KUNST

UNFOLD

Non-metric multidimensional unfolding

By Maria Thissen and
Nol Bendermacher

RTOG-FSW
Radboud university Nijmegen

July 2012

1 Table of content

2 General description

UNFOLD performs a non-metric multidimensional unfolding analysis.

If people are asked how much money they might want, their preference will grow probably with the amount: the more the better. But if they are asked how much sugar they want in their coffee, everybody has its own preferred amount: less sugar may be wrong but more sugar may also be wrong. In the latter case the model of unfolding may be useful.

The notion of *unidimensional* unfolding originates from Coombs (1969). It supposes that there is a scale (a straight line) along which the stimuli (e.g. several cups of coffee) are located and that each test subject (or case) has its own "ideal point" on this line. It is assumed that the subjects agree on the positions of the stimuli on the scale.

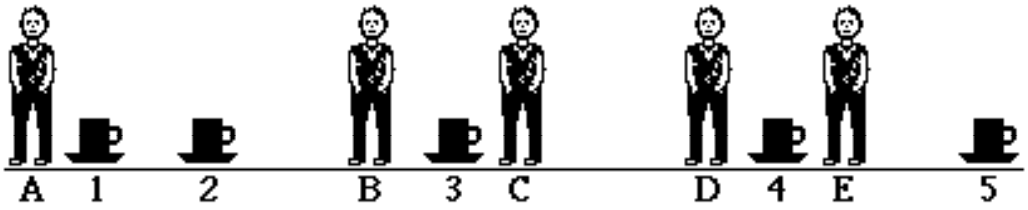

Figure 1: Stimuli and cases on a one-dimensional scale

Figure 1 shows a line with 5 cups of coffee, different in their amount of sugar, and the ideal points of 5 subjects. If the subjects are asked to order the stimuli according to their preference each will give its own order:

A:	1	2	3	4	5
B:	3	2	1	4	5
C:	3	4	2	1	5
D:	4	3	5	2	1
E:	4	5	3	2	1

One finds an individuals ranking by taking the line at his/her ideal point and folding it (see figure 2).

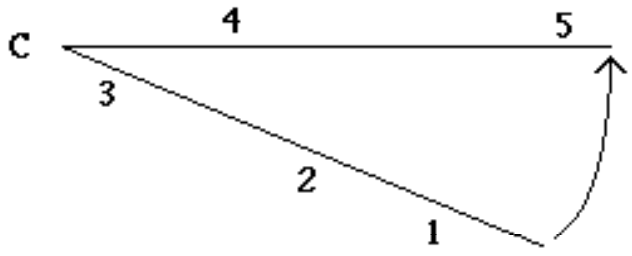

Figure 2: A folded line

It is impossible to reconstruct the original line if only one folded line is given. But with different preference orders from different subjects it is possible to find more or less accurately the original locations of stimuli and subjects, if the assumption holds that all subjects agree on the places of the stimuli. And that is what unfolding does: reconstructing the positions of stimuli and cases from the preference orders as they are given by the cases. UNFOLD, however, allows the use of a more-dimensional space. It is especially useful if one wants to find out what are the relevant characteristics that a group of subjects uses to determine their preferences among several stimuli.

3 Unfolding in detail

Consider a set of cases and a set of stimuli where the cases indicate their preferences for the stimuli (the judgments need not to express preferences; any asymmetric relation is acceptable). The aim of the program is to position both stimuli and cases as points in a space of minimum dimensionality such that, for each case, the distances to the stimuli reflect the order of the preferences as closely as possible. The point that corresponds to a case is called its "ideal point". That is to say that each case, or "judge" is represented in the solution space as a point positioned at its point of maximum preference. The stimuli are also represented by points in the same space such that a stimulus that is nearer to a case is more preferred by that case.

Following the terminology developed by Carroll and Arabie (1980) **UNFOLD** may be described as:

Data:	Two-mode	Model:	Euclidean distance incorporating
	Two-way		Two sets of points in
	Ordinal		One space
	Row conditional		The solution is internal
	Complete or incomplete		
	One replication		

3.1 The model

The program takes data of different form as described in chapter 5 and seeks to position both sets of objects (cases and stimuli) as points in a space of minimal dimensionality. The cases are positioned at their points of maximum preference: their ideal points. For each case the distances to the stimuli will reflect the order of preference as revealed by the data: the most preferred stimulus will be represented by the nearest stimulus point to a case's ideal point, the least preferred the farthest away.

Strictly speaking, this will hold only if a perfect solution is found. Otherwise some inversions will occur.

The number of dimensions of the solution space can be chosen by the user. Since one may not know what the appropriate number should be, the program offers the opportunity to specify a range of dimensionalities, for instance 1 to 3. UNFOLD will first search for a solution in the largest number of dimensions and then reduce the number of dimensions step by step until the fit becomes too poor to go on.

There is also some freedom in the choice of the distance formula to be used. The everyday's or Euclidean formula for the distance between two points P and Q is:

$$D_{ij} = \sqrt[2]{\sum_{d=1}^{k} \left(P_{id} - P_{jd}\right)^2}$$

where P_{id} is the coordinate of point i on dimension d.

But there is a more general formula where the constant 2 is replaced by a so-called Minkowski parameter m:

$$D_{ij} = \sqrt[m]{\sum_{d=1}^{k} \left(P_{id} - P_{jd}\right)^m}$$

In UNFOLD one can choose any Minkowski parameter between 0 and 15 as long as it is positive. Even non-integer values are allowed. It is however not advised to choose a value less than one, since then the theorem of the triangle inequality does not hold anymore: $d_{xy} + d_{yz} \geq d_{xz}$

Three of these distance parameters are of special interest:

city block metric: If the Minkowski parameter is 1 the distance between two points is just the sum of the differences between their corresponding coordinates. The metric owes its name to the fact that it is the distance to walk if one is forced to follow a rectangular street pattern in a modern city (and the streets are parallel to the coordinate system).

Euclidean metric: This is the common metric with Minkowski parameter 2. The coordinate differences are weighted according to their size, so large differences are more important than small ones.

dominance metric: As the Minkowski parameter increases, the distance between two points will come closer and closer to the largest coordinate difference, thereby outruling all other coordinates. Therefore the metric with an infinite Minkowski parameter is called a dominance metric.

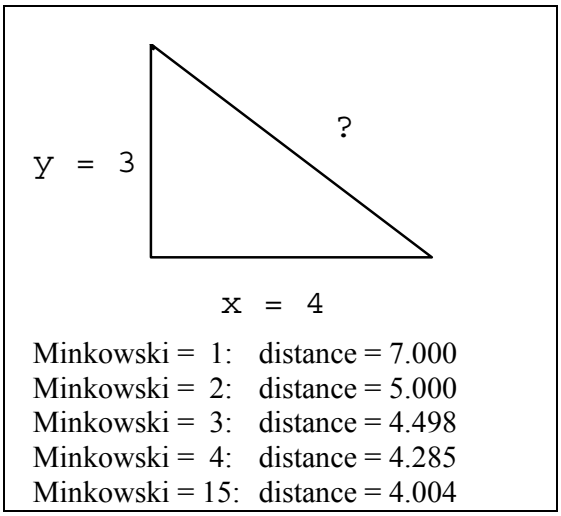

Figure 3: Distance in a two-dimensional space dependent on the Minkowski parameter

Figure 3 shows two points with coordinate differences 4 and 3 and their distances corresponding to several choices of the Minkowski parameter.

UNFOLD does not require that the cases and the stimuli can be represented perfectly in the required space. If they do not, it tries to find a solution that is as good as possible. It uses a loss- or stress function to evaluate the fit between the order in the distances to the preferences in the data. It starts from a provisional map (the trial configuration) and then moves the stimuli and the cases in small steps trying to minimize this stress-function.

3.2 Fitting values and stress

In order to measure the quality of a solution, intermediate values are computed between the distances in the configuration (distances between the cases and the stimuli) and the preferences in the data. These intermediate values stay as close to the distances as possible, but they are not allowed to violate the order in the data. The intermediate values are called *fitting values*. They are weakly monotonous with the data. The quality of a solution is measured from the differences between the distances (D) and the fitting values (F) according to the formula:

$$S_j^2 = \frac{\sum_{i=1}^{n} (D_{ij} - F_{ij})^2}{\sum_{i=1}^{n} (D_{ij} - \overline{D}_j)^2} \qquad (1)$$

where S_j^2 is the squared stress of case j, n is the number of stimuli and $\overline{D}_j$ is the mean distance for case j.

The overall stress S is the square root of the average of the individual squared stress values. For a perfect solution its value is zero.

$$S = \sqrt{\frac{1}{n} \sum_{j=1}^{n} S_j^2} \qquad (2)$$

where n is the number of cases.

There are two possible definitions of the fitting values:

$\hat{d}$ or DHat: he values are chosen such that the enumerator of the stress per case in (1) is minimal. They are called monotone regression values (Kruskal, 1964a). These are the fitting values that are used in the final stage of the minimalization process, since $\hat{d}$ is used in the function the program finally wants to minimize. The differences between the distances and these fitting values are as small as possible according to the least squares criterion.

d^* or DStar: The values are obtained by sorting the distances and the data within each case and matching them according to this order. They are called rank image values. These fitting values are used at the start of the minimalization process.

3.3 The treatment of ties

A tie in the data means that some of the stimuli are preferred equally by a case. Tied stimuli can be handled in two ways: the program can allow them to have different distances to the case (the *primary approach*) or it can try to keep their distances to the case equal (the *secondary approach*). If the primary approach is used we say that *ties may be broken*. The secondary approach will in general lead to a higher stress, since differences between distances for tied data will contribute to the stress.

In the program the primary approach is realized by sorting tied elements according to the order in their distances before the computation of the fitting values.

The secondary approach tries to represent tied data by equal distances. Therefore fitting values within a tie are replaced by their average value.

3.4 Versions of stress

UNFOLD will adjust the configuration step by step in order to decrease the value of S. To improve the configuration, the method of steepest descent is used. This procedure is iterative. For each case coordinate P_{ik} and for each stimulus coordinate Q_{jk} in the configuration we compute a gradient value G_{ik} (and H_{jk}) in such a way that the stress decreases most rapidly when a new configuration is formed by:

$$P_{t+1} = P_t - \alpha_t G_t \text{ and } \quad Q_{t+1} = Q_t - \alpha_t H_t$$

with a small step size α; t is the iteration count. In practice α is chosen according to the method described by Kruskal (1964a).

The success of the algorithm can be expressed by several versions of the stress-function. The program computes the following results:

Stress DHat: the stress S with $\hat{d}$ as the fitting values.

Stress DStar: the stress S with d^* as the fitting values.

3.5 The main algorithm

The main algorithm performs the following steps:

a The data within each case are ordered by preference.

b If the user does not provide a trial configuration, the program generates one. The algorithm is based on a principal components analysis of a matrix related to Guttman's C-matrix (see 0). The formula for the fitting values is set to d^* since it works faster and has less chance to lead to a local minimum or a trivial solution (see 3.7).

c (Phase 2:) The configuration is normalized such that the average of the coordinates for the stimuli is zero for each of the dimensions and the sum of squares of the stimulus coordinates is equal to the number of stimuli. From this configuration the distances between the cases and the stimuli are computed. From the distances the fitting values (first d^*, later on $\hat{d}$), the stress per case and the overall stress are computed.

d Now a decision has to be made if the process must go on and how. The decision depends on a number of criteria. These criteria are described in the next section (see 2.7). If the process ends while d^* is used, the formula for the fitting values is changed to $\hat{d}$; otherwise, the iteration process stops (continue with step f). After the change of the fitting-formula the process continues from the best configuration thus far (possibly not the most recent one). Even if the stress increases, the process continues a few steps (never more than 4) to see if the increase is just temporary.

e (Phase 1:) The configuration is adjusted by moving each point a small step such that the stress probably decreases (using partial derivatives of the stress with respect to the configuration). This adjustment is performed in a number of steps (at most 5 if d^* is used and only 1 if $\hat{d}$ is used).

Go on with phase 2 (step c).

f At the end of the process the final configuration is normalized and, if the metric is Euclidean, the configuration is rotated to an principal axes orientation.

3.6 Evaluation of the stress

To decide if the iteration process must end, the following criteria are evaluated in the order mentioned:

- Stop if the stress is less then the criterion value given by the user.
- Stop if the stress is more or less stable, i.e. if the new stress differs very little from the best stress so far:

$$\left|\frac{S_{best} - S_{new}}{S_{new}}\right| < 10^{-8}$$

- Stop if the maximum number of iterations has been reached.
- Stop if the stress is deteriorating (the best solution so far was found more than 5 iterations ago), unless the minimum number of iterations is not yet reached.

3.7 Trivial solutions

The requirement that the distances of a case to all stimuli have the same order as the data can be met very easily by making them all equal. The easiest way to do so is to give all stimuli the same position. But, by its denominator, the stress-formula counteracts a movement towards such a solution. Nevertheless, it can happen that at least groups of points fall together or come very close to each other. Also UNFOLD may construct groups of points with almost equal distances to each other. A configuration that suffers from such a defect is called a *trivial solution*. There is no general way to prevent trivialization. One may try another trial configuration (or a start in more dimensions).

Sometimes a case cannot be located in the configuration because of its peculiar order of preferences. Such a case may be pushed towards infinity by the other cases and by the stimuli. If this happens the program will stop after it has shown the last configuration. The best solution may be to remove the difficult case and to redo the analysis.

Occasionally you may have quite a clear idea how the cases and stimuli should be arranged in the space, while the program generates a completely different solution. Then you can provide your own trial configuration with coordinates as close as possible to what you suppose they should be.

3.8 The trial configuration

By default the trial configuration will be generated by the program.

It may be helpful to start in a large number of dimensions and then to use the coordinates on the first dimensions of the solution as the trial configuration in fewer dimensions. If the Euclidean metric is used the program will perform a principal axes rotation on its intermediate solutions and then drop the last dimension. Starting from a higher dimensionality reduces also the chances of landing in a local minimum.

Although the program can generate a trial configuration, the user may wish to define his own. When you present a trial configuration of your own, but not all elements are filled, the program will generate a complete trial configuration and then replace the appropriate elements with the given values. The given and/or generated configuration will be normalized and principal axes will be taken before usage.

With a configuration of your own you can also fix some cases or stimuli in that configuration or even specific coordinates within a case or stimulus. This means that these elements will keep there values, apart from rescaling and/or normalisation. This is useful when, from a previous analysis, you know the positions of some stimuli and you perform an analysis with other cases to see where they fit in.

3.9 Patterns

If the number of cases is large as compared to the number of stimuli, there will probably be groups of cases with the same order of preferences. Such a group is called a pattern. In the input data the group can be represented by one single row with a frequency count. This option not only makes the preparation of the data easier but it will also speed up the computations.

4 Technical details

Define:

n = The number of cases

n^* = the number of patterns; $n^* \leq n$

m = number of stimuli

i = index for pattern

j = index for stimulus

k = index for dimension

f = vector with (n^*+m) frequencies, such that $n = \sum_{i=1}^{n^*} f_i$

$f_j = 1$ for all $j > n^*$

If the data are not compressed we have:

$n^* = n$

$f_i = 1$, for $i = 1,...,n+m$

d = number of dimensions in the configuration

P = $n^* \times d$ configuration of the patterns

Q = $m \times d$ configuration of the stimuli

n_i = number of stimuli specified in row i of the data matrix

u = The Minkowski parameter

D = $n^* \times m$ matrix of distances between patterns and stimuli

$$D_{ij} = \sqrt[u]{\sum_{a=1}^{k} \left| P_{ia} - Q_{ja} \right|^u}$$

F_{ij} = The fitting value corresponding to pattern i and stimulus j

S_i^2 = The squared stress of pattern i:

$$S_i^2 = \frac{T_i}{N_i} = \frac{\sum_j \left(D_{ij} - F_{ij}\right)^2}{\sum_j \left(D_{ij} - \overline{D}_i\right)^2}$$

S = The overall stress:

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^{n^*} S_i^2 f_i}$$

4.1 Adaptation of the configuration

The adaptation is based on the partial derivatives of the overall stress S with respect to the cases and with respect to the stimuli. In the first stages of the analysis $F = d^*$ is used; in the final stages $F = \hat{d}$ is used.

UNFOLD uses the partial derivatives:

$$G_{ik} = \frac{\partial S}{\partial P_{ik}} = \frac{1}{S} \sum_j \left| \frac{P_{ik} - Q_{jk}}{D_{ij}} \right|^{u-1} \operatorname{sign}\left(P_{ik} - Q_{jk}\right) \frac{D_{ij} - F_{ij} - S_i^2 \left(D_{ij} - \overline{D}_i\right)}{n_i}$$

$$H_{jk} = \frac{\partial S}{\partial Q_{jk}} = \frac{1}{S}\sum_i f_i \left(\left| \frac{P_{ik} - Q_{jk}}{D_{ij}} \right|^{u-1} \operatorname{sign}\left(Q_{jk} - P_{ik}\right) \frac{D_{ij} - F_{ij} - S_i^2\left(D_{ij} - \overline{D}_i\right)}{n_i} \right)$$

Using initial estimates $P_{ik.0}$ and $Q_{jk.0}$ for P_{ik} and Q_{jk} and a suitable value for α one may iteratively compute the right part of the equations and assign it to the left part; the stress will thereby converge towards a minimum (local or global).

$P_{ik.t+1} = P_{ik.t} - \alpha_{ik.t} G_{ik.t}$ and $Q_{jk.t} = Q_{jk.t} - \alpha_{jk.t} H_{jk.t}$

For α the program uses

$$\alpha_{ik.t} = \frac{\beta_t}{\frac{1 - bS_i^2}{n_i} \sum_{j=1}^{ni} W_{ijk.t}} \text{ and } \alpha_{jk.t} = \frac{\beta_t}{\sum_{i=1}^{n^*} \frac{1 - bS_i^2}{n_i} f_i W_{ijk.t}}$$

where b = 0.25 when d^* is used and b = 1 when $\hat{d}$ is used and where

$$W_{ijk} = \left(\frac{\left|P_{ik} - Q_{jk}\right|}{D_{ij}} \right)^{u-2}, \beta_0 = 1 \text{ and } \beta_t = 4^{(q^3)} \beta_{t-1}^{\frac{1}{3}}$$

q is the correlation between successive gradients t and t-1 (the cosine of the gradient angle). The use of q is to see whether the last step is too large or too small by looking at the angle between the successive gradients. If the gradients point in almost the same direction (q is close to 1) the step was too small. If the new gradient points back almost in the direction one came from (q is close to -1) the last step was too large.

For the dominance metric $W_{ijk} = 1$ if $P_{ik} - Q_{jk}$ is (one of) the largest coordinate difference(s) and $W_{ijk} = 0$ otherwise.

4.2 The generation of the trial configuration

Define:

α = $\dfrac{n + \frac{1}{2}(m+1)}{n+m}$

I_x = x × x identity matrix for any x

R_{ij} = rank of stimulus j for case i; tied stimuli receive their average rank.

D = m × m diagonal matrix with $D_{jj} = \dfrac{\sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{mi} R_{ij}}{n+m}$

V = n × m matrix with $V_{ij} = \dfrac{1}{n+m}\left(1 - \dfrac{R_{ij}}{n_i}\right)$

C = (n+m) ×(n+m) matrix: $C = \begin{bmatrix} \alpha I_n & V \\ V^T & D \end{bmatrix}$

The trial configuration will be based on the second and following eigenvectors of C and the corresponding eigenvalues. However, usually n will be much larger than m.
So for numerical robustness and speed it would be desirable to have a smaller matrix C^* and still to find the wanted eigenvectors and eigenvalues. Such a C^* can be found as follows.

For an eigenvector y and its eigenvalue λ we have:

$$\begin{bmatrix} \alpha I_{n^*} & V \\ V^T & D \end{bmatrix} \begin{bmatrix} y_{n^*} \\ y_m \end{bmatrix} = \begin{bmatrix} \lambda y_{n^*} \\ \lambda y_m \end{bmatrix} \quad (1)$$

with y_{n^*} = first n^* elements of y and y_m = last m elements of y.

This leads to two equations:

$$(1)\quad \alpha I_{n^*} y_{n^*} + V y_m = y_{n^*}\lambda \;\rightarrow\; y_{n^*} = \frac{1}{\lambda - \alpha} V y_m \quad (2)$$

$$(2)\quad V^T y_{n^*} + D y_m = y_m \lambda \quad (3)$$

Substituting the right part of (2) in (3) we find:

$$\left(V^T V y_m\right)\frac{1}{\lambda - \alpha} + D y_m = y_m \lambda \quad (4)$$

This is an eigenvalue-eigenvector equation for a m-vector y_m.

Now we use Cholesky triangularization to find an m × m matrix W, such that $W^TW = V^TV$ and replace in (1) V by W and I_{n^*} by I_m:

$$\begin{bmatrix} \alpha I_m & W \\ W^T & D \end{bmatrix} \begin{bmatrix} y^*_m \\ y_m \end{bmatrix} = \begin{bmatrix} \lambda y^*_m \\ \lambda y_m \end{bmatrix}$$

Solving this equation results in the values of y_m and λ; y_{n^*} can be found by (2)

A trial configuration in d dimensions is built from the second through (d+1)-th eigenvector-eigenvalue combinations from C as described above.

1) Let F_a be a $(n^*+m) \times d$ matrix containing the second through (d+1)-th eigenvector.

2) Each column j in F_a is standardized to a length equal to its corresponding eigenvalue:

$$x = \sqrt{\frac{\lambda_{j+1}}{\sum_{i=1}^{n^*+m} f_i F^2_{a(i,j)}}}$$

$$F_{b(i,j)} = x F_{a(i,j)}$$

3) If the user has specified parts or all of the trial configuration, these values will overwrite the corresponding values in F_b.

4) The stimulus configuration is centered: for each column j in F_b the mean m_j of its stimulus coordinates is computed:

$$m_j = \frac{1}{m} \sum_{i=n+1}^{n+m} F_{b(i,j)}$$

This mean is subtracted from all entries in column j; the result is F_c:

$$F_{c(i,j)} = F_{b(i,j)} - m_j \quad \text{for } i = 1, .. , n^*+m$$

5) The whole matrix F_c is normalized, with result F_d:

$$x = \sqrt{\frac{m}{\sum_{i=n+1}^{n^*+m} \sum_{j=1}^{m} F_{c(i,j)}^2}}$$

$$F_d = F_c x$$

6) If the number of dimensions is greater than one, F_d is rotated to the principal components of the stimulus configuration (i.e. the last m rows of F_d). The result of this rotation is the final trial configuration.

4.3 Monotone regression

The fitting values according to the monotone regression method are computed for each case as follows:

First a copy F of the distances of the case to all stimuli is taken, ordered by the corresponding data elements. If the stress is zero these values form an ascending order. Usually they do not. For the explanation in the sequel we will indicate the fitting values with one single subscript like F_j to indicate the fitting value corresponding to stimulus j within a certain case.

We start from the first fitting value F_1 and if the next one (F_2) is smaller we replace them both by their average. Then we check if F_3 is less than this average. If it is not, the average is *up-satisfied*; if it is, we replace the first three values by their average.

In general, for every F or average of F's we check if it is less than the preceding F or average of F's. If it is not, it is *down-satisfied*. We look also if it is larger than the next F or average of F's. If it is not, it is up-satisfied. If it is not down-satisfied we merge with the preceding F or average of F's and assign the average to the whole block. If it is not up-satisfied it is merged with the subsequent F or block and take the average.

If the secondary approach to ties is used (so ties in the data must be reflected by ties in the solution) the initial F-values within a tie are replaced by their average before the procedure starts.

As soon as an F or block of F's is down-satisfied and up-satisfied the procedure switches to the next F and so on until the end of the list is reached. The result is a series of increasing or at least non-decreasing fitting values that minimize the stress $\hat{d}$ for a certain case.

5 The data

UNFOLD takes data in a "row-conditional" format. In the simplest case, a group of n cases or patterns might be asked to rank a set of m stimuli in order of preference. The judgment may, of course, be a ranking (or rating) in terms of any suitable criterion of which preference is the intuitively most obvious example. The data matrix, then, consists of n rows each of which reflects a particular case's order of preference for the stimuli.

Preference judgments may be represented in four distinct ways:

1 a rank order:	a list of stimulus numbers from most preferred to least preferred, for instance *2 5 3 4 1* to indicate that stimulus number 2 is the most preferred, followed by stimulus 5 and so on until stimulus 1, which is the least preferred. With m stimuli each data row must contain m numbers in the range 0 through m. If the position of a particular stimulus is unknown, the value 0 can be used to fill the gap.
2 a reversed rank order:	a list of stimulus numbers from least preferred to most preferred, for instance *1 4 3 5 2* to indicate that stimulus 1 is least preferred, followed by stimulus 4 and so on until the most preferred stimulus 2. With m stimuli each data row must contain m numbers in the range 0 through m. If the position of a particular stimulus is unknown, the value 0 can be used to fill the gap.
3 rank scores:	the first number corresponds to stimulus 1 and defines its rank, the second number corresponds to stimulus 2 and defines its rank and so on. The stimulus with the lowest rank is the most preferred. The list *5 1 3 4 2* indicates that stimulus 1 is the least preferred (score 5), stimulus 2 is the most preferred (score 1) and so on. It should be noted that not the actual numbers are important, but only their order.
4 reversed rank scores:	the first number corresponds to stimulus 1 and defines its rank; the second number corresponds to stimulus 2 and defines its rank and so on. But now the stimulus with the lowest rank is the least preferred. The list *1 5 3 2 4* indicates that stimulus 1 is the least preferred (score 1), stimulus 2 is the most preferred (score 5) and so on.

The program will always translate these four formats to the first one: rank order data.

5.1 Compressed data, missing values and selection:

If there are cases with identical preference orders, the input data can be compressed to a list of case ‘patterns’. Then each row will contain the preference data as described above plus a frequency count.

The input data may contain ‘missing values’ to indicate that some stimulus numbers or a rank scores are unknown. With rank scores and reversed rank scores this means that the corresponding score takes no part in the computations.

For rank order and reversed rank order data there are two possible ways to treat the missing values, depending on the option ‘replace missing values’ chosen by the user:

- If the option ‘replace missing values’ is chosen it is assumed that the omitted stimuli are less preferred than the chosen ones (with rank orders) or more preferred (with reversed rank orders). These missing stimuli then are considered to form a tie.

- If the option is not chosen it is assumed that it is completely unknown where the omitted stimuli should be inserted in the list.

Although the number of values in each input row must be exactly the same as the number of stimuli plus optionally a frequency count, it is possible to exclude one or more stimuli from the analysis.

5.2 Examples

Table 1 gives some examples of the input types and their interpretation.

Input type	replace missing values	Input values - = missing value	Interpretation: rank order [..] = tied stimuli
rank order		2 5 3 4 1	2 5 3 4 1
reversed rank order		1 4 3 5 2	2 5 3 4 1
rank scores		5 1 3 4 2	2 5 3 4 1
reversed rank scores		1 5 3 2 4	2 5 3 4 1
rank order,	yes	2 5 3 - –	2 5 3 [1 4]
reversed rank order	yes	3 5 2 - -	[4 1] 2 5 3
rank scores		8 1 6 8 2	2 5 3 [1 4]
rank order	no	2 5 3 4 –	2 5 3 4
reversed rank order	no	1 4 3 5 -	5 3 4 1
rank scores		5 - 3 4 2	5 3 4 1
reversed rank scores		1 - 3 2 4	5 3 4 1

Table 1: Examples of input rows and their interpretation.

6 Files

There are five file types that may play a role for this program:

data files:
The files that contain the data to be analyzed. One program run can analyze several files and each file can contain several tests.
There may also be a file with the trial configuration.

settings files:
These files are used to save the options as they are specified by a user. A settings file contains all information about the analysis to be performed, including the description of the data, but not the data themselves. It is possible to have more than one settings file. By default, their extension is '.setunf'.

listing files:
A listing file contains the main results of an analysis in a nice layout for human readers. There will be one listing file for each data file. The name will borrow its first part from the data file and end on '.LST' (or on '1.LST', '.2LST', ... and so on). You can inspect it by any editor, but in order to have an orderly layout you must view it in a small non-proportional font like Courier New 9.

raw output files:
Depending on the chosen options, the program may produce one file with “raw” output for each input file. These raw files are meant to be input for other programs. Their names will take their first part from the data files and end with '.OUT' (or '1.OUT', '2.OUT', ... and so on).

plot files:
If the program produces any plots (graphical representations) the listing files will contain coarse versions of the plots, but the program will also produce a more refined bitmap for each plot. The names of these bitmap files will take their first part from the data files and on Windows machines they will end with '.BMP' (or '1.BMP', '2.BMP', ... and so on).

7 Installing the program on Windows

The installation of the program is very simple:

Copy the file *Unfold.exe* to any place on your hard disk. Optionally you may make shortcuts on the task bar and/or the desktop.

After the first time you have used the program, double click on the listing file. Windows will ask you to select the program to be used when opening the file. Select a simple text editor like Notepad or WordPad.

After the first time you have saved the program settings, double click on the settings file. Windows will ask you to select the program to be used when opening the file. Select the program *Unfold.exe* or any shortcut to it.

That is all: from now on, you can start the program by double clicking the exe-file, one of its shortcuts or a settings file.

8 Running the program

To run UNFOLD you must double click on its executable file (for Windows: *Unfold.exe*) or, if you have used the program before, on one of its settings files (for instance *Current.SetUnf*). The first thing you will see then, is the main window of the program, as shown in figure 4.

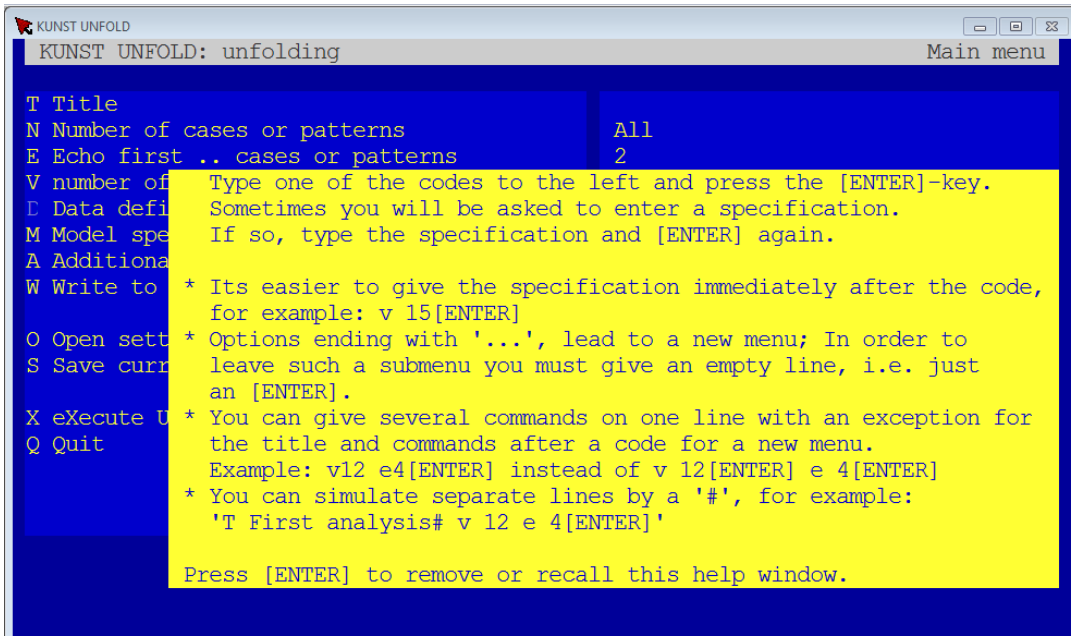

Figure 4: The main window.

On the screen the light part with the text 'Type one ...' is a yellow text window. Windows like that contain hints and explanations. If you have read the text (or do not need it), you can press the Enter-key and the yellow window will vanish (see figure 5).

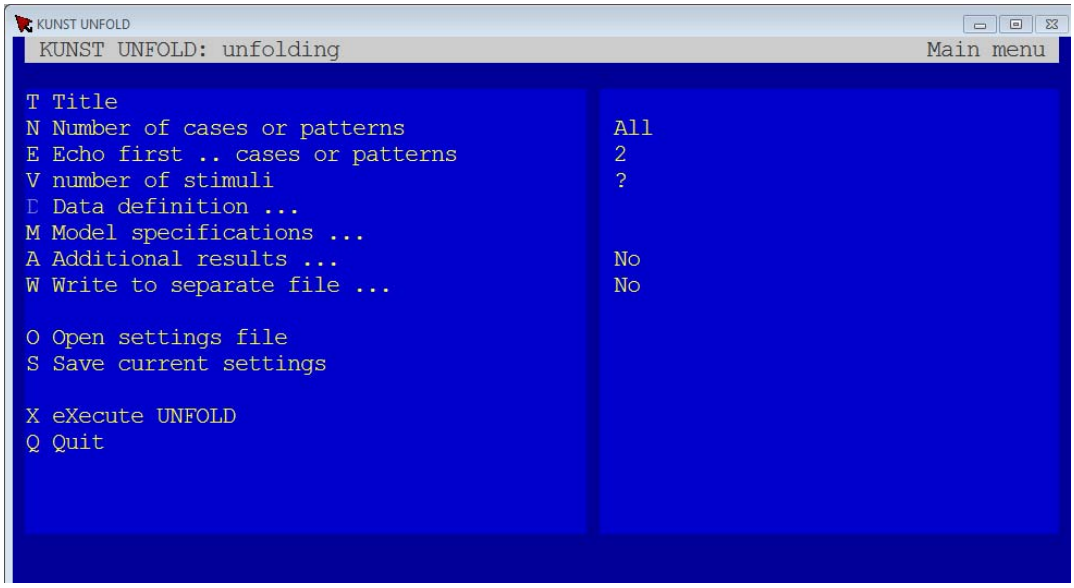

Figure 5: The main window without help-window.

Now you can see the entire main window. The left part is the *main* menu. It consists of a list of options, each preceded by a one-character code. To select an option, you must type the code, followed by the information you want to give and then the Enter-key. From now on, we will indicate the Enter-key as: Enter. You may for instance type: `T Analyzing first session`Enter to define the title that will appear as a header in the listing files. If you do not know what the meaning of an option is, you may just enter its code and Enter. There will appear a question on the screen and, if helpful, a yellow window to give you information. Press Enter to remove the yellow window and then give the specification belonging to the code, or ignore the yellow window and give the specification at once. **Do not repeat the code itself!**

The right part of the window gives a short review of the options, as they are currently set. In the example of figure 5, you can see the following:

- No title line is defined.
- The number of cases or patterns is still undefined.
- The number of stimuli is not yet specified, but must be given.
- The data definitions cannot yet be chosen because the number of stimuli must be known first (although it is difficult to see in this print, the option character 'D' is gray).
- The model specifications are not echoed here, but in the corresponding submenu.
- No additional results will be reported.
- No raw output file will be produced.

9 Menu options

9.1 The main menu

The *main* menu (see figure 5) contains the following options:

`T Title`
This option allows you to specify a header to be used in the listing files.

`N Number of cases or patterns`
By this option you may specify the number of cases or patterns to be analyzed. You may answer '`all`' to indicate that the program must read all cases or patterns in the file and count them itself. However, if you want to build a trial configuration for the cases/patterns during the first program phase, you must specify the exact number here.

`E Echo first .. cases or patterns`
If you type `e ##Enter` the first ## cases or patterns will be shown in the listing file. This may help you to check if the input specifications are correct.

`V number of stimuli`
You must use this option to specify the number of stimuli. This number does not include a possible frequency count.

`D Data definition ...`
If you type `DEnter` the *main* menu will be replaced by the *data definition* menu. This menu allows you to define the input data. It will be discussed in 9.2.

`M Model specifications ...`
This option leads to a new menu where you can choose several options that influence the process, like the number of dimensions, the minimum stress and the handling of ties. It will be treated in 9.3.

`A Additional results ...`
This option leads to a new menu that allows you to request additional information in the listing file. It will be treated in 9.4.

`W Write to separate file ...`
This option leads to a new menu that allows you to write several results to a "raw" text file. It will be discussed in 9.5.

`O Open settings file`
If you have ever saved the options for UNFOLD or if you received a settings file from someone else, you can retrieve the options from the settings file. If you type `OEnter`, a file-selector box will appear on the screen that allows you to select the settings file. By default settings files from UNFOLD have the extension '`.setunf`'. It may be handy to save your current settings before collecting new ones. The program may remind you of that. After you have collected information from a settings file, its name will be visible on the upper right part of the main window.

`S Save current settings`
If you want to save the options and specifications that you have made so far, you can enter `S`Enter. If you do so a file-selector box will appear that allows you to specify the path and the name of the file to which the settings must be written.

`X eXecute UNFOLD`
If you have specified all options you can type `X`Enter to start the computations. The program will check if all obligatory options are specified and if there are no inconsistencies. If everything is right, the computations will start. If the program has been correctly installed and runs without problems, it will, when it is finished, automatically open the last (or only) listing file it has made. If it fails to do so, you can open it yourself by any text editor like *WordPad*, *Notepad* or *Word*. In order to have a nicely outlined text, you must select a small non-proportional font like Courier New 9.

`Q Quit`
The option `Q` is an emergency exit. If you choose it, UNFOLD will halt without performing any calculations and without producing any output files.

9.2 Definition of the data

From the *main* menu, you may type the option `D`Enter to enter the *data definitions* menu. After you have filled out the options in this menu you must press Enter to return to the *main* menu.

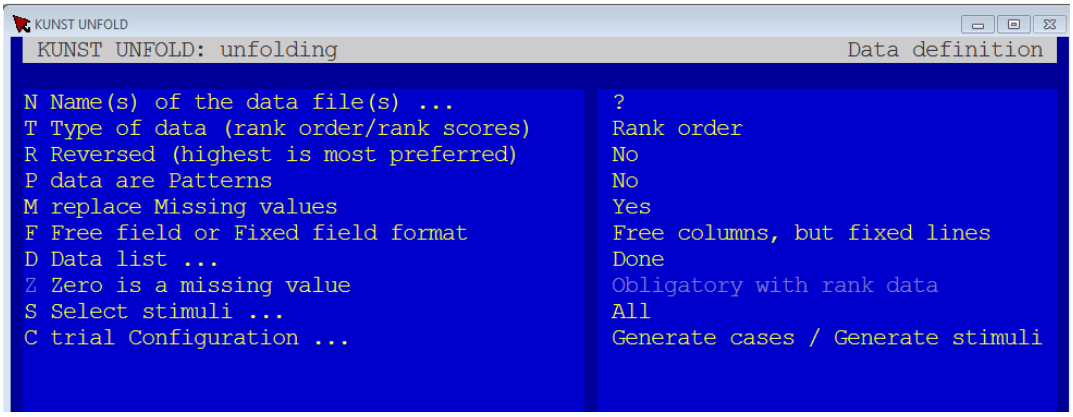

Figure 6: The data definition menu.

With m stimuli each data row must contain exactly m or m+1 values. The (m+1)-th value is the frequency count for condensed data. If any data are missing they must be represented by special values, called *missing values*. With rank order or reversed rank order data the only possible missing value is zero. With rank scores or reversed rank scores it is possible to define upper limits: any value equal to or greater than such an upper limit is treated as a missing value.

In the *data definition* menu (see figure 6) the following options can be chosen:

`N Name(s) of the data file(s)`
If you choose this option a file-selector box will appear that enables you to select one or more data files to be analyzed.

`T Type of data (rank order/rank scores)`
With this option you can switch between two types of data:

Rank order: Each row in the input data contains a list of stimulus numbers from most preferred to least preferred or the other way around, dependent on the option Reversed.

Rank scores: In an input row the first value defines the preference rank of the first stimulus, the second value the rank of the second stimulus and so on. The stimulus with the lowest value is the most preferred or the least preferred, dependent on the option Reversed.

`R Reversed (highest is most preferred)`
`R Reversed (last is most preferred)`
With this option you can switch between rank order (`No`) and reversed rank order (`Yes`) or between rank scores (`Yes`) and reversed rank scores (`No`).

Note: If the data are *rank orders* or *reversed rank orders*, subsequent *positions* in a data row do not represent the stimuli but the preference levels. (Their *values* correspond to the stimuli). Nevertheless names given to the columns by the option datalist in the data menu will be used as names for the stimuli irrespective of the data type.

`P data are Patterns`
By switching this option to `Yes` you specify that each data row contains a frequency count. Such a count means that it is possible to replace k cases with the same preferences by a single pattern with its frequency set to k. The frequency count must be the last variable in the list of variables. If there are m stimuli, the frequency is given by the (m+1)th variable. In the data list this variable will be marked by type `F`.

`M replace Missing values`
This option is only available if the data are of the types rank order or reversed rank order. Data of these types may contain zeros to indicate missing values. If the option replace missing values does not apply, these zeros are simply ignored. But if the option applies the stimuli that are missing in a row will be handled as if they form a tie at the end (with rank order data) or at the beginning (with reversed rank order data) of the row.

`F Free field or Fixed field format`
This option defines where the values in a case are to be found:

In a *free* format, the positions of the values may be different from row to row, although their order must be always the same. You can still choose whether line numbers are specified or not.

If the input file contains one single line per case and the scores are separated by spaces, commas or tabs, you can select the free format option and skip the data list, with a few exceptions:
(1) By default the case identification (the label that identifies the case or the pattern) is taken from positions 1-8 of the first input line of each case or pattern. If this identification is absent or to be found elsewhere, you must use the data list to adjust its location.
(2) If you need to define missing values other than zero for (reversed) rank scores you must do so in the data list.
(3) If you want to give names to the stimuli you must use the data list to do so.

In general, the use of free format is highly recommended since it is less sensitive to irregularities in the data or mistakes in the specifications.

In a *fixed* format, each case consists of the same number of lines (most probably just one) and each value has a precisely defined position in that line. This format is especially useful if the values are typed one after the other without intervening spaces, tabs or commas.
Each time you type `F`Enter the choice switches.

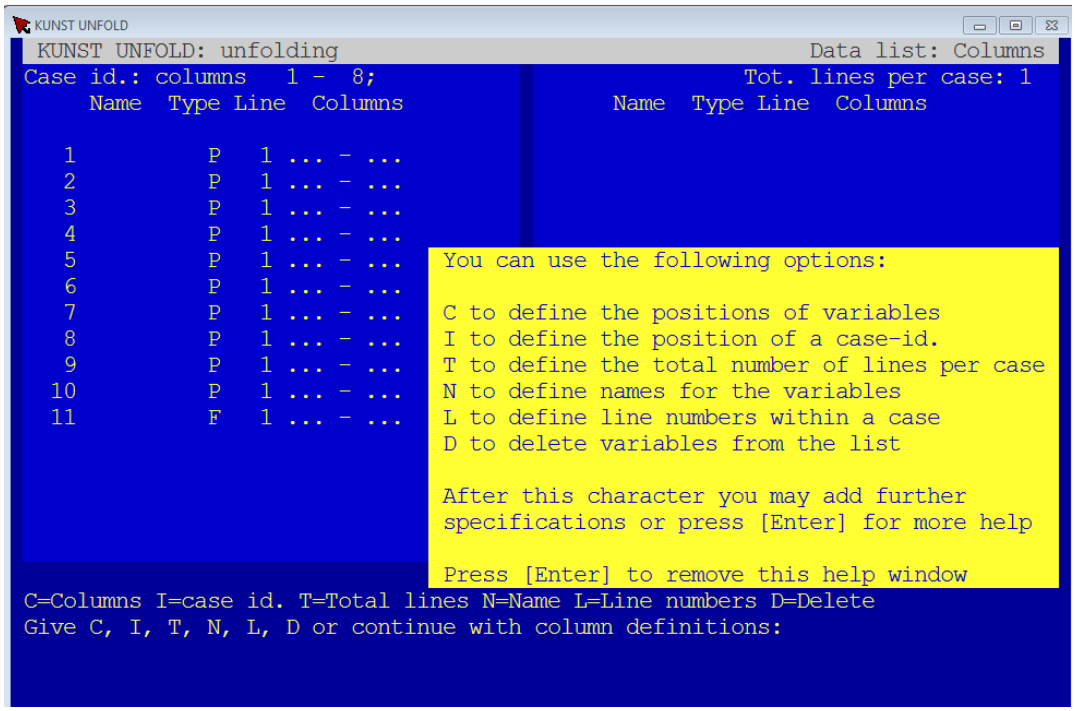

Figure 7: A data list for fixed format rank order data.

`D Data list ...`
If you choose this option, a new window will open with a layout that differs from the usual menu layout. Its layout depends on the chosen format type. Figure 7 shows a data list for fixed format data of type (reversed) rank order. The column type defines the role of the data columns: '`S`' for *stimulus* (in case of rank scores or reversed rank scores), '`P`' for *preference* (in case of rank orders or reversed rank orders) or '`F`' (for *frequency count*).

Now you can type one of the indicated characters (`C`, `I`, `T`, `N`, `L`, `D`) followed by a sequence number or a range of sequence numbers and then followed by the corresponding information. The possibilities can best be clarified by some examples:

You type: `c1 9-11`Enter
Thereby you specify that the first stimulus number is in positions 9 through 11 (right adjusted). In this example the first eight positions contain a pattern identification string, as you can see from the line starting with '`Case id.`' at the top of the window.
From now on, you can leave out the code `C`; it will be assumed as long as you do not select another code.

You type: `2-4 12-20`Enter
This means that preferences 2, 3 and 4 occupy positions 12 through 20, so preference 2 is in position 12-14, preference 3 is in positions 15-17 and preference 4 in positions 18-20.

You type: `5 21-24`Enter
This means that the fifth preference occupies positions 21-24.

You type: `L6 2`Enter
Now you have entered a different code. The code **`L`** indicates that you are defining line numbers. You specify that preference 6 is contained in the second line of a pattern. Line numbers must be in ascending order. Therefore, the program will adjust the line numbers of preference 7 and higher to 2 if they are still 1. From now on, you may leave the code `L` out, until you switch to another code.

You type: `N1 Dressing`Enter
The code **`N`** indicates that you are specifying names for the stimuli. Stimulus 1 will receive the name '`Dressing`'. Names will be truncated to 8 characters. Note that the names are linked to the stimuli, even if the data are preferences.

You type: `2-5 Other`Enter
Variable 2 through 5 will all be called '`Other`'

You type: `6-10 Dress6`Enter
Variables 6 through 12 will be called '`Dress6`', '`Dress7`', ..., '`Dress10`'. As you see, if the name ends on a number, subsequent names will have their numbers incremented.

You type: `D5`Enter
The option **`D`** allows you to remove a row from the data list. All sequence numbers and all other definitions will be adjusted accordingly. In this example, you remove the fifth row. You can also specify a range of row numbers, for instance '`D5-7`', to remove the rows 5, 6 and 7.

You type: `I1-4`Enter
With the option **`I`** you define an area in the first line of each row that contains an alphanumerical case label (or *case identification*). These labels will be used in the listing file. Specify `I0` to indicate that no labels are present.

You type: T4Enter
If you don't use the option **T**, the program will assume that the number of lines for each case is equal to the last line number in the data list. If there are more lines in a row, you must specify so by the option T. In this example, you specify that there are 4 lines in each pattern.

With (reversed) rank score data there is also the option M to define missing values by upper limits:

You type: M3-5 99Enter
The option **M** lets you define upper limits for the rank scores. It is only available with the data types rank scores and reversed rank scores. If the value of a stimulus is equal to its upper limit or greater, its value will be treated as missing. In this example, you specify that for the stimuli 3, 4 and 5 values that are 99 or larger must be treated as missing. Upper limits must be positive numbers. To remove a missing value use X instead of a number.

You type: FEnter or BEnter
If there are more than 32 variables, they will not fit at once on the data list screen. Therefore, you have the possibility to scroll **f**orward and **b**ackward with the options F and B.

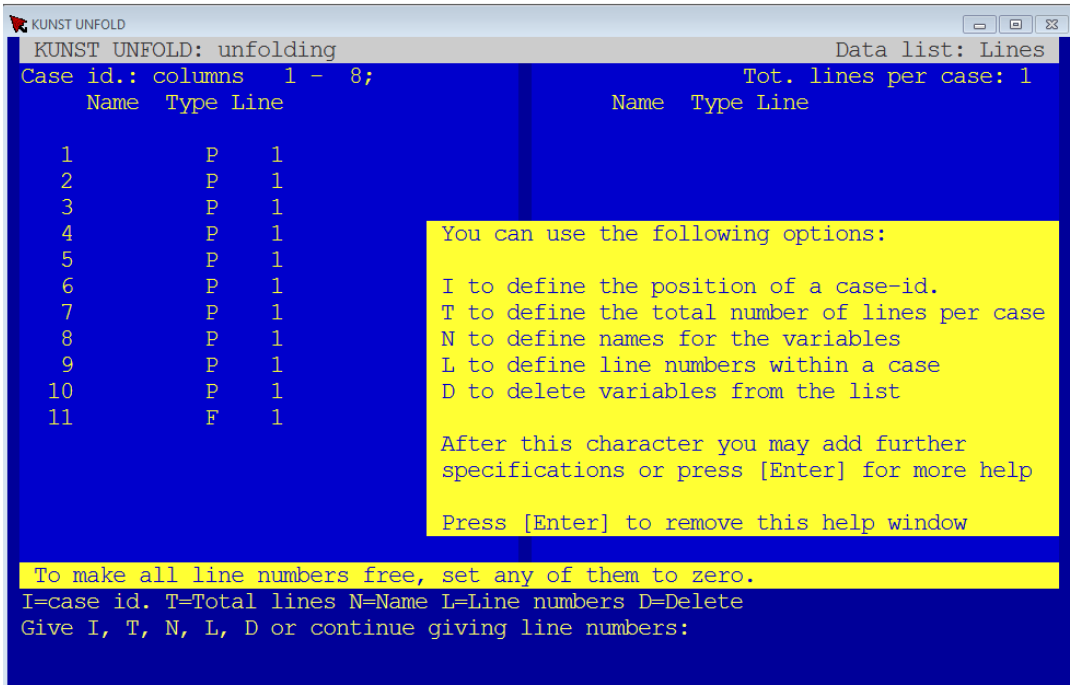

Figure 8: A data list for fixed format rank order data.

If you use free format data (see figure 8), you need none of the options in the data list, apart from a few exceptions:

(1) By default the case identification (the label that identifies the case or the pattern) is taken from positions 1-8 of the first input line of each case or pattern. If this identification is absent or to be found elsewhere, you must use the data list to adjust its location.
(2) If you need to define missing values other than zero for (reversed) rank scores you must do so in the data list.
(3) If you want to give names to the stimuli you must use the data list to do so.
(4) You may specify on which line of a case each variable is recorded. So you can use the option `L`. The options `T`, `N`, `D` and `M` are also available. Their meaning and use are the same as with fixed format data (see above).

If you have finished the data list, you can go back to the *data* menu by entering an empty line (just `Enter`). If the yellow window is still visible, you must enter two empty lines (`Enter` `Enter`): one to remove the yellow window and one to return to the *data* menu.

`Z Zero is a missing value`
With this option you can define whether zero is to be treated as a missing value or not.

`S Select stimuli`
When choosing this option a new window appears with a list of all stimuli (see figure 9). If an arrow follows a variable it is selected in the analysis.

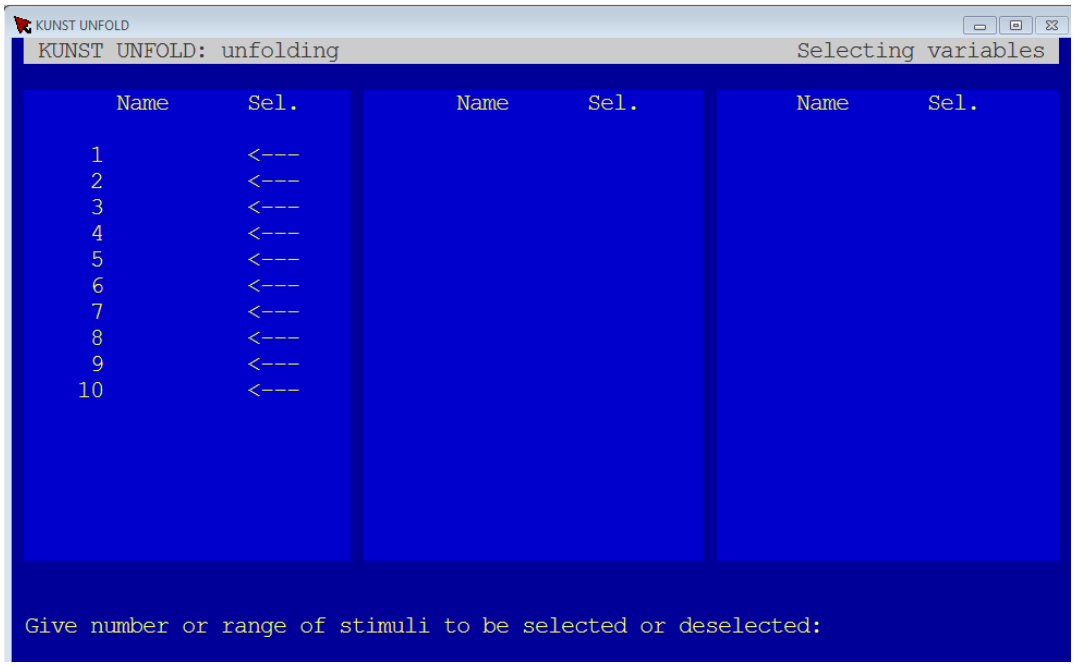

Figure 9: Selection of stimuli.

To select or deselect a variable you just have to give its number and Enter . If it was selected it will be deselected now. If it was deselected it will be selected now. You can also use a range of variables. What will happen depends on the first variable in the range. All variables take the switched value of the first variable.
For example 3-5Enter .
You go back to the *data* menu by entering an empty line (just Enter).

`C trial Configuration ...`
If you type CEnter the *data definition* menu will be replaced by the *trial configuration* menu. This menu allows you to select the method for entering the trial configurations for cases/patterns and stimuli. It will be treated in 9.4. By default the trial configurations will be generated by UNFOLD.

9.3 Model specifications

In the *main* menu, typing mEnter leads to a submenu as shown in figure 10. It allows you to set some parameters that influence the computations to be performed.

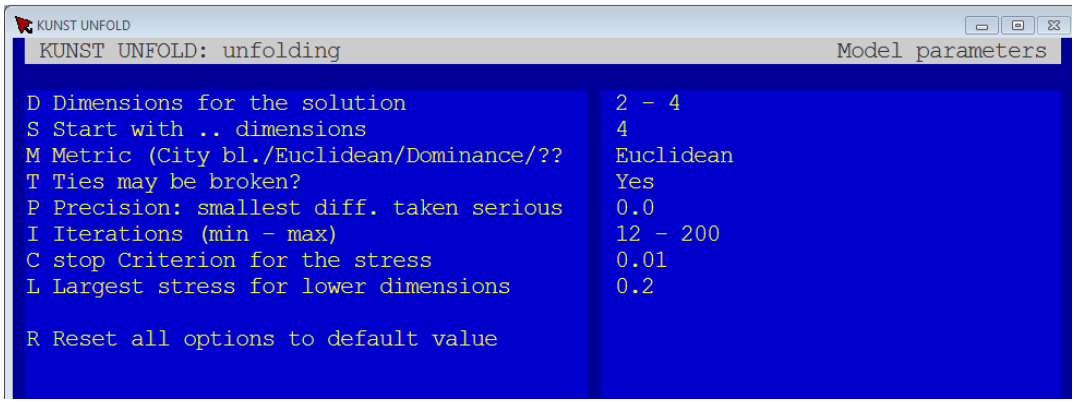

Figure 10: Model specifications.

`D Dimensions for the solution`
With this option you can give the minimum and maximum number of dimensions for which a real solution must be found. You cannot ask for a solution in more than 8 dimensions. First the program seeks a solution in the maximum number of dimensions. If a solution is found, the last dimension is dropped and a new analysis is performed with the reduced configuration as a trial configuration. This reduction of dimensions will however be interrupted if the stress of the last found solution is higher than the criterion given with option L.
With the Euclidean metric the solutions will be rotated to their principal axes, so that a minimum of information is lost when the last dimension is dropped.

`S Start with .. dimensions`
Even if one expects a solution in few dimensions it may be helpful to start with more, because that may lead to a better trial configuration for the lower dimensional solutions! In that case one must use this option to specify the number of dimensions from which UNFOLD must start. The program will then perform analyses in higher dimensions with a minimum of output in the listing file and only start reporting results as soon as the maximum of the actual range (according to option D) is reached.

`M Metric (City bl./Euclidean/Dominance/??)`
The option Metric offers the opportunity to define the distance to be used. The given value will act as the Minkowski-parameter (see 3.1). One may give any positive number less than or equal to 15, but strange things may happen with a parameter below 1. The Minkowski-parameter is 2 by default, thereby defining the usual Euclidean metric. Instead of a 1 one may choose a `C` (for City block), instead of a 2 an `E` (for Euclidean) and with a `D` one asks for the Dominance metric.

`T Ties may be broken?`
The option `T` determines whether the program must try to represent equal rankings in the data by equal distances (the secondary approach) or not (the primary approach). The secondary approach will generally lead to higher stress (see 0).

`P Precision: smallest diff. taken serious`
By typing `p ##Enter` one can indicate that two rank scores are considered to be tied if their difference is smaller than ##. There may be some ambiguity in this definition of ties: if the data are ordered there may be a chain of small differences, each smaller than the criterion. In that case the program scans the data from small to large. The first value starts a tie. All subsequent values that differ less than the criterion from the first value are added to the tie. The first value that differs too much from the starting element will act as the start of a new tie and so on.
Choose zero unless you have good reasons to give another value.

`I Iterations (min - max)`
UNFOLD ends its iteration process when the configuration does not improve any more (see 3.6). But the user may put some restrictions on this mechanism by defining a minimum and a maximum number of iterations to be performed. By default the minimum is 12 (do not give up too soon) and the maximum is 200. By this option one may change these minimum and maximum values. Because the process consists of two phases, one using $\hat{d}$ and one using d^*, the iterations are evenly distributed among them: for both phases the minimum and the maximum are set to half the limits that are given here.

`C stop Criterion for the stress`
The process of finding a solution continues until the stress becomes lower than the value given with this option. See 3.6 for all the possible reasons to end the iteration process.

`L Largest stress for lower dimensions`
The reduction of the dimensions as given in option `D` will be stopped if the stress of the last found solution is higher than the criterion given with this option.

`R Reset all options to default values`
By typing `rEnter` one resets all the other options in this menu to their default values.

9.4 Additional results in the listing file

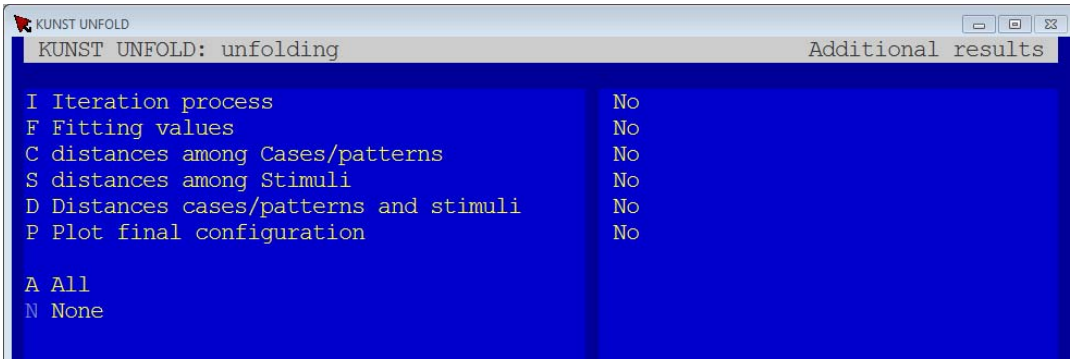

Figure 11: The additional results menu

Entering `A`Enter in the *main* menu leads to the *additional results* menu (see figure 11). The options in this menu offer the probability to require additional information in the listing file besides the default results.

The menu contains the following options:

`I Iteration process`
By typing `i`Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain some information after each iteration step. It is not advised to use this option unless there isa good reason to look into the details of the process.

`F Fitting values`
By typing `f`Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain a list of the fitting values, which includes also the stress per case or pattern and contributions of the stimuli to the squared stress per case or pattern.

`C distances among Cases/patterns`
`S distances among Stimuli`
`D Distances cases/patterns and stimuli`
By choosing one of these options you switch the option from yes to no or the other way around. If it is yes, the listing file will contain the corresponding distances.

`P Plot final configuration`
By typing `p`Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain a plot of each final configuration. If the number of dimensions is greater than 2, a separate plot will be shown for each pair of dimensions. In the listing file, these plots are rather coarse images build as characters in a grid. More detailed pictures is be stored in separate bitmap files.

`A All`
`N None`
If you type `a`Enter, options `I`,`F`,`C`,`S`,`D` and `P` will be switched to yes all together.
If you type `n`Enter, options `I`,`F`,`C`,`S`,`D` and `P` will be switched to no all together.

9.5 Additional results in the raw output file

In the *main* menu the option W leads to the *write to separate file* menu as shown in figure 12.

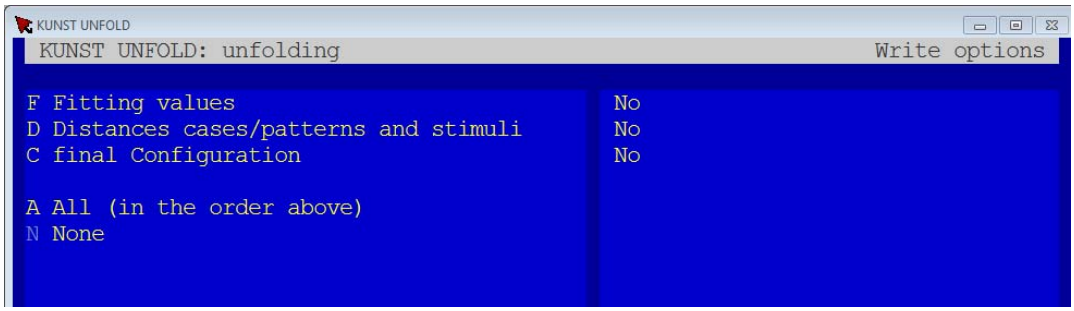

Figure 12: The Write-to-separate-file menu.

This menu contains a list of flags. If you type `F`Enter, `D`Enter or `C`Enter, the corresponding option switches from no to yes or back. The options all lead to output in a separate file without any layout. This output file makes it relatively easy for other programs to collect the information for their own use.

If any output option is chosen, a raw output file will be produced. If more than one kind of output is produced, the file will contain the results in the order in which they are enumerated in the menu. In the output file a single labelling line will precede each option. If solutions are sought in more dimensions, the options will be written for each solution.

`F Fitting values`
If this option is set to yes, the raw output file will contain the fitting values for each solution.

`D Distances between cases/patterns and stimuli`
If this option is set to yes, the raw output file will contain a matrix of the distances between the cases or patterns and the stimuli.

`C final Configuration`
If this option is set to yes, the raw output file will contain the final configurations.

9.6 Defining the trial configuration

From the *data definitions* menu, you may type the option `C`Enter to enter the *trial configuration* menu as shown in figure 13. After you have filled out the options in this menu you must press Enter to return to the *data definitions* menu.

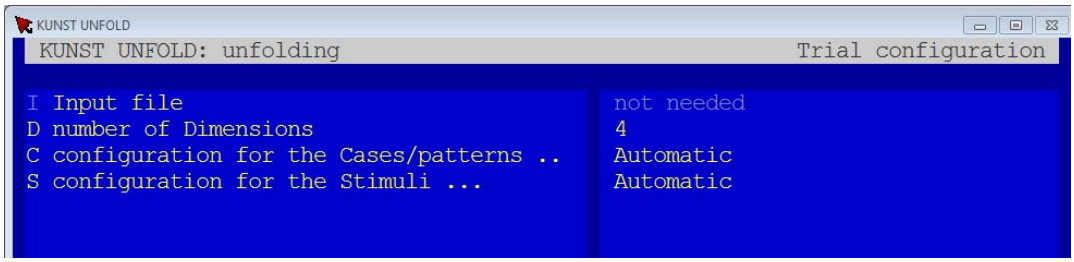

Figure 13: The trial configuration menu.

`I Input file`
If you choose this option a file-selector box will appear that enables you to select the file that contains the trial configuration for the cases or patterns and/or the trial configuration for the stimuli. In figure 13 this option is disabled because the options `C` and `S` specify that both trial configurations must be generated by the program.

`D number of Dimensions`
With this option one may define how many dimensions the trial configuration contains. It will only be used if one or both trial configurations are given in a file or during the first phase of the program.
By default this number is set to the number of dimensions to start with, as set in the *model specifications* menu (see 7.3).

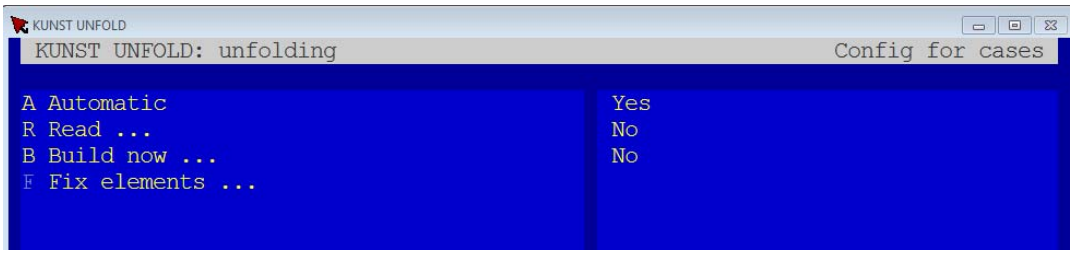

Figure 14: Defining the trial configuration for the stimuli

`C configuration for the Cases`
`S configuration for the Stimuli`
Typing `C`Enter leads to a menu where you can specify how the trial configuration for the cases or patterns must be defined. Figure 14 gives an example.
Typing `S`Enter leads to a menu where you can specify how the trial configuration for the stimuli must be defined.

`A Automatic`
This is the default choice. It means that the trial configuration for has to be generated by the program. However, you may have reasons to define it yourself.

- Maybe you are not satisfied with a solution from a previous UNFOLD run and you want to guide the program in a certain direction.
- There may be theoretical grounds to expect some type of configuration to be the best.
- If several data matrices are analysed you may want to use the same trial configuration for all of them, or maybe you want to use the outcome of one analysis as the start for another.

`R Read ...`
This option is used to indicate that the trial configuration for the stimuli must be read from a file. The option leads to a special submenu that will be discussed in 7.7.

If both trial configurations have to be read, they must be in the same file and the configuration for the cases/patterns must be in front of that for the stimuli. If the data are also in the same file, these must come at the end of the file.

`B Build ...`
This option is used to define the trial configuration immediately. The corresponding sub menu is described in 7.8.

When R or B is chosen, the resulting trial configuration may contain too few dimensions. With B it may even contain 'holes'. UNFOLD will fill these missing values as well as possible, but the resulting configuration may be far from optimal.

The options A, R and B are mutually exclusive. If one of them is chosen, the others are cleared and the only way to clear one is to choose an other.

It is important to know that the trial configurations (given or generated) are normalized in two ways:

1 Both configurations are moved so that the centre of the stimuli is at the origin.
2 Both configurations are rescaled so that the mean of the squared distances of the stimuli from the origin is 1.

`F Fix elements ...`
If you give your own trial configuration (option R or B) you can also fix some elements in that configuration. Fixed elements will not be moved during the computations. Choosing this option will open the Fix-menu (see 9.9).

9.7 Reading a trial configuration

Entering R Enter in the menu defining one of the trial configuration leads to the *reading trial* menu (see figure 15).

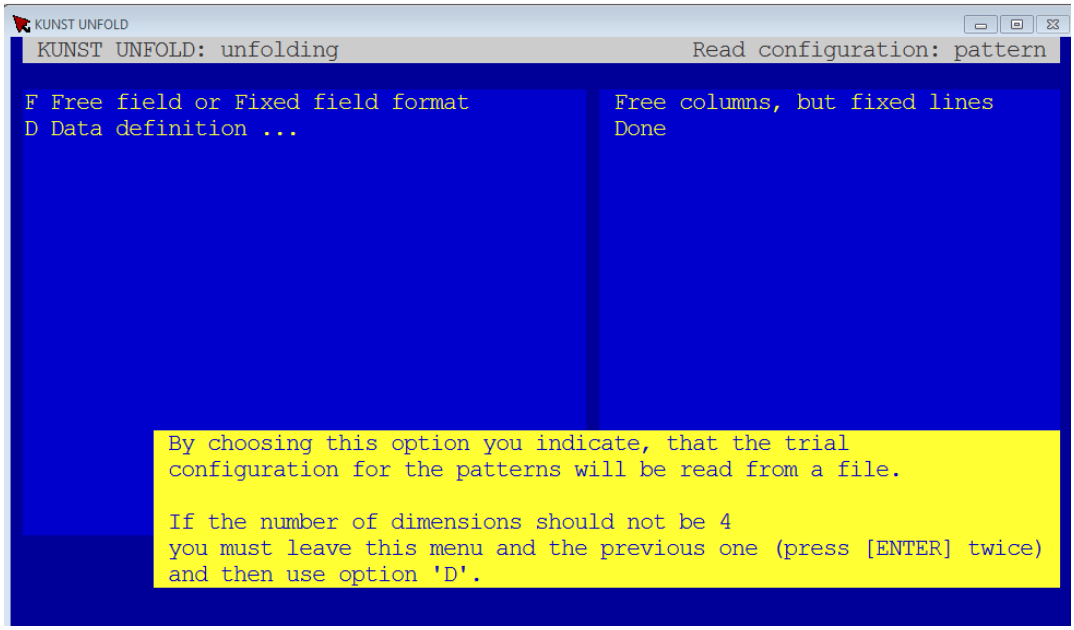

Figure 15: Reading the trial configuration for the stimuli from a file

The options F and D are the same are similar to those for the data (see 7.2). If both the trial configuration for the cases and the configuration for the stimuli must be read from a file, they must both be stored in the same file and the cases/patterns must precede the stimuli. If this file also contains the preference data these should come after the trial configurations.

9.8 Building a trial configuration

Entering B`Enter` in the menu *defining the trial configuration for cases or patterns* leads to a window with a layout that differs from the usual menu layout. In a similar way the option B`Enter` in the menu *defining the trial configuration for stimuli* opens a *build* window for the stimulus configuration. These windows are almost identical. Therefore we will describe only one: the build window for cases or patterns. Figure 16 shows a *build* window for more than 15 cases and 4 dimensions.

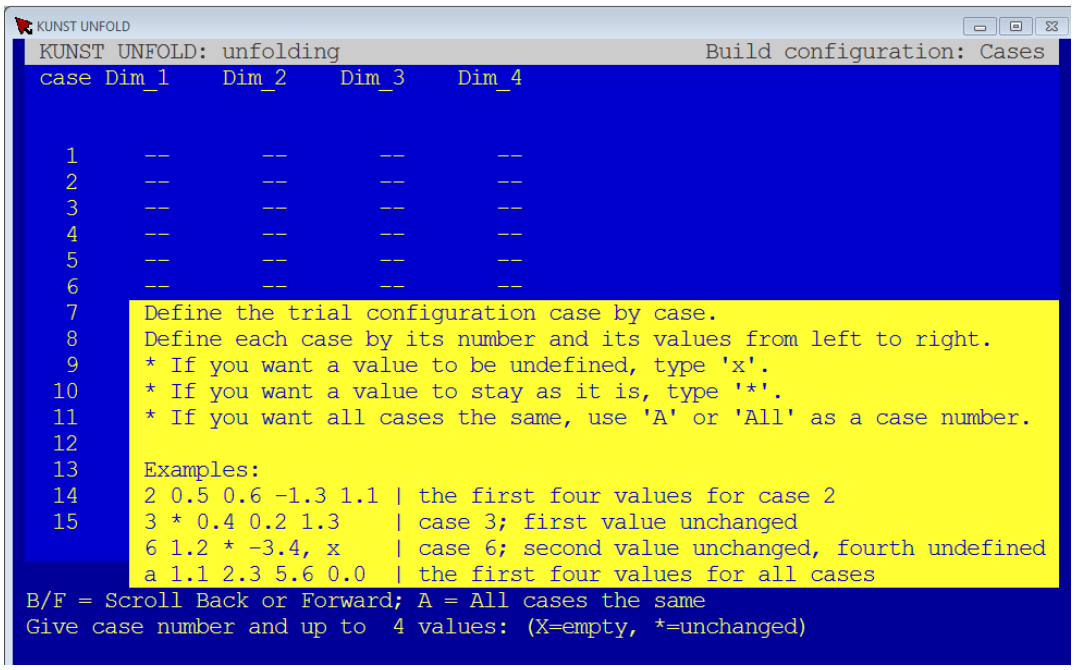

Figure 16: Building a trial configuration for cases or patterns.

In this window you may define all or part of the trial configuration. You may leave some cases (or patterns) empty and even within one case/pattern you may skip dimensions. The program will generate the missing elements in the second phase.

You must define the trial configuration case by case (pattern by pattern). Give a case/pattern number followed by a list of coordinates. In order to give the same values for all cases/patterns you can use "`A`"or "`All`" instead of a case or pattern number. An asterix (*) indicates that an element must be left unchanged. If you want to 'undefine' a value, give it the value '`X`'. Here are some examples:

`1 1.4 3.5 -2.3`
In row 1 the coordinates on the first 3 dimensions are 1.4, 3.5 and -2.3.

`3 * 0.3 0.7 1.2`
In row 3 the first coordinate must be left as it was and the coordinates for dimensions 2 to 4 are 0.3, 0.7 and 1.2.

`1 1.5 x 2.5`
In row 1 the coordinates on dimensions 1 and 3 are 1.5 and 2.5 and the second coordinate must be made unknown.

`all 4 5 6 7`
For all rows the coordinates on the first 4 dimensions are 4, 5, 6 and 7.

Because the number of cases or patterns is too large to show the whole configuration in the window there are the options `F`Enter and `B`Enter to scroll forward and backward.

9.9 Fixing elements in a trial configuration

Entering `F`Enter in the menu *defining the trial configuration for cases or patterns* or in the menu *defining the trial configuration for* stimuli leads to a window that allows you to fix some elements in the trial configuration. Figure 17 shows such a *build* window for 10 stimuli and 4 dimensions.

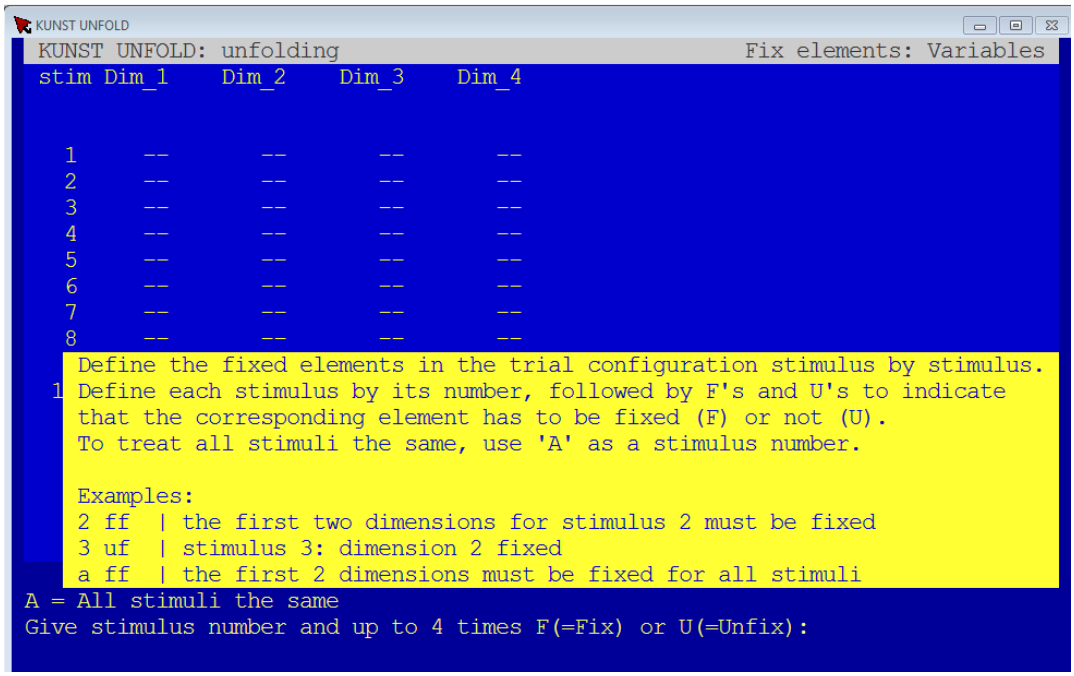

Figure 17: Fixing elements in the trial configuration for the stimuli

You can only fix elements that are not generated by the program. So, if you did choose to build a configuration, but did not fill all the coordinates, only these coordinates can be fixed for which a value was given. If you did choose to read the configuration from a file, all elements can be fixed.

You must enter a row number, defining the row, and a list of `F`- and `U`-characters (for Fix and Unfix) on the dimensions.

If you want all stimuli to have the same pattern use '`A`' or '`All`' as a row number followed by a list of `F`- and/or `U`-characters.

10 Results

After execution of UNFOLD, you will find one or more new files in your working directory:

- the listing file (`.lst`)
- if you asked for additional output in a separate file (`W` in the *main* menu) the raw output file (`.out`)
- possibly some bitmap files

We will explain the results by an example run on a small dataset.

10.1 Example run

Our example data are borrowed from Frisenfeldt Thuesen (2007). They contain the preferences of 32 consumers for 4 salad dressings. In our terms the data are rank order patterns. Figure 18 shows the input data. The first 3 positions of each pattern contain a pattern identification code. Then follow 4 stimulus numbers in order of preference and a frequency count.

```
C1    1 2 3 4   2
C7    2 1 3 4   1
C8    2 1 4 3   2
C9    2 3 1 4   1
C13   3 1 2 4   2
C14   3 1 4 2   1
C18   3 4 2 1   1
C19   4 1 2 3  11
C20   4 1 3 2   6
C21   4 2 1 3   3
C23   4 3 1 2   1
C24   4 3 2 1   1
```

Figure 18: Example data

The main results of the analysis will appear in the listing file. Its precise content depends on the specifications by the user and the input data. The lines in the listing have a length of 80 characters or less. View (and print) this file with a small non-proportional font like Courier New 9. Hereafter we will show the most important parts of this listing in a somewhat condensed form.

After a header part the listing file contains an overview of the user options, as in figure 19.

```
================================================================

   1.  Summary of the user specifications and input data.

================================================================

 The data will be read from file:
   C:\MyData\Salad_Dressings.dat
 They are numbers of stimuli in order of preference.
 The first one mentioned is the most preferred.
 Unmentioned stimuli will be treated as least preferred and tied.

 The number of variables is 5.
 Variable 5 contains the frequencies of the data patterns.

 The number of patterns will be computed.
 The number of patterns to be echoed is 2.

 Ties in the data may lead to unequal distances in the solution.

 A solution will be sought in 2 dimensions.
 Solutions in 3 to 3 dimensions are performed to find a trial configuration
   in 2 dimensions.

 The number of iterations will be at least  12 and at most  200:
   - At least   6 iterations with D* and at least   6 with D^.
   - At most  100 iterations with D* and the rest with D^.

 The iteration process stops when the stress is less than 0.001

 The smallest difference to be taken seriously is 0.0

 The euclidean metric will be used (Minkowski parameter 2).

 In addition to the standard results this listing will contain:

   - A report of the iteration process
   - A plot of the final configuration

 The following items will be stored in a separate file with the name:
   C:\MyData\Salad_Dressings.out
   - The final configuration


 The input data:
 ---------------
 Data type F marks the frequency variable.
 The input is supposed to contain the following variables:

 Var.   Name Type Line      <--- Missing values --->

     1 Dress1   P         1     0.
     2 Dress2   P         1     0.
     3 Dress3   P         1     0.
     4 Dress4   P         1     0.
     5 Freq     F         1     0.


 For a full row of 5 variables 1 line will be read.

 The pattern identification will be taken from columns 1 to 3 of each line.
```

Figure 19: Overview of the user options.

The next part of the listing gives feed back on the data and shows how they are interpreted. Whatever the original data type may be, the data will always be transformed into a matrix of rank orders. Figure 20 shows the feed back.

```
The preferences (stimulus numbers) will be read now:

First patterns: (frequencies in parentheses)

  1 C1      : (  2) 1      2      3      4

  2 C7      : (  1) 2      1      3      4

The number of patterns read is 12.


Matrix of preferences:
______________________


The first stimuli are the most preferred.

  Patterns    Freq.    Stimuli

  1 C1      : (  2)    1   2   3   4

  2 C7      : (  1)    2   1   3   4

  3 C8      : (  2)    2   1   4   3

  4 C9      : (  1)    2   3   1   4

  5 C13     : (  2)    3   1   2   4

  6 C14     : (  1)    3   1   4   2

  7 C18     : (  1)    3   4   2   1

  8 C19     : ( 11)    4   1   2   3

  9 C20     : (  6)    4   1   3   2

 10 C21     : (  3)    4   2   1   3

 11 C23     : (  1)    4   3   1   2

 12 C24     : (  1)    4   3   2   1
```

Figure 20: Feed back on the input data.

The following part shows the trial configuration before and after normalization and rescaling. See figure 21.

```
The trial configuration:
------------------------

A trial configuration for the patterns will be generated by the program.
A trial configuration for the stimuli will be generated by the program.

The trial configuration before normalization:

   Patterns      Dimensions

                  1            2            3

C1          1     0.10203     -0.29675      0.38675
C7          2     0.10108     -0.44205      0.02126
C8          3    -0.11071     -0.41105     -0.01781
C9          4     0.27661     -0.39005     -0.27347
C13         5     0.45404     -0.04747      0.16279
C14         6     0.41873      0.18082      0.19449
C18         7     0.38153      0.11853     -0.50480
C19         8    -0.18132      0.04554      0.04559
C20         9    -0.00484      0.24283      0.11636
C21        10    -0.18227     -0.09976     -0.31990
C23        11     0.17069      0.29482     -0.17837
C24        12     0.16975      0.14953     -0.54386

    Stimuli      Dimensions

                  1            2            3

Dress1      1    -0.02125     -0.01179      0.08411
Dress2      2    -0.02294     -0.18854     -0.04648
Dress3      3     0.29127      0.05146     -0.02119
Dress4      4    -0.08580      0.08917     -0.03515


The final trial configuration:

   Patterns      Dimensions

                  1            2            3

C1          1    -0.18253      1.23722      2.24825
C7          2    -0.36691      2.21725      0.42184
C8          3    -1.38278      1.76271      0.12763
C9          4     0.64034      2.43083     -1.09784
C13         5     2.01634      0.68435      1.03529
C14         6     2.19211     -0.53665      1.04779
C18         7     1.99912      0.20243     -2.59252
C19         8    -1.03125     -0.67927      0.15264
C20         9     0.16038     -1.44572      0.45936
C21        10    -1.21563      0.30077     -1.67377
C23        11     1.16763     -1.23214     -1.06033
C24        12     0.98325     -0.25211     -2.88674

    Stimuli      Dimensions

                  1            2            3

Dress1      1    -0.31784     -0.17198      0.44325
Dress2      2    -0.58645      0.80205     -0.13210
Dress3      3     1.37386      0.06500     -0.04486
Dress4      4    -0.46958     -0.69507     -0.26629
```

Figure 21: The trial configuration.

Then follows a report of the main computations, as shown in figure 22.

```
Four salad dressings

==========================================================================

   2. Looking for a solution in 3 dimensions.

==========================================================================

 Algorithm switches to monotone regression because stress is less than 0.001

 Iterations stop because stress is less than 0.001

 Performing a principal axes rotation in 3 dimensions.

 Stress D* is   0.0000     after     4 iterations.
 Stress D^ is   0.0000     after     2 iterations.


 Four salad dressings

==========================================================================

   3. Looking for a solution in 2 dimensions.

==========================================================================

 Minimizing raw stress ('soft squeeze').

 Fitting values are rank images (D*).

 Numbered lines:   computation of fitting values and evaluation of stress.
 Unnumbered lines: moving the points in the configuration.
 '#':              indicates that the gradient angle is used.


  Iter   Stress D^      Stress D*   Coef.alienation   Cosinus of
                                         D*          Gradient angle
  ____________________________________________________________________

    1    0.19150        0.36913
     #                  0.33094         8.4774         0.0000
     #                  0.31686         1.3782        0.34863E-01
     #                  0.31070        0.29609        0.42662
     #                  0.30621        0.12398        0.52542
     #                  0.30636        0.62250E-01    0.27045
    2    0.18767        0.35930
     #                  0.36567         2.5711       -0.19406
    3    0.16443        0.32636
     #                  0.32365         2.6649        0.78581
     #                  0.32956         1.0492       -0.35038
    ....... and so on
   36    0.51321E-03    0.10264E-02
     #                  0.99114E-03    0.36306E-04    0.30714
     #                  0.98481E-03    0.37328E-05    0.79955
     #                  0.98334E-03    0.14685E-05   -0.69229E-01
     #                  0.98275E-03    0.55441E-06   -0.13271
     #                  0.98249E-03    0.21338E-06   -0.92163E-01
   37    0.42984E-03    0.85967E-03

 Algorithm switches to monotone regression because stress is less than 0.001

 Minimizing standardized stress ('hard squeeze').

 Fitting values are based on monotone regression (D^).

 Iterations stop because stress is less than 0.001
```

Figure 22: The main computations.

If a detailed report of the iteration process is requested the listing contains at each iteration step the iteration number or #, the stress $\hat{d}$, the stress d^*, the coefficient of alienation and the cosine of the angle between successive gradients (the correlation between the two gradients). The coefficient of alienation is an overall measure of the decrease of the stress function: $\sum_{k=1}^{dim}\sum_{i=1}^{N} G_{ik}^2 + \sum_{k=1}^{dim}\sum_{j=1}^{N} H_{jk}^2$

The lines with the iteration number contain only the two stresses as a result from the computation of the fitting values (phase 2) and evaluation of the stress. The lines preceded by # contain the stress d^*, the coefficient of alienation and the cosine of the gradient angle as the result of moving the points in the configuration (phase 1).

The next part of the listing gives the final results, as shown in figure 23.

```
Performing a principal axes rotation in 2 dimensions.
Stress D* is  0.42984E-03 after    39 iterations.
Stress D^ is  0.85967E-03 after    39 iterations.


Final configuration
___________________

   Patterns       Dimensions

                   1           2

C1         1    -0.30260      0.93271
C7         2    -0.69445      1.68012
C8         3    -1.45655      1.10503
C9         4     0.33119      1.91531
C13        5     1.33685      1.05121
C14        6     1.54912      0.88366
C18        7     1.55423      0.36605
C19        8    -0.45249     -0.69228
C20        9     0.35657     -1.00548
C21       10    -0.89431     -0.65104
C23       11     0.99696     -0.85628
C24       12     1.40333     -1.17866

If patterns are not weighted by their frequency:

Mean:             0.31065      0.29586
St.Dev.:          1.01425      1.06412

If patterns are weighted by their frequency:

Mean:            -0.03828     -0.20666
St.Dev.:          0.82422      0.92133


    Stimuli       Dimensions

                   1           2

Dress1     1    -0.70849      0.28712
Dress2     2    -0.71105      0.28875
Dress3     3     1.39878      0.30529
Dress4     4     0.02077     -0.88115

Mean:             0.00000      0.00000
St.Dev.:          0.86089      0.50878
```

Figure 22: The iteration process and the resulting configuration.

Finally a plot of the final configuration is shown. In the listing file this plot is simply build as a grid of characters. See figure 24. A graphical equivalent is given in a separate bitmap file.

```
 Plot of the final configuration
 _______________________________


           The stimuli are marked with "=".

 Dimensions 1 and 2.                          Scaling factor = 100 / 1.9153
      -100   -80   -60   -40   -20     0    20    40    60    80   100
       ..+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.
   100 :                                ::     4                        :  100
    96 :                                ::                              :   96
    91 :                                ::                              :   91
    86 :                       2        ::                              :   86
    81 :                                ::                              :   81
    77 :                                ::                              :   77
    72 :                                ::                              :   72
    67 :                                ::                              :   67
    62 :                                ::                              :   62
    58 :          3                     ::                    5         :   58
    53 :                                ::                              :   53
    48 :                             1  ::                        6     :   48
    43 :                                ::                              :   43
    39 :                                ::                              :   39
    34 :                                ::                              :   34
    29 :                                ::                              :   29
    24 :                                ::                              :   24
    20 :                                ::                         7    :   20
    15 :                   =1=          ::                       =3=    :   15
    10 :                                ::                              :   10
     5 :                                ::                              :    5
 Dim. 1 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::    0
    -5 :                                ::                              :   -5
   -10 :                                ::                              :  -10
   -15 :                                ::                              :  -15
   -20 :                                ::                              :  -20
   -24 :                                ::                              :  -24
   -29 :                                ::                              :  -29
   -34 :                 10             ::                              :  -34
   -39 :                        8       ::                              :  -39
   -43 :                                ::               11             :  -43
   -48 :                               =4=                              :  -48
   -53 :                                ::     9                        :  -53
   -58 :                                ::                              :  -58
   -62 :                                ::                    12        :  -62
   -67 :                                ::                              :  -67
   -72 :                                ::                              :  -72
   -77 :                                ::                              :  -77
   -81 :                                ::                              :  -
   -86 :                                ::                              :  -86
   -91 :                                ::                              :  -91
   -96 :                                ::                              :  -96
  -100 :                                ::                              : -100
       ..+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+.
      -100   -80   -60   -40   -20     0    20    40    60    80   100

 Overlap:  +++   =1=  =2= +++

 A graphical representation of the plot of dimension 1 against dimension 2
 is given in file:
 C:\Kunst2012\Unfold\TestUnfold\Salad_DressingsP.bmp

 ======= Normal end of analysis =======
```

Figure 24: Plot of the configuration.

A graphical equivalent is given in a separate bitmap file, as shown in figure 25.
If the solution has more than 2 dimensions a plot will be shown for each pair of dimensions.

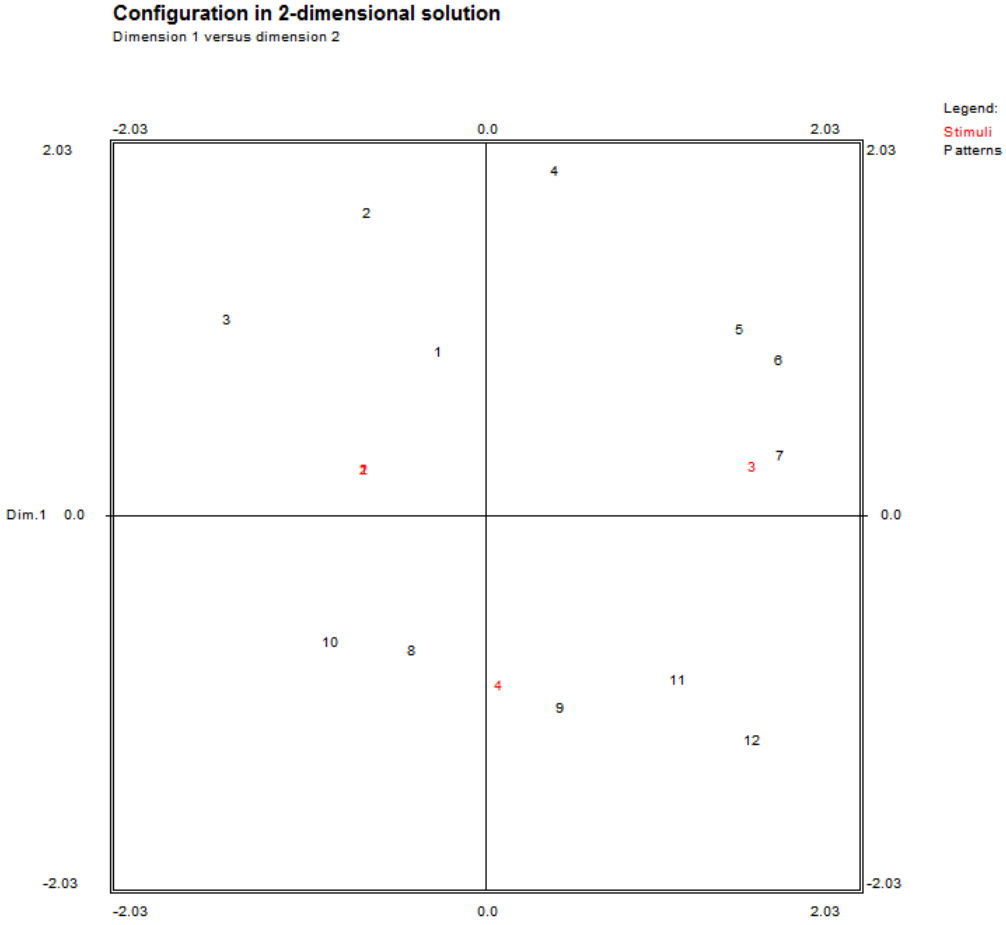

Figure 25: Plot of the final configuration in a bitmap file.

10.2 Results in the raw output file

If any "raw" output is asked (the option `W` in the *main* menu) it will be written to the raw output file. Depending on the chosen option the output may contain three parts:
(1) the fitting values, (2) the distances between cases/patterns and stimuli and (3) the final configuration. The three parts are given in this order. If solutions are given for different dimensionalities the three parts are given for each dimensionality from highest to lowest ; first all parts for the highest dimensionality, then all parts for the next dimensionality and so on.

Part 1, the fitting values, starts with a line like this:
'`>> Fitting values in 2 dimensions: ( 1 line per pattern)`'
Then follows one row for each case or pattern. Each row consists of one or more lines. The first line contains an identifying text in the first eight positions, followed by five groups each consisting of a stimulus number of 4 positions, a space and a fitting value of 10 positions (including 4 decimal digits). The first number starts at position nine. If there are more lines per row, the second and following lines start with eight spaces again followed by five groups starting at position nine. The stimulus numbers are important because the fitting values are given in increasing order.

Part 2, the distances between cases/patterns and stimuli starts with a line like this:
'`>> Distances between patterns and stimuli in 2 dimensions: ( 1 line per pattern)`'
Then follows one row for each case or pattern. Each row consists of one or more lines. The first line contains an identifying text in the first eight positions, followed by seven fitting value of 10 positions (including 4 decimal digits). The first number starts at position nine. If there are more lines per row, the second and following lines start with eight spaces again followed by seven values of 10 positions starting at position 9.

Part 3, the final configuration, starts with a line like this:
'`>> Final configuration in 2 dimensions: (first the patterns, than the stimuli)`'
Each row consists of one or more lines. The first line contains an identifying text in the first eight positions, followed by seven fitting value of 10 positions (including 4 decimal digits). The first number starts at position nine. If there are more lines per row, the second and following lines start with eight spaces again followed by seven values of 10 positions starting at position 9.

Figure 26 shows a shortened version of the raw output file corresponding to the example analysis

```
>> Fitting values in 2 dimensions: ( 1 line per pattern)
C1         1     0.7626   2     0.7626   3     1.8134   4     1.8425
C7         2     1.3915   1     1.3931   3     2.5043   4     2.6593
C8         2     1.1055   1     1.1084   4     2.4754   3     2.9652
.....
C23        4     0.9765   3     1.2291   1     2.0533   2     2.0563
C24        4     1.4142   3     1.4840   2     2.5722   1     2.5722
>> Distances between patterns and stimuli in 2 dimensions: ( 1 line per
pattern)
C1          0.7626    0.7626    1.8134    1.8425
C7          1.3931    1.3915    2.5043    2.6593
C8          1.1084    1.1055    2.9652    2.4754
.....
C23         2.0533    2.0563    1.2291    0.9765
C24         2.5707    2.5737    1.4840    1.4142
>> Final configuration in 2 dimensions: (first the patterns, than the stimuli)
C1         -0.3026    0.9327
C7         -0.6945    1.6801
C8         -1.4565    1.1050
.....
C23         0.9970   -0.8563
C24         1.4033   -1.1787
Dress1     -0.7085    0.2871
Dress2     -0.7111    0.2887
Dress3      1.3988    0.3053
Dress4      0.0208   -0.8812
```

Figure 26: Shortened version of the raw output file.

11 Literature

Bezembinder, Thom.G.G., Van rangorde naar continuum, een verhandeling over datastructuren in de psychologie, Van Lochum Slaterus, Deventer, 1970.

Carroll, J.D. and Arabie, P., *Multidimensional scaling*, in Rosenzweig, M.R. and Porter, L.W., (Eds.), Annual Review of Psychology, Palo Alto, California, Annual Reviews, Inc.,1980, 31, 607-649.

Coombs, C.H., *A theory of data*, Wiley, New York, 1969.

Coxon, A.P.M., *The mapping of family-composition preferences: a scaling analysis*, Social Science Research, Vol 3, pp 191-210, 1974.

Davidson, J.A., A geometric analysis of the unfolding model: nondegenerate solutions, Psychometrika, Vol 3, pp. 193-216, 1972.

Frisenfeldt Thuesen, Kristine, Analysis of ranked preference data, 42, 2007, http://www.google.nl/#hl=nl&sclient=psy-ab&q=frisenfeldt+thuesen&oq=frisenfeldt+thuesen&gs_l=hp.12...29202.29202.1.30934.1.1.0.0.0.0.181.181.0j1.1.0...0.0...1c.IdzpiPaNlVg&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.&fp=6960b29b621983be&biw=1280&bih=832

Greene, P.E. and Carmone, F.J., Multidimensional scaling: an introduction and comparison of nonmetric unfolding techniques, Journal of Marketing Research, Vol 6, pp. 330-341, 1969.

Greene, P.E. and Rao, V.R., *Applied multidimensional scaling*, Holt Rinehart and Winston, New York, 1972.

Kruskal, J.B., *Nonmetric multidimensional scaling: a numerical method*, Psychometrika, Vol 29, pp. 45-129, 1964a.

Kruskal, J.B., Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika, Vol 29, pp. 1-27, 1964b.

Lingoes, J.C., Roskam, E.E.Ch.I., & Borg, I.,, *Geometric representations of relational data*, Ann Arbor: Mathesis Press, 1979.

Roskam, E.E.Ch.I., *Nonmetric data analysis: general methodology and technique with brief description of mini-programs*, Internal Report 75-MA-13, Dept. of Mathematical Psychology, University of Nijmegen, The Netherlands, 1975.

12 Index