Readme irMF V4.3.1

Revision date: August 23, 11
Contact Information: genetree@bellsouth.net, paul.fogel@wanadoo.fr

Contents

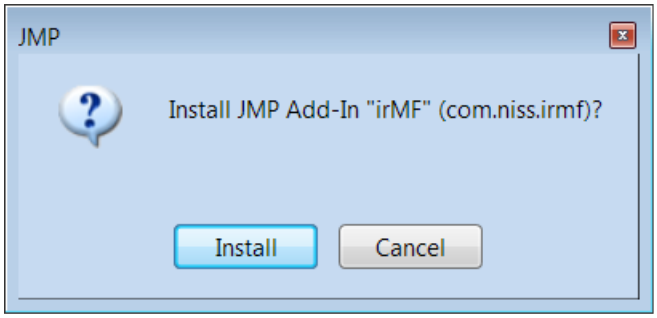
1. Installation and Update

1.1 Installation

Note: irMF V4.3.0 runs only under MS Windows and operates as a SAS JMP script (version 9.0.0 or higher).

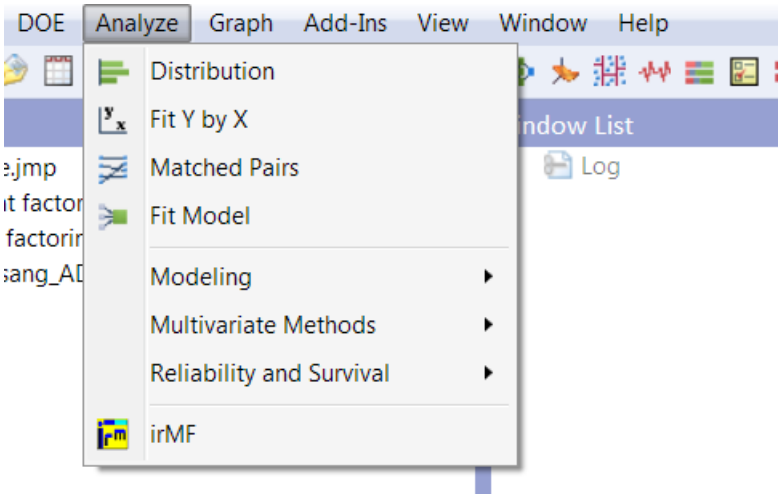
irMF 4.3.0 is encapsulated in a jmp addin setup file named `irMF.jmpaddin`. Simply double-click this file

and the add-in will be automatically installed into JMP. A dialog will pop up to confirm installation.



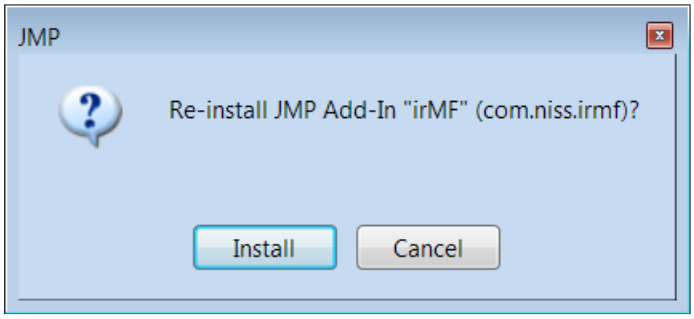
You should now see the irMF icon at the end of the *Analyze* tool bar.

irMF option menu appears also in jmp main menu at the end of the *Analyze* menu.



1.2 Update

To install an update of irMF, follow the same procedure as for installation. A dialog will pop-up to confirm reinstallation.

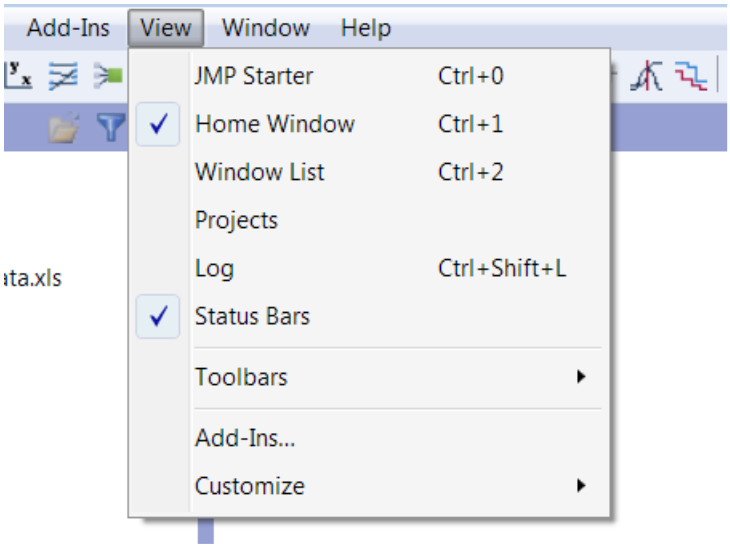


1.3 Uninstallation

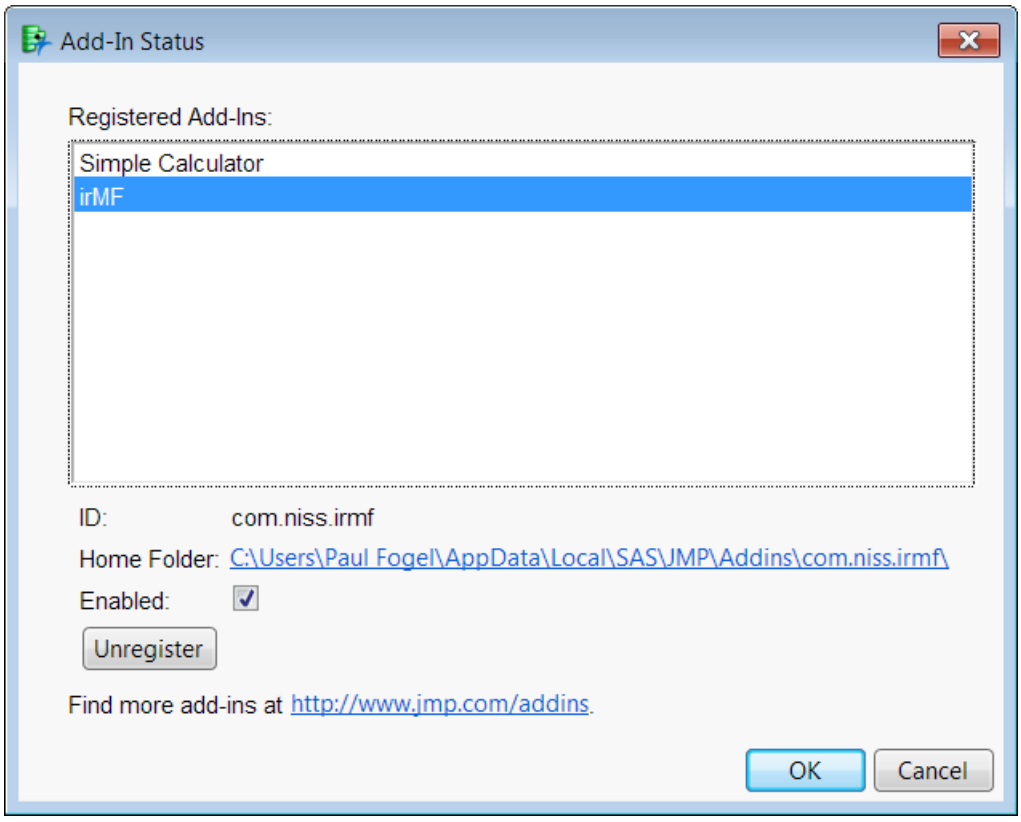
If you wish to uninstall irMF, select the menu *Add-ins* (under the *View* option in JMP main menu):

1.4 Professional versus Limited version

Certain tabs are inactivated in the limited version which is downloadable from NISS site. Please contact authors in order to receive the “Pro” unlimited version.



A list of registered add-ins is displayed:



Select irMF add-in and click the button Unregister.

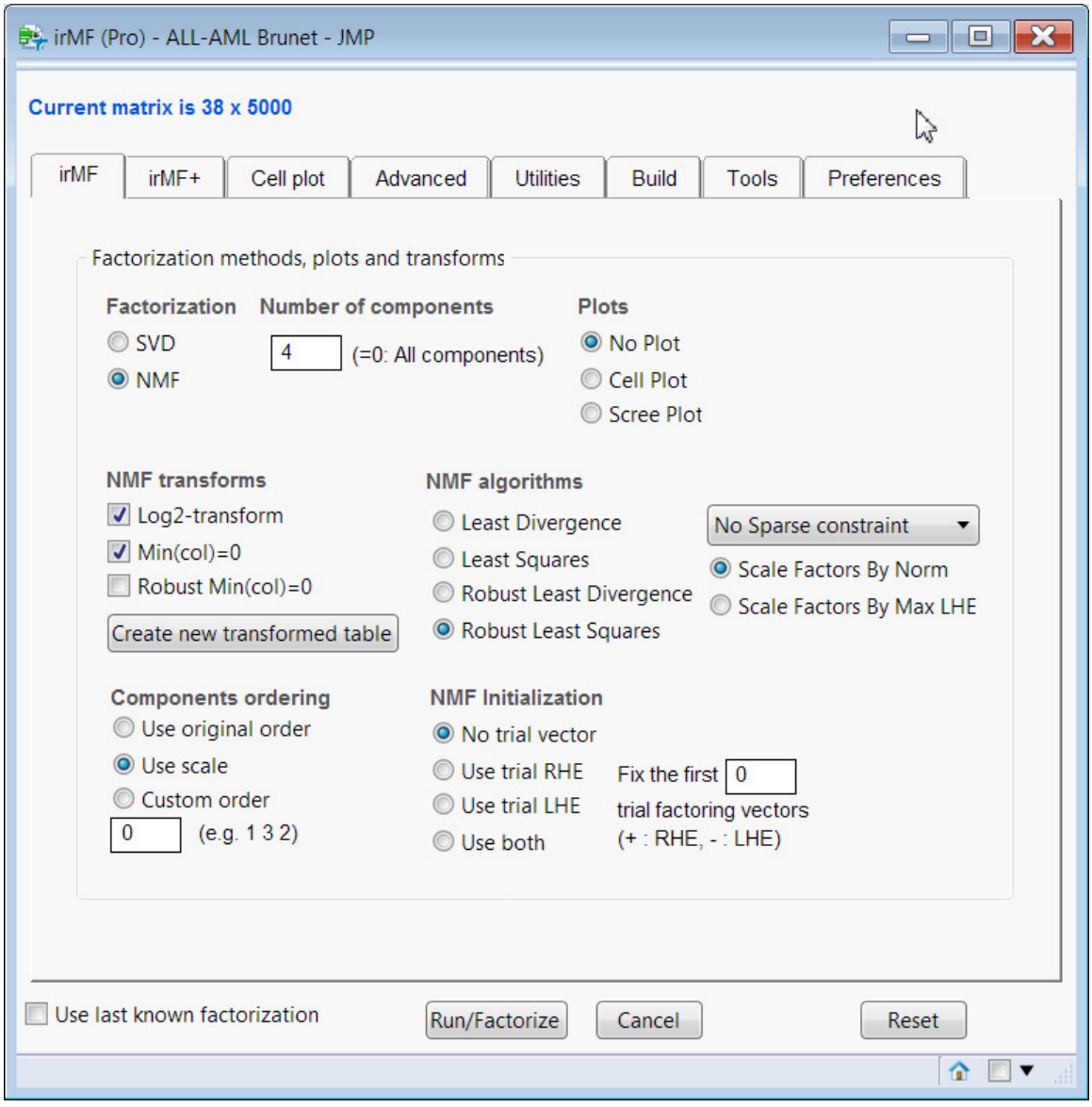
2. Data file format

The following data file format must be strictly followed:

Column 1: sample label
Column 2: group label (= sample label if no group information available)
Other columns: variables/predictors must be numeric. All will be used in the analysis.

Both the sample label and the group label must have character type.

3. irMF main dialog tab



Type of Matrix Factorization: irMF provides two types of matrix factorization: Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF). Note that only SVD accepts matrices with missing cells. Robust SVD is also implemented, following Liu & al (2003). We suggest applying robust SVD (rSVD) and

clean up data (see Build tab) by replacing missing cells and outliers by rSVD model-fitted values prior to running standard NMF.

Note: Options that are specific to the chosen factorization method are displayed. Options that are specific to the un-chosen factorization method are hidden.

Number of components: Number of components to be used in the factorization model must be chosen. The scree plot can be used to help you chose this number.

Note: Each NMF component defines a cluster of rows or columns, thus a too high number of components may lead to empty clusters (i.e. the resulting classification 0function does not cluster any row or column into this cluster). In such case, irMF sends a warning or stops trying higher component models when a scree plot is requested.

Plots: Two types of plots are available: Cell Plots or Scree Plots. Additional options for Cell Plot can be found in the 'Cell plot' tab. Additional options for the Scree plot can be found in the 'Advanced' tab.

NMF scree plots

Volume:
We introduce a novel complementary test that takes advantage of the non-orthogonality of NMF factoring vectors. For a number of components r, we calculate the volume of a matrix Z having r columns $\hat{X}_k, 1 \le k \le r$, where $\hat{X}_k$ is the approximation to X obtained with k components, reshaped into a column vector and normalized. To calculate the volume, we take the determinant of Z'Z. irMF provides a simultaneous plot of:
(i) The volume achieved with models corresponding to 1, 2, ..., r components (max volume = 1 means that components are orthogonal). Volume decrease indicates that components are correlated.
(ii) Mean square residuals (MSR), which are normalized to have max MSR = 1 and are overlaid with volume.

The volume criterion can evolve in different ways, depending on the nature of underlying mechanisms:

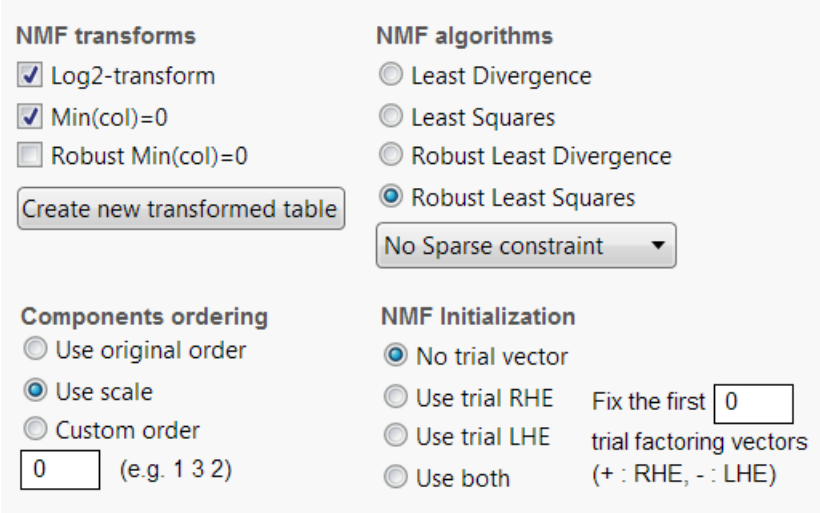
- Independent mechanisms (e.g. data is a mixture of independent sources): Volume remains stable as long as the number of components is smaller than the number of mechanisms. The volume shows a sharp decrease once the number of components exceeds the number of mechanisms.
- Dependent mechanisms (e.g. metagenes with common pathways): Components being associated with mechanisms are correlated. Therefore, the volume decreases substantially until all existing mechanisms can be associated with their own component. Volume stops decreasing (or keeps only decreasing slightly) when extra components which explain noise rather than signal are added, due to the orthogonality of added noise.

Robustness:

The stochastic nature of the *robust* version of the NMF algorithm is described in section 4.1.1, options for NMF. This algorithm provides a means to assess whether a given rank r provides a biologically meaningful decomposition of the data. Each sample being clustered with a given frequency into a particular cluster, the mean frequency over all samples is an indicator of the robustness of the clustering: the higher the mean frequency, the more consistent the clustering across all runs. Similarly, an indicator can be constructed regarding the robustness of the clustering of variables. Both indicators tend to become unstable when r becomes too high.

Note: NMF scree plots are time-consuming. However, in datasets with numerous redundant variables, like microarrays, experience shows that there will be little difference if NMF scree plots are obtained from a smaller subset of variables sampled at random.

! In the following, we assume that NMF option has been selected



NMF transforms: Min(col)=0 removes the lowest value from each column so all columns become positive and the smallest value in each column equals 0. Both options are recommended if the dataset is log-distributed and Least Square method is used. The option Robust Min(col)=0 prevents outliers from distorting the profile of column minimums, by projecting the original profile of minimums onto the profile of column 10% quantiles. Using the latter option can yield negative values in the transformed matrix, which are all replaced by 0 values.
Create transformed table: Use this option to output a table with transformed values.

NMF algorithms: Choose between Least divergence, Least squares, Robust Least divergence or Robust Least squares. The robust approach is described below. We recommend using Least squares, unless the data is Poisson distributed, in which case Least divergence is recommended (although much slower than LS).

Sparse factoring vectors: A sparse vector has many elements at or near zero. Sparse factoring vectors are potentially useful: sparse vectors are easier to interpret and it is

unlikely that all predictors are involved in a specific mechanism. Our approach is simple and effective: We first run a standard NMF and then subtract, for each factoring vector, the minimum of the vector elements to let their minimum be 0. The obtained factoring vectors are then used as fixed initial vectors to a second-step NMF which is automatically run.

Scaling options: The default option is to scale factoring vectors by L1 or L2 norm, depending on the algorithm. It is possible to scale by the max element of each left factoring vector in order to have left factoring vector elements ranging from 0 to 1 (as done in score plot visualizations).

Components ordering: NMF does not guarantee that factoring vectors are found in decreasing order of their respective scale. Check the option 'Keep vectors in original order' if you want factoring vectors in the order of computation. Check the option 'Use scale' if you want factoring vectors to be ordered by descending scale. Check the option 'Custom order' if you want factoring vectors to appear in a specified order (e.g. if you want one particular cluster to appear next to another one).

Note: The Custom order option can be used along with the 'Use last known factorization' option in the following way: (i) Run NMF with scale ordering and with 'Use last known factorization' option unchecked (ii) Based on the output, enter the most appropriate components ordering, e.g. 2 1 4 3 (iii) Re-run NMF this time with 'Use last known factorization' option checked (this option will be automatically checked after entering a custom ordering).

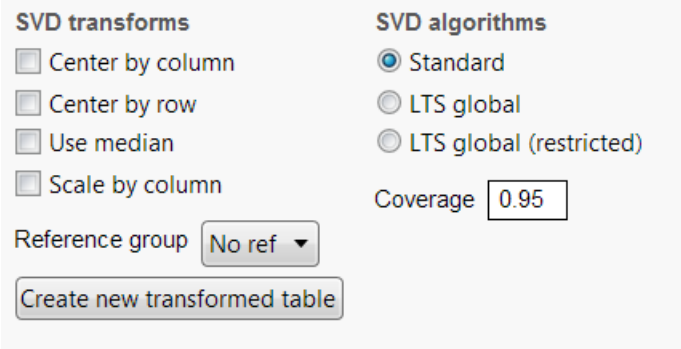
NMF initialization: By "NMF initialization", irMF means the process of running NMF twice, the first time to initialize the model (using "options for NMF" as described above), and the second time to impose some additional constraints on the factorization.

Trial factoring vectors: irMF allows the user to initialize the factoring vectors. Run first irMF with no trial vectors. Examine the factoring vectors output table Right and/or Left factoring vectors, make all desired changes and choose between one of the options 'Use trial RHE', 'Use trial LHE' or 'Use both' (RHE/LHE stand for Right/Left Hand factoring vEctors). If there are specific "known" factoring vectors, they can be fixed provided that they appear first in the factoring vectors output table. In this case, the sign convention determines whether left or right vectors should be fixed when both trial LHE and RHE are used. If necessary permute columns of the table of factoring vectors to have fixed factoring vectors first in the table. Only non-fixed trial vectors will be updated. irMF assumes that trial factoring vectors are equally important so scales are all initialized to one before the fitting process starts.

Note: If some of the trial vectors are fixed, the option 'Keep factoring vectors in original order' (see below) is automatically turned on.

Note: Another strategy is to take the effects of "known" factoring vectors out of the matrix. The residual matrix can be obtained from the "build" tab (see below) and then be used as input to irMF to further analyze the data set.

! In the following, we assume that SVD option has been selected



SVD transforms: These options correspond to standard normalization procedures. The default is no centering. If option 'By row' or 'By column' is checked, then the default is to use the mean of rows or columns. The median is used instead if the option is checked. A particular group can be selected in order to normalize rows. In such case, the mean or median of this group will be subtracted.
Build transformed table: Use this option to output a table with transformed values.

SVD algorithms: Standard and two robust SVD options are currently implemented, following Liu & al (2003). Standard SVD is computed using alternating least squares. Least trimmed squares replaces the least squares regression by using the k data points that have the lowest sum of squared residuals.

! The following boxes are displayed at the bottom of the dialog if any of the first four tabs is activated.

The option **Use last known factorization** is accessible from all tabs: Last factorization results can be used to bypass calculations and apply changes in output options. This option is particularly useful with large datasets.

Run/Factorize: Run irMF with selected options.

Cancel: Disregard selected options and close irMF dialog.

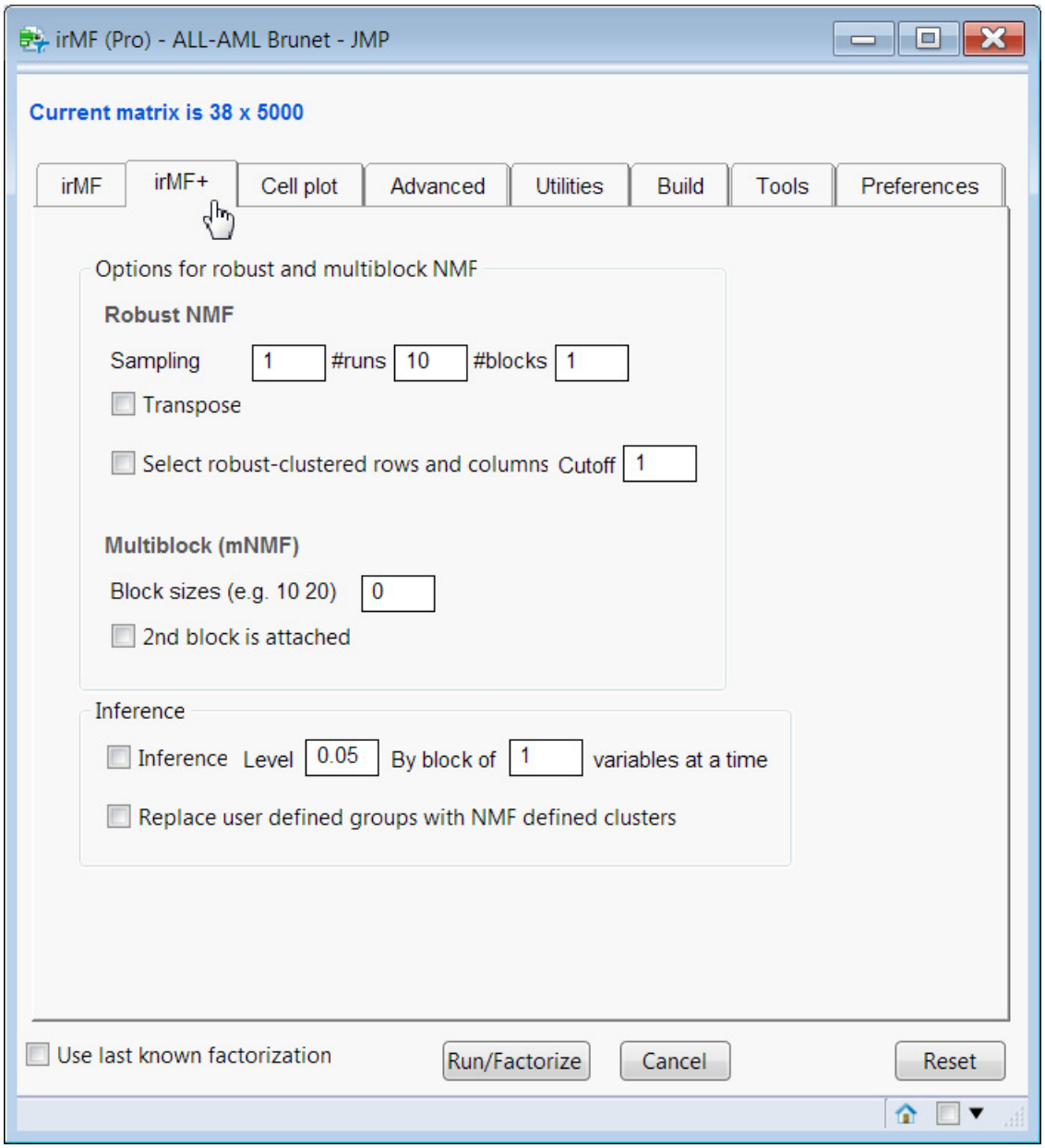
Reset: Click this button to clear the irMF environment (tables and specific namespace variables). This option is particularly useful to reinitialize NMF models that are stored in memory.

4. Other dialog tabs

4.1 irMF+ tab

! Inactivated in limited version

The irMF+ tab has two panels: “Options for robust and multiblock NMF” and “Inference”. This tab is accessible only if PRO version is activated.



4.1.1 Options for robust and multiblock NMF

Robust NMF: The stochastic nature of most clustering algorithms has been shown to be rather useful in providing methods for evaluating the consistency and robustness of their

performance (Devarajan, 2008). The central idea is to perform numerous runs of the algorithm starting from different random initializations. On each run, the algorithm groups the samples into clusters, allowing for the calculation of the frequency at which two different samples fall into the same cluster. In contrast to random initialization, we first calculate a pseudo-unique NMF solution based on SVD initialization, which is itself unique. In order to evaluate the robustness of the clusters, samples are bootstrapped and right factoring vectors are re-estimated on each run. Since different right factoring vectors give rise to different clusters of variables, the frequency at which a variable falls into a particular cluster can be calculated, the highest frequency determining the most reliable, robust cluster for any variable. The right factoring vectors which are obtained on each run of the bootstrap can be used in the reverse way to re-estimate left factoring vectors, which in turn determine sample clusters. Similarly, the frequency at which a sample falls into a particular cluster can be calculated, the highest frequency determining the most reliable cluster for any sample. Note that other algorithms using random initialization do not guarantee any consistency in the ordering of the clusters, i.e. the same cluster can appear as number 1 or 3 depending on the initialization. As a consequence, it is not possible to assess directly the frequency at which a sample falls into a particular cluster, as we do here.

The three options below are all related with the robust algorithm:

- Sampling: Columns can be sampled to obtain initial estimates of the left factoring vectors more rapidly. These initial estimates will further be used to update right factoring vectors. This option is particularly useful when the number of columns is very large and variable redundancy occurs. (For example: 1.0 complete sampling; 0.5 50% sampling.)
- #runs: Enter here the number of bootstrap runs to be used by the robust NMF algorithm.
- #blocks: When the number of columns is very large, it may be necessary to update right factoring vectors in a piecewise manner in order to prevent memory overflow. Note that for each block, only the variables in the block of the matrix of the matrix to be factorized are used, ignoring other parts.
- Transpose: Check this option if you want to swap roles between columns and rows during the robust estimation process. This option is disabled if multiblock NMF is used or trial factorial vectors are used (see below).

mNMF (Multiblock NMF): Since version 3.0.6, irMF has implemented a new algorithm for multiblock NMF called mNMF, which can be (roughly) thought of as an adaptation of CPCA-W (Smilde et al., 2003) using an NMF factorization module instead of SVD. Enter block sizes, separated by a comma or a blank - irMF assumes that blocks are subsets of contiguous columns.
mNMF produces one table of ordered right factoring vectors (“profile” table) for each block.
Notes:

- Left factoring vectors are common to all blocks, thus only one clustering of samples is produced.
- Cell plots are organized by block.

2nd block is attached: Check this option if the current table has 1/X or the positive part of –X attached to X (after using the build option 'Attach element-wise inverse' or 'Attach pos and neg parts', see Build section). Note that irMF will reverse colors in the cell plots for the attached block.

Select robust-clustered rows and columns: Check this option to select rows and columns which are most of the time clustered into the same cluster. The minimal frequency for consistent clustering can be tuned by the cutoff.

4.1.2 Inference

Inference: Assume that rows of the matrix correspond to samples of different types or classes (e.g. control, disease or drug groups) and columns correspond to response variables. As it is likely that only a limited number of response variables are linked to each class label, we are interested in finding such variables or "predictors" that would allow predicting the class of any sample. Matrix factorization is used to create ordered sets of response variables. Here each set corresponds to a particular right factoring vector, which has been ordered by decreasing values (absolute values for SVD) of its elements. It is important to note that the ordering is totally unsupervised (the sample group information is not used). In order to identify predictors, we take advantage of the ordering of variables within each set and test each variable sequentially so there needs be no correction for multiple testing (since the ordering is not supervised). We apply ANOVA on each variable to test differences between groups. The testing process stops whenever the null hypothesis is not rejected at the defined level, i.e. variables that appear further in the ordered list are not tested. irMF allows testing blocks of consecutive variables, instead of single variables (option 'By block of') in order to prevent that the procedure stops at an early state. To calculate P-values with respect to blocks, P-values of single variables are pooled using Fisher's method.

Replace user defined groups with NMF defined clusters
By contrast with the traditional approach of running first a supervised ANOVA in order to reduce the number of variables before, before applying a non-supervised clustering approach, such as Hierarchical Clustering, irMF can perform a "blind" selection of variables. When the option is checked, NMF clusters – which are found in a totally non-supervised way – replace group information in the ANOVA. The selection can subsequently be used to obtain a clear Heatmap where only variables of interest are displayed, i.e. "noisy" genes, with respect to the achieved clustering, are left out of the heatmap.

Note that if clusters are highly associated with groups, the "blind" selection will be very close to a supervised one. In such case, we suggest the following procedure for selecting variables:

(i) Apply a blind selection using sequential ANOVA, pooling large blocks of consecutive p-values (e.g. block size = 5, see Inference in section irMF main

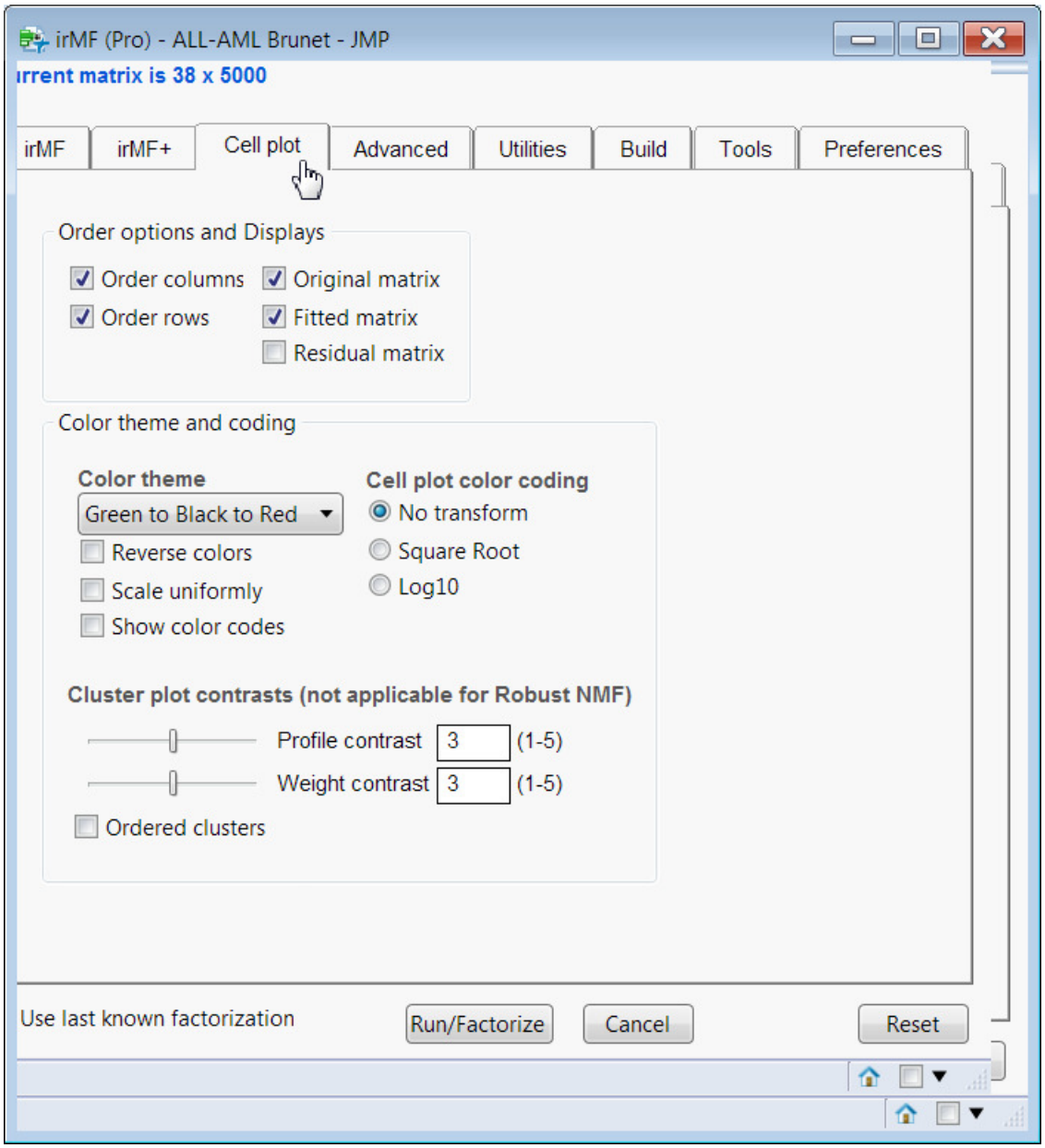
dialog) to obtain a list of candidate variables that is large enough to minimize the risk of leaving out interesting ones.

(ii) Uncheck the option, i.e. apply sequential ANOVA this time with real group information, restricting the set of candidate variables to the blind selected ones.

The main advantage of using this procedure over standard sequential ANOVA is that we expect the ordering of candidate variables to be improved once noisy variables are left out of the dataset. Thus, smaller blocks of consecutive p-values can be used in step ii) of the procedure (e.g. block size = 3).

Note that Inference can not be activated when Scree plot is checked.

4.2 Cell plot tab



Order options and Displays: Rows and columns of the original matrix can be reordered by decreasing values of the elements of the right and/or left factoring vectors. Thus there are as many possible permutation matrices as there are components in the model. These permutations may apply to original, fitted, or residual matrices. The permuted matrices can be displayed in the form of a cell plot (heat map).

Color coding: The color coding is based on the standard deviation of the data columns. If the cell plots appear uniformly green or red, then try using log-normal or square-root transform. Note that the transform applies only to the way colors will be coded in the cell

plot. The matrix factorization itself applies to original or normalized data if these options are checked in the main dialog tab.
Note: This option is disabled if at least one of the NMF normalization scheme (irMF tab) is checked.

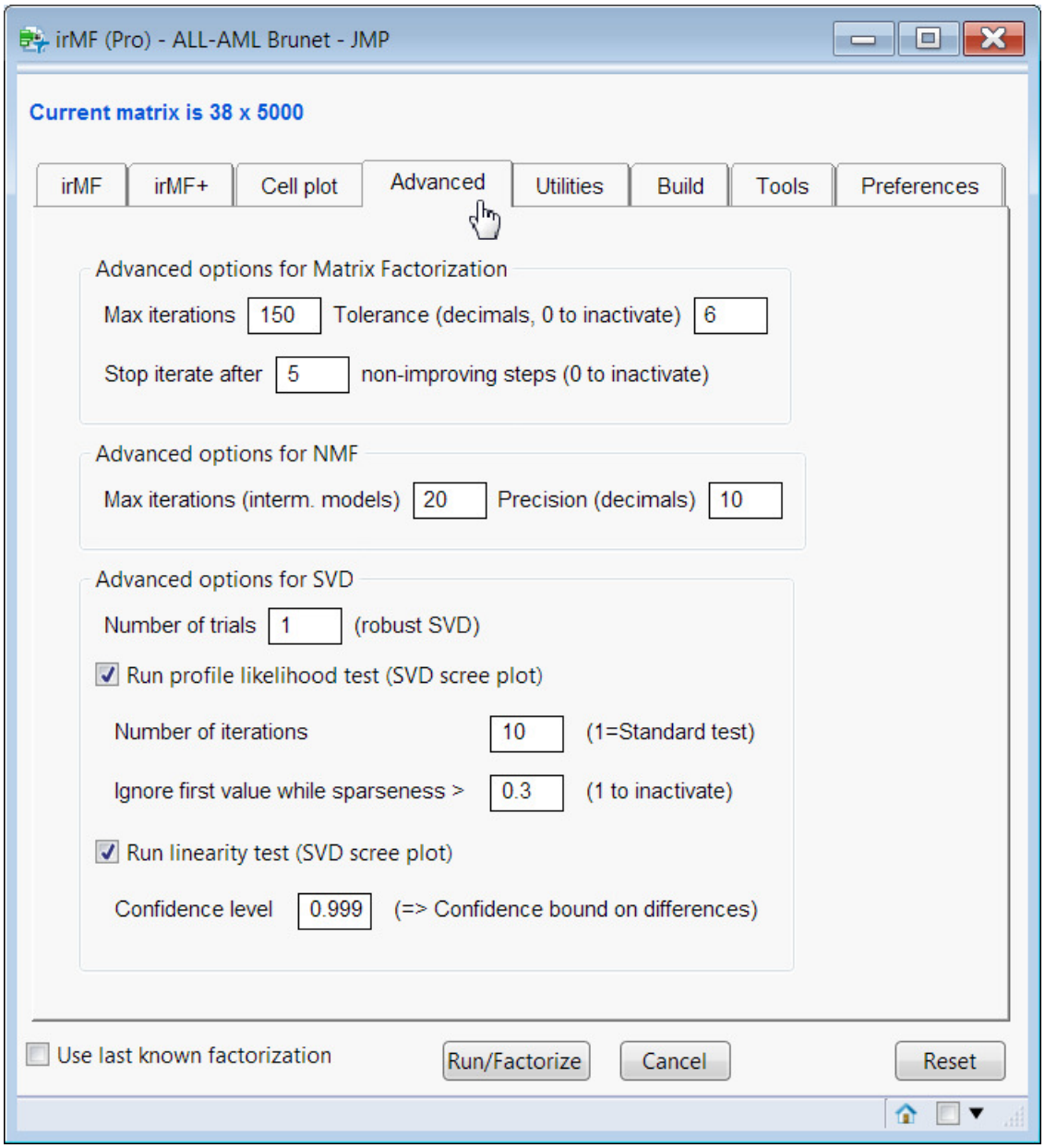
Cluster plot (NMF): The cluster plot is specific to NMF. NMF can be used to cluster a data set. Consider the left factoring vectors. Each row is assigned the cluster number of the left component with the highest element. Likewise the columns can be assigned the number of the right component with the highest element so the matrix can be double clustered. The matrix is permuted in the order of decreasing values of the elements of the factoring vectors, cluster by cluster.

Note: In the special case of 2 clusters, ascending order is used to order the elements of the second factoring vector. This way, a continuous map is created in both directions, e.g. on the X-axis going progressively from most up- to most down-regulated genes.

Profile and weight contrasts: The contribution of a row to a particular cluster is defined by the ratio of the left component element associated with the cluster over the mean of the other left component elements. Likewise, the contribution of a column to a particular cluster can be calculated. When components are sparse, it may be useful to enhance the contrast using the slider box.

Ordered clusters: Check if clusters have been ordered in order to achieve a continuous change in patterns. The ordering of rows and columns within each cluster will be affected. However, the clustering itself remains unchanged.

4.3 Advanced tab



Advanced options for Matrix Factorization:

- **Max iterations**: 200 are recommended, although convergence is generally ensured within 100 iterations. If least divergence is used, the actual max number of Li-Seung iterations is multiplied by 10, since convergence is known to be much slower.

- **Tolerance**: This parameter controls the convergence; we stop when the Mean Square error does not change more than the tolerance level.

Advanced options for NMF:

- **Max iterations (intermediate models):** When the least square method is used, this parameter defines the number of Lee-Seung preliminary iterations that are necessary to initialize projected gradient.

- **Precision**: When the least divergence method is used, 0 values will be replaced by a small positive value 1e-**Precision**. Replacing 0 values is important to prevent calculation underflow.

Advanced options for SVD:

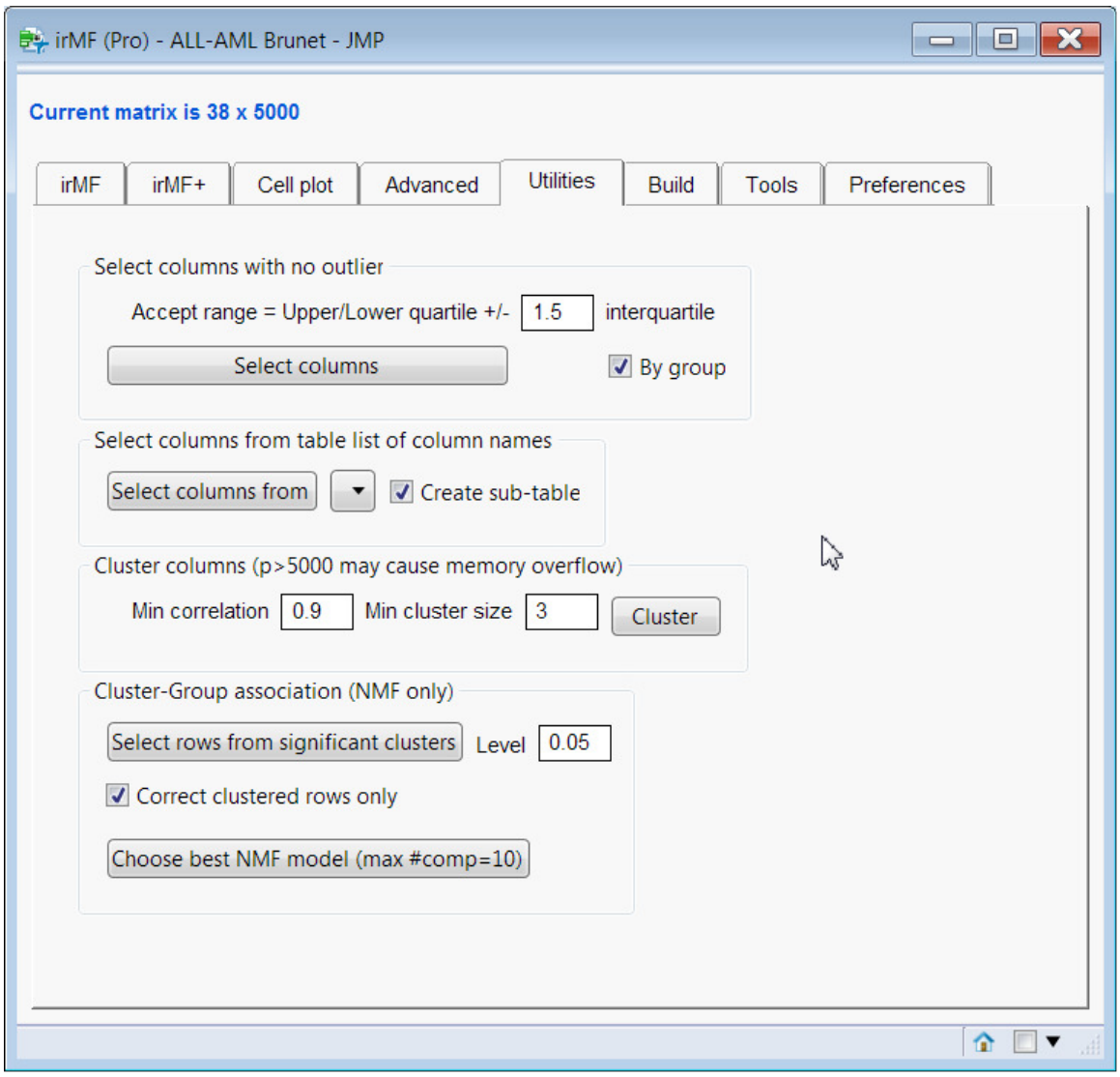
- **Number of trials**: Several trials can be performed when using the robust algorithm, which is initialized in a stochastic way. The best solution will be returned. Our experience is that using only one trial is generally enough to get a solution close to optimal.

- **Profile likelihood and linearity tests**: In developing an approximation to the matrix X, the number of right and left factoring vectors, k, needs to be specified or determined.

  The profile likelihood test for finding the optimal number of components, k, is adapted from Zhu and Ghodsi (2006). One assumes that eigen values follow a mixture distribution of two normal distributed populations; one group corresponding to real components (the larger eigen values) and the other noise (the smaller eigen values). We calculate the profile likelihood for any hypothesis of k significant components under the assumption that both populations have same standard deviation. It makes sense to select the hypothesis number that has the highest profile likelihood.
  The linearity test is performed on the differences between consecutive eigen values, which should be constant for those components which are noise. We compute an upper confidence bound for those differences. The linearity test plot should be read from right (smallest component) to left (largest component): the first time the difference between consecutive eigen values exceeds the upper bound line can be taken as the optimal number of components.

4.4 Utilities tab

! Inactivated in limited version



For the JMP active data set, utilities are provided to help generate tables for future use with irMF or other platforms:

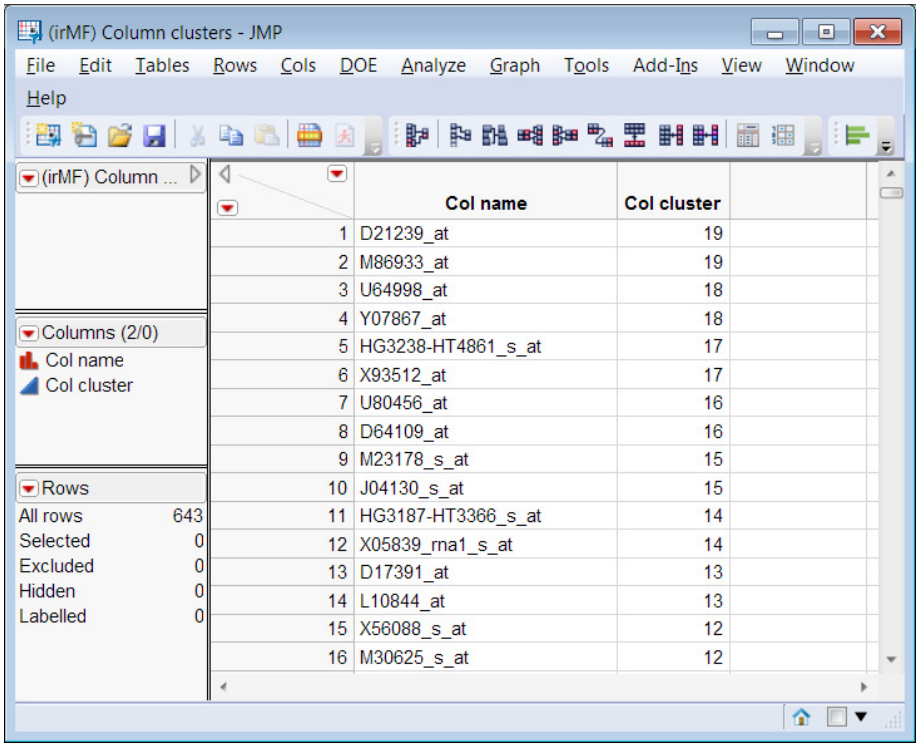
Select columns with no outlier: Removing columns with univariate outliers can be useful for pre-processing before NMF or if one wants to reduce the number of columns. If there are treatment groups, it makes sense to check for outliers within each group.

Select columns from list of column names: irMF generates tables where column names are ordered according to one or another model-component. Column names can be selected from these tables to generate a smaller table that will contain the more

interesting columns. This option can be used with any table (not just irMF tables) that contains a list of column names.

Cluster columns: We give a simple "Leader" clustering algorithm proposed by J.Liu (personal communication), in which we use pairwise correlations between columns:
Step 0: select or filter the variables list
Step 1: Starting with no clusters, calculate all the pairwise correlations and select the pair of variables with the highest correlation.
Step 2: Find the strongest correlation in the unclustered columns to the leader and form a new cluster for those two correlated columns.
Step 3: From those unclustered columns find the column that has the best average correlation with the cluster formed in step 2. If the average correlation is larger than a user-specified cutoff (e.g. absolute correlation is larger than 0.9), add the column to the cluster; otherwise go to step 2 and restart a new search.
Step 4: Repeat procedure 2 and 3 until no pairwise correlation is found to be larger than the cutoff.

This algorithm can be used to exclude columns that do not fit in any of the found clusters (cluster 0). To do that, use the "Select columns" option as described above and select the table "(irMF) Column clusters" which is created by the clustering program.



	Col name	Col cluster
1	D21239_at	19
2	M86933_at	19
3	U64998_at	18
4	Y07867_at	18
5	HG3238-HT4861_s_at	17
6	X93512_at	17
7	U80456_at	16
8	D64109_at	16
9	M23178_s_at	15
10	J04130_s_at	15
11	HG3187-HT3366_s_at	14
12	X05839_rna1_s_at	14
13	D17391_at	13
14	L10844_at	13
15	X56088_s_at	12
16	M30625_s_at	12

Select cells from the "Col name" column which adjacent cells in the "Col cluster" column are not zero.
Select rows from significant clusters: Recall that for each cluster, the cluster sample with the highest LHE element points to the associated experimental group. We use the

hypergeometric distribution to test this association. If significant, rows from the cluster are selected.

Correct clustered rows only: Check this box if you want only correct clustered rows within significant clusters to be selected.

Choose best NMF model: In order to choose an appropriate model, one strategy is to run NMF with 2, 3, 4, etc. components and optimize the association between found NMF clusters and groups. irMF performs permutation runs to see how "lucky" the found associations can be. The procedure is following:
First test all the models (up to 10 components accepted). The resulting clustering's are stored automatically into memory. Then let irMF choose the best model according to an association score (described below) and assess the significance of the association over all tested models.

Note: To start with a fresh series of models, all internally stored NMF models can be erased from memory through the option "Close irMF tables" (see irMF main tab)

The internal algorithm is as follows:
For each run:
1) Permute row group labels.
2) For each model, calculate a score associated with the quality of the association.
3) Take the max score across all models and compare with the original max score found without permutation. If larger, then increment a counter.

After 10000 runs, p-value = count/10000

Note: The number of runs can be redefined in the Preferences dialog tab (see below).

One difficulty is in defining an appropriate score to assess the quality of the association, that can be compared between different permutation runs and models (there is a risk of model over-fitting, pointing to non-relevant clusters). irMF essentially uses a chi-square score of independence between the rows and columns of a table. Here the rows are the clusters, the columns are the groups, and each cell contains the number of observations that belong to the corresponding group and cluster. The ordering of rows within a cluster is important. For this reason, each count is weighted by the value of corresponding element in left hand vector, giving rise to a *weighted* score.

For each cluster, we add an association significance (based on the hypergeometric distribution) in the cell which corresponds to the main group within cluster.

Group size by cluster & significance of main group:
[using hypergeometric distribution]

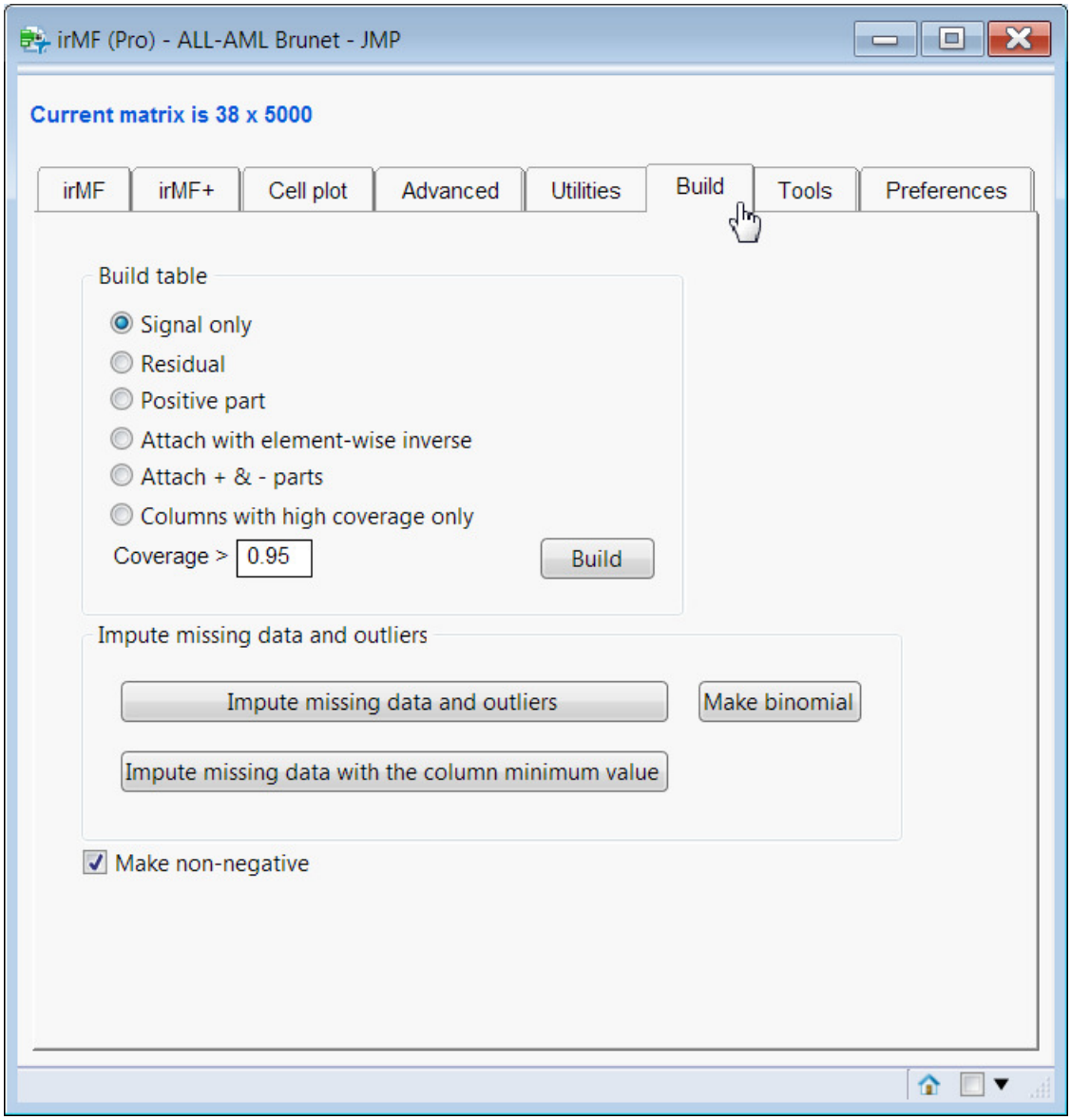
Cluster(Size)	ALL_B1	ALL_B2	ALL_T	AML
C 1 (9)	8 [<0.0001*]	0	0	1
C 2 (10)	2	8 [<0.0001*]	0	0
C 3 (8)	0	0	8 [<0.0001*]	0
C 4 (11)	1	0	0	10 [<0.0001*]

Weighted score: 43.14
Global significance using 10000 permutations: 0.0001

Note: NMF produces for each model a global p-value of the association between clusters and groups. This p-value is calculated in a similar way, but is associated with the model being tested, not *all* the models that have been tested on the same dataset.

4.5 Build tab

! Inactivated in limited version



Signal only: Use this option to output a table with model-fitted values.

Residual: Use this option to output a table with model-residual values.

Positive part: Use this option is you choose to ignore negative values, which will be replaced by 0.

Attach with element-wise inverse: In micro array experiments we are typically interested in both up and down regulation. NMF focuses on the large positive values. If every element in the matrix is replaced by $1/x_{ij}$ then small values become large. Use this option to run mNMF on both tables simultaneously (see option '2nd block is attached' in NMF tab).

Attach + and - parts: This option is useful when a subject effect has been subtracted from the original signal, resulting in positive and negative values. The matrix is then duplicated in positive and negative parts, and negative values are replaced by their absolute value. Use this option to run mNMF on both tables simultaneously (see option '2nd block is attached' in NMF tab).

Columns with high coverage only: Use this option to output a table with only columns having less than xx% outlier (as detected by robust SVD) or missing cells.

Impute missing data and outliers: Use this option to impute and replace missing cells and outliers using rSVD model-fitted values. It is often useful to "clean up" a matrix prior to run standard NMF.

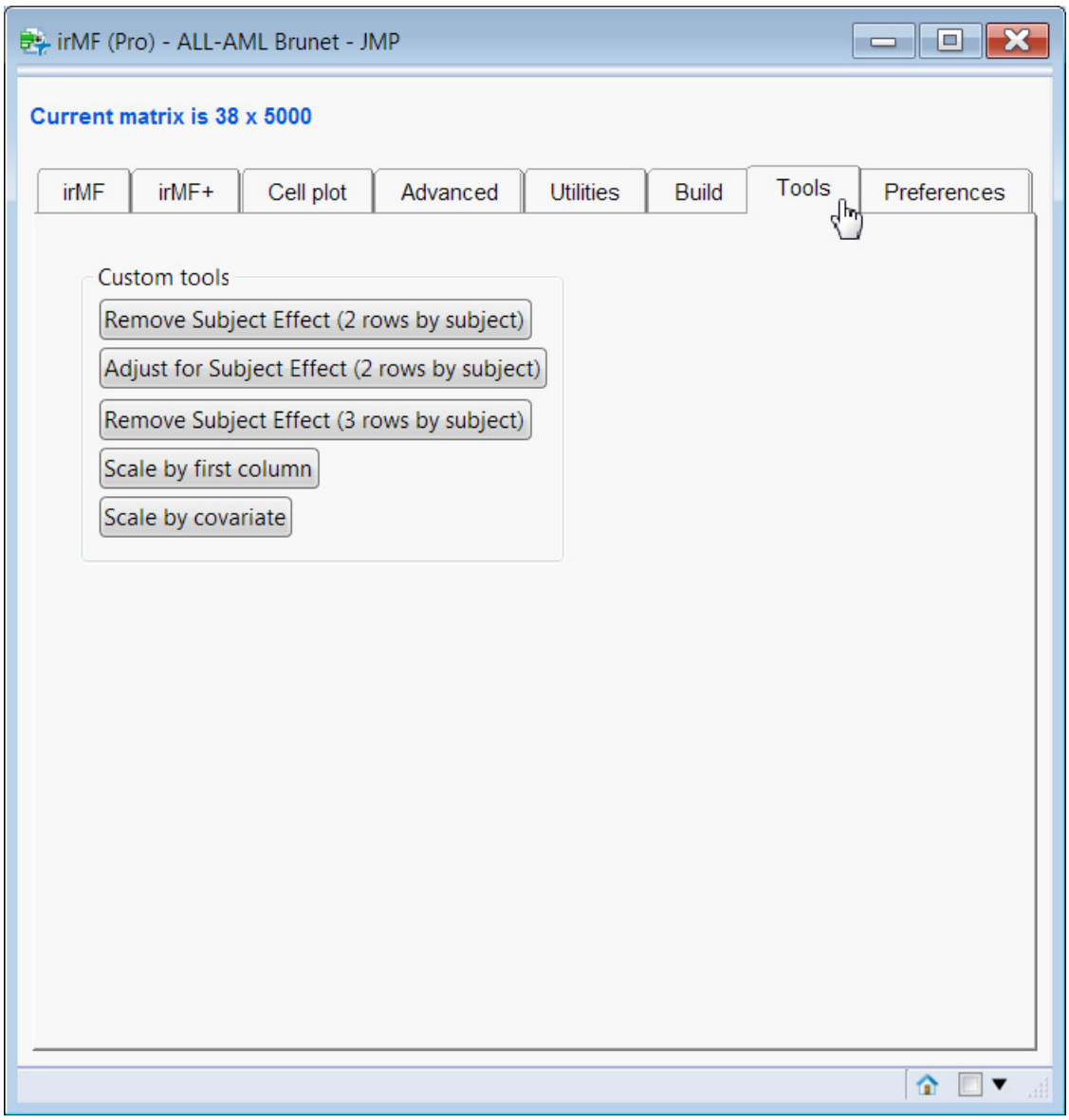
Make binomial: Replace matrix values with 1/0, 1 for all non-missing and 0 for all missing data.

Impute missing data with the column minimum value: Use this option to impute and replace missing cells within a column by the column minimum value.

Make non-negative: SVD based fitted values are not guaranteed to be non-negative. Check this option to replace negative values by 0 (default).

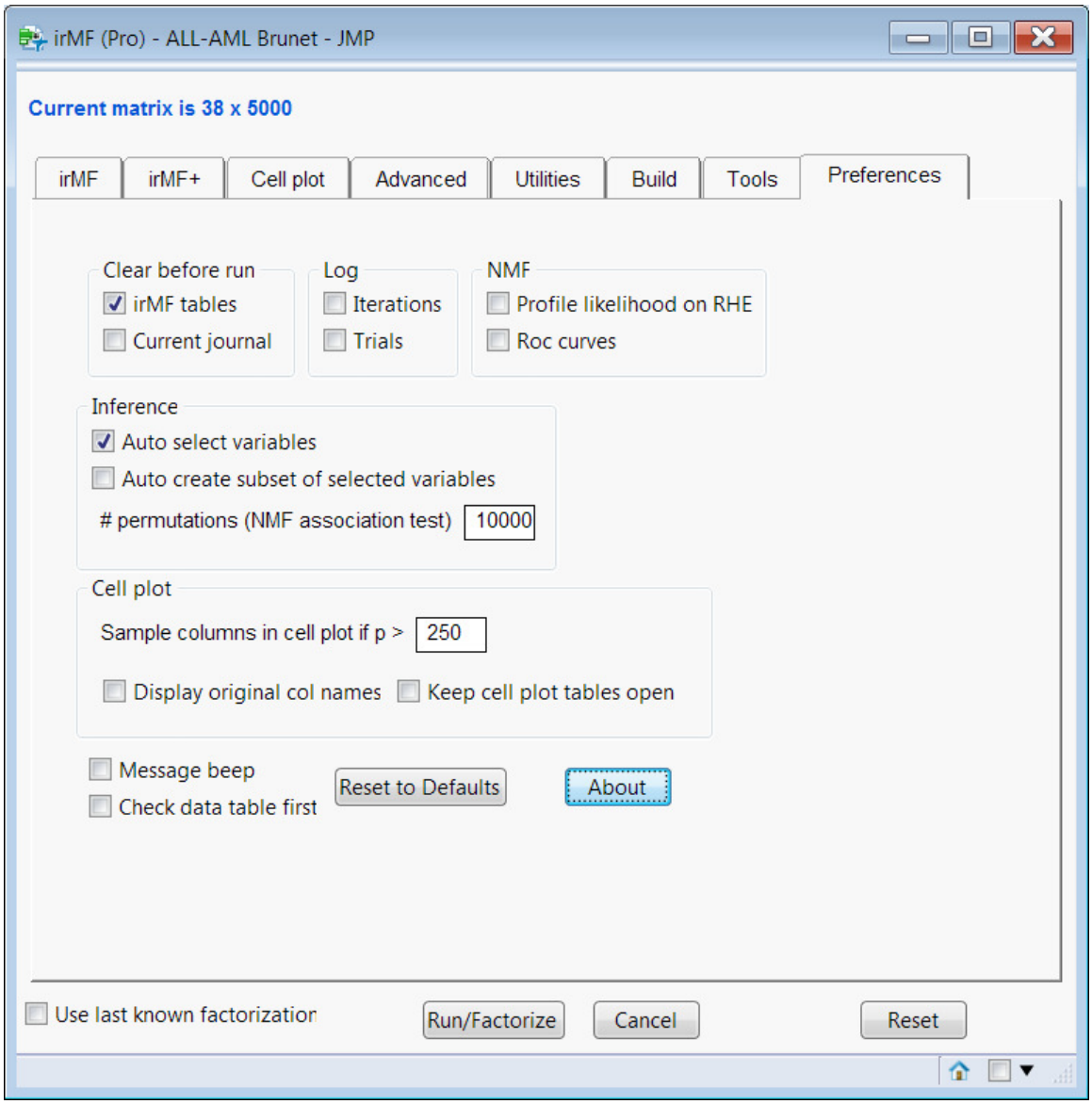
4.6 Tools

! Inactivated in limited version



Tools are experimental functionalities that can be easily accessed and modified (the jsl scripts are not encrypted). These tools are still not part of irMF itself and are therefore not described in this document.

4.7 Preferences tab



Note: Preference parameters are not stored in data table file as they are not specific to the dataset under stuty. When starting JMP (or after resetting irMF workspace), default parameters as stored in file irMFini.jsl are applied. This file can be retrieved by Windows search function and is editable - but must be handled with care.

Clear irMF tables: irMF creates a number of tables while running. These tables will be automatically cleared each time irMF is run if this option is checked.

Clear current journal: Check this option if you do not want results to be appended to the journal each time irMF is run.

Iterations and trials residuals (Mean Square Error) can be logged in JMP log window.

NMF:

- Profile likelihood on RHE: Same approach as profile likelihood (see above advanced tab, Scree plot section), but applied on each profile vector.
- Roc curves: In the special case where the number of components is set equal to the number of groups, Roc curves are built for all pairs of components if the option is checked in the Preferences tab. For each pair, the discriminant function is the ratio of the second over the first left component. Recall that the Roc curve represents false versus true positive rates as a function of a cutoff in the discriminant function. In order to calculate these rates, each cluster must be associated with an experimental group. The cluster sample which has the highest LHE element points to the associated experimental group.

Inference:

Auto select variables: Check if you want irMF to select automatically variables in the table.

Auto create subset of selected variables: Check if you want irMF to create automatically a subset table of selected variables.

permutations (NMF association test): Set the number of runs which are performed during the permutation test used to assess the association between found NMF clusters and actual groups.

Sample columns: When the number of columns is too large to be displayed on a single page (due to screen resolution), a random sample of columns is displayed. Note that the calculations are done on the full data set.

Display original col names: Check this option to display column names as they appear in the table. Otherwise column numbers will be displayed. Note that if there are many columns, names will be unreadable. Do not check this option if you want the rows of the cell plot match with the row labels on the right side of the plot.

Keep cell plot tables open: Cell plot tables are temporary tables associated with cell plots. These tables are automatically closed unless this checkbox is checked.

Message beep: Checking this option will cause irMF messages to beep.

Check data table first: When checked, this option can substantially slow down the opening of irMF dialog when the number of rows in the data table is very large (e.g. > 1000). If unchecked, preliminary checks - like ensuring non-negativity before running NMF - are disabled.

Reset to default: irMF default parameters are defined in the jsl script file irMFini.jsl. Use a text editor to edit this file if you want to modify the defaults (however back up the original file before).

About: Clicking on this button opens the 'about' window:

5. Outputs

Scree plots, cell plots and Roc curves are all stored in a JMP journal.
A number of tables are created:

- Scaling factors (Table name: '(irMF) Scaling factors'). These factors are all equal to 1 if NMF is applied since NMF factors are scaled in a particular way (see below).
- Left hand factoring vectors (Table name: '(irMF) Left factoring vectors').
- Right hand factoring vectors (Table name: '(irMF) Right factoring vectors'). An additional column (baseline) is added if the NMF transform Min(col)=0 or Robust Min(col)=0 has been activated. This column corresponds to the min or robust min vector of original variables.
- Selected variables (inference, Table name: '(irMF) Selected variables'). If mNMF is used, as many such tables will be created as there are blocks.
- Ordered right hand factoring vectors with likelihood profile (NMF only, Table name: '(irMF) Right factoring vectors (profile)')
- Percent of outliers by row or column (robust algorithm only, Table name: '(irMF) Marked outliers by column/row')

Table scripts

- Tables of Left and Right factoring vectors contain scripts that allow for useful visualizations. Specifically, the Right factoring vectors table contains a script named

"Select" which allows for selecting variables that discriminate most between NMF components. This script uses a normal mixture to model the distribution of the relative difference between two right factoring vectors into a mixture of left, central and right normal distributions. It then uses a control chart (with control limits based on average and standard deviation of the central normal distribution) to detect Up and Down variables (vs. the reference component), which are beyond control limits. Control limits are based on false discovery rate, which is calculated according to the mixture model parameters.

- Another script, 'Select variables', applies profile likelihood on each right vector, which is contained in table '(irMF) Right factoring vectors – Profile'. RHE elements are plotted for each factoring vector, along with profile likelihood to determine where to stop selecting variables. Also a table 'Profile most contributing variables' is opened, which contains the names of the selected variables and their respective component.
  Note: The option 'profile likelihood on RHE' must be activated in the preferences tab in order to obtain these tables.
- If the option 'clear irMF tables' is checked, output tables which are already open are automatically replaced when irMF is run again. Otherwise, new output tables are created.

6. Notes

Options are saved in a table script named SetirMFoptions. Save your dataset before closing it, to allow irMF retrieve automatically last used options whenever the dataset will be open again.

Modifying the irMFini.jsl: This script contains default parameters that will automatically appear in irMF dialogs the first time it is called with a given data table. These parameters can be changed by the user.

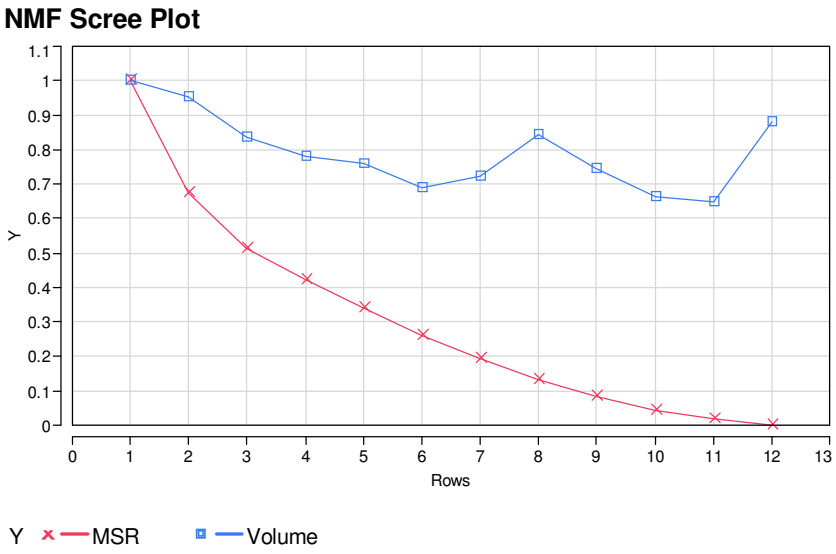
Note: irMF preference parameters apply to all tables and are not saved in table script. Within a session, preference parameters can be changed though the preference dialog. To make these changes permanent, edit the irMFini.jsl file. Preference parameters appear on the top. You can change the values taken by any of the parameters, but the "if (isEmpty(…" conditional statements must not be changed.

7. A small example

Scientists often encounter 2-way data tables. In our case, we have 86 Scotch whiskies that have been rated on a five-point scale for 12 flavor characteristics: Body, sweetness, smoky, medicinal, tobacco, honey, spicy, winey, nutty, malty, fruity, and floral. This data set comes from a wonderful book on the classification of Scotch whisky based on flavors by David Wishart (2002). The first comment is that good classification usually involves subject knowledge. The Wishart book provides considerable background knowledge of Scotch whisky. In particular, production methods are described in some depth so that factors that contribute to ultimate flavor are made clear. David's premise is that people are interested in single malt whiskies for their different flavors and can benefit from a

refined analysis of the factors of flavors (as opposed to a simple single "quality" scale). If the consumer understands the different dimensions of flavor then exploring different flavors and finding addition single malts of interest is possible. But a 12 dimensional world is very big. How the 86 readily available single malts cluster is a major topic of David's book. He provides 10, 6, and 4 level clusterings of the single malts.

How can NMF be used for clustering this data set? In many ways the data is ideal to display some of the potential advantages of NMF. The X matrix is positive with each flavor scaled from 0 (not present) to 4 (pronounced). Water and grain neutral alcohol have no flavor so all the flavors of Scotch whisky are designed or engineered in. For example, the still master controls the "cut" of the distillation process to give fewer or more fermentation secondary compounds, e.g. aldehydes, esters, ketones, acids, etc. The whiskey is aged in a wide variety of oak casts used to impart flavors, e.g. old, young, European or American, previously used for wine, port, other whiskies, etc. It is not unreasonable to think of the flavor components being "layered onto" the starting unflavored product, water and alcohol. Some of the production methods might add several flavor components in essentially fixed ratios. Might we find some prototypical flavor patterns and that the individual single malts are combinations of these flavor patterns? So our left factoring vectors will be the mixing levels, W, and our right factoring vectors will be the prototypical flavor patterns, P. Let's see how it works out. First, how many flavor patterns are present? For NMF a Scree plot can be computed:



Note: Least square is used.

Obviously considerable care has been given to the various flavors chosen to characterize Scotch whisky as there is no dramatic clustering of the flavors. Even so, we note erratic variations in the volume, which even increases if we add more than 6 components, indicating that additional components essentially explain noisy data. We also note a linear trend in the decrease of MSR if we add more than 3 components, so we choose to

go with four major flavor factors (but we could also have chosen five or six major factors). A look at the four resulting right factoring vectors is instructive.

Col name 1	Comp 1	Col name 2	Comp 2	Col name 3	Comp 3	Col name 4	Comp 4
Sweetness	1	Body	1	Smoky	1	Nutty	1
Floral	0.92287912	Winey	0.99544741	Body	0.89579761	Floral	0.31242299
Fruity	0.75636826	Honey	0.6022709	Medicinal	0.89387148	Malty	0.29524257
Malty	0.6426054	Sweetness	0.55618489	Spicy	0.41705693	Fruity	0.25147832
Spicy	0.51867136	Spicy	0.39179929	Nutty	0.25192624	Honey	0.2307033
Honey	0.2671121	Smoky	0.34974321	Malty	0.25188554	Smoky	0.11744747
Body	0.24347497	Malty	0.32150296	Sweetness	0.24905905	Body	0.09139205
Smoky	0.2295555	Fruity	0.29094332	Fruity	0.17787716	Sweetness	0.09126265
Tobacco	0.0039877	Tobacco	0.0063726	Tobacco	0.14974995	Winey	0
Nutty	0	Nutty	6.74569e-6	Winey	0.05217703	Spicy	0
Winey	0	Floral	0	Floral	0	Tobacco	0
Medicinal	0	Medicinal	0	Honey	0	Medicinal	0

Component 1 contains sweetness, floral, fruity flavors and to a lesser extent malty and spicy flavors. (We normalized each component so that the largest element is one.) Component 2 has the winey flavor (2nd element) isolated from the other components. This suggests that winey flavor might be added specifically to the single malt, perhaps through the use of oak barrels previously used for wine aging. Smoky and body flavors are captured by component 3. Component 4 mainly contains nutty, floral and malty characteristics. To a lesser extent component 3 contains nutty and malty flavors as well. Note that these components are not mathematically orthogonal. Pure flavors are not expected to be available to the designer of single malt; it seems weird that a mathematical detail like orthogonality should intrude into the data analysis process!

Are there single malts that appear to be relatively pure embodiments of these four flavor profiles?

Distillery	Comp 1	Comp 2	Comp 3	Comp 4	Cluster
Bladnoch	0.16472792	0.02304583	0.00855898	0.00754729	1
Glenfiddich	0.1642158	0	0.01824516	0	1
Glendronach	0	0.30790596	0.02179138	0.12754792	2
Macallan	0.04442146	0.30177278	0	0.13550328	2
Ardbeg	0.0165995	0	0.38740094	0.00168604	3
Laphroig	0	0.02965167	0.35400281	0	3
Edradour	0.08319047	0.10723692	0	0.26743463	4
GlenGrant	0.05620948	0.06288304	0	0.13369549	4

These eight single malts, 2 for each cluster, have flavor profiles very close to the flavor profile for the four flavor profile vectors that can be used to reconstruct all the single malt flavor profiles. (Here we normalized each component so that the sum of weights for each weighting component is one and assigned each single malt to its closest prototypical component. More specifically, we assigned each row of the matrix (single malt) to the number of the component, which has the highest element.)

Some wild ideas: these eight single malts might form a small, cost-effective inventory to keep your single malt drinking friends happy; one of each pair might serve as the basis

for making Scotch blends; it might be instructive to map these four flavor profiles back to the manufacturing methods.

Some final comments: SVD is mathematically driven to maximize the coverage of the variance of the measurements and have orthogonal components whereas non-negative matrix factorization aims to decompose the matrix into profile vectors and the weighting of those vectors to reconstruct the observed, positive data. Here and in other examples, not shown, the resulting right factoring vectors appear to point to underlying parts of a mixture or mechanisms. We think NMF can offer easier insight into the problem behind the non-negative data matrix.

Note: The data set used in the above example is included in the package.

8. References

Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *PNAS* 101, 4164–4169.

Devarajan, K. (2008). Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. *PLoS Computational Biology*, Vol. 4, Issue 7.

Donoho, D., Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? *Advances in Neural Information Processing System*

Fisher, R. A. (1948), Combining independent tests of significance, *American Statistician*,2, 30. (In response to Question 14)

Fogel, P., Young, S. S., Hawkins, D. M., Ledirac, N. (2007). Inferential, robust non-negative matrix factorization analysis of microarray data. *Bioinformatics*, Vol. 23 no. 1 2007, 44–49.

Gabriel, K.R., Zamir, S. (1979). Lower Rank Approximation of Matrices by Least Squares with any Choice of Weights. *Technometrics* 21, 489–498.

Good, I.J. (1969). Some applications of the singular decomposition of a matrix. *Technometrics* 11, 823-831.

Hoyer, P.O. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research* 5, 1457–1469.

Kim, P.M. and Tidor, B. (2003). Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Res.* 13, 1706-1718.

Lee, D.D., Seung H.S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788-791.

Liu, L., Hawkins, D.M., Ghosh, S., Young, S.S. (2003). Robust singular value decomposition analysis of microarray data. *PNAS* 100, 13167-13172.

Smilde, A.K., Westerhuis, J.A., Jong,S. (2003). A framework for sequential multiblock component methods. *J. Chemometrics* 17: 323–337

Wishart, D. (2002). *Whisky Classified, Choosing Single Malts by Flavor*. Pavilon, London

Zhu,M. and Ghodsi,A. (2006) Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Comput. Stat. Data Anal.*, 51, 918–930.