output which is regulated via the options on the TUCKALS2 print/plot options screen. Even with the defaults, the output can be quite voluminous if there are many levels in one of the modes. The output increases even further when more than four components are specified for any one mode due to the way the component matrices are printed. The Conversion of Output option is disabled (see the general entry for details).

The option to produce browser output is designed to make navigation through the output easier. Some newer types of output may not yet be available in browser output.



Options
- Conversion of output (disabled but can be made available upon request)
- Output in browser format

- Printing of notes
- Printing of data
- Printing of initial configurations
- Printing of iteration table
- Printing of contributions to the fit
- Printing of extended core array
- Printing of latent covariance matrix

Navigable browser output

Introduction

When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser for the standard output.

The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.

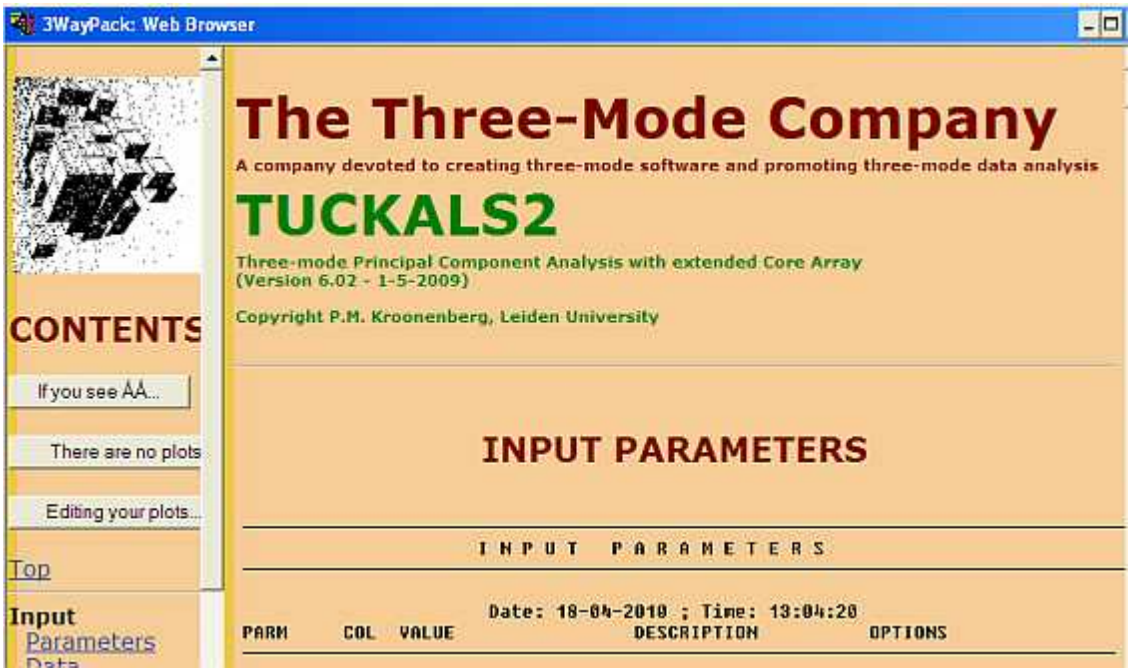
If plots have been requested and they have been created on the Main Screen via the View/ Create button, they will display in the browser as well.

The program will write three files with HTML code for displaying in a browser:

<jobname>.htm	When displaying the output outside 3WAYPACK, this file should be double-clicked, it contains the basic frame for displaying the HTML output.
<jobnameBr>.htm	This file is the side bar containing the navigations
<jobnameBd>.htm	This file contains the actual output.

If plots are available in png-format the browser will automatically insert them into the output. Some newer types of output may not yet be available in browser format.

To print options*; *To plot options



Print explanatory notes

Introduction

If this option is checked some basic interpretational comments with the respect to the output are printed in the output file.

Notes for some newer types of output may not yet be available.

To print options*; ;*To plot options

Print complete data set

Introduction

If checked the complete data file will be printed per frontal slice. This may generate quite a bit of output.

By default the first five and the last five lines/records of the data file are printed to allow checking whether the input data have been correctly read.

To print options*; *To plot options

Print initial configurations

Introduction

The algorithm for computing the parameters of the Tucker3 model needs starting values for the contents of the component matrices. The program will internally create such initial configurations or these configurations can be read into the program.

The default initial configuration is an approximation to an ordinary PCA on tall combination-mode matrices (e. g. $K \times I$ by J), and its results correspond to the output from the original algorithm by Tucker (1966).

The initial configuration, whether it is an internally determined or an external one, may be inspected to investigate the (provisional) Tucker solution, or to ascertain the correct reading of an external configuration.

To print options*; *To plot options

Print iteration table

Introduction

Depending on your theoretical interest, you may want to see how quickly the program converges. As the progress of the iteration is not only written to the output file, but is also visible on screen in a DOS box, you have something to look at while the program is running.

When the acceleration technique is used please check that the algorithm is converging correctly.

When there are missing data you may notice that the function is not converging monotonically (negative signs appear), but this is no reason for concern. This is due to an inaccurate, but efficient, calculation of the loss function during iteration; eventually the negative signs will disappear. If they do not you are in trouble, possibly due to too many missing data points or inadequate starting values.

If this option is not checked, a counter appears on the screen in a DOS box, and in the output file only the final results of the iterative process will be displayed but not the details of the iterations themselves

To print options*; *To plot options

Print fit contributions

Introduction

The program will print by default how well each subject, variable, condition has been fitted by the Tucker2 model specified. In addition, the fitted and residual sums of squares are plotted in a sums-of-squares plot.- SS(Res) versus SS(Fit); see for details Kroonenberg (2008, Section 12.6).

Sometimes you may not be interested in this information or the output becomes too voluminous. In such cases the printing of these fit measures may be suppressed.

Note that when external configurations are used the contributions will not be printed, because the total sums of squares are not necessarily partitioned into the fitted sums of squares and the residual sums of squares.

To print options*; *To plot options

Print latent covariance matrix

Introduction

The latent (or core) covariance matrix is an PQ by PQ matrix with mean squares for each combination of components p and q, summed over the components of the third mode (see Kroonenberg, 2008, Chapter 15, for a detailed explanation).

Checking this option will produce these latent or core covariances. The latent covariance matrix has not seen many practical uses, so far.

(This option is not yet operational)

To print options*; *To plot options

Print extended core array

Introduction

The core array in the Tucker model is based on the reduction of the data along the first two ways rather than all three ways as is the case in the Tucker3 model. Therefore the core array has dimensions $P \times Q \times K$ and has the shape of a cake with K slices of order $P \times Q$. It is called the *extended core array*.

Printing extended core arrays

The extended core array can be printed in two ways

1. *Per frontal $P \times Q$ slice* (Option: ***slices***); Here $P = Q = 2$

k=1

g_{111}	g_{121}
g_{211}	g_{221}

k=2

g_{112}	g_{122}
g_{212}	g_{222}

k=..

etc.

This option is most advantages if one want to evaluate the off-diagonal elements with respect to the diagonal ones.

2. *Corresponding elements below each other* (Option: **compact**). *Default*

k=1	g_{111}	g_{121}	g_{211}	g_{221}
k=2	g_{112}	g_{122}	g_{212}	g_{222}
k= .	...	...	...	...

This option is most advantageous when one wants to evaluate elements in the same position in the K slices, say all g_{pqk}, k = 1,..,K.

4.4.3 Plot options

Introduction
The program has two options to produce plots for components:

Types of plots
- *paired-components plots*: pairs of components from one mode are plotting against each other
- *all-components plots*.: all components of a mode are plotted against the level labels or level numbers if no labels are present.

Scaling plots
There are also three ways to scale component plots:
- *normalised coordinates*: components have length one
- *unit-mean-square coordinates*: the mean-squared lengths of components have length one
- *principal coordinates*: components have lengths equal to the square roots of their eigenvalues.

Plots are included in the *standard output* but also *publication-quality plots* via the plotting program GNUPLOT can be produced.

Minimum spanning trees
The minimum-spanning tree can be calculated to determine whether points in the component space lie close together. At present only the information to draw these minimum-spanning trees are provided but they are not yet plotted by the program itself.

Arrows or points
One also has a choice to plot the levels of the mode either as points or as arrows emanating from the origin.

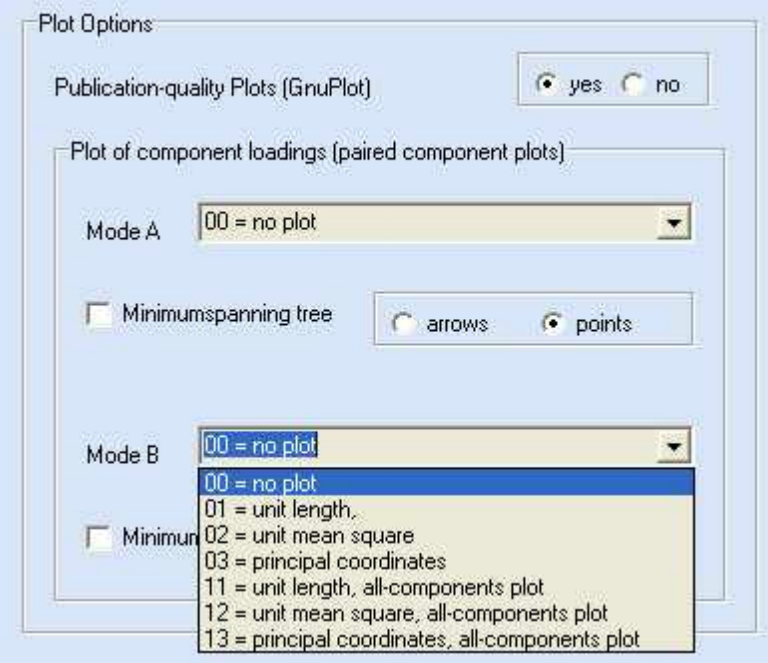
Component plots

Introduction
TUCKALS2 has various options to produce plots for components: paired-components plots and all-components plots. The components are printed using three different scalings but only one of them

can be plotted in a single run. When *P* (or *Q*) is less than or equal to 4, all components will be plotted against one another. Components 5 and higher will only be plotted against the first component.

GnuPlot
Publication-quality plots can be specified via the GnuPlot option. (see the Utilities section for more details on GNUPLOT).

Go to Options; Arrows or points; Printing



Options

- *None*: no component plots; default.

- *Standard coordinates*: Unit lengths: lengths components = 1
  The lengths of all components are equal to one. Thus, the scatter in the plots is not adjusted to the explained variability (eigenvalues). This differs from the situation for the component loadings (column objects) in an ordinary two-mode PCA, but it is the standard way for component scores of the subjects (row objects).

- *Unit-mean-square coordinates*: lengths components = number of levels
  The lengths of all components are equal to the number of levels in a mode. The outward appearance of the plots is identical to that of the unit-length scaling above, due to the automatic adjustment of the scales of the plots. This scaling has the advantage that the values across modes are not influenced by different numbers of levels in the modes.

- *Principal coordinates*: lengths = square root of eigenvalues
  The lengths of the components are equal to the square root of standardised component weights. If the total sum of squares of the data is equal to the number of levels of a mode, then this scaling corresponds to the situation in regular two-mode PCA with standardised variables. However, the values are not necessarily comparable with regular 'loadings' (variable-component correlations),

because this will depend on the centring and normalisation of the original variables. The effect of this scaling is that the scatter in the plot will depend on the differences in variability accounted for by the components.
Principal coordinates have the advantage that distances between the levels in the space are correctly represented.

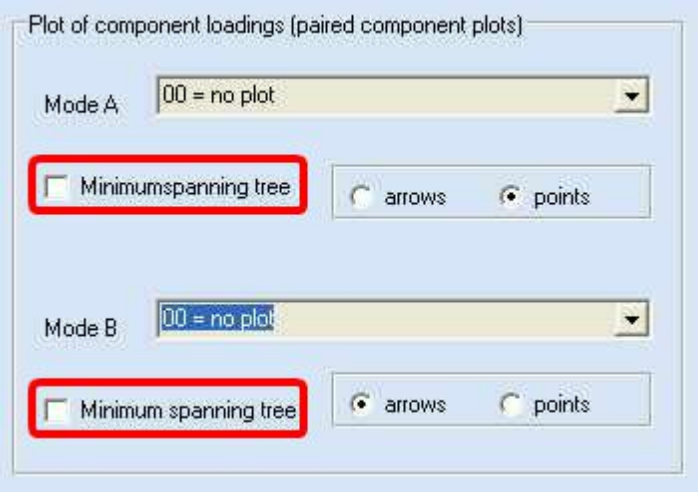
Arrows or points
For each of the modes the user can specify whether the levels of the modes should be represented by arrows or by points. Generally arrows are chosen if the components are in principal coordinates and points are selected in the other two cases, but there is no obligation to do so.

All-components plots
Each of the above options can be combined with an all-components plot. This can be done by choosing the same options but the value of the options is increased with 10.

Minimum-spanning tree

Introduction
The minimum-spanning tree is a path diagram which connects the points in a multidimensional space such that the sum of all distances between the points is as short as possible. Effectively this means that each point is connected with its nearest neighbour. With this information it is possible to assess in a lower dimensional subspace whether two points are close together in the higher-dimensional full space.

Thus one can use the minimum spanning tree to evaluate the closeness of two points in the two-dimensional plots which are produced by the program while the fitted component space is three-dimensional or more. This option is thus only useful for three-dimensional and higher dimensional spaces.

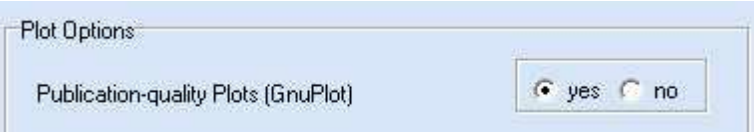


The program prints which points are connected at which distance based on the component space. At present there are not yet plots to display the minimum spanning tree in the component plots produced by the program.

Publication-quality plots

Introduction
When plots are requested they will always be printed in the standard output, but publication-quality

plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create button on the Main Screen the plots are created. See the Utilities section for more details on GNUPLOT.



When browser output has been selected only png-bitmap graphics are created which can be displayed in the browser..Editable graphics in so-called vector format are only produced when there is no browser output.

Please associate in Windows the extension .plt with the GNUPLOT program so that double clicking on the .plt file in the data directory the plots can be created outside 3WAYPACK.

For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

4.4.4 Type of CPC file

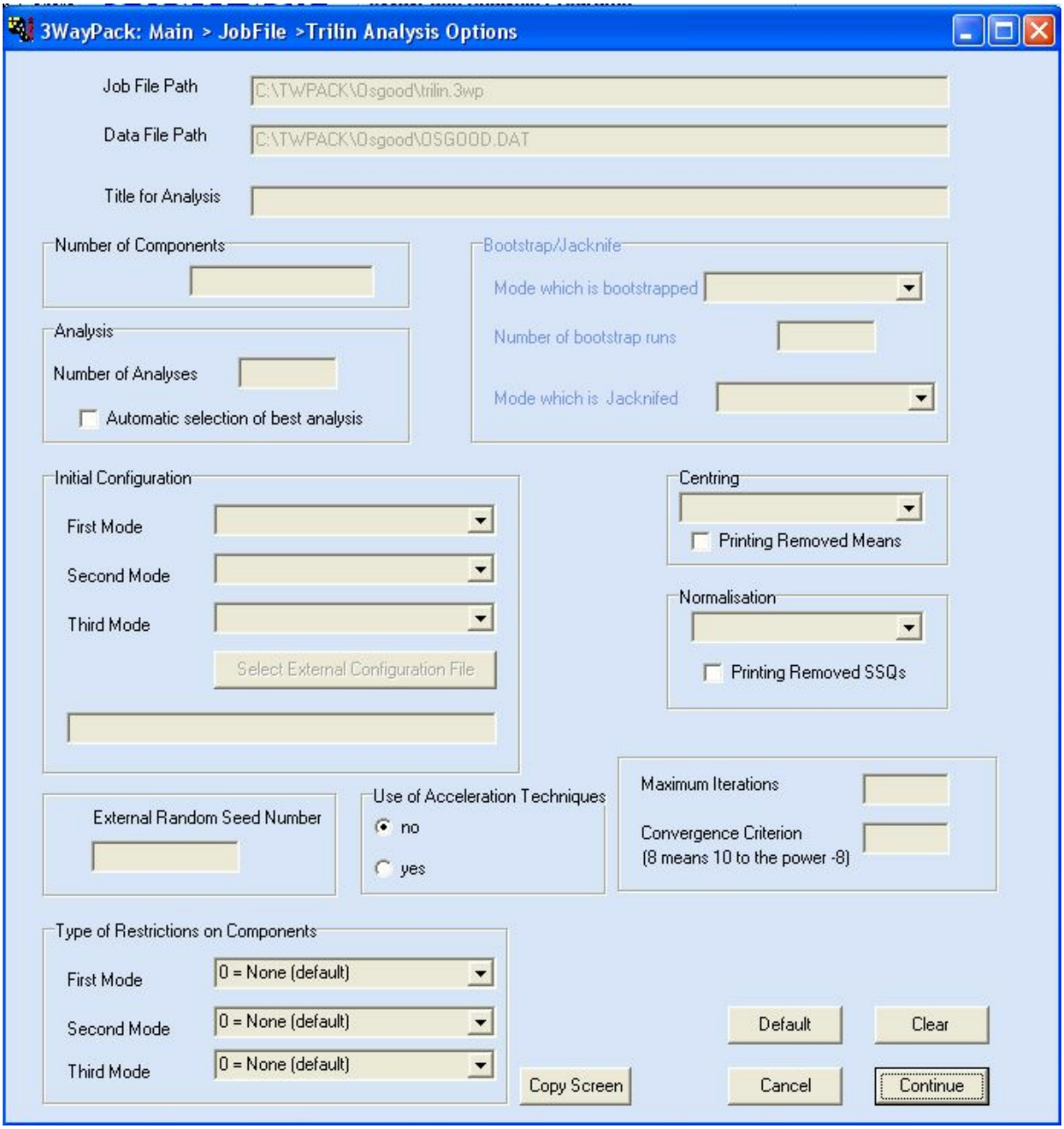
Introduction

The analysis programs write a CPC-file, which contains the coordinates of the components and the raw core array as well as the total and fitted sum of squares. This file stores this information for use in the postprocessing programs. By default the normalised coordinates should be used. For very specific core rotations the principal coordinates are required.

4.5 Trilin analysis options

Introduction

The Trilin Analysis Options screen is accessed via the Job Specification screen via the Analysis Options button. On the present screen the details of the analysis can be specified such as the number of components, the number of analyses, execution of an external analysis, bootstrapping, jack-knifing, specification of centring, normalisation, type of initial configurations, algorithm specifics, etc.



4.5.1 Job, data, and title

Introduction

The Trilin analysis options screen first provides information about the current job file, the data file and gives the opportunity to specify a title for labelling the output.

Job file
This text box indicates the name and location of the current job file. It cannot be edited in this screen. For another job or job name please go back to the Main Screen

Data file
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file.

Title for the analysis
The title specified in this text box will be used to label the output. It is suggested to specify here any specific details which cannot easily be read from the output, such as specifics about the data and their origin, etc.

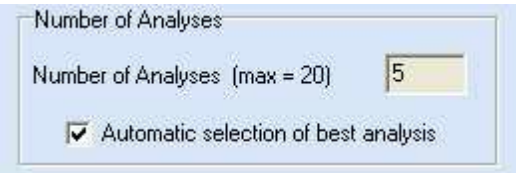
4.5.2 Number of components

Introduction
In the Parafac model as estimated by TRILIN all modes have the same number of components. By default the number of components has been set at two.

Specifications
- The specification of the number of components without selecting any other option is sufficient for an analysis, be it a trivial one.
- The appropriate number of components is a complicated issue in a Parafac analysis. In general one chooses this number to be less than the number of levels of any mode, but there are situations in which a larger number can be used.
  Occasionally a Tucker3 core array is analysed and then the number of components is larger or equal to the number of levels of one of the modes. For theoretical expositions why this may be the case one is referred to Kruskal (1976).

4.5.3 Number of repeated analyses

Introduction
By default, TRILIN will do a single Parafac analysis based on the number of components supplied.



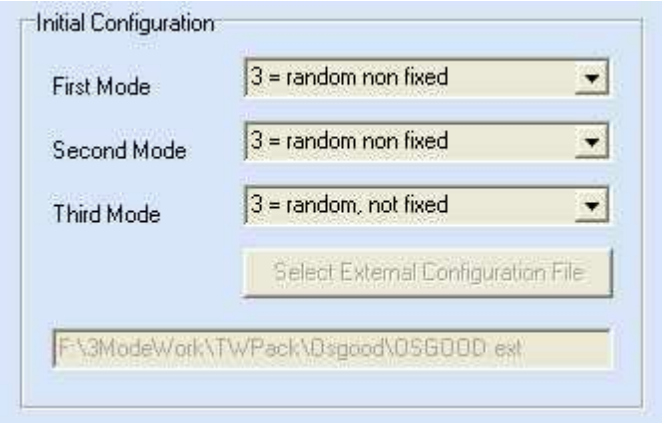
Number of analyses
Because the Parafac algorithm easily gets bogged down in suboptimal analyses, it is advisable to do a number of analyses with the same number of components to discover the existence of multiple solutions. Often five analyses will be sufficient for a good indication whether one has obtained an acceptable optimum. However, if multiple solutions occur one should consider increasing the number

of analyses and rerun the program.

Automatic selection of best analysis
By also checking the box `Automatic selection of best analysis`, the program will produce a table of all analyses and redo the best one. Detailed output will only be printed for this best analysis. Leaving this box unchecked will produce more intermediate output.

4.5.4 Initial configurations

Introduction
The iterative procedure to solve the parameter estimation of the Parafac model, `TRILIN`, needs initial configurations for the component matrices **A**, **B**, and **C**. If there is no special reason to do otherwise, it is advised to use random starting configurations (the default option). The initial configurations of the three modes have the same options but it is not necessary to use the same one for every mode. For full details see Kroonenberg (2008, Section 5.7.3).



Options
0 = Tucker, not fixed

- The initial configuration is derived from the data.
- The program computes an initial configuration for each mode by calculating ten steps towards a solution as described in Tucker (1966). This involves stacking the slices underneath each other into a tall matrix, e.g. $J \times K$ by I (a process called *matricisation*), and performing a PCA on this tall matrix. .

1 = External, fixed

- An external configuration is used and it is not modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, this configuration may be read in and kept fixed during the analysis. In other words, the rest of the solution is fitted around the external and fixed configuration.
- Comparing the fits without and with the fixed solution will allow an assessment of how well the external configuration fits the data.
- The external solution is assumed to be orthonormal (orthogonal with unit-length components). If this is not the case the program will issue a warning, but will proceed as if the configuration is correct.
- The external configuration should be in a file accessible to the program (preferably `<datasetname>.ext`) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.

- For full details see Kroonenberg (2008, Section 4.10.7).

2= External, not fixed.

- An external configuration is available and it is modified during iteration.
- If an (orthonormal) solution is available from another analysis or from the literature, you may request the program to read this configuration and use it as a starting configuration.
- The external configuration should be in a file accessible to the program (preferably <datasetname>.ext) in the data directory, and should be in free format, i.e. there should be at least one blank between numbers.

- For full details see Kroonenberg (2008, Section 4.10.7).

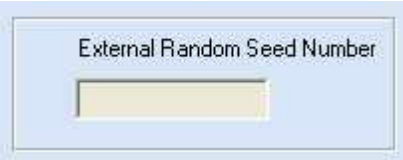
3 = Random, not fixed (default).

- A random initial (orthonormalised) configuration will be used.
- When multiple analyses have been requested new random initial configurations will be created for each analysis.
- To create different starting configurations for each run, change the random seed.

External configurations

- External configurations for different modes should be together in one external configuration file in the order of the modes.Thus in the file a first-mode configuration should always precede a second-mode one and a second-mode configuration a third-mode one.
- Once an external configuration is specified for any mode, the program will not rest until you have supplied a file name.
- The numbers in the file should be in free format, i.e. there should be at least one blank between numbers. The more decimals the better.

External random seed number



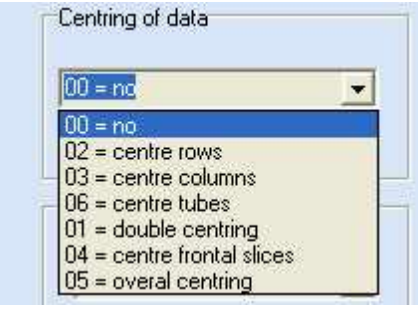
The algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. To search for different solutions for the parameters different random initial configurations can be specified via using different random seed numbers.

4.5.5 Centring

Introduction

Centring is the process of removing one or more means from the data. For three-way data there are several possibilities of which the more useful ones are included in the three-mode programs. Some additional options are available in PREPROC3.

Centring across a mode will cause the components of that mode to be centred. Note that centring fundamentally changes the data, so that it is virtually impossible to compare analyses of differently centred data. For further centring options see PREPROC3 centring.



Options

None: no centring: Default

Row centring: Fibre centring across columns per row-tube combination

- Row centring indicates that for all $I \times K$ rows the means are calculated by summing over the index j and dividing by J. These means are subtracted from the data values, so that all rows are in deviation scores.
- As variables are mostly in the second mode, this is a unusual option for multiway profile data. It is more common but still rarely used for multiway monopolar rating scales. One should never centre across bipolar rating scales due to the ambiguity of their orientation.

Column centring: Fibre centring across rows per column-tube combination

- Column centring indicates that for all $K \times J$ columns the means are calculated by summing over the index i and dividing by I, and that these means are subtracted from the data values, so that all columns are in deviation scores.
- As variables are often in the second mode, this is the recommended common option for profile data.

Tube centring: Fibre centring across tubes per row-column combination

- Tube centring means that for all $I \times J$ tubes the means are calculated by summing over the index k and dividing by K. These means are subtracted from the data values, so that all tubes are in deviation scores.
- This option is especially useful if subjects are in the third mode which occurs for multiway rating scales or multiway scaling data (individual differences scaling).

Double centring: Centre both columns and rows

- Centring across the first and second mode, also called double centring, means that for both the $K \times I$ and the $J \times K$ means are calculated by summing over the index j plus dividing by J, and summing over i plus dividing by I, respectively. These means are subtracted from the data values (and the overall mean is added in again to maintain the overall size of the data), so that all rows and all columns are in deviation scores.
- This option is particularly useful to transform (squared) distances or dissimilarities into scalar products, for instance if one wants to perform an individual differences scaling analysis.

Frontal slice centring: per frontal slice

- For each frontal slice the mean is calculated by summing over all indices (i,j) and dividing by IJ . Within each slice, this mean is subtracted from the data values.

Overall centring

- The mean is calculated over all data values, and subsequently subtracted. This option is rarely used.

Printing removed means

The program offers the opportunity to print the removed means. Inspecting them can be a very useful exercise to get a feel of the variability in those means. An analysis of such means can be executed in PREPROC3 where an analysis of variance can be performed on the means.

4.5.6 Normalisation

Introduction

Normalisation is the process of equalising the fibre or slice sums of squares. After normalisation either the fibres or slices have a total sum of squares of one.

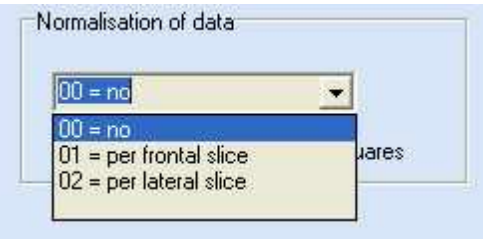
Normalisation is entirely independent of the centring, so that the removed values do not generally represent variances.

No attempt has been made to divide the sum of squares by degrees of freedom, so that the removed values are not mean squares. This means, for instance, that the total sum of squares of the normalised three-way data is equal to I (frontal-slice normalisation) or equal to J (lateral-slice normalisation).

For horizontal-slice normalisation the ways should be swapped via PREPROC3, which program can also perform the required normalisations. When both centring and normalisation are selected, centring will always be carried out first.

Centring and normalisation should not be done orthogonally as the normalisation will destroy the centring.

Normalisation is the process of equalising the sums of squares of a fibre or slice. After normalisation either the fibres or slices have a total sum of squares of one.



Options

None: no normalisation. Default

Frontal slice normalisation

- Each frontal slice is normalised such that the sum of squares of its elements is equal to one.
- This option is generally used for symmetric frontal slices in multidimensional scaling designs, or when the variables, attributes, etc. are in the third mode.

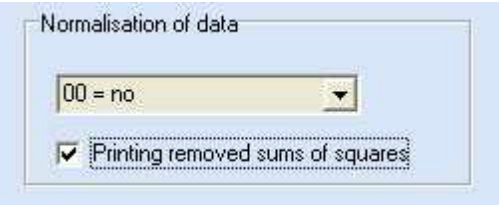
Lateral slice normalisation

- Each lateral slice is normalised such that the sum of squares of its elements is equal to one.
- This is the standard option for multiway profile data if the second modes are variables, attributes, etc.

Column fibre normalisation

- All $J \times K$ columns are normalised such that their sums of squares are equal to one.
- This option is not yet operational and is not generally recommended, but there are situations in which it might be useful.

Printing removed sums of squares



To check whether the normalisation was carried out properly the removed sums of squares may be listed.

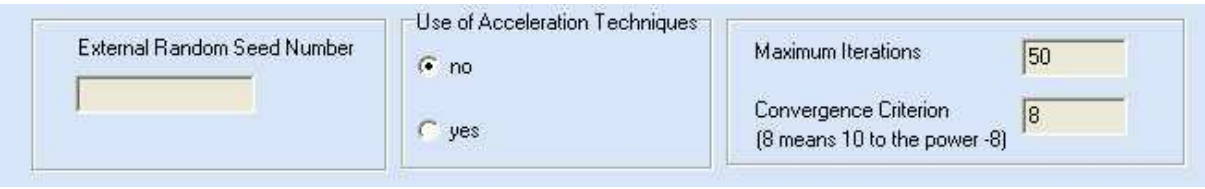
4.5.7 Iteration parameters

Introduction

The algorithm to estimate the parameters of the Parafac model is the heart of the program.

During the iterations the parameters of the model are improved step by step until no further improvement is possible according to the convergence criteria set by the user or until a preset maximum number of iterations is reached. The iterative algorithm needs a set of mostly random starting values.

In some cases the iterations can be speeded-up via a special procedure called successive relaxation.



Convergence

The default for the convergence of the discrepancy or loss function is 10^{-8} (10 to the power -8). From this value the criterion for the norm of component matrices is derived.

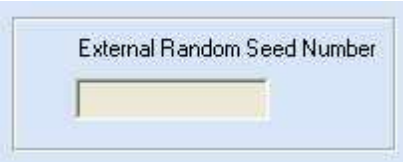
To change the convergence criterion specify the size of the exponent, e. g. 10 for 10^{-10}. On most machines the criterion should not be less than roughly 10^{-16}.

Maximum number of iterations

When the maximum number of iterations is reached, the program will finish as if it had converged and will print all relevant information.

The default is set at 100, the maximum value is 99999.

External random seed number



As mentioned above the algorithm to compute the parameters of the model needs starting values. These values are specified via the initial configurations. To search for different solutions for the parameters random initial configurations can be specified via using different random seed numbers.

This random seed specification is only necessary if a random initial configuration is requested. If the random seed is internally determined, the same random number will be used for each run of an analysis. Thus specifying the random seed allows changing the initial seed to create different initial configurations.

Use of acceleration technique

The program generally needs an acceleration technique due to its slow convergence. Therefore several facilities have been built in to speed up convergence. One of those is the use of successive overrelaxations. Due to the overrelaxation the algorithm should converge faster, but in case of suspicious behaviour, such as consistent negative differences of the discrepancy function, please rerun the analysis without the acceleration.

Not uncommonly at the beginning of the overrelaxation some negative differences in the loss function occur, but they should disappear rather rapidly. If not, one should inspect the output to see what is the problem and rerun the analysis without acceleration.

Another method used to speed up the process is limited convergence checking and reduced output during iteration.

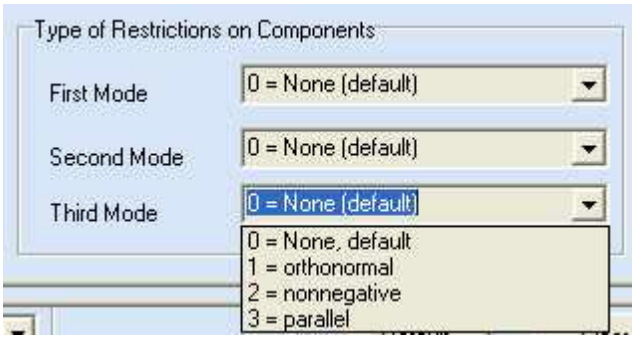
4.5.8 Type of restrictions on components

Introduction

There are four different algorithms for estimating the Parafac model in TRILIN. The standard alternating least squares algorithm and three algorithms which place restrictions on entire component matrices.

Each of these restrictions can be independently applied to any of the component matrices. A specific advantage of these restrictions is that they generally prevent degenerate solutions.

At present the program does not contain options to put restrictions on individual components or values within a component.



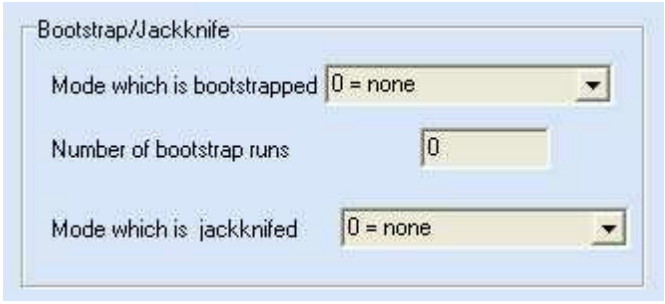
Types of algorithms

1. *Standard algorithm*. The standard alternating least square algorithm without any restrictions
2. *Orthogonality*. The components of a mode are restricted to be orthogonal. In the output the components will be orthonormal as in the default printing they are scaled to have lengths one.
3. *Nonnegativity*. The components of a mode are restricted to be nonnegative. There are many research designs which specify that components should be nonnegative, for instance wave frequencies in spectroscopy.
4. *Parallel components*. Because the Parafac model itself does not allow for identical components within a component matrix the standard algorithm breaks down when this is almost the case. By specifying parallel components in the model this problem is circumvented without loosing the uniqueness of the components (details can be found in Krijnen, 1991, Chapter 2; a pdf of this document is available from the website of The Three-Mode Company).

4.5.9 Bootstrap and jackknife analyse

Introduction

It will be possible to perform bootstrap and jackknife analyses with the TRILIN program, but these parts are still in their development stage.



Bootstrap

The bootstrap option performs the requested number of bootstrap analysis. It is always assumed that the variables or attributes are in the second mode, but the (stochastic) mode to be bootstrapped may be the first or the third mode. Be prepared for a serious wait if your problem is at all large. For full details see Kroonenberg (2008, Section 8.8.1).

Number of bootstrap runs

The number of draws with replacement from the requested mode can be specified. For each draw a complete analysis is performed, which can be time consuming.

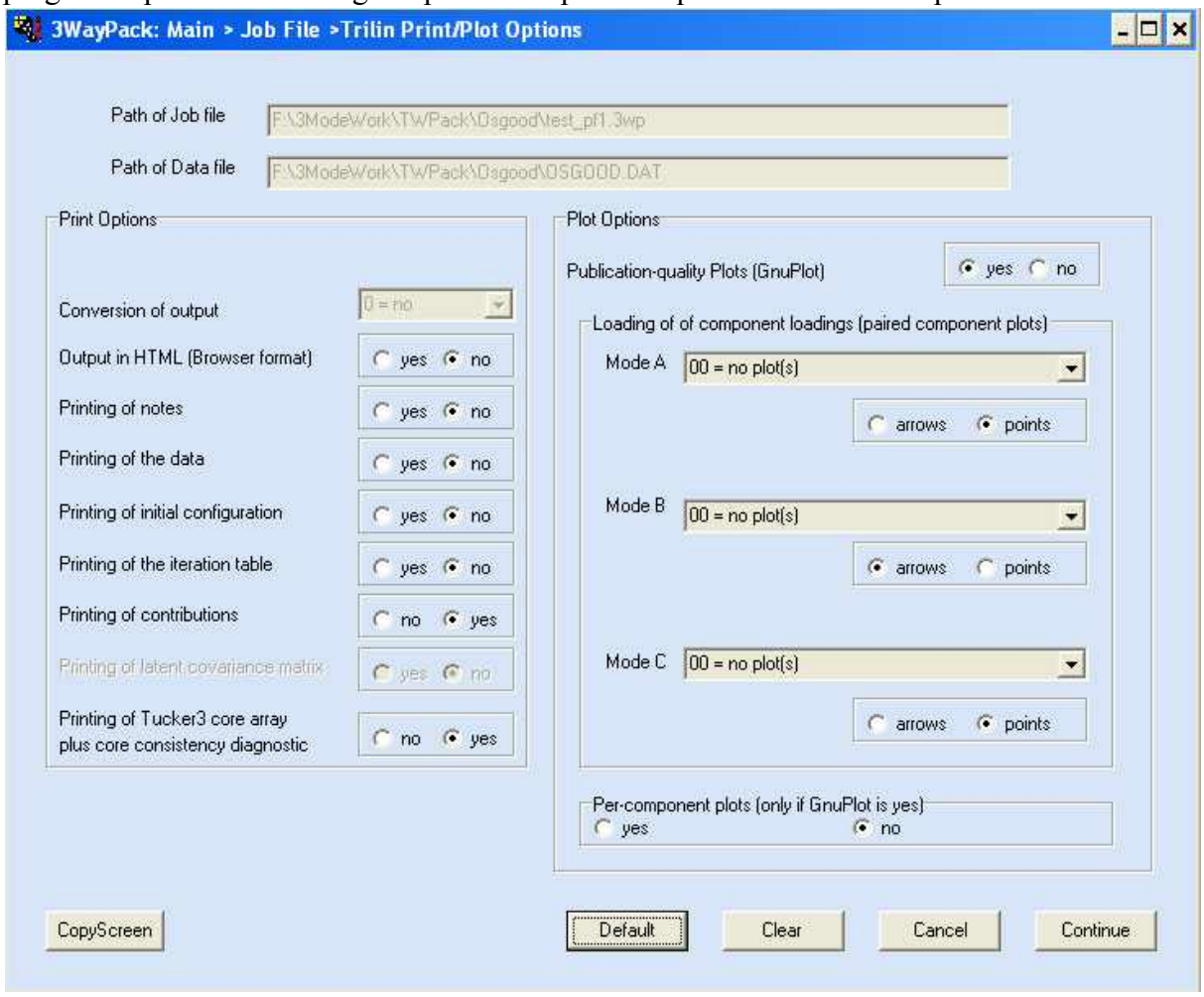
Jackknife

The jackknife option allows for a jackknife procedure on the data to establish its statistical stability. For the requested mode a random half of each slice is left out of the analysis. Afterwards its values are estimated with the model derived. Therefore 2**I* analysis for a horizontal-slice jackknife or 2**K* analyses for a frontal-slice jackknife are performed. For full details see Kroonenberg (2008, Section 8.8.2) and his references.

4.6 Trilin print/plot options

Introduction

The Trilin print/plot screen allows you to specify the kind of print and plot output you want the program to produce. Selecting all options can produce quite a voluminous output.



Print options

The standard output is always written to a file <jobname>.out which can be viewed in the internal viewer or by an external editor. There is also an option to produce browser compatible and navigable output which can include plots.

Plot options

Basic line plots, if requested, will always be printed in the standard output file, but publication-quality plots can only be obtained by choosing the GNUPLOT option. TRILIN writes special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create

button on the Main Screen the plots are created.

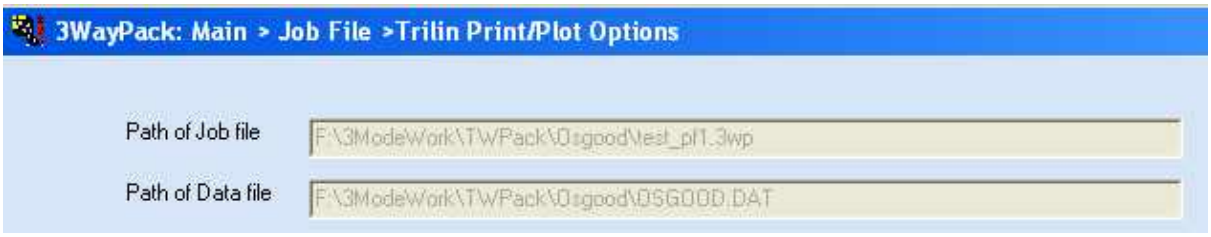
When also browser output has been requested .png-bitmap graphics but no editable graphs are created which can be displayed in the browser.

Editable graphics in so-called vector format are produced when there is no browser output. For further details see the Gnuplot section.

4.6.1 Job and data files

Introduction

The TRILIN print/plot options screen first provides information about the current job file and the data file. .



Job file

This text box indicates the name and location of the current job file. It cannot be edited in this screen. To use another job or job name please go back to the Main Screen

Data file

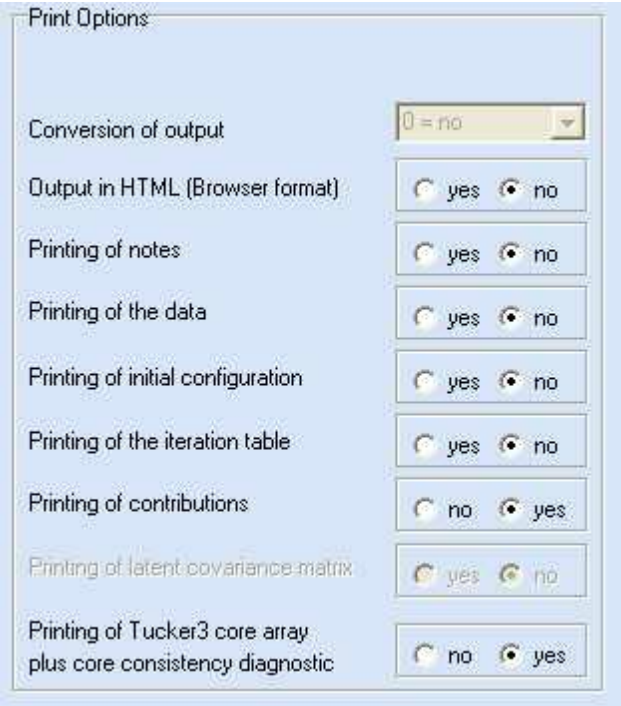
The path of the data file cannot be changed in this screen. To use a different data set, please use a different job file

4.6.2 Print options

Introduction

The program can produce a large amount of output which is regulated via the options on the TRILIN print/plot options screen. Even with the defaults, the output can be considerable if there are many levels in one of the modes. The output increases even further when more than four components are specified for any one mode due to the way the component matrices are printed. The conversion of output is disabled (see the general entry for details)

The option to produce browser output is designed to make navigation through the output easier. Some newer types of output are however not yet available in browser output.



Options

- Conversion of output (disabled but can be made available upon request)
- Output in browser format
- Printing of notes
- Printing of data
- Printing of initial configurations
- Printing of iteration table
- Printing of contributions to the fit
- Printing of core and core consistency

Navigable browser output

Introduction

When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser for the standard output.

The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.

If plots have been requested and they have been created on the Main Screen via the View/ Create button, they will display in the browser as well.

The program will write three files with HTML code for displaying in a browser:

<jobname>.htm	When displaying the output outside 3WAYPACK, this file should be double-clicked, it contains the basic frame for displaying the HTML output.
<jobnameBr>.htm	This file is the side bar containing the navigations
<jobnameBd>.htm	This file contains the actual output.

If plots are available in png-format the browser will automatically insert them into the output. Some newer types of output may not yet be available in browser format.

To print options; *To plot options*

options

Print explanatory notes

Introduction

If this option is checked some basic interpretational comments with the respect to the output are printed in the output file.

Notes for some newer types of output may not yet be available.

To print options; *To plot options*

Print complete data set

Introduction

If checked the complete data file will be printed per frontal slice. This may generate quite a bit of output.

By default the first five and the last five lines/records of the data file are printed to allow checking whether the input data.have been correctly read.

To print options; *To plot options*

Print initial configurations

Introduction

The algorithm for computing the parameters of the Tucker3 model needs starting values for the contents of the component matrices. The program will internally create such initial configurations or these configurations can be read into the program.

The default initial configuration is a random start matrix but an alternative is an approximation to an ordinary PCA on tall combination-mode matrices (e. g. $K \times I$ by J), and its results correspond to the output from the original algorithm by Tucker (1966).

The initial configuration, whether it is an internally determined or an external one, may be inspected to investigate the (provisional) Tucker solution, or to ascertain the correct reading of an external configuration.

To print options; To plot options

Print iteration table

Introduction

Depending on your theoretical interest, you may want to see how quickly the program converges. As the progress of the iteration is not only written to the output file, but is also visible on screen in a DOS box, you have something to look at while the program is running.

When the acceleration technique is used, which is the default option, please check that the algorithm is converging correctly.

When there are missing data you may notice that the function is not converging monotonically (negative signs appear), but this is no reason for concern. This is due to an inaccurate, but efficient, calculation of the loss function during iteration; eventually the negative signs will disappear. If they do not you are in trouble, possibly due to too many missing data points or inadequate starting values.

If this option is not checked, a counter appears on the screen in a DOS box, and in the output file only the final results of the iterative process will be displayed but not the details of the iterations themselves

To print options; To plot options

Print fit contributions

Introduction

The program will print by default how well each subject, variable, condition has been fitted by the Parafac model specified. In addition, the fitted and residual sums of squares are plotted in a sums-of-squares plot.- SS(Res) versus SS(Fit); see for details Kroonenberg (2008, Section 12.6).

Sometimes you may not be interested in this information or the output becomes too voluminous. In such cases the printing of these fit measures may be suppressed.

Note that when external configurations are used the contributions will not be printed, because the total sums of squares are not necessarily partitioned into the fitted sums of squares and the residual sums of squares.

To print options; To plot options

Print core array + core consistency

Introduction

The Parafac model specifies a superdiagonal core array, i.e. only the g_{sss} elements of the $S \times S \times S$ core cube are none zero. However, it is possible to calculate a Tucker3-type core array with the Parafac components Sometimes called a PFCORE procedure. From this core array it can be determined to what extent the core array is 'really' superdiagonal. A measure for the superdiagonality is the core consistency index of Bro & Kiers. Kroonenberg presented a normalised version of this index (see Kroonenberg (2008. Section 8.6.3).

To print options; To plot options

4.6.3 Plot options

Introduction
The program has three options to produce plots for components:

Types of plots
- *paired-components plots*: pairs of components from one mode are plotting against each other
- *all-components plots*.: all components of a mode are plotted against the level labels or level numbers if no labels are present.
- *per-component plots*: the same component from each mode plotted in a single plot. Thus all first components are in one plot, all second components in the next, etc.

Scaling plots
There are also three ways to scale component plots:
- *normalised coordinates*: components have length one
- *unit-mean-square coordinates*: the mean-squared lengths of components have length one
- *principal coordinates*: components have lengths equal to the square roots of their eigenvalues.

Plots are included in the *standard output* but also *publication-quality plots* via the plotting program GNUPLOT can be produced.

Minimum spanning trees
The minimum-spanning tree can be calculated to determine whether points in the component space lie close together. At present only the information to draw these minimum-spanning trees are provided but they are not yet plotted by the program itself.

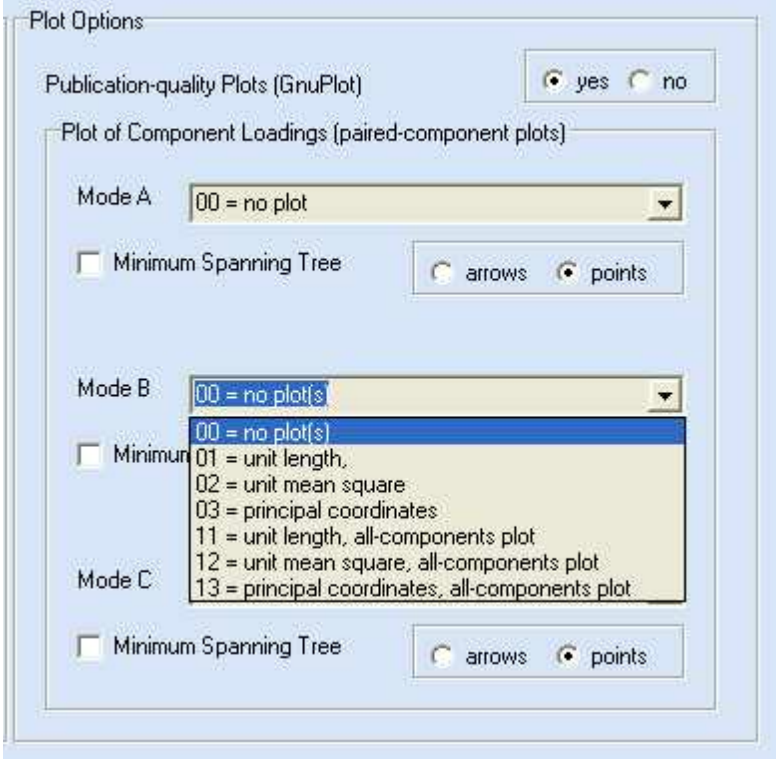
Arrows or points
One also has a choice to plot the levels of the mode either as points or as arrows emanating from the origin.

Component plots

Introduction
TRILIN has various options to produce plots for components: paired-components plots, all-components plots, and per-component plots. The components are printed using three different scalings but only one of them can be plotted in a single run. When S is less than or equal to 4, all components will be plotted against one another. Components 5 and higher will only be plotted against the first component.

GnuPlot
Publication-quality plots can be specified via the GnuPlot option. (see the Utilities section for more details on GNUPLOT).

Go to sections Options; Arrows or points; Printing



Options

- *None*: no component plots; default.

- *Standard coordinates*: Unit lengths: lengths components = 1
  The lengths of all components are equal to one. Thus, the scatter in the plots is not adjusted to the explained variability (eigenvalues). This differs from the situation for the component loadings (column objects) in an ordinary two-mode PCA, but it is the standard way for component scores of the subjects (row objects).

- *Unit-mean-square coordinates*: lengths components = number of levels
  The lengths of all components are equal to the number of levels in a mode. The outward appearance of the plots is identical to that of the unit-length scaling above, due to the automatic adjustment of the scales of the plots. This scaling has the advantage that the values across modes are not influenced by different numbers of levels in the modes.

- *Principal coordinates*: lengths = square root of eigenvalues
  The lengths of the components are equal to the square root of standardised component weights. If the total sum of squares of the data is equal to the number of levels of a mode, then this scaling corresponds to the situation in regular two-mode PCA with standardised variables. However, the values are not necessarily comparable with regular 'loadings' (variable-component correlations), because this will depend on the centring and normalisation of the original variables. The effect of this scaling is that the scatter in the plot will depend on the differences in variability accounted for by the components.
  Principal coordinates have the advantage that distances between the levels in the space are correctly represented.

Arrows or points
For each of the modes the user can specify whether the levels of the modes should be represented by arrows or by points. Generally arrows are chosen if the components are in principal coordinates and points are selected in the other two cases, but there is no obligation to do so.

All-components plots
Each of the above options can be combined with an all-components plot. This can be done by choosing the same options but the value of the options is increased with 10.

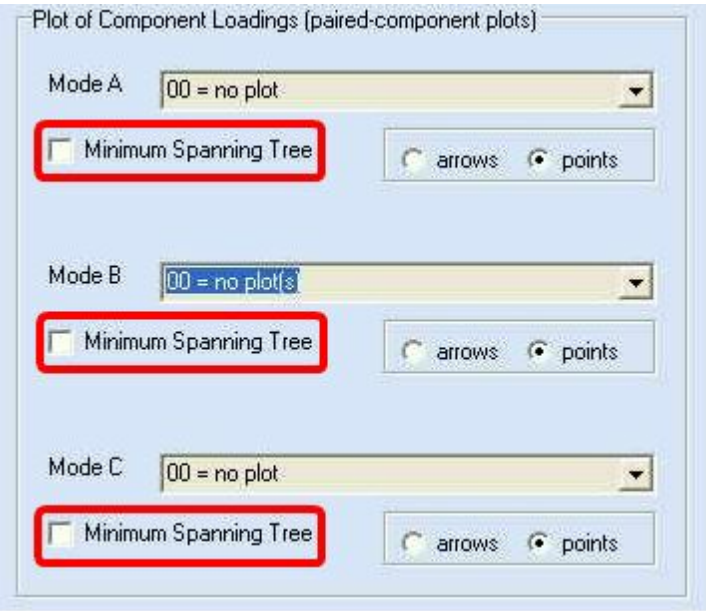
Per-component plot

Introduction
The Parafac model is component oriented rather than component space oriented. The first components of the modes are exclusively linked to each other, so it is useful to display them together in a single plot. Such plots are called per-component plots and also line plots, but the latter term is also used in other senses; for details see Kroonenberg (2008, Section 11.5.1).

Minimum-spanning tree

Introduction
The minimum-spanning tree is a path diagram which connects the points in a multidimensional space such that the sum of all distances between the points is as short as possible. Effectively this means that each point is connected with its nearest neighbour. With this information it is possible to assess in a lower dimensional subspace whether two points are close together in the higher-dimensional full space.

Thus one can use the minimum spanning tree to evaluate the closeness of two points in the two-dimensional plots which are produced by the program while the fitted component space is three-dimensional or more. This option is thus only useful for three-dimensional and higher dimensional spaces.

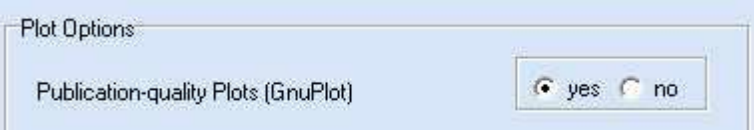


The program prints which points are connected at which distance based on the component space. At present there are not yet plots to display the minimum spanning tree in the component plots produced by the program.

Publication-quality plots

Introduction

When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>.plt. By clicking the View/Create button on the Main Screen the plots are created. See the Utilities section for more details on GNUPLOT.



When browser output has been selected only png-bitmap graphics are created which can be displayed in the browser. Editable graphics in so-called vector format are only produced when there is no browser output.

Please associate in Windows the extension .plt with the GNUPLOT program so that double clicking on the .plt file in the data directory the plots can be created outside 3WAYPACK.

For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

4.6.4 Type of CPC file

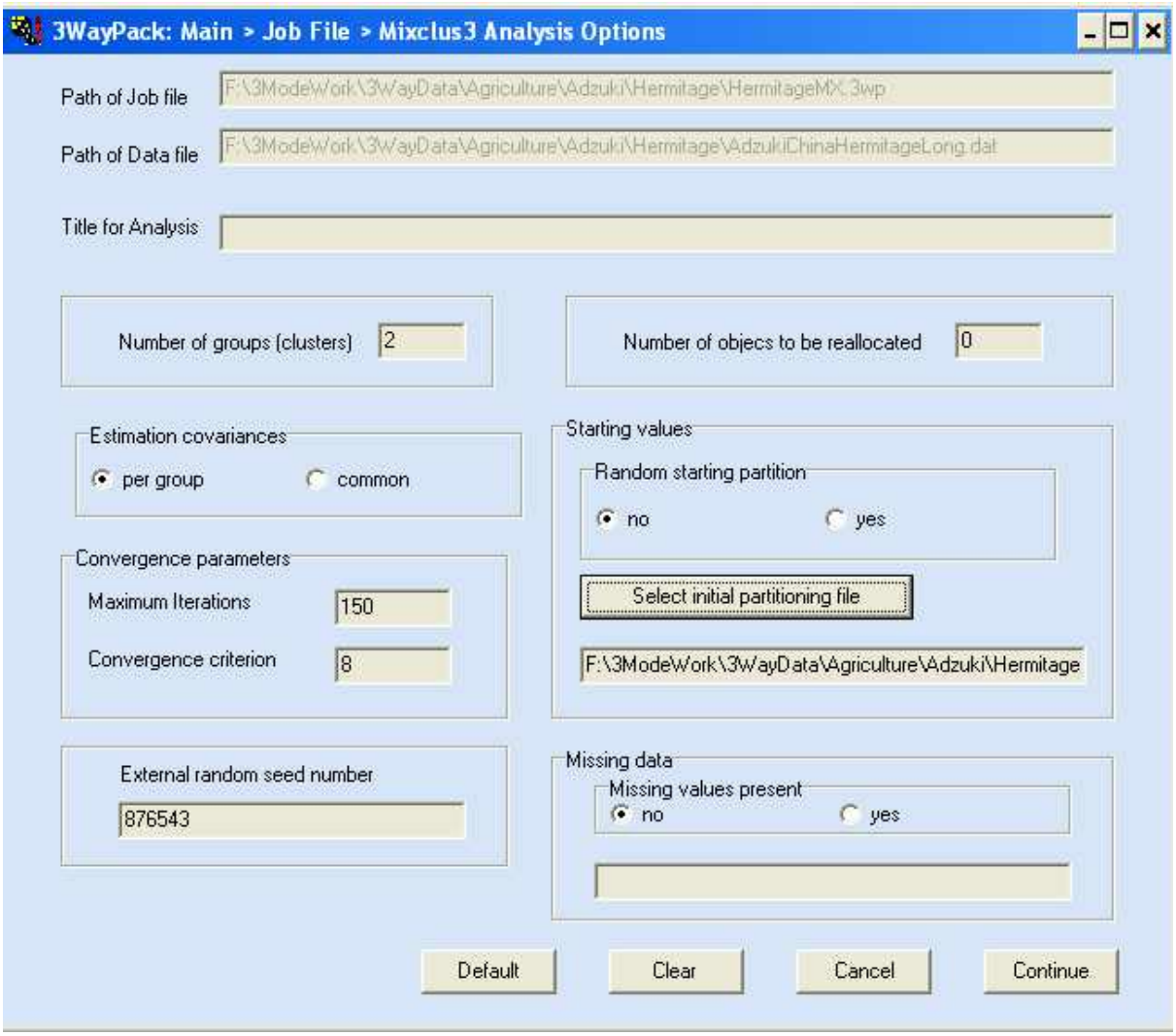
Introduction

The analysis programs write a CPC-file, which contains the coordinates of the components and the raw core array as well as the total and fitted sum of squares. This file stores this information for use in the postprocessing programs. By default the normalised coordinates should be used. For very specific core rotations the principal coordinates are required.

4.7 Mixclus3 analysis options

Introduction

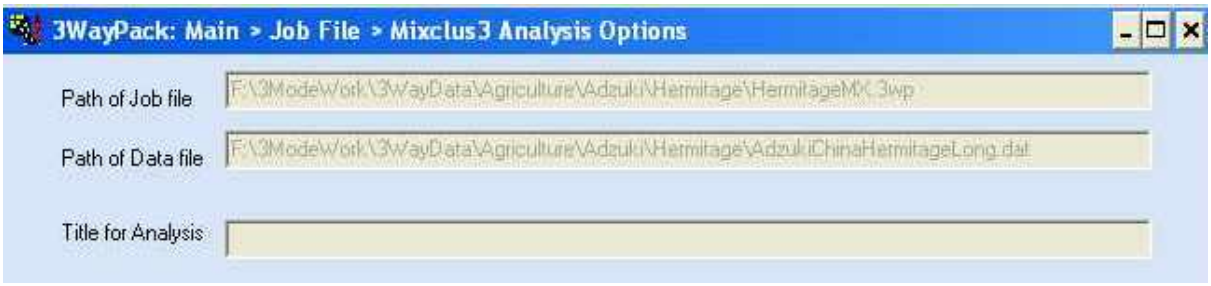
Mixclus3 is the program for mixture method of clustering for three-mode data. It was originally devised and programmed by Kaye E. Basford of the University of Queensland. The present program is both a derivative and an extension of her program.



4.7.1 Job, data, and title

Introduction

The Mixclus3 Analysis Options screen first provides information about the current job file, the data file and gives the opportunity to specify a title for labelling the output.



Job file

This text box indicates the name and location of the current job file. It cannot be edited in this screen. If you want another job please go back to the Main Screen

Data file

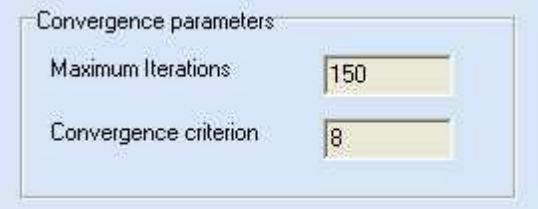
The path of the data file cannot be changed in this screen. To use a different data set, please use a

different job file.

Title for the analysis
The title specified in this text box will be used to label the output. It is suggested to specify here any specific details which cannot easily be read from the output, such as specifics about the data and their origin, etc.

4.7.2 Convergence parameters

Introduction
The convergence of MIXCLUS3 is regulated by two parameters: The number of iterations (Maximum Iterations) and the size of successive differences between the loss function during the iterations (Convergence criterion).



Maximum number of iterations
The default number of iterations is 150, but large problems may need more. The format for this parameter allows for 99999 as an absolute maximum number.

Convergence criterion
The size of the criterion is indicated by the negative exponent of 10. The default value of 8 is 10^{-8} = 0.00000001. Larger values than 12 or may be 16 do not make sense as this is more accurate than the machine precision.

4.7.3 Number of groups

Introduction
The desired number of groups or clusters need to be specified beforehand. To determine the "proper" number of clusters the user needs to run the program several times to find the solutions for each of these numbers. At present no automated procedure is in place to compare the solutions from several solutions in a single run.

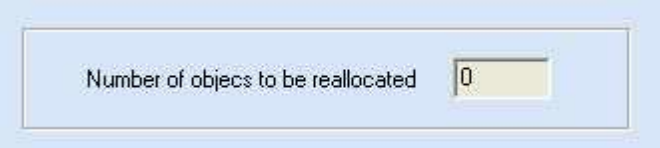
Note that fitting the model with separate covariance matrices for the groups using increasingly larger numbers of groups is more difficult than doing so for a common covariance matrix. In the former case the number of parameters increases rather rapidly leading to unstable solutions.

4.7.4 Number of reallocations

Introduction
One of the problems in mixture method cluster analysis is that the algorithm for calculating the algorithm may get stranded in a local maximum. Moreover it is not always clear how well-determined a solution is. To investigate both these issues one can request that objects are reallocated

to their next closest group. The analysis is then rerun and the user can compare the two or more results to investigate whether there is a better solution, whether there are multiple solutions, and whether the solution is fairly stable. Depending on the results one can sometimes identify the objects which cause the instability.

At a later date we might automate this comparison process.



4.7.5 Type of covariance matrices

Introduction

The mixture method of clustering in the program suite has two variants:

1. each group/cluster has its own covariance matrix with the (co)variances among the variables
2. all groups have a single common covariance matrix.

As the first option has many more parameters it has the potential for fitting better but at the same time it uses up many more degrees-of-freedom aversely affecting the significance test and sometimes the stability of the solution (for further details see Kroonenberg (2008, Section 16.3.1).

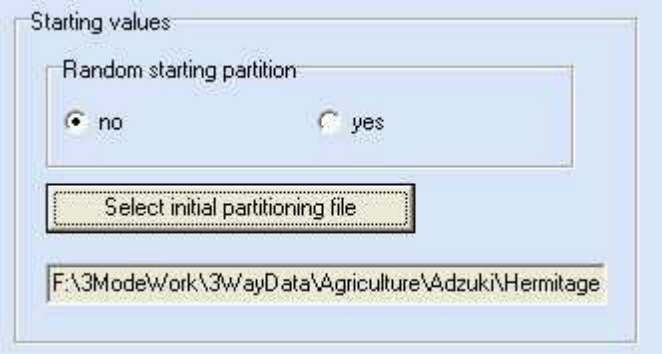
In case of doubt the advice is to try both options and compare the relative fits of the models.

4.7.6 Starting values

Introduction

The algorithm of the mixture method of clustering needs an initial partitioning of the objects into the number of groups specified. Most commonly rational starting values perform better than random starting values. A good option is to carry out a two-mode cluster analysis via Ward's method on one or more slices and use the result to specify the starting solution. Alternatively one can run a two-mode mixture method clustering separately for each replication with the present program and obtain a series starting solutions in that way.

The starting partitioning should be stored in an external file in free format. Be sure that the number of different values for the starting partitioning is the same as the number of groups you specify. If not, the analysis program, i.e. MIXCLUS3 itself, will complain. Note that the Interface itself is not able to detect the discrepancy.

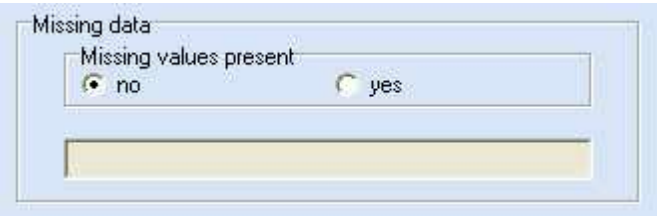


4.7.7 Missing values

Introduction

In contrast with the component analysis programs (TUCKALS2, TUCKALS3 and TRILIN) the missing values are exclusively handled by MIXCLUS3 itself. However, if desired, one may fill in the missing values using the program PREPROC3 but these values will not be reestimated in MIXCLUS3. In other words the program will treat them as ordinary observed values.

MIXCLUS3 accepts only a *single* missing-value code.

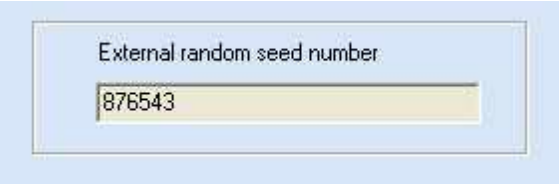


4.7.8 External random seed

Introduction

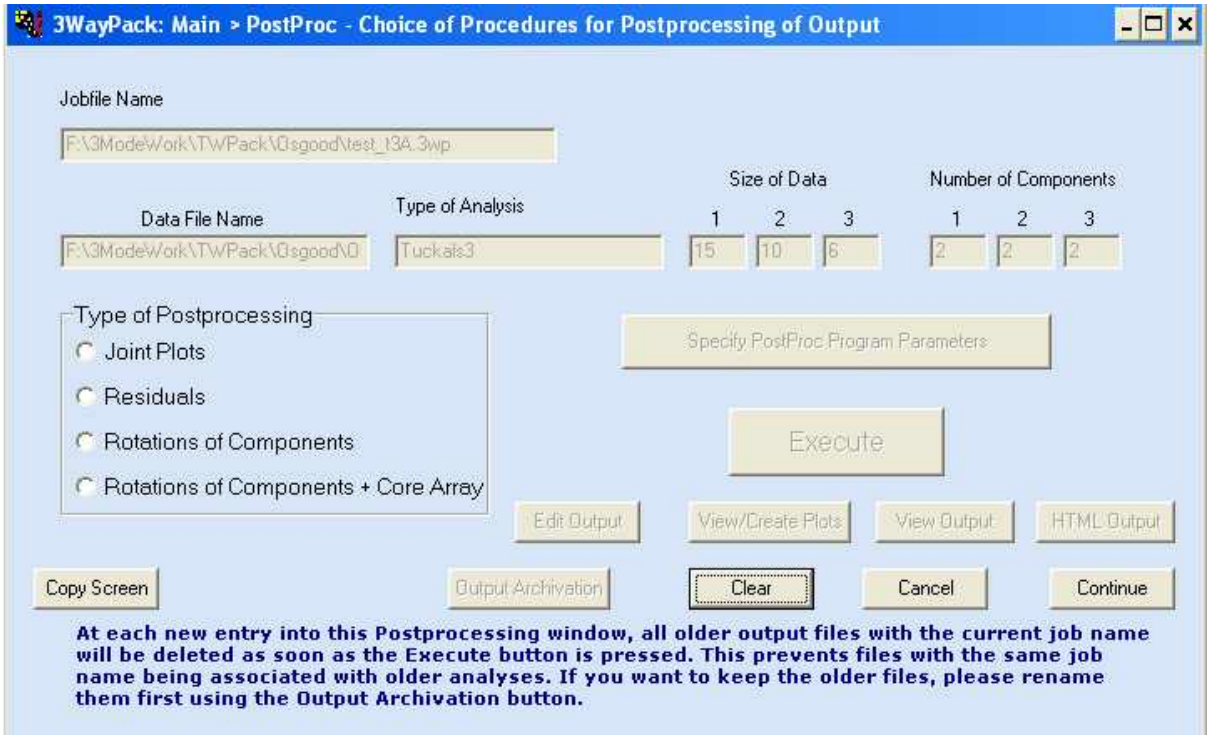
This option is only useful if a random starting configuration is requested. If the random seed is internally determined, the same random number will be used for each run of an analysis. This option allows changing the initial seed to create different starting configurations.

Please specify only positive numbers without a plus or decimal sign.



5 Postprocessing

Introduction
In order to enhance and facilitate interpretation it is desirable that the basic output of the analysis programs is further enhanced by rotation of components and/or core arrays, by plotting the outcomes, examining residuals etc. These activities are collectively known as *postprocessing*. To be able to perform postprocessing the component matrices and the core array as well as the total sum of squares need to be available to the postprocessing programs. This information is collected in the CPC file which has the internally determined name of <jobname>.cpc by the three-way analysis programs. This file contains the basic information on which the postprocessing operations are carried out by the four postprocessing programs T3ROTATE, ROCOCO, RESIDUAL and JOINTPLT.



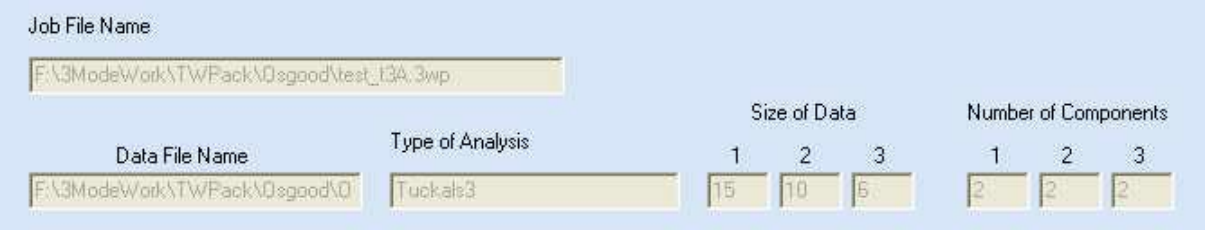
Postprocessing programs
The major programs which operate on the output are

- JOINTPLT - the construction of joint biplots of two modes given a component of the third mode and nested-mode biplots.
- RESIDUAL - the calculation of residuals to assess the element-wise fit of the model with respect to the data. Model-based estimates for the data values are compared with the observed values.
- ROCOCO - varimax rotation the components and core array jointly in order to get optimal simplicity in all matrices or in those especially selected for rotation.
- T3ROTATE - rotating components with counter-rotations of the core array. Options: varimax, oblimin and constant first component.

5.1 Information from analysis programs

Introduction
Obviously the information about the analysis program for which the postprocessing is requested cannot be changed and they are presented on the Postprocessing screen for reference purposes

only.

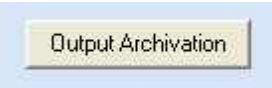


The postprocessing facilities have been constructed in such a way that the information about the postprocessing carried out are stored in the job file so that the analysis can be repeated. However, postprocessing can only be carried within the job at hand. It is thus not possible to postprocess output without having the appropriate job active. Archiving the output of the main analysis has no influence on the availability of postprocessing as the name of file containing the relevant information (<jobname>.cpc-file) is not changed.

5.2 Archiving/Renaming of output

Introduction

Output files in 3WAYPACK are named on the basis of the name of the job file, e.g. <jobfile>.out. Therefore each time an analysis is run with the same job file all existing output files with the same job name are overwritten. The Output Archivation option allows renaming the following output files.



Output file names overwritten in 3WayPack in the Postprocessing screen (if the relevant program has just been run)

<jobname>.oro	the basic output file of T3ROTATE
<jobname>.orc	the basic output file of ROCOCO
<jobname>.ore	the basic output file of RESIDUAL
<jobname>.ojp	the basic output file of JOINTPLT
<jobname>XX.plt	the plot command file
<jobname>XX.htm	the general browser command file
<jobname>XXBr.htm	the left-hand navigation browser file
<jobname>XXBd.htm	the right-hand content browser file

where XX = JP, RE, RC, RO for JOINTPLT, RESIDUAL, ROCOCO and T3ROTATE, respectively.

Note: At present the naming for T3ROTATE and ROCOCO are identical using RO, rather than RO and RC, respectively. This will be corrected as soon as possible.

5.3 Joint biplots

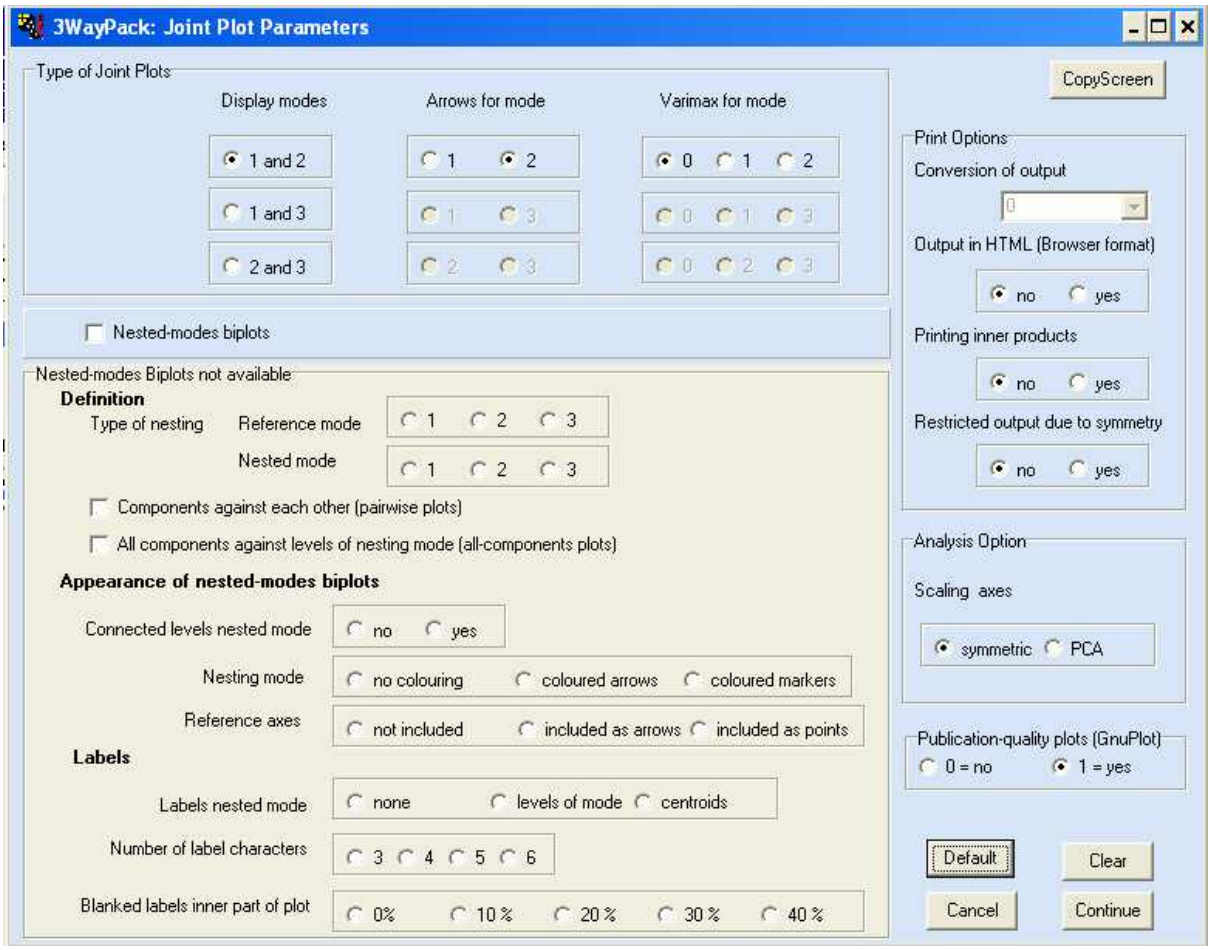
Introduction

JOINTPLT constructs joint biplots for two modes (the *display modes*) given the third mode (the *reference mode*). In addition, nested-mode biplots can be constructed which plots all first-third combination components in the space of the second mode. For some data sets the coordinates of such plots can be considered component scores.

Standard output is written to <jobname>.ojp, commands for the plotting program to

<jobname>JP.plt. For names of browser output files see Print options.

No joint biplots are available for TRILIN as they is incompatible with the Parafac model fitted by that program. Please use per-component plots instead.



Program options

Types of joint biplots.

JOINTPLT allows the choice of one type of joint biplot, i.e. biplot of two of the modes against each other. To obtain one of the two other types, the program should be run again with a new specification. In order to keep each type of joint biplot, be sure to archive the earlier output before the next run of the program. If the analysis program is TUCKALS2 or only one matrix is being analysed (K=1), only the first type of joint biplot is possible.

Joint biplots can be customised by plotting a varimax-rotated version of one of the display modes and it can also be specified which of the display modes should be indicated by vectors and which by points. For interpretation purposes variables should generally be displayed as vectors.

Note that often the later axes of the joint biplots have very small amounts of explained variability and they therefore can be safely ignored. Some subjective judgement may be required for this.

Numbers of component unequal

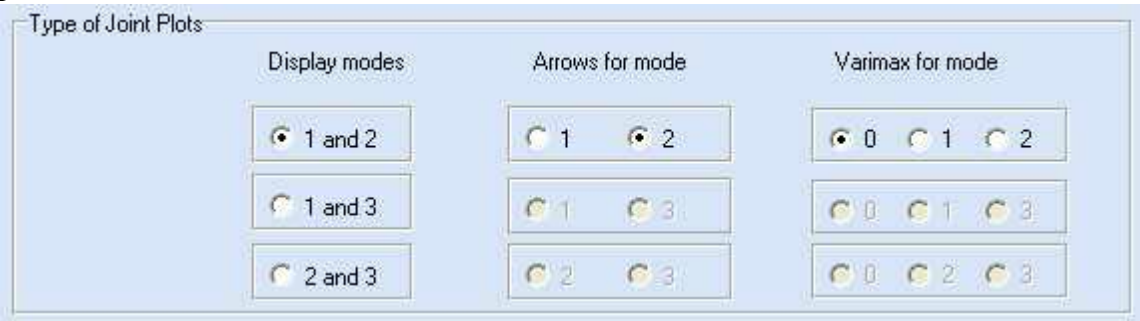
In addition note that if the number of components for the display modes are unequal (say, $P < Q$) then the joint biplot has at most P axes.

Print options.
- The conversion of the output to pure ASCII or only capitals has been disabled on purpose.
- Output in browser format
- Inner products of the vectors from the two display modes
- Restrict printing of the component matrices if two modes are the same, for instance in the case of symmetric input matrices.

Nested-mode biplots
The option of nested-mode biplots is at present only available for the Tucker3 model. If the reference mode is, say, the third mode and the nested-modes are the first mode and the third mode, a nested-mode biplot displays the combined $I \times K$ levels of Mode A and Mode C in the space of the reference mode B. Detailed explanations can be found in Kroonenberg (2008, pp. 276ff).

5.3.1 Types of joint biplots

Introduction
Joint biplots can be made for the two display modes given the third reference mode. You can only do one type of joint biplot at a time. You may indicate which of the two display modes will have arrows and which has points displayed. One of the two display modes may be rotated by a varimax procedure.



Types of joint biplots.
JOINTPLT allows the choice of one type of joint biplot, i.e. biplot of two of the modes against each other. To obtain one of the two other types, the program should be run again with a new specification. In order to keep each type of joint biplot, be sure to archive the earlier output before the next run of the program. If the analysis program is TUCKALS2 or only one matrix is being analysed (K=1), only the first type of joint biplot is possible.

Arrows & Points - Varimax
- One of the display modes will be displayed by vectors and the other by points. For interpretation purposes variables should generally be displayed as vectors.
- The interpretation of some more dimensional joint biplots can enhanced by plotting a varimax-rotated version of one of the display modes.

Inspection the explained variability of axes
Note that often the later axes of the joint biplots have very small amounts of explained variability and they therefore can be safely ignored. Some subjective judgement may be required for this.

Number of axes displayed
In addition note that if the number of components for the display modes are unequal (say, $P < Q$)

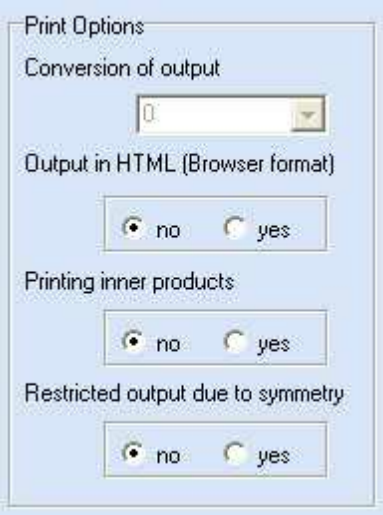
then the joint biplot has at most *P* dimensions.

5.3.2 Print options

Introduction

JOINTPLT has a number of special printing options. The conversion of output is disabled (see the general entry for details).

Standard output is written to <jobname>.ojp, commands for the plotting program to <jobname>JP.plt. For names of browser output files see below.



Options

- *Browser output*.
  When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser.
  Navigation. The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.
  Plots. If plots have been requested and they have been created on the Postprocessing screen via the View/Create button, they will be displayed in the browser as well. The plots will show up in the data directory in png-format.
- *Browser files*. The program will write three files with HTML code:

<jobname>JP.htm The browser output can be displayed outside 3WAYPACK by double-clicking this file.
<jobnameBr>JP.htm This file is the side bar containing the navigation
<jobnameBd>JP.htm This file contains the actual output

Some newer types of output may not yet be available in browser format.

- *Inner products*. The inner product of a row marker and a column marker indicates the strength of the association of the two markers. Its size is the product of the length of the projection of a row marker on the vector of the column marker and the length of the column marker. A zero value indicates no relation (projection into the origin), a positive (negative) value a positive (negative) association.

  Especially in joint plots with more than two dimensions, inner products can be useful to check the importance of the projection of a row marker on the vector of the column marker and

vice versa.

The inner products are also the coordinates for nested-mode biplots (or the component scores on the components of the reference mode).

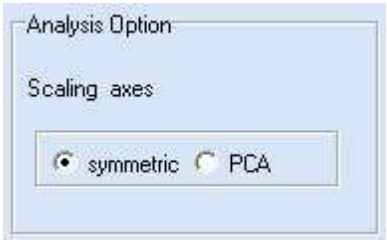
- *Symmetry*. If the input data are symmetric frontal slices, the components for the first two modes will be identical. To reduce the output one may request that the program skips printing these identical component matrices.

5.3.3 Analysis option

Introduction

There are two ways to scale the axes of a joint biplot.

- *Symmetric option*: The axes are scaled in such a way that both display modes fill the space equally well.
- *PCA option (principal coordinates)*: The first mode has normalised coordinates and the other mode has principal coordinates. In this case one of the modes may be clustered around the origin while the other is on the perimeter of the plot.

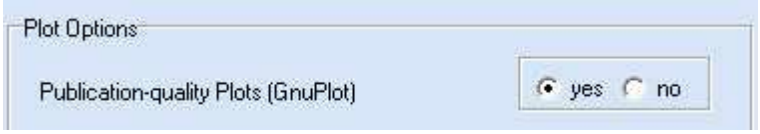


The scaling does not affect the interpretation of the inner products in the plots. However, the principal coordinate option will give a better distance representation of the second mode (in principal coordinates), while the distances in the first mode (in normalised coordinates) are not properly represented in the plot. Therefore in the latter case the elements of the first mode can only be interpreted via their projections on the second-mode vectors or biplot axes through the origin and the second-mode points.

5.3.4 Publication-quality plots

Introduction

When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>JP.plt. By clicking the View/Create button on the Postprocessing Screen the plots are created. See the Utilities section for more details on GNUPLOT.



When browser output has been selected only png-bitmap graphics are created which can be displayed in the browser.Editable graphics in so-called vector format are only produced when there is no browser output.

Please associate in Windows the extension .plt with the GNUPLOT program so that double

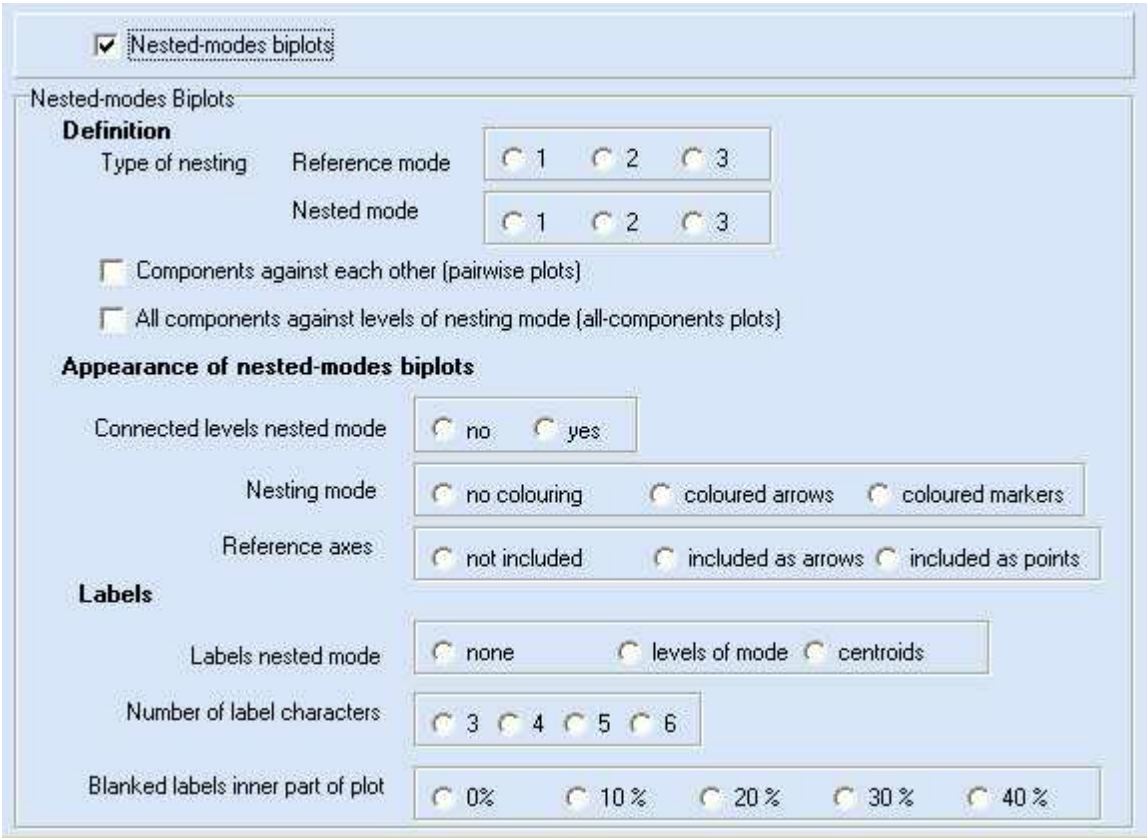
clicking on the .plt file in the data directory the plots can be created outside 3WAYPACK.

For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

5.3.5 Nested-mode biplots

Introduction

The option of nested-mode biplots is at present only available for the Tucker3 model. If the reference mode is, say, the third mode and the nested modes is the first mode and the third mode the nesting one, a nested-mode biplot displays the combined $I \times K$ levels of Mode A and Mode C in the space of the reference mode B. Detailed explanations can be found in Kroonenberg (2008, pp. 276ff).



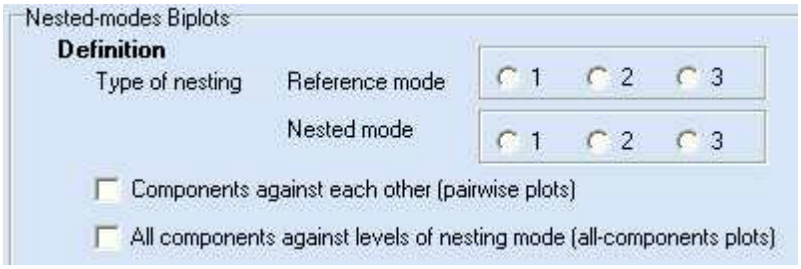
Definition

Introduction

To specify a *nested-mode biplot* it is necessary to indicate which mode is the *reference mode* and which mode is the *nested mode*. By implication the remaining mode is the *nesting mode*.

One can choose to plot:

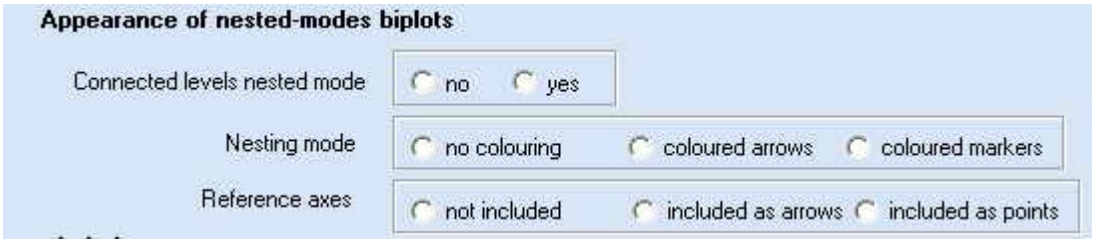
- *paired-components plo*ts: the components are pair-wise plotted against each other.
- *all-components plots*: all components of a mode are plotted against the level labels or level numbers if no labels are present..



Appearance

Introduction

A *nested-mode biplot* is based on the matrix which results from crossing the levels of two modes and projecting them into the space of the remaining mode (the reference mode). This way a single plot can be created in which all three modes are represented.



Trajectories

Assuming there is an ordinal relationship between the levels for one display mode so that this can be treated as the *nested mode*, trajectories can be drawn for each level of the other display mode (the *nesting mode*) in the space of the remaining mode (the *reference mode*).

Nesting mode

Different colouring of the points or arrows is possible to enhance interpretability.

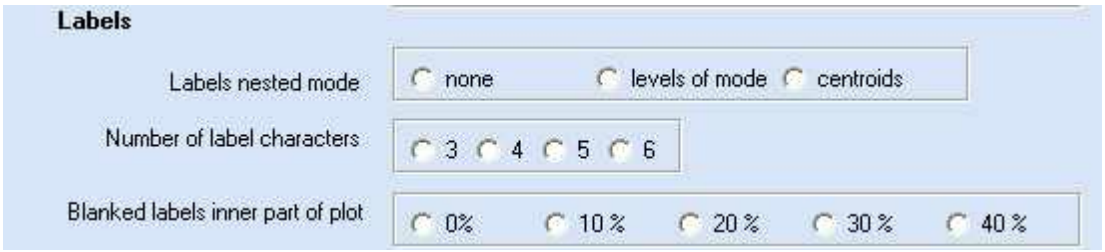
Reference axes

Finally the reference axes can also be plotted in the same graph. One may choose to include them as arrows emanating from the origin or as coloured markers.

Labels

Introduction

Nested-mode biplots can become difficult to read due to the large number of points and labels. Therefore the options exist to limit the clutter on the plots by limiting the number characters each label may occupy and by blanking labels in the central part of the plot. In addition the number of labels can be limited by only labelling the centroids rather than the individual points (= levels of the modes)

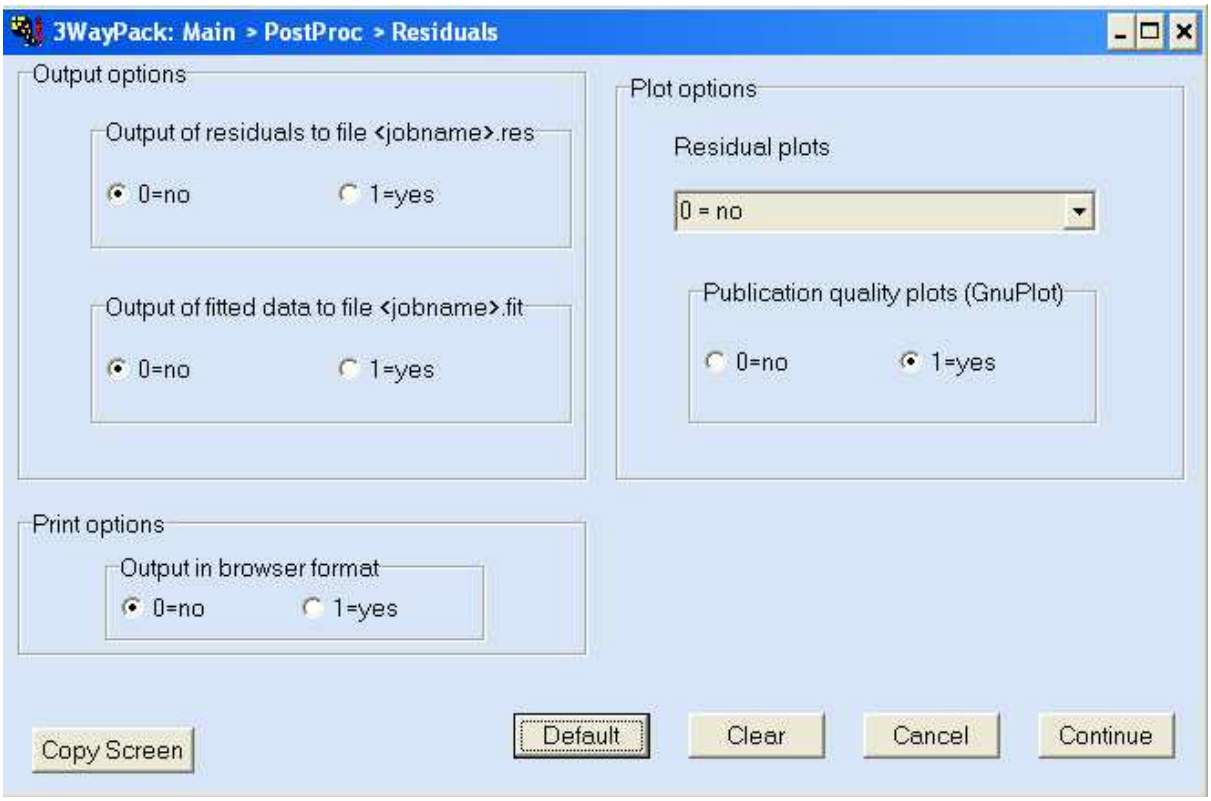


5.4 Residuals

Introduction

The postprocessing program RESIDUAL can calculate standardised residuals and plot these against the standardised fitted data. The residuals can be plotting per frontal slice or for the entire data set. This program also offers the option to write either the residuals and/or the fitted data to files called <jobname>.res and <jobname>.fit respectively. Note that the fitted data are such that they fit the model perfectly, and thus can be used for simulations etc.

Standard output is written to <jobname>.ore, commands for the plotting program to <jobname>RE.plt. For names of browser output files see Printer options.



Options

- *Output:* Residuals and/or fitted data can be written to a file
- *Printing*: Browser format or not
- *Plotting:*

  Two types of plots can be specified: a *global residual plot* or *frontal-slice residual plots*; Publication-quality plots or not

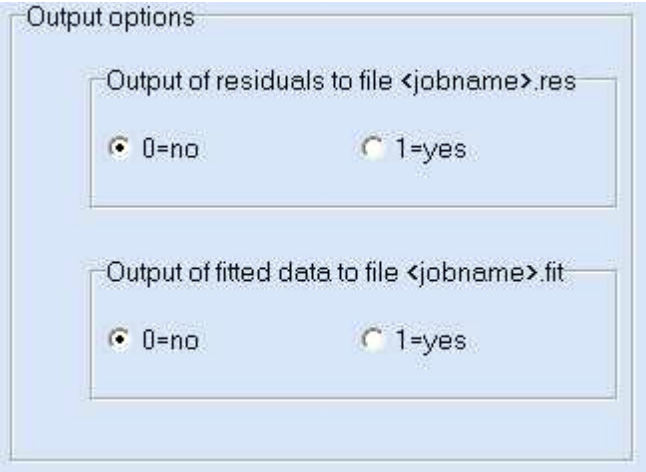
5.4.1 Output options

Introduction

RESIDUAL can write two types of information to a file.

- *Residuals*: Raw residuals may be written to file <jobname>. res, so that they can be processed by other programs. Using other programs the extent of the normality of the residuals can be assessed as well as outliers, and similar characteristics.
- *Fitted data*: The fitted data may be written to a file <jobname>. fit, so that they can be

processed by other programs. The fitted or implied data follow the three-way model perfectly. Therefore, they may be used for research purposes. In particular, the fitted data can be used as a basis for simulation studies for which perfectly fitted data are required.

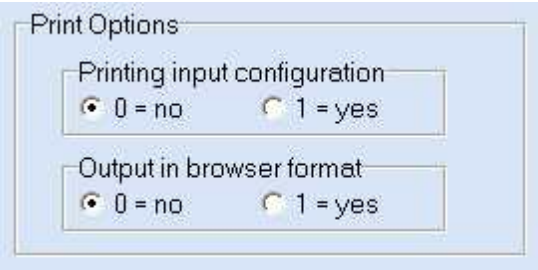


5.4.2 Print options

Introduction

There are two types of print options available: (1) printing the initial configurations, (2) generating output files in browser format.

Standard output is written to <jobname>.ore, commands for the plotting program to <jobname>RE.plt. For names of browser output files see below.



Print options: Details

If no plots have been requested only the extreme values and a stem-and-leaf plot of the residuals will be listed.

- *Browser output*.
  When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser.
  Navigation. The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.
  Plots. If plots have been requested and they have been created on the Postprocessing screen via the View/Create button, they will be displayed in the browser as well. The plots will show up in the data directory in png-format.
- *Browser files*. The program will write three files with HTML code:

<jobname>RE.htm The browser output can be displayed outside 3WAYPACK by double-clicking this file.

<jobnameBr>RE.htm This file is the side bar containing the navigation

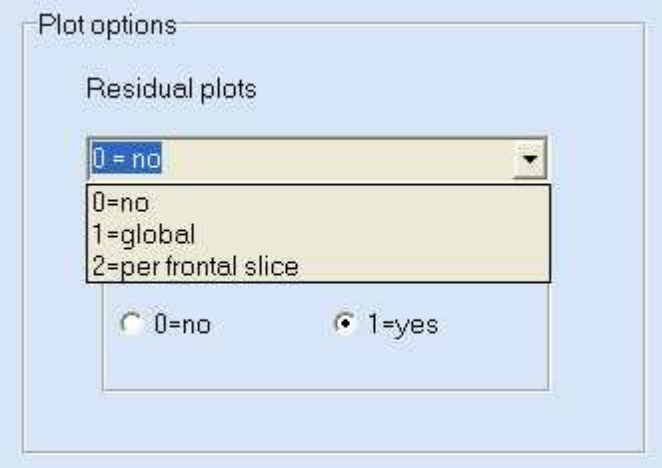
<jobnameBd>RE.htm This file contains the actual output

Some newer types of output may not yet be available in browser format.

5.4.3 Plot options

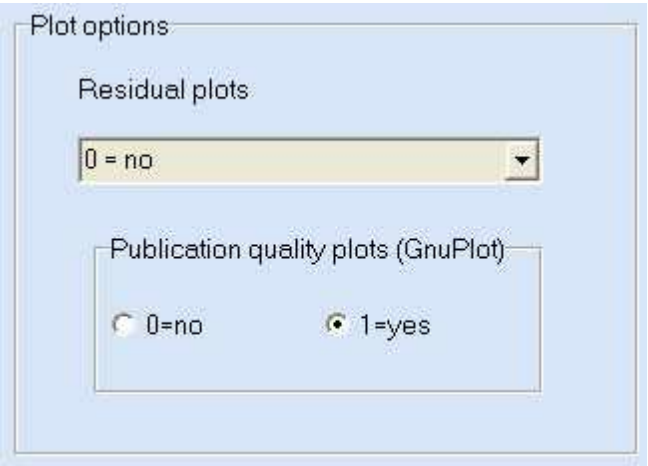
Introduction

RESIDUAL can produce two types of plots: A *global residual plot* or *frontal-slice residual plots*



Options

- *Global residual plots.* The residuals are calculated for the entire data set, and the standardized residuals are calculated on the basis of all residuals. One single plot will show the standardized residuals against the raw fitted data. A list of extreme residuals will be printed.
- *Frontal-slice residual plots.* The residuals are calculated and plotted for each *frontal slice* separately.
  If you want residuals per *lateral slice* or per *horizontal slice*, the only way to do this within 3WAYPACK is to let PREPROC3 swap the data array in an appropriate way. The standardized residuals are calculated for the data of each slice. A list of extreme residuals will be printed for each slice.



When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code

for the plotting program GNUPLOT and writes this to the file <jobname>RE.plt. By clicking the View/Create button on the Postprocessing Screen the plots are created. See the Utilities section for more details on GNUPLOT.

When browser output has been selected only png-bitmap graphics are created for inclusion in the HTML output. Editable graphics in so-called vector format are only produced when no browser output has be requested.

If you associate in Windows the extension .plt with the GNUPLOT program you may double click on the .plt file in the data directory to create the plots outside 3WAYPACK.

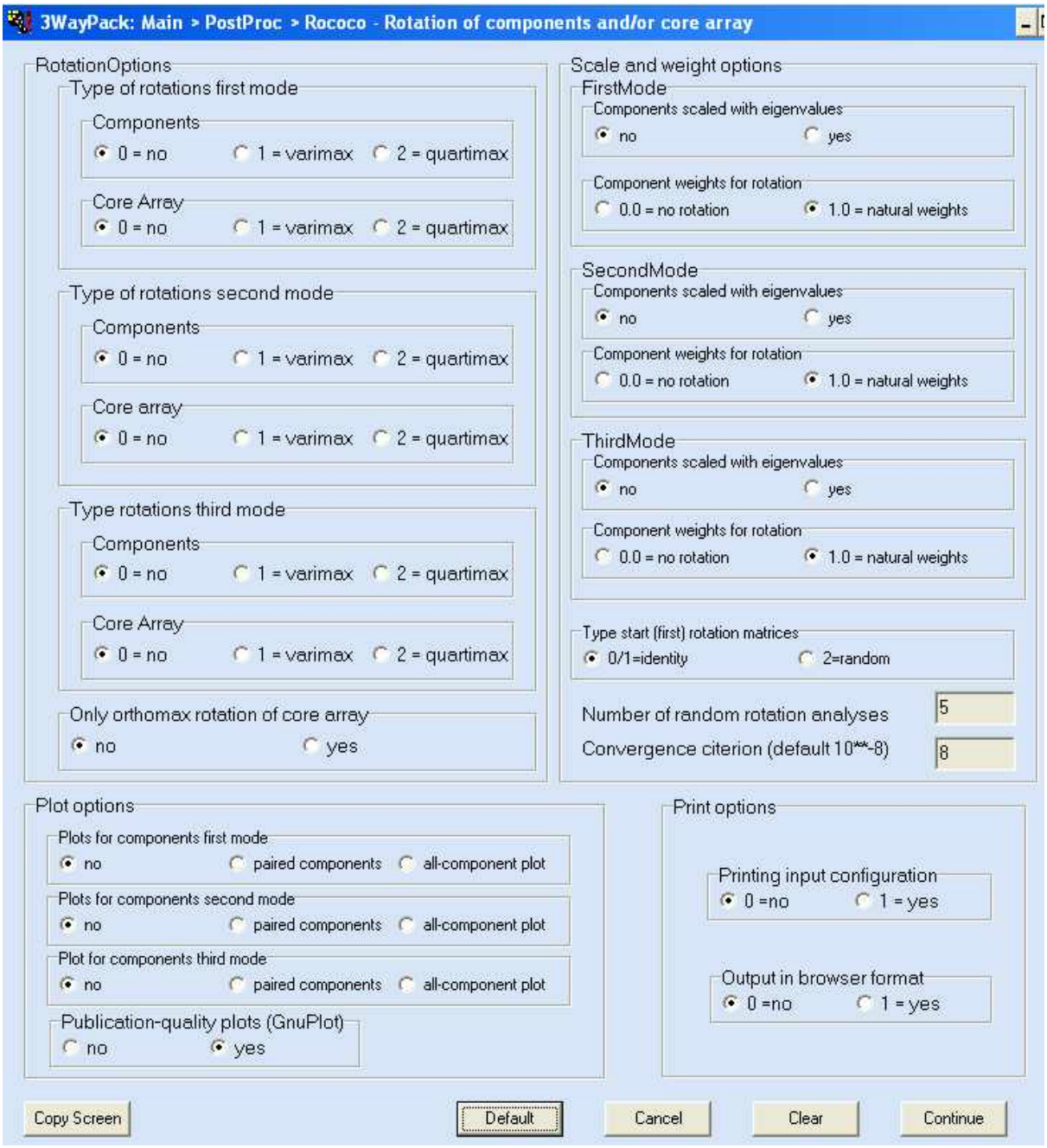
For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

5.5 T3Rotate

Introduction

T3ROTATE is a program to rotate components from a three-mode analysis, in particular from a TUCKALS2 or TUCKALS3 analysis, fitting the Tucker2 model and the Tucker3 model, respectively. No rotations are available for components from a TRILIN analysis as this is against the basic properties of the Parafac model fitted by that program. In the case of TUCKALS2 all the options related to the third mode are disabled.

Any rotation of components is compensated by a counter-rotation of the core array.



Available rotations.

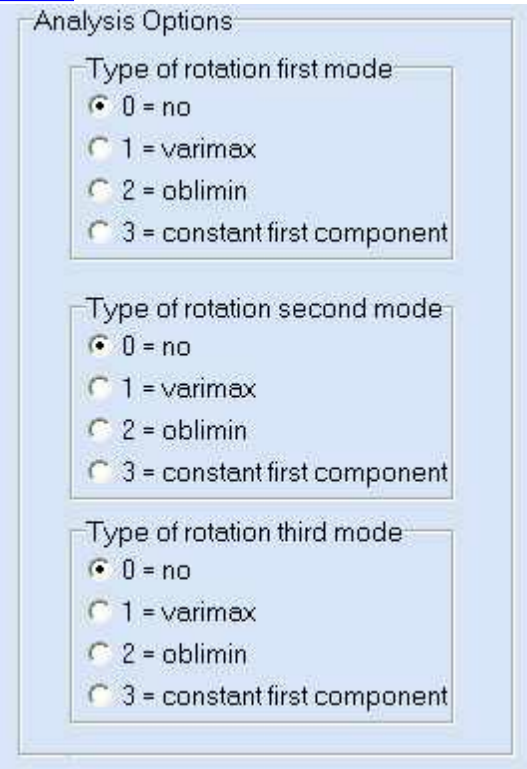
1. *Varimax* rotation on the orthonormal components (equivalent to the Harris-Kaiser independent cluster rotation; see Kiers & Ten Berge, 1994;
2. *Oblimin* transformation of the orthonormal components;
3. *Optimally constant first component*. This option can sometimes be used to facilitate interpretation, especially of reference modes for joint plots.

5.5.1 Rotation options

Introduction

The component matrices of the three modes can be independently transformed, either by varimax, or oblimin, or in such a way that the first component of a component matrix is as constant as possible using a nonsingular transformation. Note that the default for transformation is the *orthonormal*

component matrix. This is different from two-mode PCA where the basis is the matrix with component-variable correlations (loadings). To have the latter option please specify the scaling option for the desired mode.



Options (Same for each mode)

There are four choices: no rotation, varimax, oblimin, and constant first component

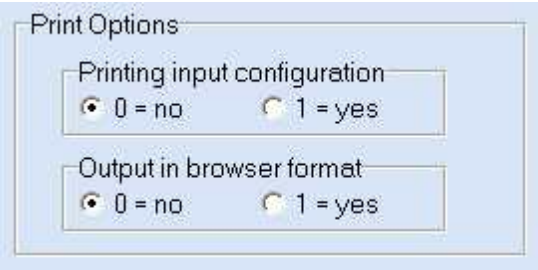
- *No rotation*: Default
- *Varimax - orthonormal rotation*: The Varimax routine used is based on the code by Velleman. It operates on an orthonormal component matrix . Because of the orthonormality, the Varimax is equivalent to the Harris-Kaiser independent cluster rotation; see Kiers & Ten Berge, 1994.
- *Oblimin - oblique rotation*: The Oblimin rotation or rather transformation is based on an algorithm by Clarkson & Jennrich. It operates on the orthonormal component matrix. The correlations between the components are printed as well. Due to the nonsingularity of the transformation and hence the nonsingularity of the inverse transformation on the core matrix, the sums of squares partitioning of the core matrix is no longer valid.
- *Constant First Component*:: Optimize equality of the first component coefficients. The Constant First Component transformation is based on code by Arbuckle and Friendly. The output will give a measure of the improvement of equality of the elements in the first component.

For detailed explanations see Kroonenberg (2008, Section 10.3).

5.5.2 Print options

Introduction

There are two types of print options available in T3ROTATE: (1) printing of the initial configurations are desired, and (2) whether output in browser format is desired.



- *Browser output*.
  When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser.
  Navigation. The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.
  Plots. If plots have been requested and they have been created on the Postprocessing screen via the View/Create button, they will be displayed in the browser as well. The plots will show up in the data directory in png-format
- *Browser files*. The program can write three files with HTML code:

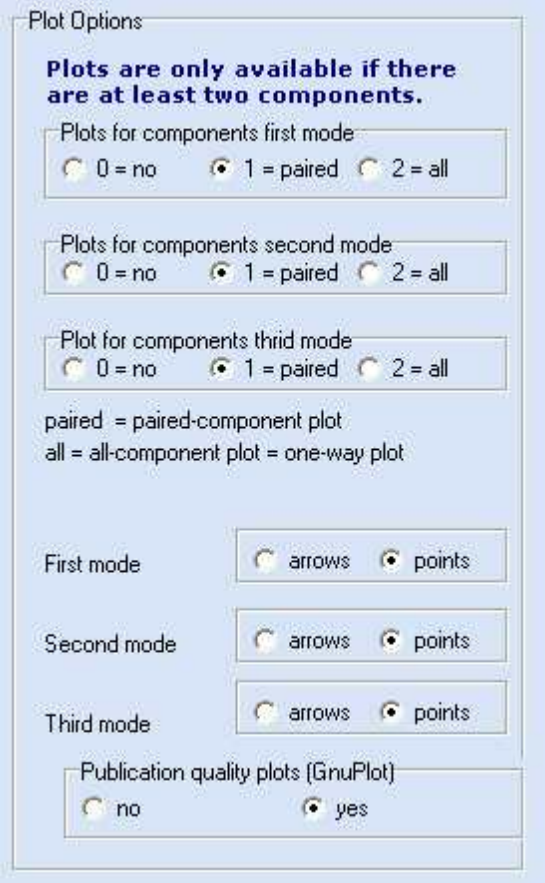
<jobname>RO.htm	The browser output can be displayed outside 3WAYPACK by double-clicking this file.
<jobnameBr>RO.htm	This file is the side bar containing the navigation
<jobnameBd>RO.htm	This file contains the actual output

Some newer types of output may not yet be available in browser format.

5.5.3 Plot options

Introduction

T3ROTATE does not make any plots for a particular mode if there is only one component in that mode. In all other cases one may request a plot of the components one against another (*paired-components plot*) or all components in a single plot against the levels of the mode (*all-components plot*). Publication-quality plots will be produced by default but can be suppressed.



Options

- *No*: Default.
- *Paired-components plots*: Pairs of components from one mode are plotting against each other
- *All-components plots.*: All components of a mode are plotted against the level labels or level numbers if no labels are present

Arrows or points

One also has a choice to plot the levels of the mode either as points or as arrows emanating from the origin.

Publication-quality plots

When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>RO.plt. By clicking the View/Create button on the Postprocessing Screen these plots are created. See the Utilities section for more details on GNUPLOT.

When browser output has been selected only png-bitmap graphics are created for inclusion in the HTML output. Editable graphics in so-called vector format are only produced when no browser output has be requested.

If you associate in Windows the extension .plt with the GNUPLOT program you may double click on the .plt file in the data directory to create the plots outside 3WAYPACK.

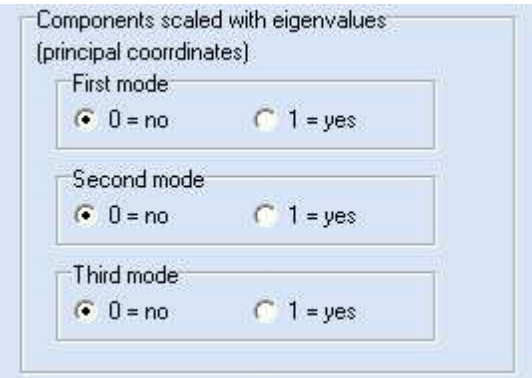
For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs

such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

5.5.4 Scaling components

Introduction

In standard component analyses rotations are generally carried out on matrices for which the columns are in principal coordinates. This is not usually done in three-mode analysis where the rotations are often carried out on normalised coordinates. However, this option offers the opportunity to apply the appropriate scaling of the components so that they are in their principal coordinates. In certain types of interpretations it is advantages to do so, especially if one wants to have a 'Murakami core array'; see Kroonenberg (2008, Section 10.5.1).

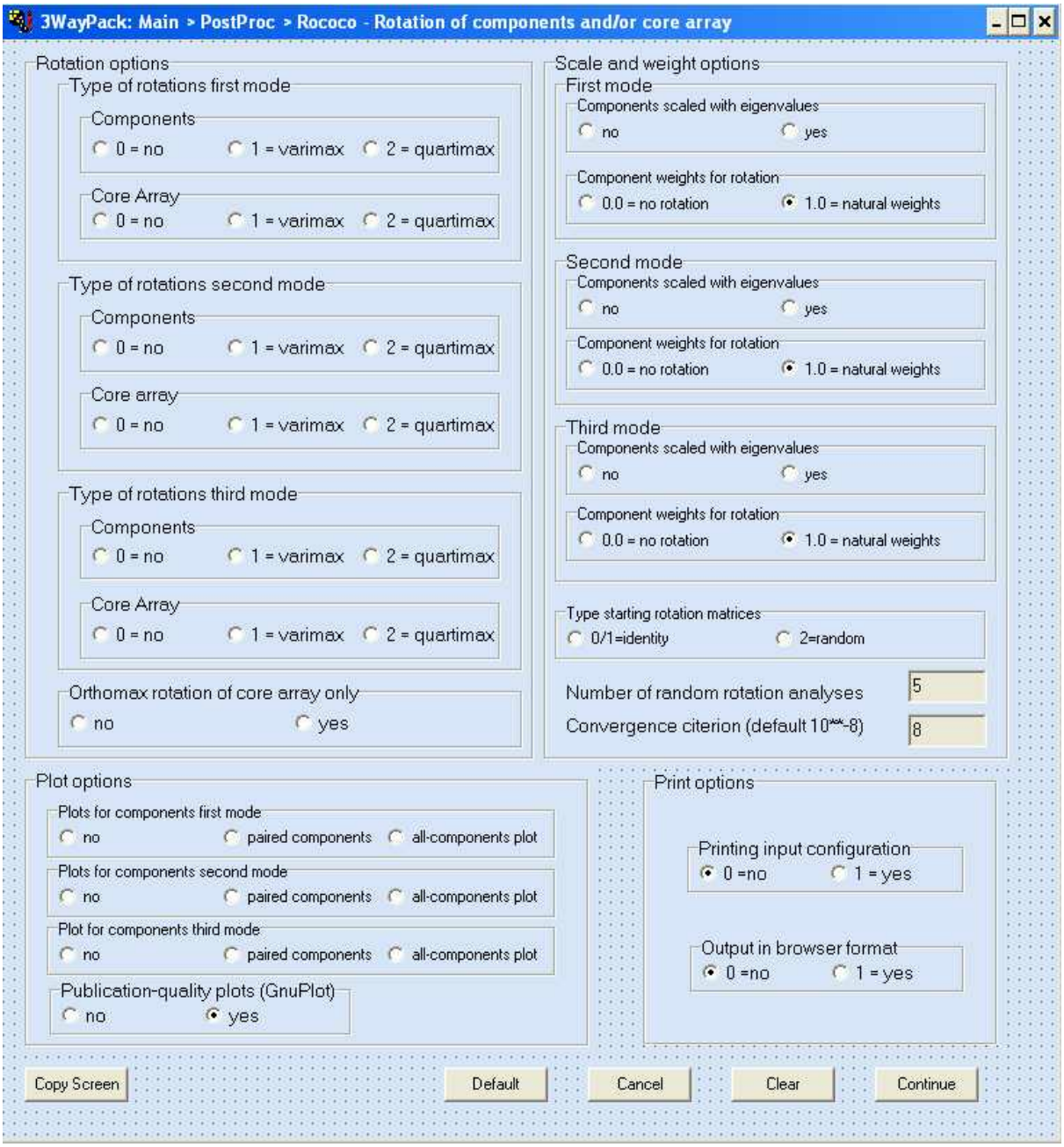


5.6 Rococo

Introduction

ROCOCO (Rotating core and components) is a program to rotate one or more component matrices and/or the core array simultaneously using a varimax procedure. The ROCOCO program in 3WAYP ack is a Fortran version of a MATLAB program developed by Henk. A.L. Kiers, University of Groningen (h.a.l.kiers@rug.nl)

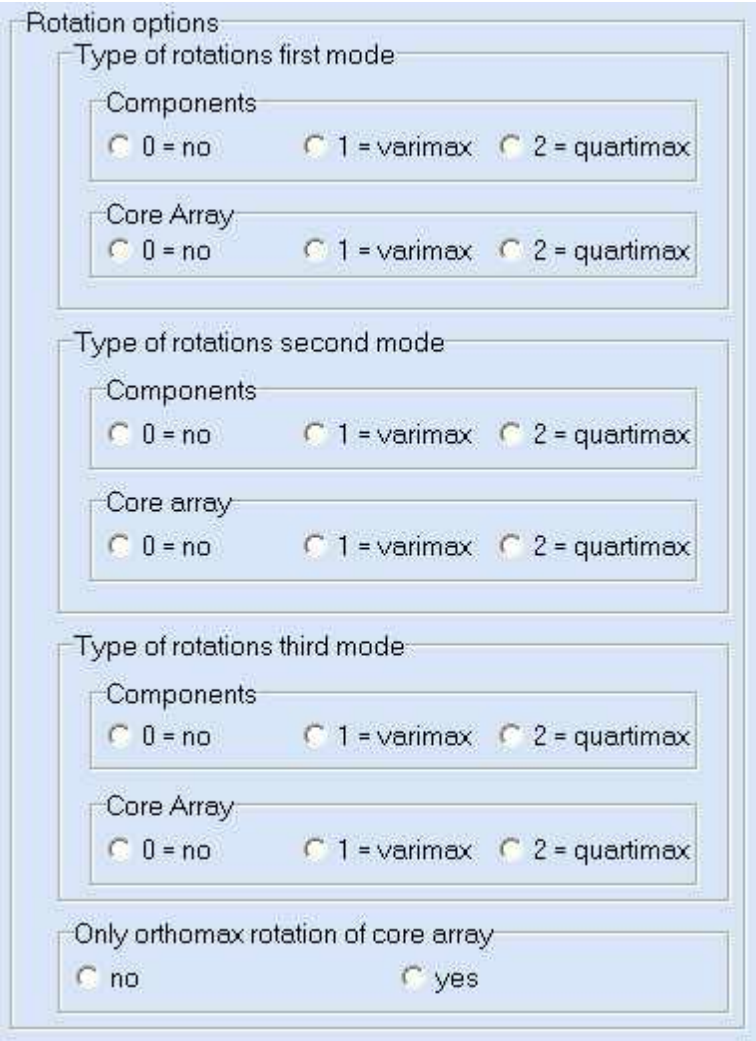
The use and interpretation of the program ROCOCO is not completely straightforward and careful study of the original paper (Kiers (1998)) and the comments in Kroonenberg (2008, Section 10.4) should be very useful.



5.6.1 Rotation options

Introduction

ROCOCO allows for simultaneous orthonormal rotation of more of more component matrices and/or core array with the explicit aim to simplify the structure in the core array and each or some of the component matrices. The rotations are successively carried out for each component matrix and for the core array, whereby each of the four entities can be given a weight to emphasise its importance in the fitting process. The idea is that the loss functions referring to entities with higher weights will be more important in the minimisation of the loss function. For example, zero weights for the components will lead to optimally simple core arrays but no simplification of the components and vice versa.



Options

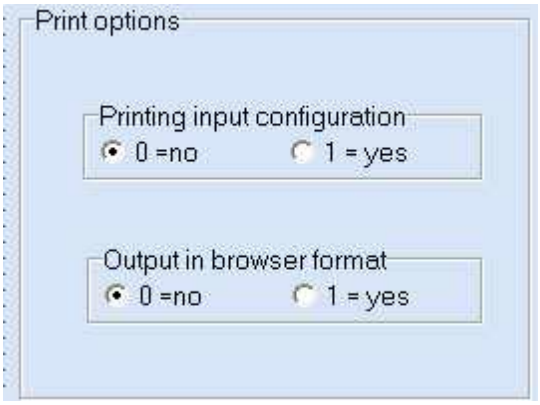
The rotation options are the same for all component matrices. Exclusive rotation of the core array can only be done with an varimax transformation.

- *No rotation*: Default
- *Varimax rotation*: This is the standard rotation option and it is generally recommended.
- *Quartimax rotation*: For details see the original publication by Kiers (1998)

5.6.2 Print options

Introduction

There are two types of print options available in ROCOCO: (1) printing the initial configurations, (2) generating output files in browser format.



- *Browser output*.
  When this option is checked, the program will produce HTML-output suitable to be displayed in any browser. By default the program will use its internal browser.
  Navigation. The HTML-output is set up such that one can easily navigate the, sometimes voluminous, output file.
  Plots. If plots have been requested and they have been created on the Postprocessing screen via the View/Create button, they will be displayed in the browser as well. The plots will show up in the data directory in png-format.
- *Browser files*. The program will write three files with html code:

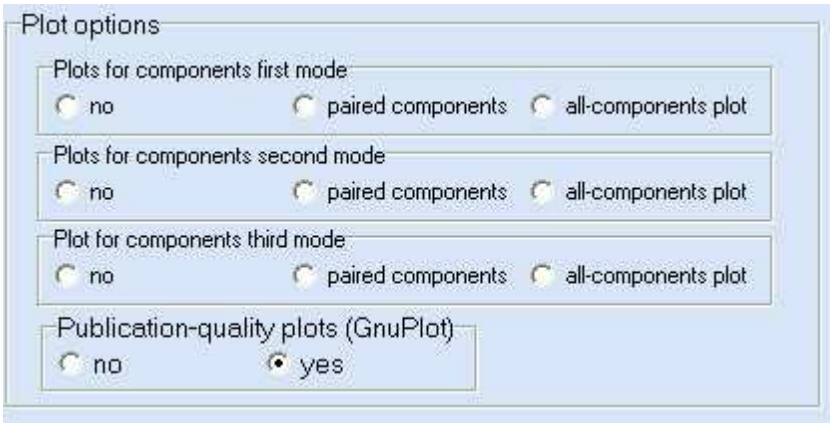
<jobname>RC.htm	The browser output can be displayed outside 3WAYPACK by double-clicking this file.
<jobnameBr>RC.htm	This file is the side bar containing the navigation
<jobnameBd>RC.htm	This file contains the actual output

Some newer types of output may not yet be available in browser format.

5.6.3 Plot options

Introduction

T3ROTATE does not make any plots for a particular mode if there is only one component in that mode. In all other cases one may request a plot of the components one against another (*paired-components plot*) or all components in a single plot against the levels of the mode (*all-components plot*). Publication-quality plots will be produced by default but can be suppressed.



Options
- *No*: Default.
- *Paired-components plots*: Pairs of components from one mode are plotting against each other
- *All-components plots*.: All components of a mode are plotted against the level labels or level numbers if no labels are present

Arrows or points (*not yet implemented*)
One also has a choice to plot the levels of the mode either as points or as arrows emanating from the origin.

Publication-quality plots
When plots are requested they will always be printed in the standard output, but publication-quality plots can only be obtained by choosing the GNUPLOT option. The program generates special code for the plotting program GNUPLOT and writes this to the file <jobname>RC.plt. By clicking the View/Create button on the Postprocessing Screen these plots are created. See the Utilities section for more details on GNUPLOT.

When browser output has been selected only png-bitmap graphics are created for inclusion in the HTML output. Editable graphics in so-called vector format are only produced when no browser output has be requested.

If you associate in Windows the extension .plt with the GNUPLOT program you may double click on the .plt file in the data directory to create the plots outside 3WAYPACK.

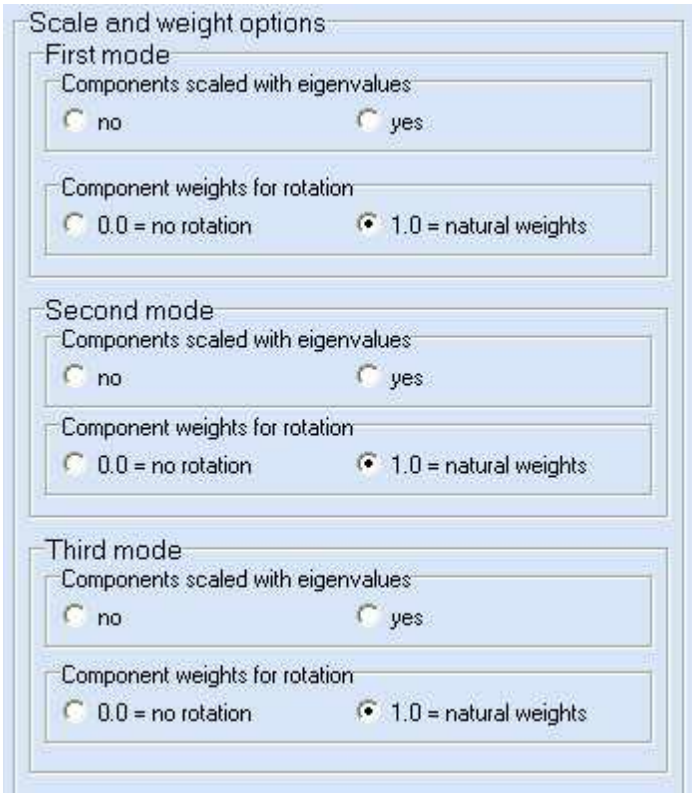
For further details see the special section on GNUPLOT, where it is also described how to save the graphics as window-metafiles (.wmf/.emf) which can be transferred to presentation programs such as Microsoft POWERPOINT, Lotus FREELANCE or Adobe ILLUSTRATOR for further editing and beautification.

5.6.4 Scaling components

Introduction
In standard component analyses rotations are generally carried out on matrices for which the columns are in principal coordinates. This is not usually done in three-mode analysis where the rotations are often carried out on normalised coordinates. The user has to indicate whether a component matrix is going to be rotated. The options are the same for each mode.

Scaling
The scaling option offers the opportunity to scale the components so that they are in their principal coordinates. In certain types of interpretations it is advantages to do so, especially if one wants to have a 'Murakami core array' ; see Kroonenberg (2008, Section 10.5.1).

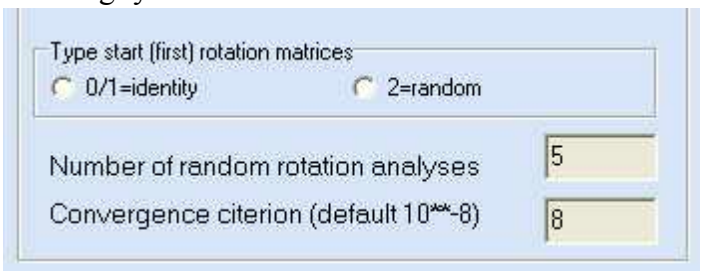
Weights
The weights determine the role each component matrix plays in creating simple structure for their components and/or core array. Zero weights prevent simplifications in a component matrix or core array, while larger values for some of the modes will emphasise their importance in reaching a simple structure. At present it is only possible to indicate whether a component matrix should be rotated or not by indicating that *natural weights* are to be used.



5.6.5 Iteration options

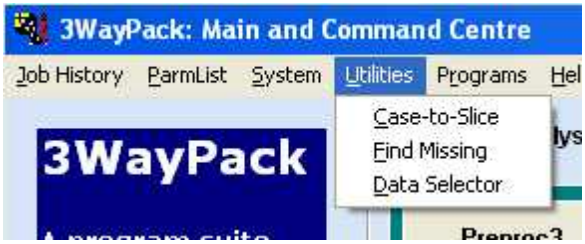
Introduction

To steer the iterations one can specify the type of initial component matrices, i.e. whether they are identity matrices (default) or random matrices. In addition one can specify whether only a single analysis needs to be run or a larger number of them. In general at least five starting configurations are recommended as the loss function can be a bit bumpy leading to local minima. Finally the convergence criterion can be set a larger or smaller accuracy. To change the convergence criterion specify the size of the exponent, e. g. 10 for 10^{-10}. On most machines the criterion should not be less than roughly 10^{-16}.



6 Utilities

Introduction
Standard statistical programs often do not cater for the kind of data manipulation that is required for three-mode data. Under the Utilities heading on the Main Menu bar several data management programs are available. In addition, the suite contains a few utility programs which also do a certain amount of preprocessing.



Utility programs
Three utility programs are available from the Utilities entry on the Main Menu bar.

- *Case-to-slice*: Change the input data from case-mode to frontal-slice mode.
- *Find Missing*: A utility to locate the missing data in a three-way array.
- *Data Selector*: A utility to select subsets from a three-way array by eliminating designated slices.

General explanations of facilities in use by many programs in 3WayPack.
- GNUPLOT: A program to produce publication-quality plots on the basis of command file with extension .plt, produced by virtually all programs included in the program suite.
- *Input format specification*. Details on specifying the input format of a data to be used by the analysis programs in TWPACK.
- *Conversion to ASCII or capitals*: A facility for reformatting raw output files to basic ASCII or all capital letters.
-

6.1 From case mode to slice mode

Introduction
Conceptually a three-way array is mostly depicted as in the left panel of the figure below (Figure **A three-way array**). However, physically data files come in one of two shapes. The data can be arranged as in all major statistical packages i.e. in *case-mode* (a wide matrix). In this case all data of each case are together on one or more records (middle panel; Figure **case-mode**). The alternative is *frontal-slice mode* (a tall matrix), i.e. the data are slices vertically arranged (right-hand panel; Figure **Frontal-slice mode**). The three-way programs in this package need the data in frontal-slice mode.

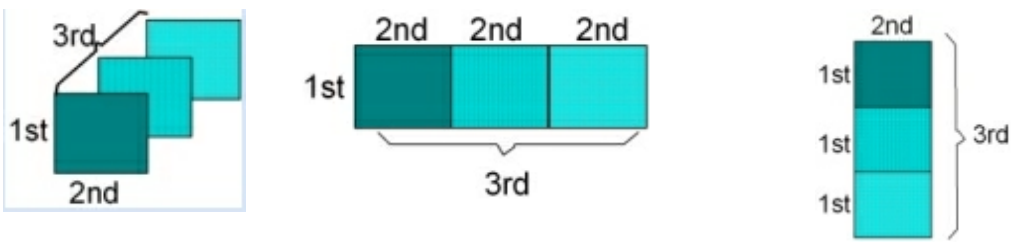
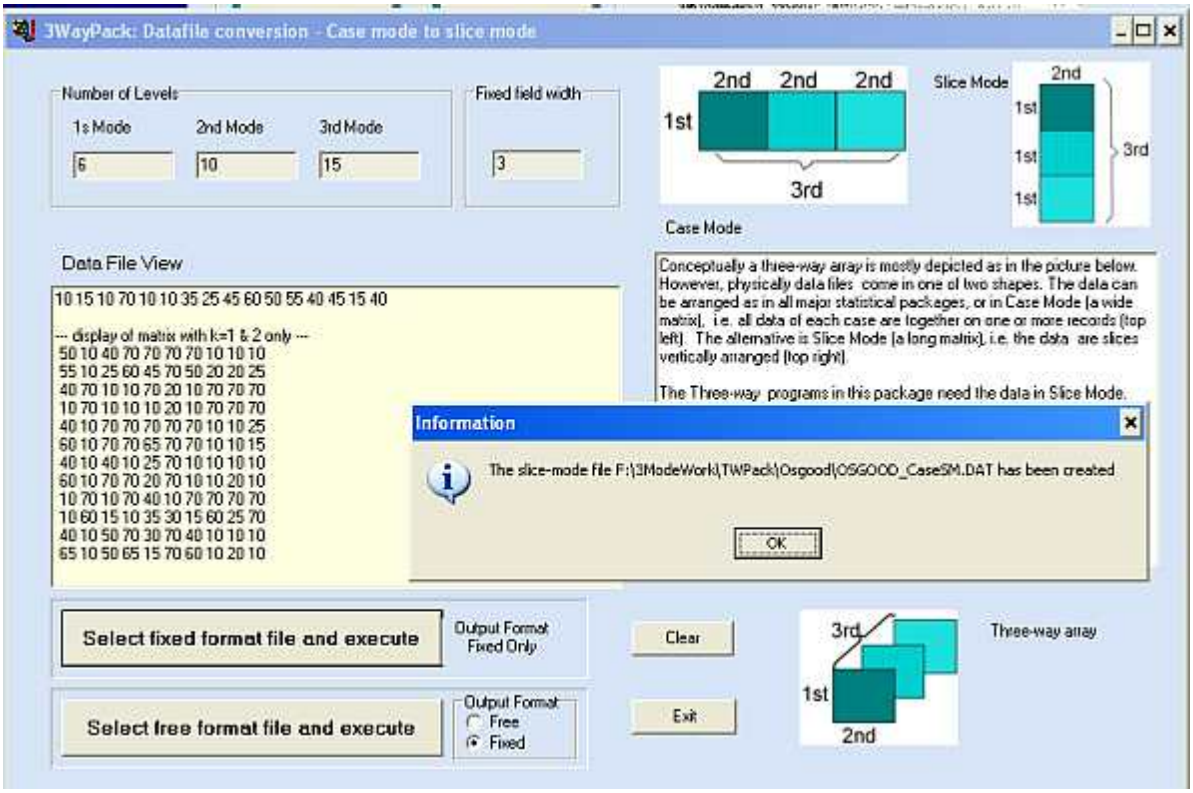


Figure: A three-way array **Figure: case-mode (wide matrix)** **Figure: Frontal-slice mode (tall matrix)**

In the middle panel above a (1st Mode) by (2nd Mode nested in the 3rd Mode) data set is shown in case-mode. Suppose the first mode has 15 levels, the second mode 10, and the third mode 3, then the data form a 15 by 30 wide matrix.

The Case-to-Slice program converts this data matrix into a (1st Mode nested in the 3rd Mode) by 2nd Mode matrix as is shown in the right-hand panel. Thus the tall matrix has the shape of a 45 by 10 matrix. This is the correct shape for the input of the three-way programs.



Options

Number of levels
Provide the program with the correct dimensions for each of the modes of the data set. Note that the program cannot detect whether the nesting has been correctly specified, i.e. whether the numbers are supplied so that the second mode is nested in the third mode or vice versa.

Fixed field width
This option can be used if the data are in a fixed format in which each number takes up a fixed amount of space on all records. If such a fixed format is not available, the data can be read in as unformatted data with spaces between the numbers. Note however that in free format no information should precede the data such as row and column numbers, id information etc. In free format only the first *J* numbers will be read. Any other information at the end of the record will be ignored. Therefore the identification information can be added at the end of a record.

Data File View
In the data field the first record of the original data are displayed as well as the first slices of the data

in their new arrangement.

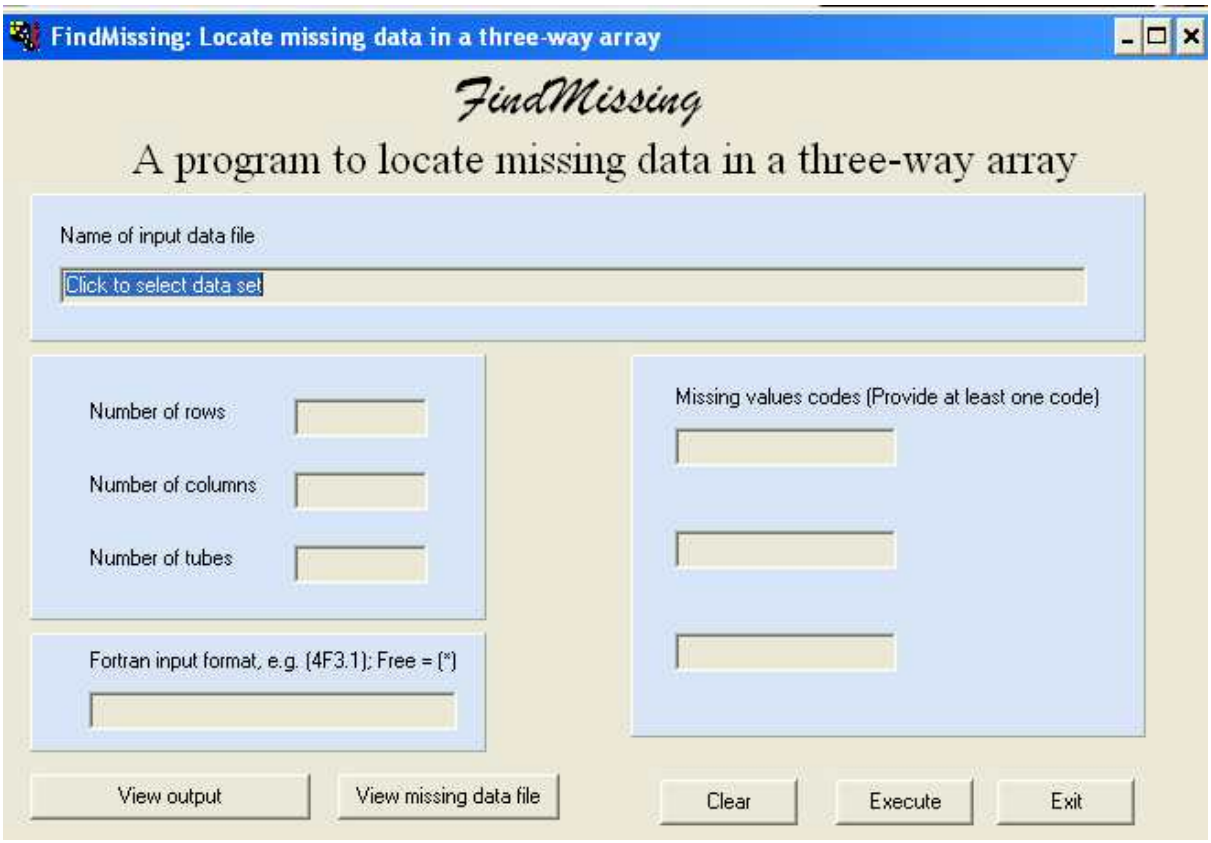
Select fixed format file and execute
Use this option for fixed format data

Select free format file and execute
Use this option for free format data. The output data set will be either in fixed format or in free format.

Error checking
We have made an attempt to prevent the user making incorrect specifications for the size of the data. Most cases have been covered but there may still be specifications which will cause the program to crash. Therefore, please be sure your dimension specification of the data is correct.

6.2 Creating a missing-data file

Introduction
If missing data are present, the analysis programs except MIXCLUS3 need a special file (the missing-data file) containing in free format the coordinates (i,j,k) in the data array where the missing data are located. A file with the name <datasetname>.mis in which these locations are stored has to be created. For additional information on missing data in 3WAYPACK, especially on starting values in the analysis programs, see Missing-value specification on the Job specifications screen.



The program FINDMISSING is solely designed to create the missing-data file in which the *locations of the missing data* in the data array are listed. It also produces information on the amount of missing data for each levels of each mode.

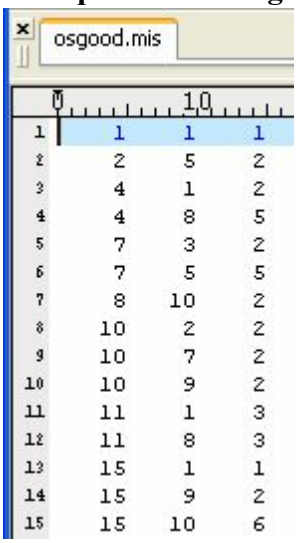
Specifications
The specifications are rather straightforward.

- *Select a data set*. The missing-data file will have the name <datasetname>.mis and will be located in the data directory.
- *Specify the dimensions* of the data array.
- Provide up to three *missing-data codes*. Note that missing-data codes can only be specified for the complete data array. Thus the same number should not be a valid value for one variable and a missing-data code for another.
- *Provide the data format*, either as fixed or as free. The latter is specified as (*). Most (even complicated) valid Fortran specifications are acceptable (see the section on Input format specification).
- *Execute* the program.
- *View* the output and/or the missing-data file.

Usage of missing-data file
The analysis programs except MIXCLUS3 need the missing-data file for their operations.

Rationale
The rationale for using a missing-data file with locations of missing data is that it makes the analysis programs more efficient especially when there are relatively few missing data because it circumvents the need to check for each data point whether there it is a valid observation. Moreover it saves space as no binary array is needed to indicate which elements of the array are missing and which are valid.

Example of a missing-data file



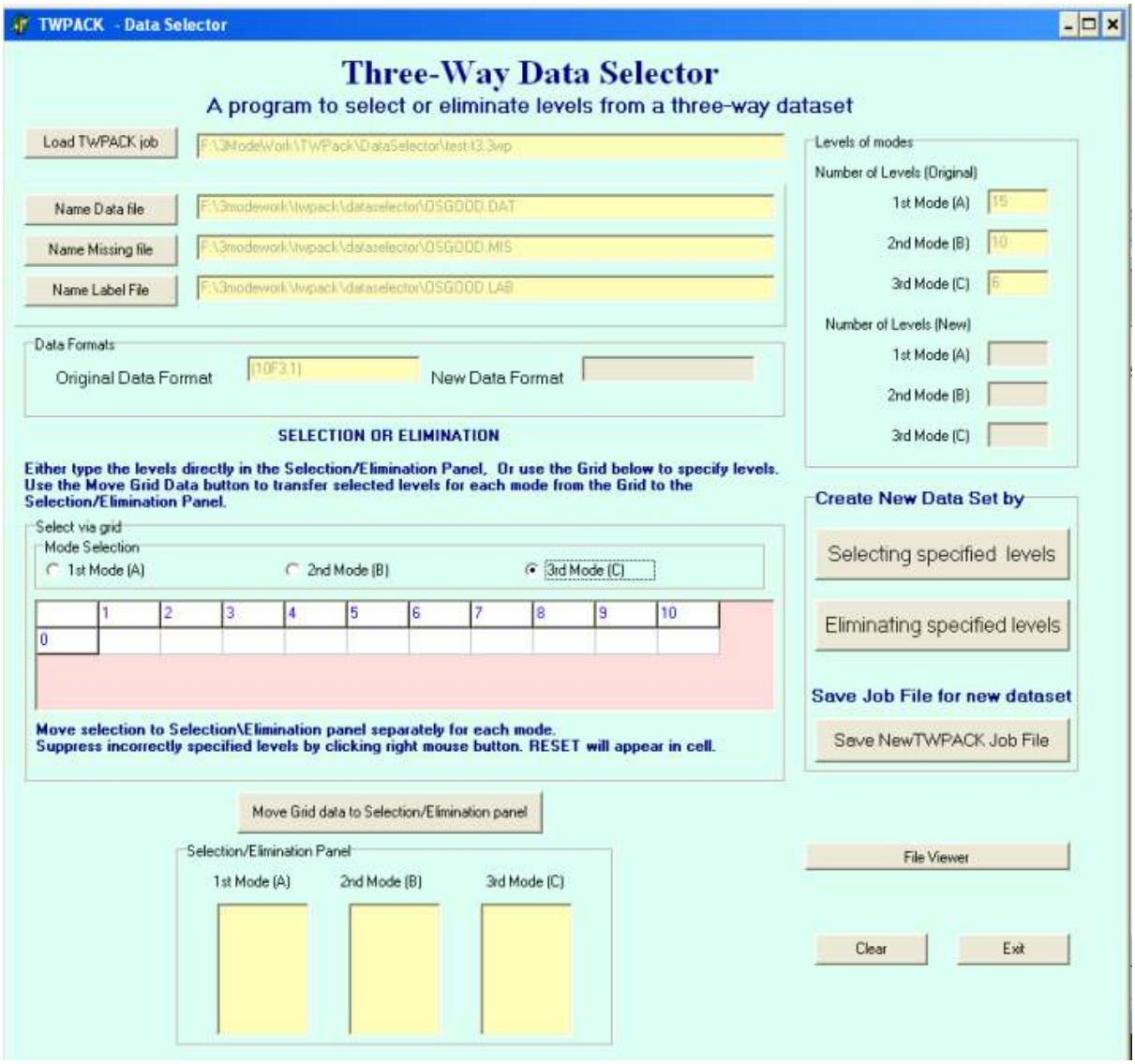
1	1	1	1
2	2	5	2
3	4	1	2
4	4	8	5
5	7	3	2
6	7	5	5
7	8	10	2
8	10	2	2
9	10	7	2
10	10	9	2
11	11	1	3
12	11	8	3
13	15	1	1
14	15	9	2
15	15	10	6

Figure: Missing-data file

6.3 Data subset selection

Introduction
The main purpose of the program DATA SELECTOR is to select or eliminate levels from a three-way dataset. Being part of the TWPACK utilities, the program also creates a new TWPACK jobfile which can be used to run the changed data set. If a missing-data file <datasetname>.mis and/or a label file <datasetname>.lab are present new versions are created in accordance with the new format of

the data set.

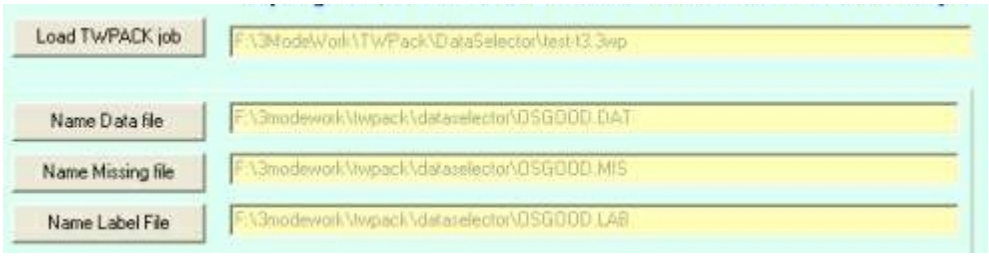


6.3.1 Basic operation

Introduction

The DATA SELECTOR is meant to assist in making subsets of existing data sets. This is done either

1. by specifying which levels should be included in the new data set.
2. by eliminating those levels which should be eliminated.



Operation of DATA SELECTOR

There are two ways to start the subsetting procedure.

1. *A previous job file is available* for the data, so that it can be read and the appropriate

information about the data set will be transferred to the program. After the information has been read in the parameters cannot be changed (the appropriate entries will be shown in grey).

2. *No previous job file is available*. The user will first have to fill in the number of levels of the modes and the input format of the data. Then the name of the data file and if available the missing-data file and the label file can be specified. For details about specifying the data format, see the special section on such formats.

TWPACK jobfile

A TWPACK job file holds all relevant data of the past analyses with the data set mentioned in the job file. Creating a subset of the data set mentioned in the job file has an effect on the number of levels of the modes, the input format, the label file and the missing-data file. DATA SELECTOR will read all the necessary information from the job file and adapt them wherever necessary, so that it can create a new job file and associated files which correspond to the new situation. In the DATA SELECTOR screen all the relevant information from the job file will be copied and displayed. The user will not be able to change these values manually.

The numbers of levels and the input format after selection/elimination will be computed by the program and cannot be set by the user.

Please be aware that if many types of analyses are described in the original job file these will not be copied into the new job file.You will have to rebuild this anew. This is necessary because TWPACK requires a unique coupling of an analysis and a data set.

Files names for the new reduced data set

If available, the missing-data file and/or the label file belonging to the data set will also be adjusted so that they correspond with the new data set. The original files will stay intact and new ones will be created with an addition of _sub to the file name. Thus

- The new data file becomes <datasetname_sub>.dat
- The new label file becomes <datasetname_sub>.lab:
- The new missing-data file becomes <datasetname_sub>.mis:

Moreover, if a job file from a TWPACK analysis was used, a new job file is made corresponding to the new subset data file. The naming convention is the same:

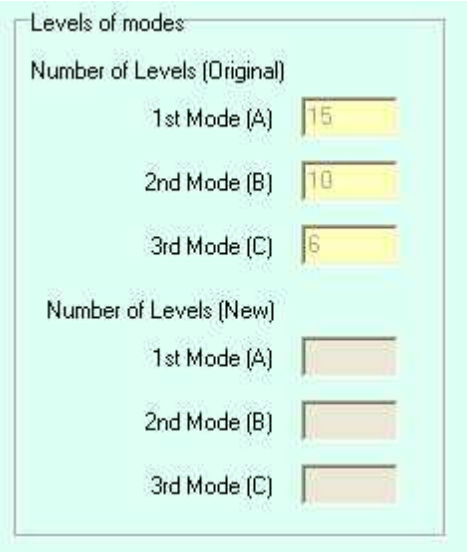
- The new job file becomes <jobfilename_sub>.3wp.

6.3.2 Level and format specification

Introduction

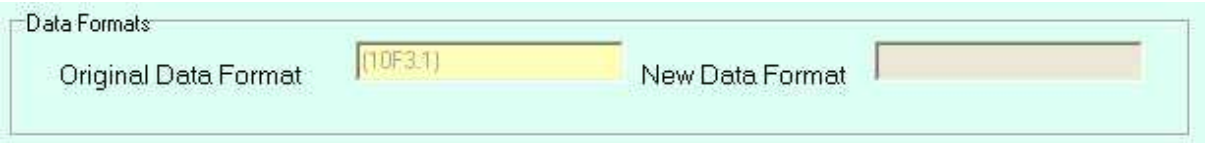
When no TWPACK job file is available the user has to specify the levels of the modes and the input format for the data. After that the data set itself can be specified.

Number of levels of the three modes



The number of levels for each mode need to specified by the user if no previous job file is available. This has to be done *before* the data file is selected. The same is true for the input format. After selection or elimination the program lists the new numbers of levels and the new data format.

Data input formats



Please note that before execution the original format is checked against the length of the physical data record. Since this is the only check, it is important to make a correct format specification. The file viewer can be used after the data set has been specified to check whether the format specification was correct.

Visualising levels as slices

Removing a level of the first mode removes an: horizontal slice; removing a level of the second mode removes a lateral slice; removing a level of the third mode removes a frontal slice.

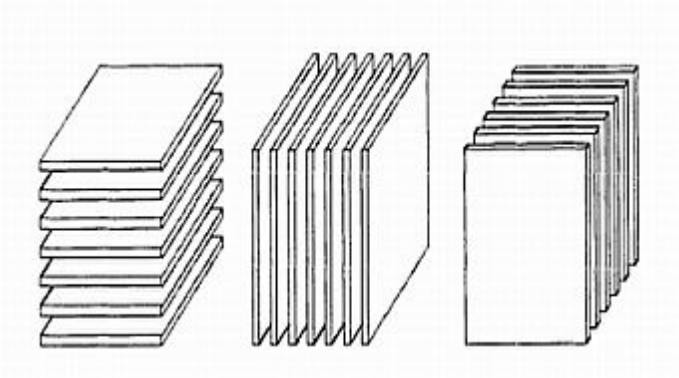


Figure: Data slices: Horizontal, Lateral and Frontal slices.

6.3.3 Selection/elimination of levels

Introduction

The purpose of the DATA SELECTOR program is to remove complete levels from the data.

Removing a level implies that an entire slice is taken out of the data array (see Figure Data slices). Depending on the number of slices to be removed one may choose to specify either which slices/ levels have to be removed or which levels/slices are to remain. Usually it is simpler to use the first option. If no levels are specified for a certain mode, all levels for that mode will be present in the new data set.

Visualising levels as slices
Removing a level of the first mode removes a horizontal slice; removing a level of the second mode removes a lateral slice; removing a level of the third mode removes a frontal slice.

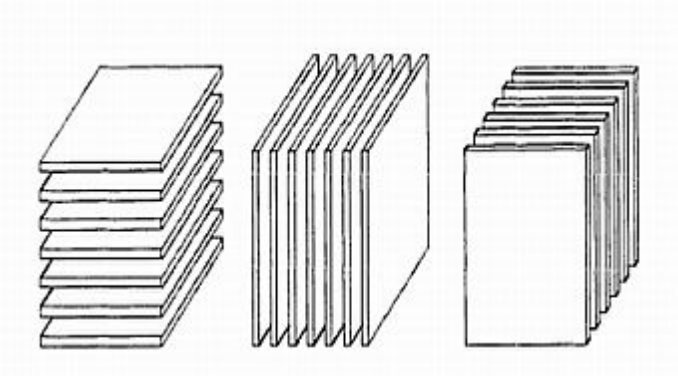
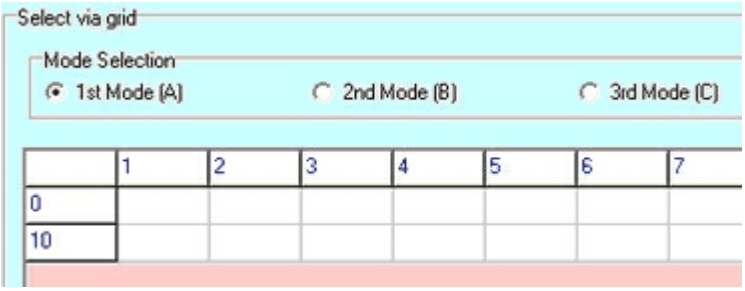


Figure: Data slices: Horizontal, Lateral and Frontal slices

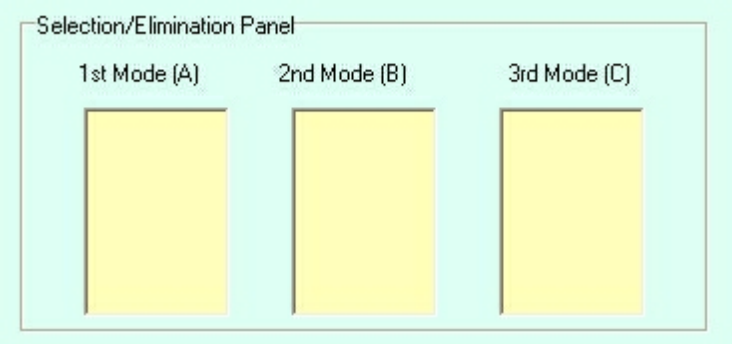
Selecting/Eliminating via the Grid
The program contains a grid which can be used to indicate which levels are involved with the selection or elimination. This indication of levels has to be done separately for each mode.



In the present grid level 2, 5, 7, and 14 are indicated. Most likely they will be eliminated from the data set. By accident level 13 was also specified but by double clicking on the 13 box the values has been RESET to inactive. If the user clicks in a box outside the number of levels, thus in box 17 if there are only 15 levels then the word ERROR appears. Double clicking will change this to RESET, but this is not required.

Once a selection of levels has been made for a mode, the levels can be transferred to the Selection/ Elimination Panel by clicking on the button Move grid data.

Selecting/Eliminating by typing
The level selection panel contains three boxes with the enumeration of the level values which have been chosen for selection or elimination. The boxes can be filled either via the grid (see above) or by direct typing.



6.3.4 Program Execution

Introduction
There are two options (Selection of levels and Elimination of Levels) for executing the DATA SELECTOR program and each has its own button. The program will give a message that the new data file has been created supplying its name as well. After the new data file has been created as well as the associated label file and missing-data file (if available), the associated TWPACK job file can be saved for later usage if an original job file was used in the program.

Note: If no levels are specified for a certain mode, all levels for that mode will be present in the new data set.

Files names for new reduced data set
If available, the missing-data file and/or the label file belonging to the data set will also be adjusted so that they correspond with the new data set. The original files will stay intact and new ones will be created with an addition of _sub to file name. Thus
- The new data file becomes <datasetname_sub>.dat

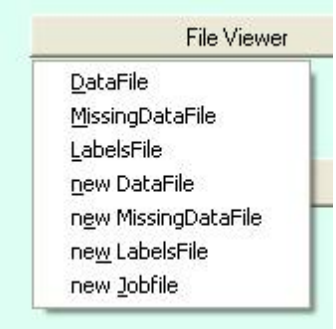
- The new label file becomes <datasetname_sub>.lab:
- The new missing-data file becomes <datasetname_sub>.mis:

Moreover, if a job file from a TWPACK analysis was used, a new job file is made corresponding to the new subset data file. The naming convention is the same:

- The new job file becomes <jobfilename_sub>.3wp.

File viewing

Both the old and new files can be inspected to ascertain whether the selection/elimination procedure has been carried out as intended. To do this one can use the File Viewer.



6.4 GnuPlot

Introduction

GNUPLOT is an open source plotting program which works with a command file which generally has the extension .plt. Here we will only indicate some very basic properties of the program. In particular what needs to be done to store the plots on the clipboard for further processing by graphical and presentation programs.For full information on GNUPLOT go to GnuPlot Central. (*www.gnuplot.info*)

The plot files produced by 3WAYPACK are themselves fully editable. Thus colours and graphical elements can be added and changed if one is familiar with the command language. Editing can be carried out with a standard text editor such as NOTEPAD or WORDPAD. Note however that there are better editors than these two. For general editing purposes an editor which can also work columnwise apart from only paragraphwise or linewise is to be preferred.

Starting GNUPLOT

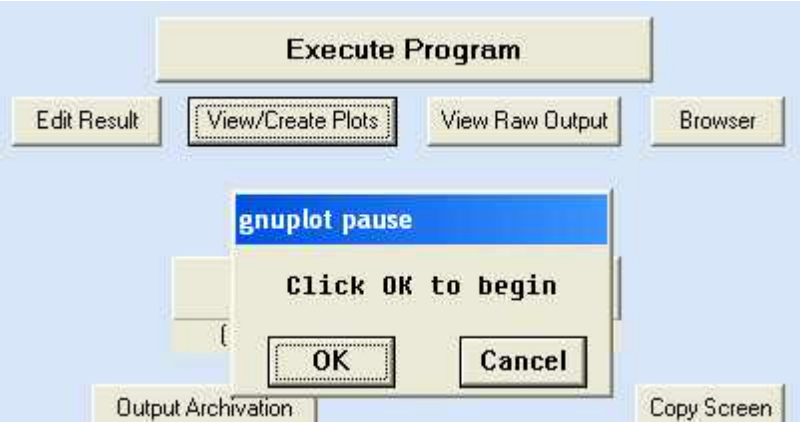


Figure. GnuPlot start

A user can request in an analysis or postprocessing program that a command file with instructions for GNUPLOT is produced. After the program is finished double clicking the View/Create Plots button will either (1) create the plots in png-format if also browser output has been specified or (2) the plots can be viewed one by one so that they can be saved for further use.

Clicking on the button will start GNUPLOT and a little information box appears with an OK and Cancel button, as shown in Figure GnuPlot start.. Depending on the analysis and the specifications one of more plots will be generated. In the case of browser output these plots will be automatically stored in the data directory in png-format. In all other cases specific measures are need to store the graphs,. as is specified below.

Transferring plots to the Clipboard.
In non-browser mode GNUPLOT will produce *vector plots* (i.e. editable plots in .wmf-format) which can be stored via the clipboard to a file or which can be directly pasted into a presentation or word-processing program.

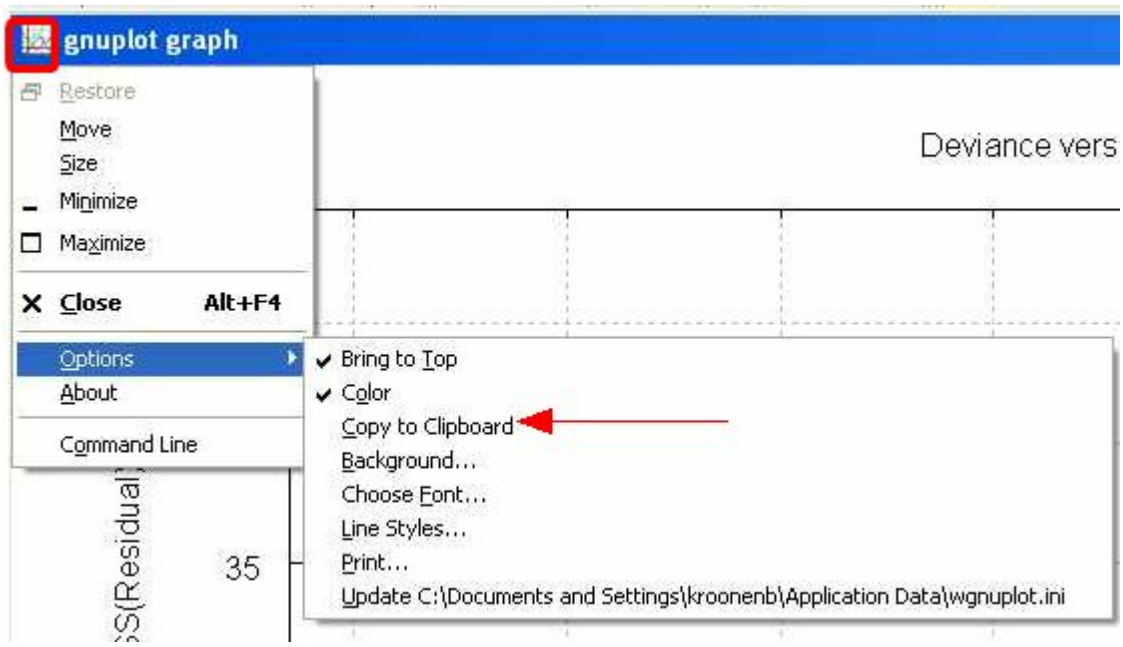


Figure. GnuPlot options

To copy a graph to the Clipboard one needs to click on the GNUPLOT icon after which a roll-down menu appears (see **Figure GnuPlot options**). Choose Options and click on Copy to Clipboard. For further editing one needs to Ungroup the graph in the application at hand. After that the graph can be edited at will.

Adapting plots in GNUPLOT
In the menu of **Figure GnuPlot options** there are also options shown for changing properties in the graph itself. The graphs themselves cannot be changed but the attributes of the objects in the graph can be changed.such font type and font size, line thickness and colour, etc. Changes in the default settings can be saved in wgnuplot.ini; this file resides in the data directory.

Stand-alone usage of GNUPLOT
If one associates in Windows the plt-extension with the program GNUPLOT, double clicking a plt-file will open GNUPLOT and execute the command file. Thus the graphs can be produced outside the 3WAYPACK program using the WINDOWS EXPLORER or similar programs. The GNUPLOT executable, WGNUPLOT.EXE, is located in the program directory of 3WAYPACK.

6.5 Input format specification

Introduction
Input of data in statistical programs without a spreadsheet is always a challenge for both the programmer and the user. In 3WAYPACK the older, but extremely flexible, Fortran format is used. Previously it was also standard in programs like SPSS but it is now mostly superseded by some sophisticated data-entry program.

Types of input formats
Fortran formats describe the records on which the data reside. There are basically two formats: Free format and fixed format.

- *Free format:* The requirement is that all data on a record are read in, or at least those at the beginning of the record. The length of each number is irrelevant, they only have to be separated by one or more spaces, or if the programs provide for this (e.g. Excel) by a comma, the so-called c (omma) s(eparated) v(alue) format or csv format. Free format is easy if there are no leading additional numbers which should not be read and all numbers are separated by delimiters (= spaces or comma's). Free-format data are often difficult to inspect by eye, because they are generally not columnwise aligned.
  Example:
  9 4.789 -0.02
  5.90 2 -0.000004
  These data can be read in with free format if three or less number per record have to be read in.
- *Fixed format*: The requirement is that the data on the record are nicely lined up so that they can also be inspected by eye with the additional advantage that one may skip certain columns.
  Example:
  9.00 4.789 text -0.020000
  5.90 2.000 cat -0.000004.
  The input format specification may indicate that some variables should be skipped.

Details of input formats
An input format describes the data on a record. A distinction can be made between integer values (indicated with an I), real numbers (indicated with an F), character input (indicated with an A) and columns to be skipped (indicated with an X). There are more possibilities but they are not implemented by 3WAYPACK. In order to describe the data on a record one needs three specifiers:
1. *field width* - indicated below by *w*
2. *number of decimals of a data value:* indicated by *d*.
3. *repetition factor*: the number of times a descriptor is to be repeated - indicated by *n*.

The rules of a Fortran-format statement using only the I, F, A and X descriptors are the following:

- *Reading integers*: *n*I*w* - read *n* integers each occupying *w* columns on the record.
- *Reading reals*: *n*F*w*.*d* - read *n* reals each occupying *w* columns of which the last *d* numbers are decimals.
- *Skipping columns*: *n*X - skip *n* columns

- *Reading characters*: *n*A*w* - read *n* character variables each occupying *w* columns.

Example of a fixed input format:
The following fixed format is an example of a valid format: (5X,3I5, 2F4.2, 4X,A). It indicates that:
1. the first five places on the record should be skipped (5X) - columns 1 through 5;
2. three integer are to be read next each taking up 5 places on the record (3I5) - 6 through 20;
3. then there are two real numbers each with a field width of 4 of which the last two digits are decimals (2F4.2) - 21 through 29;
4. five places are skipped (5X) - 29 through 33, and
5. finally a single character is read in (A) - 34 .

Below is an example of a series of data records which satisfy the above format. The first line below describes the subdivision in the columns according to the field width. The next numeric line is the ruler indicating the column numbers. Note that the number 10 is over column 9 and 10 by necessity.

```
12345123451234512345123451234123451
=================================================
....5...10...15...20...25...30...35  <= ruler
 1      10   20   302.71 1.3     X         k1 i1
 2      56   22   707.24 4.4     R         k1 i2
 3      15   70   333.63 7.0     E         k2 i1
 4      15   80   408.26 1.3     X         k2 i2
 5      40   20   808.89 1.9     V         k3 i1
 6      40   90   902.20 9.3     X         k3 i2
=================================================
```

Note that the first real number - 2.71 - will be read properly even though it is glued to the integer before it - 30 - because of the precise description of the record by the input format. Note furthermore that also 1.3 will be read in properly because the period in the record takes precedence over the specification in the format description. However the field length must be exactly correct. Note finally that only 34 columns (5+3*5+2*4+5+1) are described by the format so that the information in columns 35 and higher will be ignored.

6.6 Conversion to ASCII or capitals

Introduction
In the beginning when the programs were designed all kinds of surprises confronted program designers. This option was designed to overcome them.

Details
Given a printer which cannot handle the extended ASCII character set used in the program, one may choose to convert these characters to basic ASCII by activating this option.

Initially some printers in Japan used the higher bit ascii codes for hiragana while the program intended to print lower-case letters.. This gave rather unreadable output for Japanese and gayin alike. To circumvent this a provision was created to transform the output in capitals only.

(These options are rather archaic and have therefore been disabled, but they can be made available upon request.)

7 Generic buttons

Introduction
Several buttons appear on virtually ever screen and have the same function everywhere. They are listed here separately.

Default
In several screens a `Default` button has been provided, clicking on which will set default values for the relevant options. In this way a simple standard analysis can be run. However, some options, such as the data specifications and the number of components for each way, generally need intervention from the user to obtain a meaningful solution.

Clear
By pressing the `Clear` button all options are removed from the active screen. Mostly it makes sense to have it followed by pressing the `Default` button as a start for filling in the required options.

Continue

All data entered are consolidated in memory before returning to the previous screen.

Copy Screen
The Copy `Screen` button was primarily used during the development of the program, but it could be functional in those cases in which the user wants to report an error.

8 References

Arbuckle, J., & Friendly, M.L. (1977). On rotating to smooth functions. *Psychometrika, 42*, 127-140.

Carroll, J.D., & Chang, J.J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition.*Psychometrika, 35*, 283-319.

Clarkson, D.B., & Jennrich, R.I. (1988). Quartic rotation criteria and algorithms. *Psychometrika, 53*, 251-259.

Harshman, R.A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, *16*, 1-84.

Lundy, M. E., Harshman, R. A., & Kruskal, J. B. (1989). A two-stage procedure incorporating good features of both trilinear and quadrilinear methods. In R. Coppi & S. Bolasco (Eds.), *Multiway data analysis* (pp. 123-130). Amsterdam: Elsevier.

Kiers, H. A. L. (1998). Joint orthomax rotation of the core and component matrices resulting from three-mode principal components analysis. *Journal of Classification, 15*, 245-263.

Kiers, H. A. L., & Ten Berge, J. M. F. (1994). Hierarchical relations between methods for simultaneous component analysis and a technique for rotation to a simple simultaneous structure. *British Journal of Mathematical and Statistical Psychology*, *47*, 109-126.

Krijnen, W.P. (1991). *The analysis of three-way arrays by constrained Parafac methods*. Leiden, The Netherlands: DSWO Press

Kroonenberg, P.M. (2008). *Applied multiway data analysis*. Hoboken, NJ: Wiley.

Kruskal, J.B. (1976). More factors than subjects, tests and treatments: An indeterminacy theorem for canonical decomposition and individual scaling. *Psychometrika, 41*, 281-293.

Smilde, A.K., Bro, R., & Geladi, P. (2004). *Multi-way analysis: Applications in the chemical sciences.* Chichester, UK: Wiley.

Timmerman,M. E. (2001). *Component analysis of multisubject-multivariate longitudinal data.* Unpublished doctoral dissertation, Department of Psychology, University of Groningen, Groningen, The Netherlands. (*http://irs.ub.rug.nl/ppn/217649602).*

Tucker, L.R (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika, 31,* 279-311.

Tucker, L. R (1972). Relationships between multidimensional scaling and three-mode factor analysis. *Psychometrika, 37*, 3-27.

Velleman, P. R. (). *Computational methods for the social sciences*.

Weesie, J., & Van Houwelingen, J. (1983). *GEPCAM User's manual*. Unpublished report. University of Utrecht.(Available from The Three-Mode Company).

9 Glossary

All-components plot. A plot in which the values of all components of a single mode are plotted against their sequence numbers or labels. Especially useful for ordered modes containing time series or spectra. Also called a one-way plot. In contrast with the usage in this book, other authors sometimes call this type of plot is a line plot, especially when only one component or variable is plotted.

Array. Multiway analogue of a vector and a matrix. A vector is a single-subscripted one-way array, a matrix is a double-subscripted two-way array. A multiway array has three or more subscripts.

Aspect ratio. The aspect ratio of an image is its displayed width divided by its height. In component plots it is important that this ratio is 1, for distances and inner products to be correctly displayed.

Asymmetric scaling. As used in optimal scaling or biplot construction, the coordinates of the row and column markers in biplots are asymmetrically scaled when one of the marker sets is in normalized coordinates whereas the other is in principal coordinates. In principle, but not in practice, other types of asymmetric scalings of the coordinates can occur. See also *Symmetric scaling*.

Biplot. A biplot of a matrix of I objects by J variables is used to display the systematic patterns between rows, columns, and between rows and columns. The prefix *bi* refers to the simultaneous display of both rows and columns of the table, not to the dimensionality of the plots. The singular value decomposition is used to derive the coordinates on the dimensions.

Bootstrap. A bootstrap analysis is a nonparametric procedure used to estimate what would happen if new samples were taken and the analysis repeated; frequently, it is used to compute standard errors for parameters in a model. Starting from the assumption that the sample distribution is the best available estimate of the population distribution, repeated samples with replacement are taken from the sample distribution. The desired model is calculated for each of these samples, so that variations in their parameter estimates can be used to estimate confidence intervals.

Centring. Subtraction of a constant term, mostly the mean, from raw data. However, sometimes a neutral point on a scale is used. The most common kinds of centring are: (1) fibre centring: subtracting the mean resulting from averaging across a single mode (i.e., the mean of a single-subscripted subarray) and (2) slice centring: subtracting the mean resulting from simultaneously averaging across two modes (i.e. the means of a double-subscripted subarray); (3) double centring: subtracting the two means resulting from two fibre centrings (in any order). Generally fibre centring is preferred.

Combination-mode (ij). Cartesian product of two (observational) modes i and j; "i outer loop, j inner loop"

Core array. A multiway array that contains the linkage information between components from different modes. The size of a full three-way core array is $P \times Q \times R$, where P, Q , and R are the numbers of components of the three modes, respectively. A core array is superdiagonal if it is a (hyper)cube and only the elements with the same index, that is, g_{sss}, are nonzero. If all superdiagonal elements are equal to 1, the core array is a superidentity array. A multiway core array is slice diagonal if only the elements with two equal indices, e.g., g_{ssr}, are nonzero; this concept is mostly used for three-way arrays. See also *Extended core array*.

Core consistency. A Parafac model shows core consistency, if the core array calculated using its components has a Superdiagonal structure, or nearly does.

Correspondence analysis. A technique to analyze the dependence in contingency tables with nonnegative numbers. The two-way version is standard, three-way versions are rare.

Degeneracy. A solution of a Parafac model is called degenerate if two or more components are

becoming (nearly) congruent. A solution is divergent degenerate if the algorithm to calculate the solution produces one or more of the parameter estimates that increase or decrease without bound, and the fit of the solution can always be improved by increasing the number of iterations and the numerical accuracy. A solution is bounded degenerate if it is an "unacceptable" solution with highly congruent components, notwithstanding convergence of the algorithm; often from multiple starting points. Sometimes solutions with temporarily highly congruent components are also called temporary degenerate.

Deviance plot. A plot with the deviance or residual sum of squares plotted against the degrees of freedom. Models that are candidates for interpretation lie on or near the convex hull. Detailed inspection of the models on the convex hull can aid in model and dimensionality selection. See also *Multiway scree plot*.

Display mode. See *Joint biplot*.

Double centring. See *Centring*.

Extended core array. A core array, most commonly three-way, in which one of the ways is not condensed into its components. An extended three-way core array is slice-diagonal if the slices featuring the two condensed ways are diagonal.

Fibres. A fibre is a one-dimensional subarray or vector indexed by all but one of the indices of the full array. For three-way data the fibres are vectors with two subscripts, in particular, rows (x_{ik}), columns (x_{jk}) and tubes (x_{ij}). In the four-way case the fibres are: x_{ikl}, x_{jkl}, x_{ijl}, and x_{ijk}; the latter are also called pipes.

Fortran input format. A formal description of the content of a record using conventions defined by the Fortran language. Separate codes are available for integers, reals, characters, and columns to be skipped.

Fully crossed multiway data. Multiway data for which in principle all possible combinations of levels from all ways exist. Thus, multiset data do not fall into this class.

Individual differences scaling: INDSCAL**.** A model for sets of symmetric dissimilarity or similarity matrices. The space of the stimuli, which constitute the first and second way, is shared by the levels of the third mode (judges, subjects, etc.). The axes of their individual spaces are arbitrarily stretched or a shrunken versions of the common axes.

Individual differences in orientation scaling: IDIOSCAL. A model for sets of symmetric dissimilarity or similarity matrices similar to indscal. In this model, however, the common space may also be rotated before the shrinking or stretching of the axes.

Inner product. The inner product of two vectors produces a scalar: if $\boldsymbol{y}$ and $\mathbf{x}$ are two vectors, then their inner product in matrix notation is $z = \mathbf{y}'\mathbf{x}$. It is equal to the sum of the products of the corresponding elements: $\sum \mathbf{y}_i \mathbf{x}_i$. Geometrically, it is the product of the lengths of the vectors times the cosine of the angle between them. This product is frequently used in interpreting biplots.

Jackknife. A jackknife estimator is based on systematically recomputing a statistical estimate leaving out one or more observations at a time from the sample. From the set of values for the statistic, an estimate for the bias of the statistic can be calculated, as well as an estimate for the variance of the statistic over repeated sampling. It is closely related to, and often used for the same purposes as, the bootstrap.

Joint biplot. In three-mode analysis, a joint biplot is a biplot for two of the modes (the *display modes*) conditional on a component of the third mode (the *reference mode*).

Latent covariance matrix. For the Tucker2 model, the latent covariance matrix contains the (co) variances of the scores of the subjects on the latent-variable--prototype-condition combinations. Depending on the scaling, the covariances may be cross products, "real" covariances, or correlations. For the Tucker3 and Parafac models, this matrix contains the variances and

covariances between the scores of the 'idealized subjects' on the latent-variable--prototype-condition combinations.

Levels. Levels are the units into which a mode is divided, each represented by a distinct value of the mode's subscript (e.g., rows are the levels of Mode A). "Level" is often used as a generic term for the entities in a way or mode, such as a variable, a subject, a crop variety, and so on.

Loadings. In two-way analysis loadings are regression coefficients relating the levels of a given mode to the components extracted from that mode. Some people restrict the definition to coefficients relating levels of the "variables" mode to its components; or even further to regression of orthogonal components on standard score variables, i.e., to variable--component correlations. Such conventions are less common in multiway contexts and/or outside of the social and behavioral sciences.

Matricisation. Matricisation is the stringing out of an *I x J x K* array into a two-way matrix, mostly of the order *I* by *(J x K)*. Also referred to as stringing-out, juxtapositioning, and, especially in chemometrics, unfolding. The use of this last term, however, is now discouraged, both in chemometrics and elsewhere.

Maximum-product rule. The maximum-product rule states that if the size of one of the ways in a three-way data set is larger than the product of the other two, the largest way can be reduced without loss of information essential for the components of the two other ways and the core. Specifically, when I > *JK*, the data can be reduced to a *JK x J x K* data set. Moreover, ignoring restrictions on the parameters, the number of free parameters in the largest mode is reduced from *I x P* to *JK x P*.

Means plot.A means plot is a plot in which the values of a single variable (second mode) are indicated on the vertical axis and the conditions (the third mode) are marked on the horizontal axis. In the plot lines connect the condition means of the groups that are constituted on the basis of a cluster analysis.

Minimum-product rule. The minimum-product rule states that for a Tuckern model the product of the numbers of components of n-1 modes must always be equal or larger than that of the remaining mode, so that for the Tucker3 model $P \times Q \geq R$, $P \times R \geq Q$,and $Q \times R \geq P$.

Mode.The term "mode" refers to any one of the indexed directions of elements of an array, such as in "Mode A corresponds to rows" or "Mode B is the Subject mode". "Mode" is typically used as a synonym for "way", but the term has fewer alternative meanings. There is also a more specialized definition, by which mode means "type of classification". However, the restricted definition is still less common. See also *Way*.

Multimode covariance matrix. A patterned covariance matrix in which one or more modes are nested in another mode, and the modes are fully crossed. Multitrait--multimethod matrices are two-mode covariance matrices in which the traits and methods are fully crossed. When no normalization has been carried out, the matrix is also referred to as multimode cross product matrix.

Multiway scree plot. A plot of the deviance or residual sum of squares against the sum of the numbers of components of the modes. This plot is the multiway variant of Cattell's scree plot. Inspecting the models on the convex hull may aid model and dimensionality selection.

Murakami form. The form of the Tucker2 model ($\mathbf{AH}_k\mathbf{B}'$) in terms of first-order component loadings **B**, second-order component loadings $\mathbf{H}_k$, and second-order component scores **A**. A Tucker3 variant has been formulated as well.

Murakami core. When in a Tucker3 model the product of the number of components of two modes -1 is equal to the number of components of the third mode, the associated core array is called a Murakami core. When $QR-1=P$ (or $PR-1=Q$ or $PQ-1=R$), and $Q \geq R$ (as can be done

without loss of generality) the core array has exactly $R(Q+R-2)$ nonzero elements, provided the mode with the largest number of components is in principal coordinates.

Natural weights. A term used for the relative weights of the modes in simultaneous rotations of components and (three-way) core arrays.

Nested-mode biplot. A biplot in which the row markers are fully crossed combinations of two modes, so that one mode is nested within the other (the index of the nested or inner mode runs fastest, and that of the nesting or outer mode slowest). The column markers are the levels of the of the remaining or reference mode. The plotting space is that of the reference mode. These plots are also called interactive biplots.

Normalization. The process of equalizing the sums of squares of a subarray of a multiway array. For three-way profile data the normalization is mostly done per variable slice, so that the sum of squares of the values in a slice is equal to 1.

Normalized coordinates. A component is in normalized coordinates if it has a length equal to 1. See also Standard coordinates.

Orthogonal. Two vectors are orthogonal if they are perpendicular to each other irrespective of their lengths. An $I \times J$ matrix $\mathbf{X}$ is (column-wise) orthogonal if $\mathbf{X'X} = \mathbf{\Lambda}_J$, where $\mathbf{\Lambda}_J$ is a $J \times J$ diagonal matrix with arbitrary nonnegative numbers.

Orthonormal. Two vectors are orthonormal if they are orthogonal and have length 1. An $I \times J$ matrix $\mathbf{X}$ is (column-wise) orthonormal if $\mathbf{X'X} = \mathbf{I}_J$, where $\mathbf{I}_J$ is a $J \times J$ diagonal matrix with ones and possibly some zeroes on the diagonal. In mathematics the name "orthogonal matrix" is sometimes restricted to mean a square orthonormal matrix, with non-square matrices described as "columnwise" orthogonal or orthonormal.

Outer product. The outer product of two vectors produces a matrix: if $\mathbf{y}$ and $\mathbf{x}$ are vectors, then their outer product is the matrix $\mathbf{Z} = \mathbf{yx'}$

Paired-components plot. A plot in which the components of a mode are plotted against one another. They are especially useful to examine the spatial arrangement of the plotted points. Such plots need to have an aspect ratio of one.

Parafac model. The Parafac (parallel proportional profiles) model is a multilinear model with components for each of its modes and a superdiagonal core array. It specifies a parallel weighting of the components of one way by those of the other ways. The Parafac model is sometimes contrasted with the "Parafac decomposition" which refers to the mathematical decomposition of an array into a sum of rank 1 subarrays, analogous to the singular value decomposition. In this latter sense it is equivalent to the canonical decomposition CANDECOMP.

Parallel solution. A solution of a Parafac model with two proportional columns in any of the component matrices

Per-component plot. A plot that contains one component of each mode. Such a plot is especially useful for the Parafac models, in which components are linked exclusively to each other. Another term for such plots is across-modes plot.

Principal coordinates. If a component is in principal coordinates, its length is equal to the singular value, which is equal to the square root of its eigenvalue.

Profile data. Data in which each subject is considered to have scores for a set of variables; these scores are called the profile of a subject.

Preprocessing. The application of procedures to a multiway data set before a multiway model is fitted; especially centring, normalization, and standardization.

Postprocessing. The application of procedures to the estimated parameters of a multiway model to enhance interpretation, such as transformations, rotations, and scalings of components and/or core array.

Reference mode. The column mode for the nested-mode biplot, or the mode conditional on which joint biplots are made. See also *Joint biplot* and *Nested-mode biplot*.

Relative fit. Relative fit is an abbreviation for "relative fitted sum of squares" and is the fitted sum of squares divided by the total sum of squares. The term applies to specific parts of a model such as the levels of a mode. It is equal to 1-Relative residual sum of squares.

Relative residual sum of squares. Residual sum of squares divided by the total sum of squares. The term applies to specific parts of a model such as the levels of a mode.

Replicated PCA. A Tucker3 model with only one component in one of the modes. It results in a single diagonal core slice, given the orthogonality of the component matrices. Also called Weighted PCA

Scores. A name given to component coefficients for subjects, objects, etc.

Singular value decomposition. The decomposition of a matrix $\mathbf{X}$ into a left set of orthonormal vectors $\mathbf{U}$, a right set of orthonormal vectors $\mathbf{V}$, and a diagonal matrix $\mathbf{\Lambda}$ of singular values, which are the square roots of the eigenvalues of both $\mathbf{X'X}$ and $\mathbf{XX'}$. Thus $\mathbf{X} = \mathbf{U\Lambda V'}$. The svd represents the basic structure of a matrix and forms the basis of the generalization to methods for multiway arrays.

Slices of an array. For three-way data a slice is matrix of values with a single subscript: frontal slices $\mathbf{X}_k$, lateral slices $\mathbf{X}_j$, and horizontal slices $\mathbf{X}_i$. For *N*-way data a more general definition is: two-way subarrays of an array, that is, the set of elements defined by keeping all but two subscripts fixed and allowing the two that vary to take on all the values in their range. Slices are sometimes also called slabs.

Standardization. Combination of centring and normalisation.

Standard coordinates. A component is in standard coordinates if it has a mean of 0 and a length or variance equal to 1.

Standardised (component) weights. The eigenvalue of a component divided by the total sum of squares. Thus for PCA, SS(Fit)$_s$/SS(Total) is the standardised weight of component *s*. In the Parafac model these quantities are g_{sss}/SS(Total). In the Tucker models the weights are SS(Fit)$_v$/ SS(Total), where v = *p, q*, or *r*.

Sums-of-squares plot. A plot with the residual sum of squares on the vertical axis and the fitted sum of squares on the horizontal axis. These plots are used to assess the extent to which the data of the levels of a mode are fitted by the multiway model.

Superdiagonality. A core array is superdiagonal if it is a (hyper)cube and only the elements g_{sss} or $g_{sss..}$ are nonzero. The term body diagonality has also been used in the past. See also Core array.

Symmetric scaling. As used in optimal scaling or biplot construction, the coordinates of row and column markers are symmetrically scaled when neither of the marker sets has normalised or principal coordinates, but the scale factor is equally divided between them. The emphasis is on assessing the inner products between row and column markers, rather than the relationships among the row or column markers. The inner products are the approximations of the values in the data table. See also Asymmetric scaling.

Tall combination-mode matrix. A two-way matrix resulting from matricising a three-way array such that the matrix has *K x I* rows and *J* columns. Thus, it is a partitioned matrix whose elements are themselves matrices.

Three-mode scree plot. See *Multiway scree plot*.

Tucker models. Multiway models in which component matrices are specified for at least one of the ways, and in which the relationships between components from the ways are represented in a core array. A Tucker1 model is an ordinary PCA model but applied to a tall combination-mode matrix; a Tucker2 model has components for two of its ways with an extended core array, a

Tucker*n* model has components for n of its ways. Generally, Tucker*n* models are applied to *n*-way data arrays.

Way. The ways of an array correspond to the individual indices used to reference elements in the array. The term 'way' is adopted from similar applications in analysis of variance. Carroll and Arabie have proposed a distinction between way and mode. In particular, their proposal entails that if the contents of the ways are different, say, subjects, variables, and conditions, the (three-way) array is said to have three modes. If it has two ways with the same content, the array is said to have two modes and three ways. See also *Mode*.

Wide combination-mode matrix. A two-way matrix resulting from matricising a three-way array, such that the matrix has *I* rows and *K x J* columns.

Index

- . -

- _ -

- 3 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -

- W -

- Z -

This manual refers to the Windows version of 3WayPack, a suite of programs and utilities for three-way data analysis.

The theory behind the models in the program suite is explained in A.K. Smilde, R. Bro, & P. Geladi (2004) and P.M. Kroonenberg (2008).

Version date: 26/09/2010