Orange

February 2024

Khiops Coclustering Guide 10.2

- Khiops
  - Optimal data preparation based on discretization and value grouping
  - Scoring models for classification and regression
  - Correlation analysis between pairs of variables
  - Automatic variable construction for multi-table relational mining
- Khiops Visualization
  - Analysis of Khiops results using an interactive visualization tool
- **Khiops Coclustering**
  - Correlation analysis of two or more variables using a hierarchical coclustering model
  - Prediction models for cluster assignment
- Khiops Covisualization
  - Exploratory analysis of Khiops Coclustering results using an interactive visualization tool

This guide is about the Khiops Coclustering component.

Abstract

Khiops Coclustering is dedicated to the analysis of the correlation between two or more variables with potentially many values. Each variable is partitioned into groups of values or intervals, so as to highlight the correlation between the variables. The coclustering model is organized into a hierarchy that enables the exploration of the results at any grain level using Khiops Covisualization.

Once a coclustering model is built, it can be simplified and its clusters can be extracted for each variable. A deployment model can also be prepared to predict clusters on new data.

Khiops Coclustering is available both in user interface mode and in batch mode, such that it can easily be embedded as a software component in a data mining deployment project.

This paper describes the parameters of Khiops Coclustering and all its functionalities, mainly how to train a coclustering model, how to simplify a coclustering by applying granularity constraints and how to prepare the deployment of a coclustering.

Summary

1. Presentation

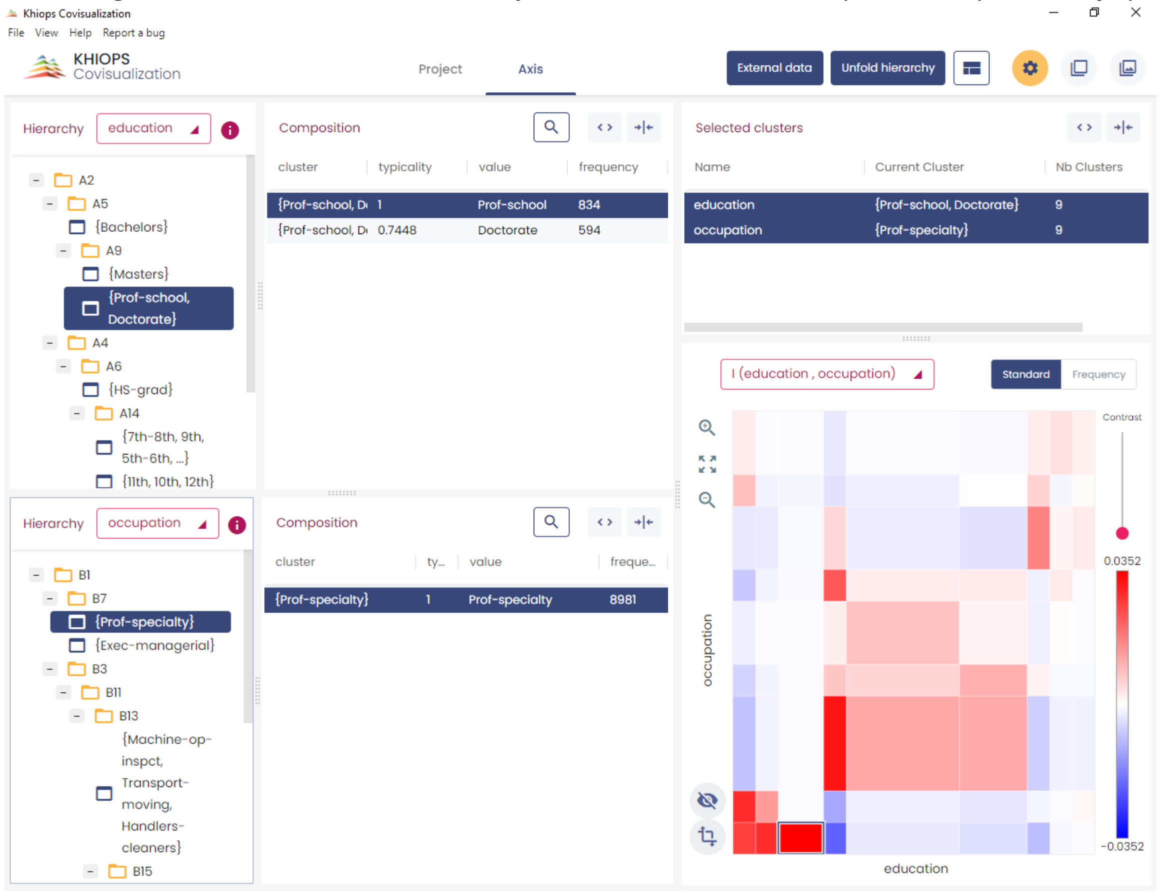
Khiops Coclustering aims at detecting highly informative patterns by the mean of hierarchical coclustering models, suitable for the task of explanatory analysis. This novel type of statistical analysis provides insights in many domains, such as:

- Text corpus analysis: clusters of texts versus clusters of words
- Market analysis: clusters of customers versus clusters of products
- Web log analysis: clusters of cookies versus clusters of web pages
- Graph analysis: clusters of source versus target nodes
- Temporal graph analysis: : clusters of source versus target nodes versus temporal intervals
- Curve corpus analysis: clusters of curves versus interval of X versus intervals of Y
- …

A coclustering model summarizes the correlation between two or more variables by simultaneously partitioning the values of each variable, into groups of value in the categorical case and into intervals in the numerical case. The cross-product of these univariate partitions forms a multivariate partition, called data grid. By counting the frequencies in the multivariate parts (called cells) of this data grid, we obtain a nonparametric estimator of the joint density of the variables. Each partition is organized into hierarchies, so as to enable an exploratory analysis of the results at any grain level.

For illustration purpose, let us consider the correlation between the education and occupation variables of the Adult database (coming from the US Census Bureau). This database contains about 50000 instances, with 14 values of occupation and 16 values of education.

Applying the **Khiops Coclustering** back-end tool, we obtain a 10*9 fine-grained data grid. The **Khiops Covisualization** tool enables the exploration of the correlation between the two variables. Displaying the mutual information highlights the over-represented cells (in red), i.e. cells with a frequency higher than expected in case of independent variables, and the under-represented cells (in blue). In the screenshot below, the selected cell indicates a high concentration of education *Prof-school* or *Doctorate* Jointly with occupation *Prof-specialty*.



Composition

cluster	typicality	value	frequency
{Prof-school, D	1	Prof-school	834
{Prof-school, D	0.7448	Doctorate	594

Selected clusters

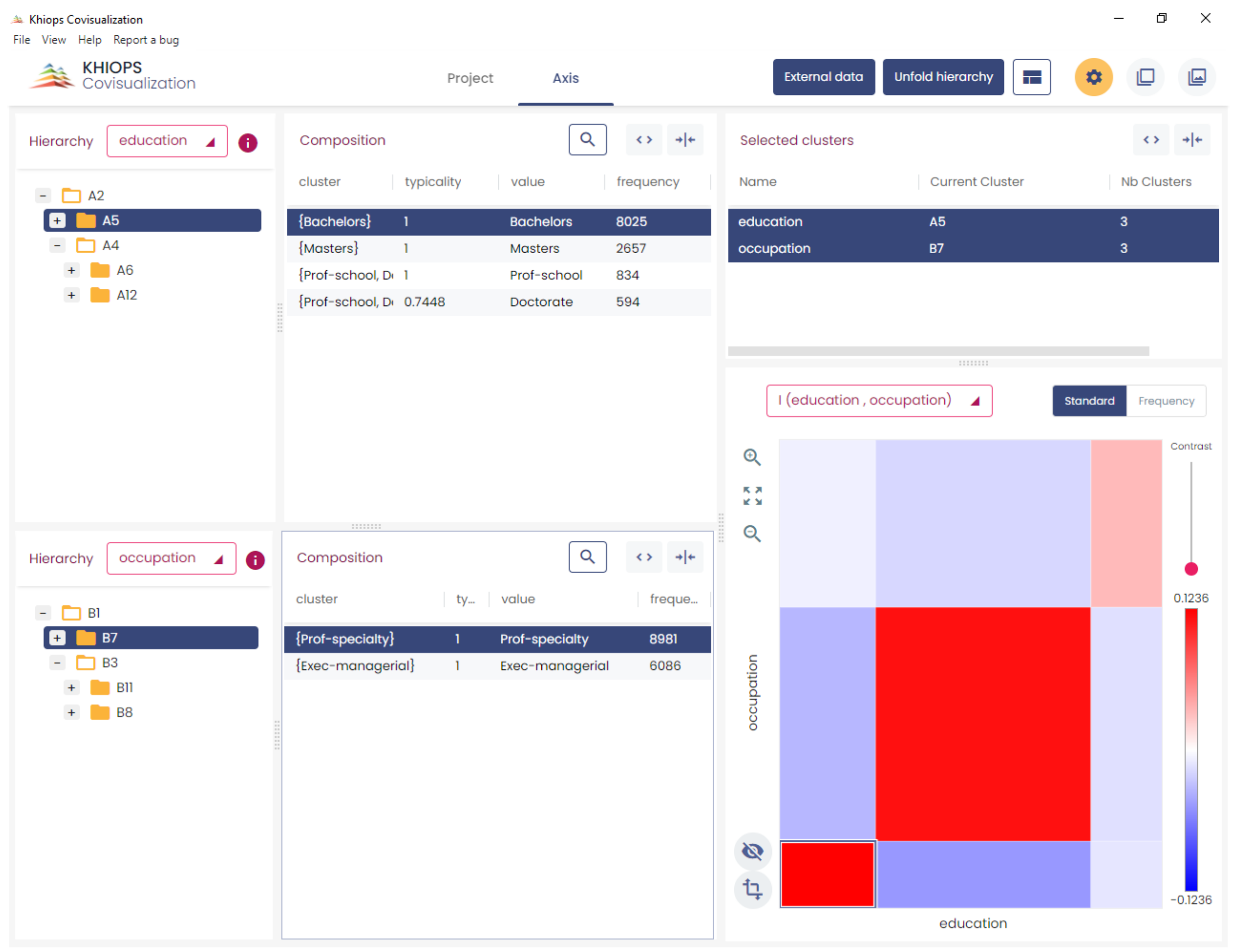
Name	Current Cluster	Nb Clusters
education	{Prof-school, Doctorate}	9
occupation	{Prof-specialty}	9

Composition

cluster	ty...	value	freque...
{Prof-specialty}	1	Prof-specialty	8981

Folding down both hierarchies allows to obtain a simplified 3*3 data grid which provides a quick summary with an easier interpretation.

In the screenshot below, the selected cell indicates a high concentration of education *Bachelor*, *Master*, *Prof-school* or *Doctorate* jointly with occupation *Exec-managerial* or *Prof-specialty*.



Composition

cluster	typicality	value	frequency
{Bachelors}	1	Bachelors	8025
{Masters}	1	Masters	2657
{Prof-school, D	1	Prof-school	834
{Prof-school, D	0.7448	Doctorate	594

Selected clusters

Name	Current Cluster	Nb Clusters
education	A5	3
occupation	B7	3

Composition

cluster	ty...	value	freque...
{Prof-specialty}	1	Prof-specialty	8981
{Exec-managerial}	1	Exec-managerial	6086

Beyond the illustrative example, this kind of analysis provides insightful summaries when applied to databases with millions of instances and variables with thousands of values.

Khiops Coclustering is the back-end tool for training and managing coclustering models.

The main functionalities are training a coclustering, simplifying a coclustering by applying granularity constraints and preparing the deployment of a coclustering.

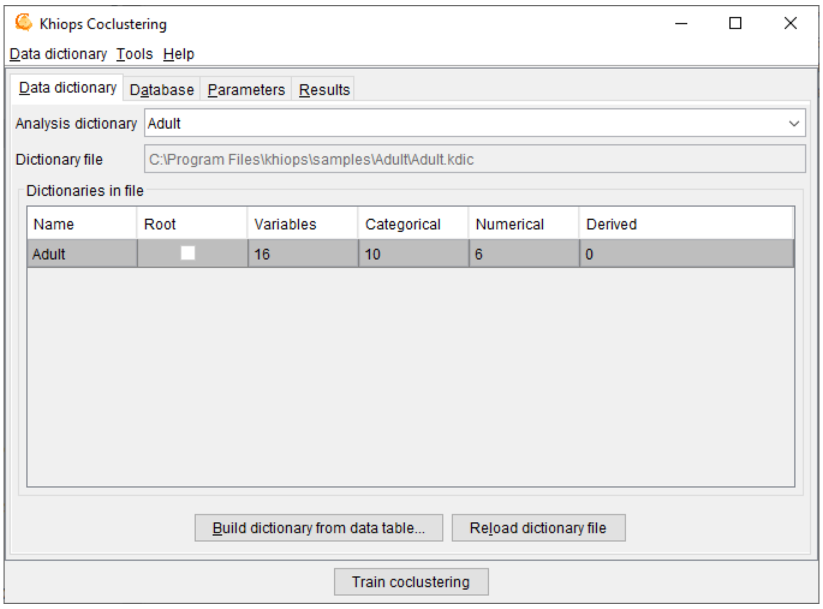
The tool and its parameters have been simplified to focus on the coclustering problem. The deployment of coclustering models is left to the back-end **Khiops** tool (see Khiops Guide).

More generally, the Khiops tool is dedicated to supervised data analysis and data management and the Khiops guide contains much more information about data dictionaries, data tables and technical limits than the present guide.

2. Training a coclustering

The main window, named *Khiops Coclustering*, enables the construction of a coclustering model from coclustering parameters, mainly from a data dictionary, a database, and the specification of coclustering variables.

2.1. Dictionary file


Khiops Coclustering
Data dictionary Tools Help
Data dictionary Database Parameters Results
Build dictionary from data table...
Reload dictionary file
Train coclustering

Analysis dictionary: Adult

Dictionary file: C:\Program Files\khiops\samples\Adult\Adult.kdic

Dictionaries in file

Name	Root	Variables	Categorical	Numerical	Derived
Adult	☐	16	10	6	0

A dictionary file is a text file with the extension .kdic. It contains the definition of one or several dictionaries, each one describing the set of variables to use in a data analysis. See **Khiops Guide** for a detailed description of dictionaries and their management.

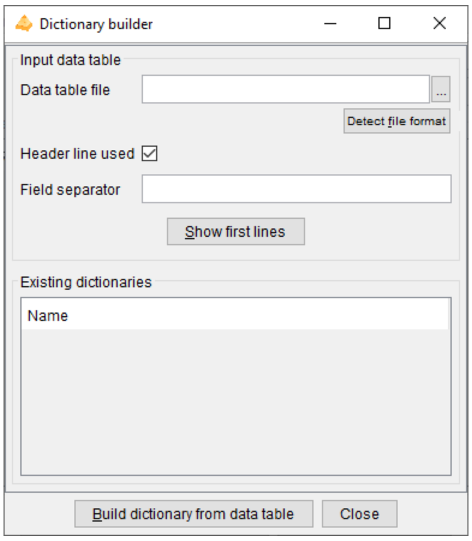
Khiops Coclustering allows to **Open** a **Dictionary file**. Opening a dictionary file amounts to loading its dictionaries into memory and making them available for data analysis. The **Save** and **Save as** actions write dictionaries to a dictionary file, whereas the **Close** action cleans the memory. The **Build dictionary from file** action builds dictionaries from data files and save them in a dictionary file. The **Reload dictionary file** action reads again a dictionary file, which may have been modified using an external text editor. The list of available dictionaries can be browsed using Khiops.

Analysis dictionary: name of the dictionary related to the data to analyse. Mandatory field.

Dictionary file: name of the dictionary file related to the data to analyse. Read-only field that shows the name of the current dictionary file.

Dictionaries in file: list of available dictionaries, with statistics describing the used variables (Name, Variables, Categorical, Numerical, Derived).

2.1.1. Build dictionary from data table



Header line used ☑

Field separator:

This action opens a window that allows to build dictionaries from data tables.

Data table file: name of the data table file to analyse. Mandatory field.

Detect file format: heuristic help that scans the first few lines to guess the file format. The header line and field separator are updated on success, with a warning or an error in the log window only if necessary.

Header line used: (default: true). If the file has a header line, Khiops will use the header line fields as variables names; otherwise, the variables will be names Var1, Var2...

Field separator: by default, if nothing is specified, the tabulation is used as the field separator.

Show first lines: show first lines of data table in log window.

Build dictionary from data table: start the analysis of the data table file to build a dictionary. The first lines of the file are analysed in order to determine the type of the variables: Categorical, Numerical, Date, Time or Timestamp. After analysis, the user can choose the name of the dictionary.

Close: closes the window. If dictionaries have been built, proposes to save them in a dictionary file

See **Khiops Guide** for more details on how dictionaries are built.

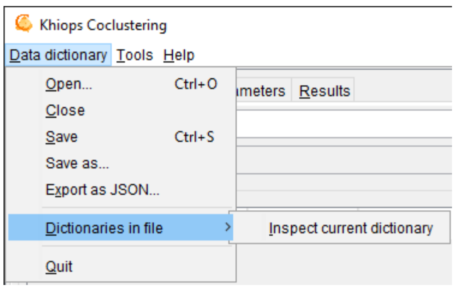
2.1.2. Reload dictionary file

Reload into memory the current dictionary file.

This action allows to modify the dictionary file using an external text editor (Notepad for example), to save the modifications, and to take them into account into Khiops by reloading the dictionary file.

In case of invalid dictionary file, the current dictionaries are kept in memory.

2.1.3. Dictionary file menu



2.1.3.1. Open

An open dialog box asks the name of the dictionary file to open.

In case of invalid dictionary file, the current dictionaries are kept in memory.

2.1.3.2. Close

The dictionaries are removed (from memory only). The potential pending modifications are lost if they have not been saved.

2.1.3.3. Save

The memory dictionaries are saved under the current dictionary file.

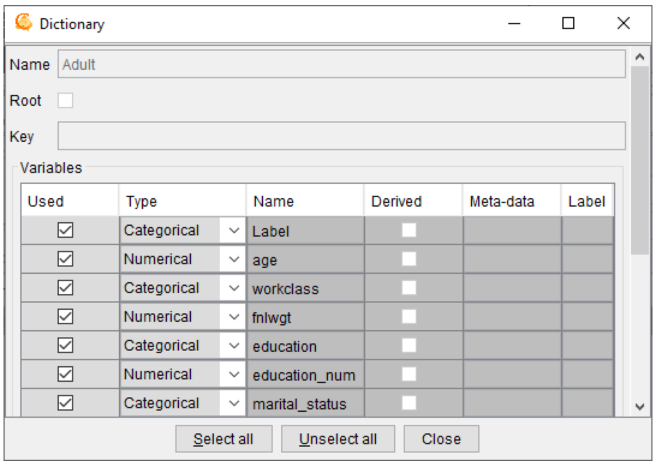
2.1.3.4. Save as

A save dialog box asks the name of the dictionary file to save.

2.1.3.5. Export as JSON

A save dialog box asks the name of the JSON file to export the dictionaries under a JSON format, with a .kdicj extension.

2.1.3.6. Dictionaries in file/Inspect current dictionary



Name Adult

Root ☐

Key

Variables

Used	Type	Name	Derived	Meta-data	Label
☑	Categorical	Label	☐		
☑	Numerical	age	☐		
☑	Categorical	workclass	☐		
☑	Numerical	fnlwgt	☐		
☑	Categorical	education	☐		
☑	Numerical	education_num	☐		
☑	Categorical	marital_status	☐		

Allows to inspect and partly modify a dictionary chosen among the list of available dictionaries. The dictionary to inspect must be selected among the dictionaries in file.

The action is available both from the menu and using a right click button on the selected dictionary.

During the inspection of a dictionary, the list of its variables can be browsed into a sub-window. For each variable, the following properties are displayed: **Used**, **Type**, **Name**, **Derived**, **Meta-data** and **Label**.

The Data Miner can choose whether to keep or not the variable for data analysis, using the **Used** property. The **Select all** and **Unselect all** buttons allow to choose all or no variables.

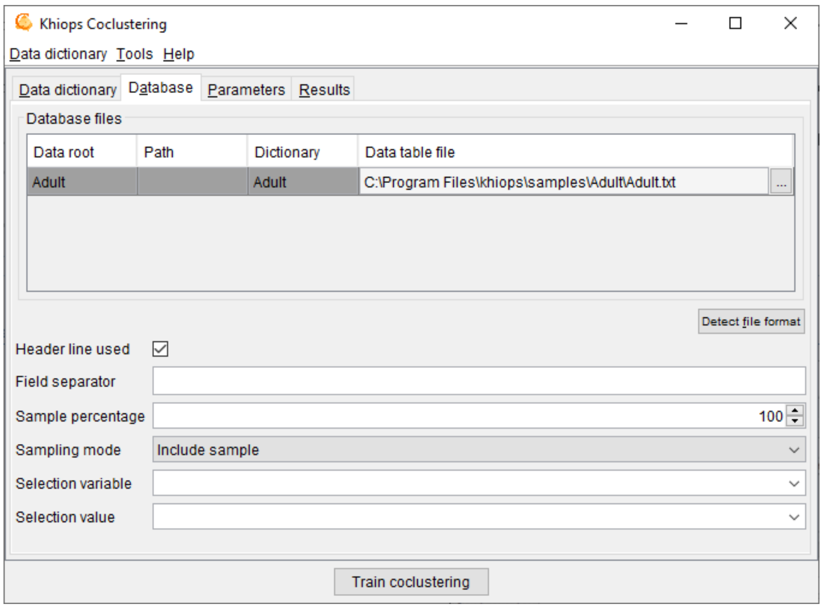
It is also possible to change the **Type** of variables: Numerical, Categorical, Date, Time or Timestamp.

Remark: for large scale modifications in a dictionary, it is preferable to update the dictionary file using an external text editor (Notepad, WordPad…), to save the file with the external editor, and then to reload the dictionary.

2.1.3.7. Quit

Quits the application.

2.2. Database


Khiops Coclustering
Data dictionary Tools Help
Data dictionary Database Parameters Results
Database files
Detect file format
Train coclustering

Data root	Path	Dictionary	Data table file
Adult		Adult	C:\Program Files\khiops\samples\Adult\Adult.txt

Header line used	✓
Field separator	
Sample percentage	100
Sampling mode	Include sample
Selection variable	
Selection value	

Database file: name of the database file to analyse. Mandatory field.

Detect file format: heuristic help that scans the first few lines to guess the file format. The header line and field separator are updated on success, with a warning or an error in the log window only if necessary.

Header line used: (default: true). If the file does not have a header line, Khiops Coclustering considers the leading variables in the dictionary to analyse the fields in the file.

Field separator: by default, if nothing is specified, the tabulation is used as the field separator.

Khiops Coclustering can be used to extract a subpart (or its exact complementary) of the records in a database file. This sampling is specified with a sample percentage of the records (to keep or to discard). The sampling is a random sampling, but is reproducible (the random seed is always the same).

Sample percentage: percentage of the samples (default: 100)

Sampling mode: to include or exclude the records of the sample (default: include sample).

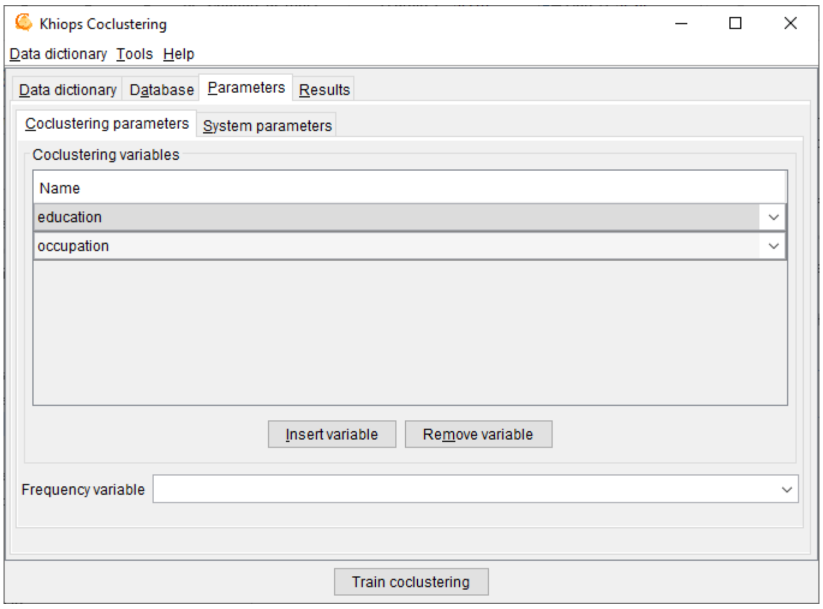
Another way to build samples is to use a selection variable and a selection value.

Selection variable: when nothing is specified, all the records are analysed. When a selection variable is specified, the records are selected when the value of their selection variable is equal to the selection value.

Selection value: used only when a selection variable is specified. In that case, the value must be a correct value (numerical value if the selection variable is a numerical variable).

2.3. Parameters

2.3.1. Coclustering parameters



Coclustering variables: list of input variables for the coclustering model.

There must be at least two numerical or categorical input coclustering variables. Up to ten variables are allowed.

Frequency variable: optional field. Name of a variable that contains the frequency of the records. Using the frequency variable is equivalent to duplicating the records in the input database, where the number of duplicates per record is equal to the frequency.

2.3.2. System parameters

Max number of error messages in log: allows to control the size of the log, by limiting the number of messages, warning or errors (default: 20).

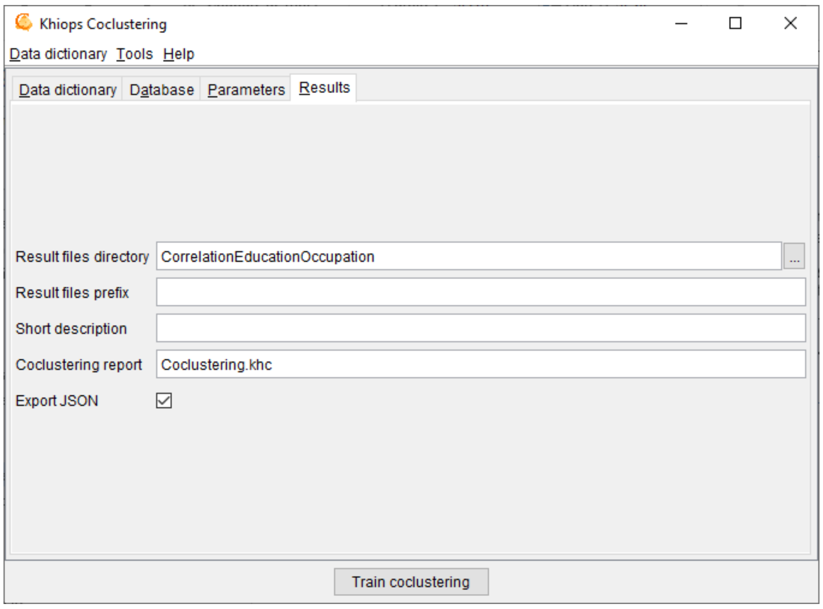
Min optimization time in seconds: allows to specify the min amount of time for the optimization algorithms. By default, this parameter is 0 and the algorithm stops by itself when no significant improvement is expected. Otherwise, the optimization is performed at least as long as specified, then stops after the next built solution.

Memory limit in Mo: allows to specify the max amount of memory available for the data analysis algorithms. By default, this parameter is set to the size of the RAM available for the Windows applications. This parameter can be decreased in order to keep memory for the other Windows application, or increased in the limit of the available RAM.

Max number of processor cores: allows to specify the max number of processor cores to use. Not used in this version: coclustering algorithms will be parallelized in future versions.

Temp file directory: name of the directory to use for temporary files (default: none, the system default temp file directory is then used).

2.4. Results



Result files directory: CorrelationEducationOccupation
Result files prefix:
Short description:
Coclustering report: Coclustering.khc
Export JSON: ✓

Result files directory: name of the directory where the results files are stored (default: empty). By default, the results files are stored in the directory of the train database. If a result directory is specified, it can be:

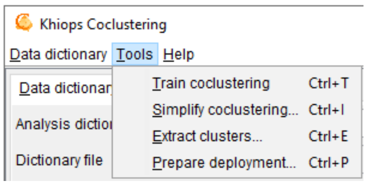
- an absolute path (example "c:\project\scenario1"): the results files are stored in this directory
- a local path (example "scenario1"): the results files are stored in a sub-directory of the train database directory
- a relative path (example ".\scenario1"): the results files are stored in a sub-directory of current directory (Khiops Coclustering executable start directory)

Result files prefix: (default: empty). This prefix is added before the name of each result file.

Coclustering report: (default: Coclustering.khc) name of the coclustering report, that contains the full definition of the coclustering model. The coclustering report is the input of the *Khiops Covisualization* tool, for versions prior to V10. This option is deprecated, and future versions will output the report in JSON format only.

Export JSON: exports the coclustering report under a JSON format. The exported JSON file has the same name as the coclustering report file, with a .khcj extension. This coclustering report is the input of the new *Khiops Covisualization* tool since version 10.1. The JSON file is also useful to inspect the coclustering results from any external tool.

2.5. Tools


Khiops Coclustering
Data dictionary Tools Help
Data dictionar
Analysis dictio
Dictionary file
Train coclustering Ctrl+T
Simplify coclustering... Ctrl+I
Extract clusters... Ctrl+E
Prepare deployment... Ctrl+P

The actions available from the tools menu are

- **Train coclustering**
  trains a coclustering model given the coclustering parameters
- **Simplify coclustering…**
  builds a simplified coclustering model
  opens a new window named *Coclustering simplification*
- **Extract clusters…**
  extracts clusters in a text file for a given coclustering variable
  opens a new window named *Cluster extraction*

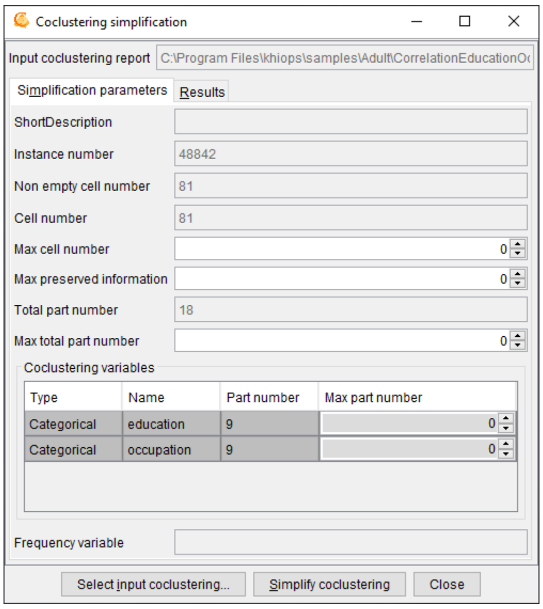
- **Prepare deployment…**
  enables the deployment of a coclustering model by the means of a Khiops deployment dictionary opens a new window named *Coclustering deployment preparation*

The first action, which trains a coclustering model from the data, is the main functionality of the tool. The required memory and computation time grow with the size of the data. As a rule of thumb, around 1 GB RAM is required per millions of data records and about one hour per million records is necessary to train the first coclustering model. This action is anytime: coclustering models are computed and continuously improved, with new solutions saved as soon as improvements are reached. The intermediate solutions can be used without waiting for the final solution, and the process can be stopped at any time to keep the last best solution.

The three last application actions exploit an existing coclustering model. They use an input coclustering model as well as granularity constraints that indicate whether the coclustering should be exploited at fine or coarse grain level.

2.5.1. Coclustering simplification

This window enables the simplification of a coclustering model given granularity constraints.


Coclustering simplification
Simplification parameters
Results
Select input coclustering...
Simplify coclustering
Close

Input coclustering report: C:\Program Files\khiops\samples\Adult\CorrelationEducationO

ShortDescription	
Instance number	48842
Non empty cell number	81
Cell number	81
Max cell number	0
Max preserved information	0
Total part number	18
Max total part number	0

Coclustering variables

Type	Name	Part number	Max part number
Categorical	education	9	0
Categorical	occupation	9	0

Frequency variable

Input coclustering report: name of the coclustering report to post-process.

Use the button **Select input coclustering** to choose an input coclustering report.
Use the button **Simplify coclustering** to build the simplified coclustering report.
The input coclustering is simplified using a bottom-up hierarchical agglomeration of the parts, until all the active simplification constraints are fulfilled (max cell number, max preserved information and max part number per variable).

2.5.1.1. Simplification parameters

Simplification parameters: recall of some coclustering statistics (read-only fields) and post-processing parameters to simplify the coclustering

- Short description
- Instance number
- Non empty cell number
- Cell number
- **Max cell number** : max number of cells to keep in the simplified coclustering (0 : no constraint)

- **Max preserved information** : max percentage of information to keep in the simplified coclustering (0 : no constraint). Low percentages correspond to weakly informative coarse models whereas high percentages correspond to highly informative detailed models.
- Total part number
- **Max total part number :** max number of total part number to keep in the simplified coclustering (0 : no constraint)
- **Coclustering variables** (in the array)
    - Type
    - Name
    - Part number
    - **Max part number** : max number of parts to keep for this variable in the simplified coclustering (0 : no constraint)
- Frequency variable

2.5.1.2. Results

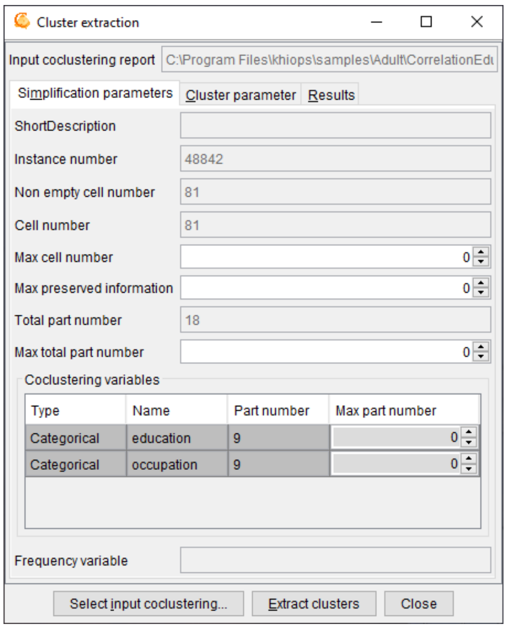
Result files directory

Result files prefix

Simplified coclustering report: (default: SimplifiedCoclustering.khcj) name of the simplified coclustering report, that is the most detailed version of the input coclustering report that meets all the simplification constraints.

2.5.2. Cluster extraction

This window enables the extraction of clusters for a given coclustering variable and given granularity constraints.



Input coclustering report	C:\Program Files\khiops\samples\Adult\CorrelationEdu
ShortDescription	
Instance number	48842
Non empty cell number	81
Cell number	81
Max cell number	0
Max preserved information	0
Total part number	18
Max total part number	0

Coclustering variables

Type	Name	Part number	Max part number
Categorical	education	9	0
Categorical	occupation	9	0

Frequency variable	

Input coclustering report: name of the coclustering report to post-process.

Use the button **Select input coclustering** to choose an input coclustering report.
Use the button **Extract clusters** to extract the clusters from the input coclustering.
The clusters are extracted for a given variable from the simplified coclustering (provided that simplification parameters are specified).

2.5.2.1. Simplification parameters

See 2.5.1.1. Simplification parameters.

2.5.2.2. Cluster parameters

Coclustering variable: name of the coclustering variable containing the clusters to extract

2.5.2.3. Results

Result files directory

Result files prefix

Cluster table file: (default: Clusters.txt) name of the text file containing the extracted clusters.

The cluster file is a text file with a header line, on record per line with tabulation as field separator.

In case of a categorical variable, the fields are:

- Cluster: name of the cluster (group of values)
- Value: name of the value contained in the cluster
- Frequency: frequency of the value
- Typicality: interest measure of the value within its cluster

  The special value " * " represents any value not seen during training the coclustering.

In case of a numerical variable, the fields are:

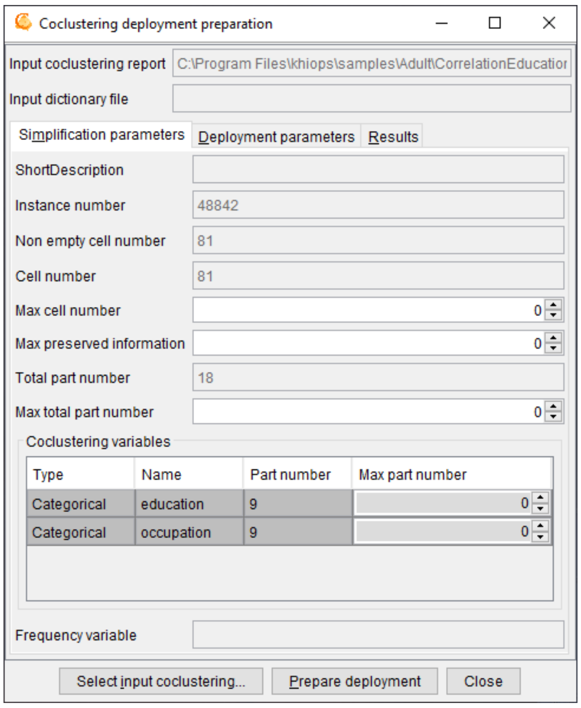
- Cluster: name of the cluster (interval of values)
- Lower bound: lower bound (excluded) of the interval
- upper bound: upper bound (included) of the interval

  Infinite lower and upper bounds are represented by empty fields. A cluster containing the missing value has empty fields for both the lower and upper bounds.

2.5.3. Prepare deployment

This dialog box deals with the preparation of the deployment of a coclustering model by the means of a Khiops deployment dictionary. Deploying a coclustering model consists in associating each instance of one variable of a coclustering model to the label of its cluster, as well as creating new variables such as the distance of the instance of each cluster.

The obtained coclustering deployment dictionary allows the user to update a database for a given entity of interest by adding new variables.


Coclustering deployment preparation
Simplification parameters
Deployment parameters
Results
Select input coclustering...
Prepare deployment
Close

Input coclustering report	C:\Program Files\khiops\samples\Adult\CorrelationEducation
Input dictionary file	
ShortDescription	
Instance number	48842
Non empty cell number	81
Cell number	81
Max cell number	0
Max preserved information	0
Total part number	18
Max total part number	0

Coclustering variables

Type	Name	Part number	Max part number
Categorical	education	9	0
Categorical	occupation	9	0

Frequency variable	

Input coclustering report: name of the coclustering report to post-process.

Input dictionary file: name of the dictionary file, that corresponds to the deployment database.

The input dictionary file must be opened from the main window using the “Dictionary file” menu.

Use the button **Select input coclustering** to choose an input coclustering report.
Use the button **Prepare deployment** to build the coclustering deployment dictionary file.

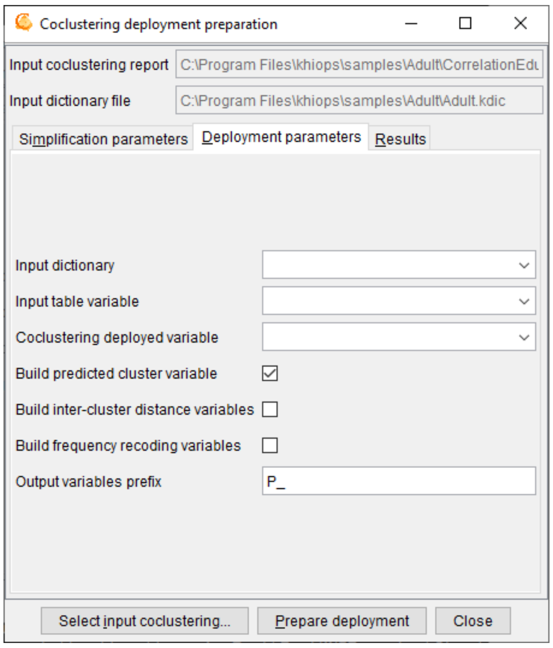
To deploy a coclustering, use the **Deploy model** functionality of the **Khiops** back-end tool and apply the deployment dictionary on new data.

A coclustering model is able to extract correlation information between two or more variables, such as Text*Word for a text corpus, Cookie*Page for a web log corpus, Curve*X*Y for a curve corpus. Let us take the example of a curve corpus, represented by a database of points with three variables, CurveId, X and Y and one record for each point in the curve corpus. The coclustering model builds clusters of curves and intervals of X and Y, such that curves distributed similarly on the intervals of X and Y tend to be grouped together. When new curves are available, it is interesting to deploy them on the basis of the trained coclustering model. Deploying a new curve consists in creating new variables to enrich the curve description: closest cluster of curve, distance to each cluster of curves, number of points per interval of X or Y.

2.5.3.1. Simplification parameters

See 2.5.1.1. Simplification parameters.

2.5.3.2. Deployment parameters


Coclustering deployment preparation

Input coclustering report: C:\Program Files\khiops\samples\Adult\CorrelationEdu

Input dictionary file: C:\Program Files\khiops\samples\Adult\Adult.kdic

Input dictionary	
Input table variable	
Coclustering deployed variable	
Build predicted cluster variable	☑
Build inter-cluster distance variables	☐
Build frequency recoding variables	☐
Output variables prefix	P_

Input dictionary: name of the dictionary that corresponds to the deployment database that contains the instances of interest.

Input table variable: name of the table variable in the input dictionary that contains the detailed record for each instance of interest.

Coclustering deployed variable: name of the deployed variable, i.e. one of the coclustering variables, which represents the entity of interest.

Build predicted cluster variable: indicates that the deployment model must generate a new variable containing the label of the cluster of the entity of interest.

Build inter-cluster variables: indicates that the deployment model must generate new variables representing the distance of the entity of interest to each cluster.

Build frequency recoding variables: indicates that the deployment model must generate new variables representing the frequency per cluster of the other coclustering variables.

Output variable prefix: (default: P_) prefix added to the deployment variables in the deployment dictionary.

Khiops guide with multi-table functionality is a prerequisite to the deployment of coclustering model.

In the case of a curve corpus, curves are represented using a multi-table schema, with curves as the root entity, in 0 to n relationship with their points.

- Root entity: dictionary Curve(CurveId), with two variables
  - Categorical CurveId
  - Table(Point) curvePoints
- Secondary entity: dictionary Point(CurveId), with three variables
  - Categorical CurveId
  - Numerical X
  - Numerical Y

The curve database consists of two data tables: one for the curves and the other for the points.

In this case, the objective is to deploy new curves, unseen during training. Whereas the coclustering model was trained using a single table point dataset, the deployments need a multi-table curve dataset, since each curve to deploy is represented by an identifier in the root table and a set of points in the secondary table.

The input dictionary is *Curve*, the input table variable is *curvePoints* and the coclustering deployed variable is *CurveId*. When the coclustering deployment model is prepared, it can be used to deploy new curves, that is to create new variables in the curve table:

- P_CurveIdPredictedLabel: predicted cluster label for variable CurveId
- P_CurveIdDistance<*CurveCluster*>: distance to curve cluster, for each cluster of curves <*CurveCluster*>
- P_XFrequency<*IntervalX*>: number of points per interval for each interval of X <*IntervalX*>
- P_YFrequency<*IntervalY*>: number of points per interval for each interval of Y <*IntervalY*>

Using the Khiops tool with its "Deploy model" functionality, a curve dataset can be deployed by the mean of the coclustering deployment model.

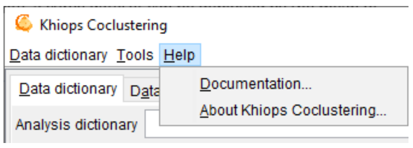
2.5.3.3. Results

Result files directory

Result files prefix

Coclustering dictionary file: (default: Coclustering.kdic) name of the deployment dictionary that contains the coclustering deployment model.

2.6. Help



The actions available from the help menu are

- **Documentation...**
  Shows a summary of the available documentation and other resources.
- **About Khiops Coclustering...**

3. Integration in information systems

3.1. Batch mode

3.1.1. Register and replay a batch scenario

For a recurrent use of Khiops Coclustering, it is interesting to register a scenario in a command file. Such a command file can easily be modified using a text editor, and then replayed by Khiops Coclustering in batch mode.

To run Khiops Coclustering, open a Shell khiops using the shortcut available in the Khiops installation directory, then type khiops_coclustering (-h for batch mode options with ability to record or replay a scenario).

To register a scenario, use the –o option on the command line.

Khiops_coclustering –o scenario._khc

All the actions or modifications of field values in the graphical user interface are registered in the command file scenario._khc.

It is a good practise to create a working directory for each Data Mining project. This directory contains all the files for databases, dictionaries, scenarios and reports.

Remark: by default, Khiops keeps a command file named scenario._kh for the last use of the tool, in the "khiops_data/lastrun" sub-directory in your user directory (%USERPROFILE% in Windows, $HOME in Linux).

The following scenario corresponds to a use of Khiops Coclustering with the Adult sample. The scenario opens the Adult dictionary and performs a coclustering analysis of the database for variables occupation and education. To replay this scenario:

- install the scenario file (named adult._khc) in the Adult directory
- Open a Shell Khiops and go to Iris directory
- Start Khiops Coclustering using –i on the command line

Khiops_coclustering –i adult._khc

Scenario file adult._khc

```
// -> Khiops Coclustering
ClassManagement.OpenFile      // Open...

// -> Open
ClassFileName C:\Program Files\khiops\Samples\Adult\Adult.kdic        // Dictionary file
OK                    // Open
// <- Open

Database.DatabaseFiles.List.Key Adult    // List item selection
Database.DatabaseFiles.DataTableName C:\Program Files\khiops\Samples\Adult\adult.txt      // Data table file
AnalysisSpec.CoclusteringParameters.Attributes.InsertItemAfter       // Insert variable
AnalysisSpec.CoclusteringParameters.Attributes.InsertItemAfter       // Insert variable
AnalysisSpec.CoclusteringParameters.Attributes.List.Key     // List item selection
AnalysisSpec.CoclusteringParameters.Attributes.Name occupation        // Name
AnalysisSpec.CoclusteringParameters.Attributes.List.Key     // List item selection
AnalysisSpec.CoclusteringParameters.Attributes.Name education        // Name
AnalysisResults.ResultFilesDirectory CorrelationOccupationEducation    // Result files directory
BuildCoclustering           // Train coclustering
Exit                  // Close
// <- Khiops Coclustering

// -> Khiops Coclustering
OK                    // Close
// <- Khiops Coclustering
```

Warning: the two last "Close" actions must be commented to replay the scenario and remain in the graphical user interface of Khiops Coclustering (otherwise, all the Khiops Coclustering session is replayed in batch mode, even the close action that exits from Khiops Coclustering).

3.1.2. Integration in a program to industrialize a data analysis process

Khiops Coclustering can easily be called from a program written in any computer language, such as C, C++, Java, or Python, in order to build an industrial data analysis process.

You first have to prepare the input data, dictionary file and datasets, write a scenario file in which you set up all needed parameters, mainly the location of the input and output data files and the data analysis operation to be performed, and finally call khiops_coclustering (in bin directory) by program using the command line options.

3.1.3. List of command line options

```
Usage: khiops_coclustering [OPTIONS]
Examples:
  khiops_coclustering -e log.txt
  khiops_coclustering -o scenario.txt
  khiops_coclustering -i scenario.txt -r less:more -r 70:90

In the first example all the logs are stored in the file log.txt
In the second example, khiops_coclustering records all user interactions in the file scenario.txt
In the last example, khiops_coclustering replays all user interactions stored in the
file scenario.txt after having replaced 'less' by 'more' and '70' by '90'
```

Available options are:

-e <file>	store logs in the file
-b	batch mode, with no GUI
-i <file>	replay commands stored in the file
-o <file>	record commands in the file
(-r <string>:<string>)...	search and replace in the command file
-p <file>	store last progression messages
-v	print version
-h	print help

3.2. Python Khiops library

The pykhiops library is delivered with the main Khiops package, under the python directory.

It allows to perform any Khiops task (data management, data preparation, modeling, evaluation…) and to exploit any khiops analysis result from a python program.

The available modules inside the pykhiops library are:

- run: to run Khiops or Khiops coclustering from python
- dictionary: python classes to inspect Khiops dictionaries (available as JSON from Khiops)
- analysisResults: python classes to inspect Khiops analysis results (available as JSON from Khiops)
- coclusteringResults: python classes to inspect Khiops coclustering results (available as JSON from Khiops)

3.3. JSON file exports

Khiops can export dictionaries as well as any analysis result in files under the JSON format.

The pykhiops modules contain python classes with a simple access to all Khiops JSON reports. These classes completely exemplify the use of the Khiops JSON reports, and provide a pedagogic example of the JSON report manipulation. The coding style in pykhiops is neither "pythonic" nor optimized but rather generic and verbose so it can be easily translated to other structured programming languages.

The structure of the Khiops JSON reports is self-documented:

- In order to be human-readable, the files are in "beautified" form, that is, with a rather comfortable spacing and indentation.
- Most information is available as key-value pairs, where the keys resemble the labels used in Khiops report files (Khiops text reports with extension ".xls") and in the Khiops covisualization tools.

3.4. Technical limitations

The computational complexity of the coclustering training algorithm depends on the number of instances and the number of values per coclustering variable. Memory and CPU requirement are about 1 GB RAM and one to ten hours of computation per million of instances, depending on the complexity of the patterns in the data.

See Khiops guide for a more details on technical limitations.

4. Appendix : coclustering report format

This technical appendix describes the format of coclustering report files (.khc) .

Coclustering reports are used internally to store and manipulate coclustering models in the Khiops Coclustering and Khiops Covisualization tools. They are not intended to be exploited directly by the data miner. Mainly, they are organized as text files with tags and sections. Tag, values are separated by tabulations (‘/’ in the following for concision purpose).

- **#khiops <version>:** header with tool version
- **Dimensions <Dimension number>**: List of coclustering dimensions(variables) in an array
  Name/Type/Parts/Initial parts/Values/Typicality/Description.
  - Name: name of dimension
  - Type: Categorical or Numerical
  - Parts: number of parts
  - Initial parts: initial number of parts (before potential simplification of the coclustering model)
  - Values: number of values
  - Typicality: interest measure for the dimension
  - Description: textual description

- **Coclustering stats**: general statistics
  - ***Instances***: total number of instances covered by the coclustering model
  - ***Cells***: total number of cells in the coclustering model
  - ***Null cost***: cost of the null coclustering model, with one single cluster per dimension
  - ***Cost***: cost of the coclustering model
  - ***Level***: normalized interest measure (between 0 and 1) for the coclustering model
  - ***Initial dimensions***: initial number of dimensions at training time
  - ***Frequency variable***: name of (optional) variable specified to store counts per record at training time
  - ***Dictionary***: dictionary specified at training time
  - ***Database***: database specified at training time

- **Bounds**: bounds of numerical dimensions (only if necessary)
  Name/Min/Max.
  - Name: name of numerical dimension
  - Min: min value
  - Max: max value

- **Hierarchy <Dimension name>**: for each dimension, links between cluster and parent cluster
  Cluster/ParentCluster/Frequency/Interest/HierarchicalLevel/Rank/HierarchicalRank
  - Cluster: cluster identifier (cluster = part, group or interval according to type of dimension)
    In the numerical case, a cluster identifier is an interval
    (e.g.]-inf;0.8],]0.8;1.75],]1.75; +inf] or Missing).
  - ParentCluster: identifier of the parent cluster in the hierarchy (empty for root cluster)
  - Frequency: number of instances in the cluster
  - Interest: interest measure of the cluster w.r.t. its whole hierarchy
  - HierarchicalLevel: normalized measure of the information kept by the coclustering model when the hierarchy is unfolded on all dimensions until reaching this level ; allows to fold and unfold the coclustering hierarchy while keeping as much as information as possible
  - Rank: rank of clusters in the top-down list of clusters, with the smallest ranks at the top
  - HierarchicalRank: rank of clusters in the hierarchy, with the smallest ranks being the closest from the root of the hierarchy

- **Composition <Dimension name>**: for each categorical dimension, list of values per cluster
  Cluster/Value/Frequency/Typicality.
  - Cluster: cluster identifier
  - Value: value of the cluster
  - Frequency: number of instances for this value

  - Typicality: interest measure of the value w.r.t. its cluster

- **Cells**: list of cells of the coclustering model
  <Dimension name 1>/<Dimension name 2>/Frequency
  - <Dimension name 1>: cluster identifier for dimension 1
  - <Dimension name 2>: cluster identifier for dimension 2
  - Frequency: number of instances for the cell (cluster1*cluster2)

The following sections are managed by the Khiops Covisualization tool, for personalization purposes.

- **Annotation <Dimension name>**: for each dimension, user data per cluster
  Part/Expand/Selected/ShortDescription/Description
  - Part: cluster identifier
  - Expand : saves whether to fold or unfold the cluster
  - Selected : saves whether to select or not the cluster
  - ShortDescription: short label, used every rename the cluster
  - Description: multi-line text to describe each cluster

- **Preferences** : list of user preferences w.r.t. what is actually shown in Khiops Visualization
  Name/Value
  - Name: name of preference
  - Value: true ou false (selection indicator)

- **ExternalDataDefinition <Dimension name>**: external data per dimension
  - **FileName**: name of external file
  - **KeyIndex**: index of key field in the header line of the external file
    Field/Type/Visible : status of each field in the external file
    - Field: name of field
    - Type: type of field (categorical, numerical, text)
    - Visible: true of false