Module 8
8.4 Unsupervised learning: clustering and association rules in KNIME

By **María Martínez Rojas**
Associate Professor CD, University of Granada

By **José Manuel Soto Hidalgo**
Associate Professor, CEAR, University de Granada

This capsule focuses on how to implement the different unsupervised learning algorithms introduced in Module 6 (*Unsupervised learning: clustering and association rules*) in *KNIME*. We will create data flows representative of the data science life cycle to solve clustering problems and create association rules. As in the previous capsule, data sets already used in Modules 2, 3, and 6 (*Bioinformatic analysis of an omics problem*, *Data science and machine learning*, and *Unsupervised learning: clustering and association rules*) will be used as examples here. Specifically, we will analyze the gene expression data discussed in Module 2 and used as example data in Module 6.

1.1. Clustering

In this section we will create data flows to solve a clustering problem which will specifically focus on identifying groups in the data without using any a priori information known about the categories, types, classes, or groups in the data. The flow represented in figure 1 shows an implementation of the two types of clustering discussed in Module 6: hierarchical and *k*-means clustering. Hierarchical clustering can be implemented in two different ways, either by including an external distance model and with its own node, or by including the distance matrix. Finally, the results are visualized through a heatmap and different visualization options (continuous or discrete) are shown.

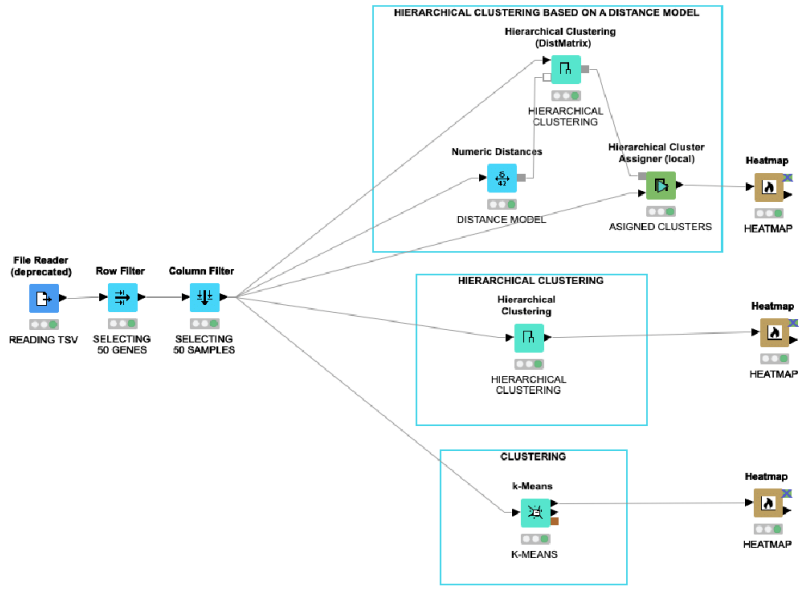

Figure 1. Data flow with hierarchical clustering and k-means analysis.

First, we read the data (expression matrix) with the "File Reader" node and, as an example used to facilitate our understanding of the dendrograms, we will select only a portion of the expression matrix (50 genes and 20 samples) with the "Column Filter" and "Row Filter" nodes. Once the 50 genes and 20 samples are selected, we demonstrate several ways to perform hierarchical clustering with *KNIME*. A distance model can be created ("Numeric Distances" node) which is used by the "Hierarchical Clustering" node ("DistMatrix") together with the data to perform the hierarchical clustering. Figure 2 shows the different configuration options of the "Numeric Distances" node in which different distances can be used (e.g., Euclidean, Manhattan, or Maximo, etc.) the variables (samples) to be considered in the clustering, and treatment of missing values.

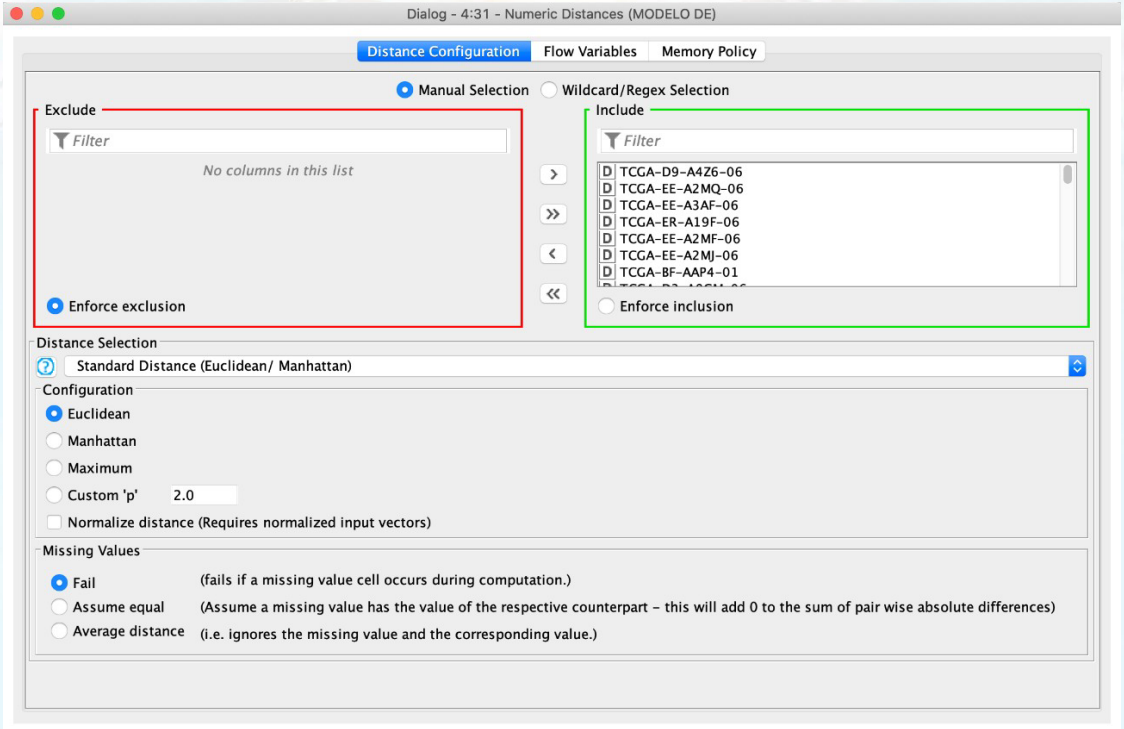

Figure 2. Options and parameters of the "Numeric Distances" node.

The "Hierarchical Clustering" node ("DistMatrix") generates a data model with the clusters that, together with the "Hierarchical Cluster Assigner" node and the data, assigns each entry to a cluster number. Finally, we can draw the heatmap with the "Heatmap" node and interact with it, for example, by visualizing it with continuous (figure 3) or discrete (figure 4) representation.

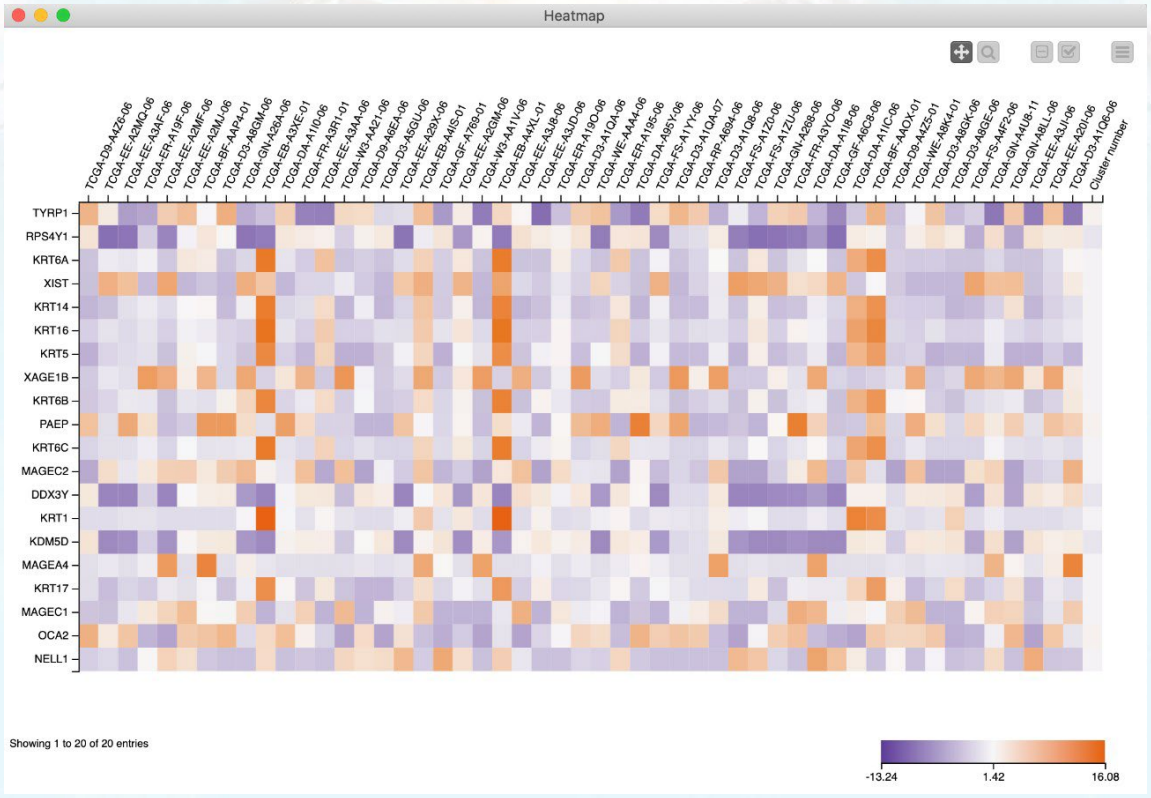

Figure 3. Heatmap visualization of results with continuous representation.

Figure 4. Heatmap results displayed as a discrete representation.

Hierarchical clustering can also be performed without using a distance model by using the "Hierarchical Clustering" node. This node presents options for the type of distance to use, number of output clusters, and type of union, as well as the variables to consider (figure 5).

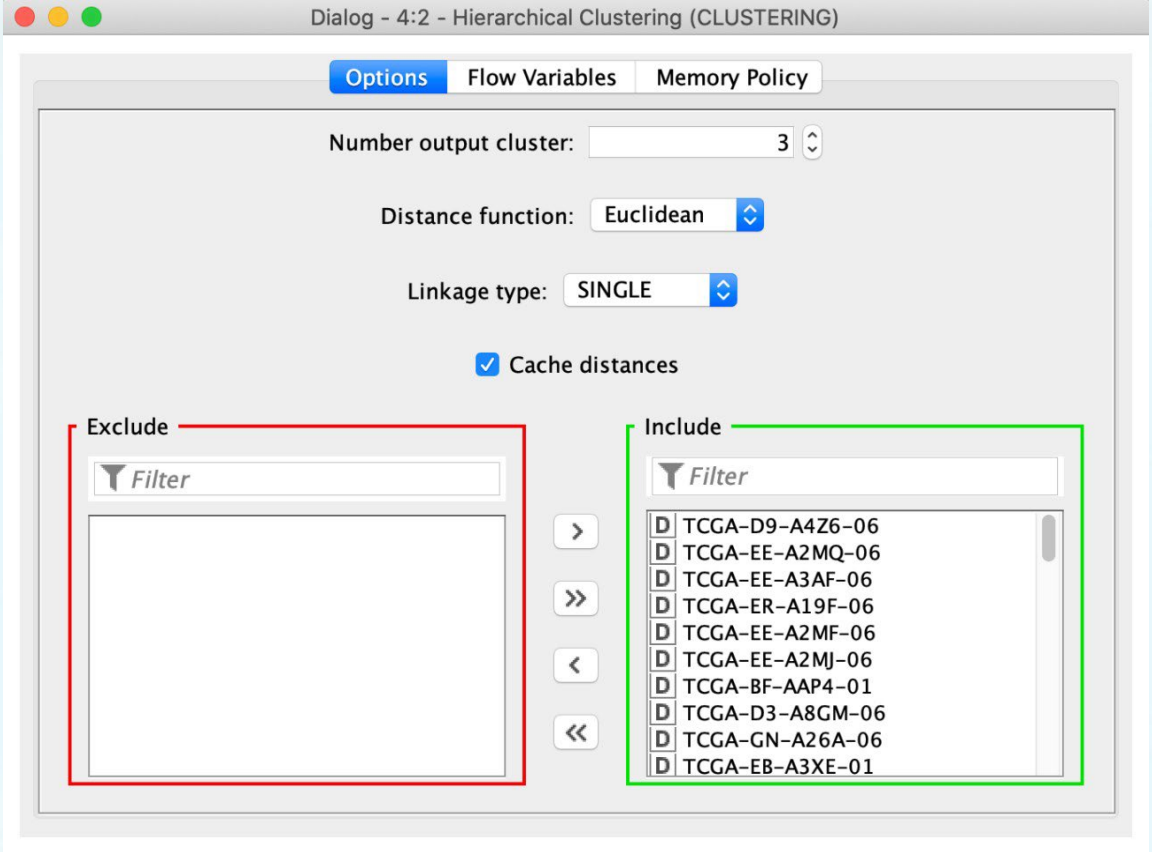

Figure 5. The "Hierarchical Clustering" node options and parameters.

We can also visualize and interact with the dendrogram by employing this node. To do so, right click on the node once it has been executed and select the "View: Dendrogram"/"distance view" option to show the result (figure 6).

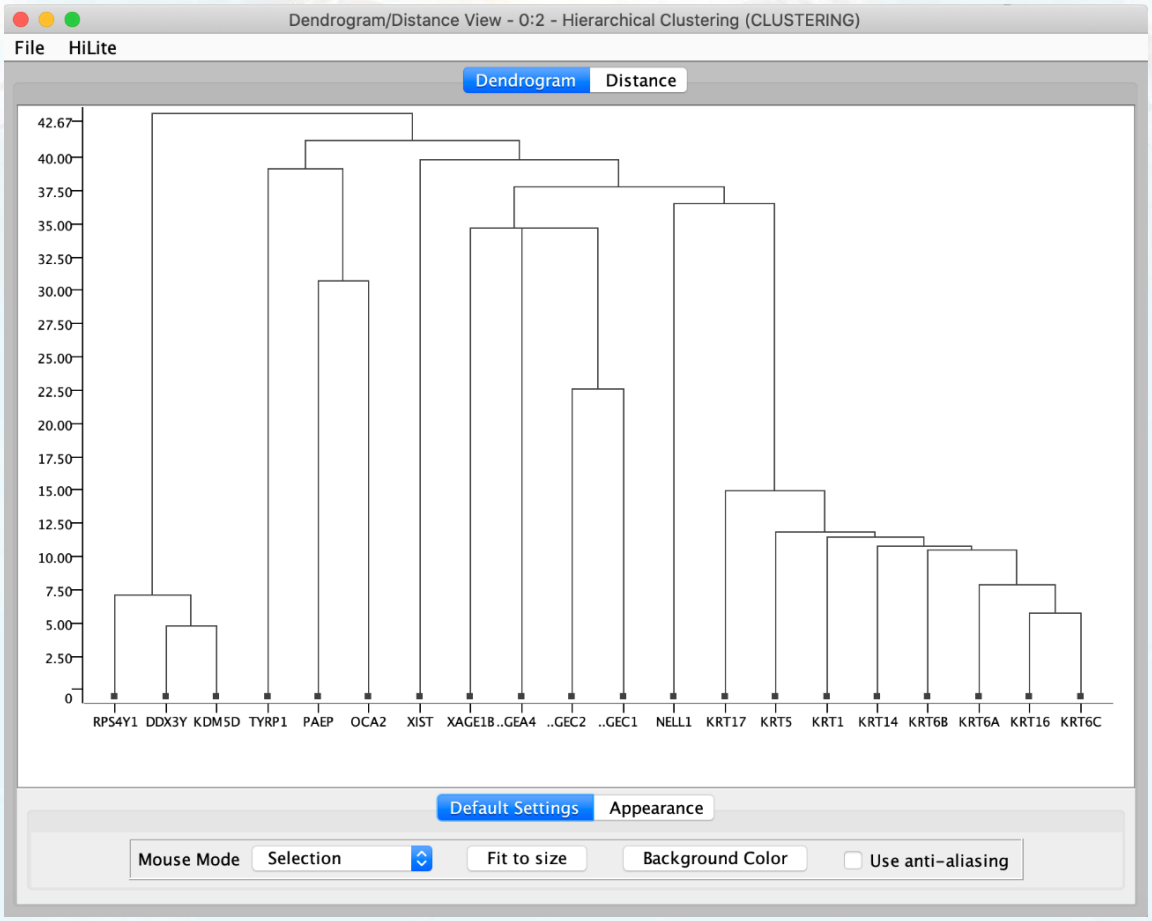

Figure 6. Visualization of the results as a dendrogram.

Finally, the "*k*-means clustering" option is also shown in the flow shown in figure 1; the "k-Means" node allows us to perform such clustering in a very simple way. As shown in figure 7, it allows us to set the number of clusters to obtain and randomly initiate a maximum number of iterations, etc.

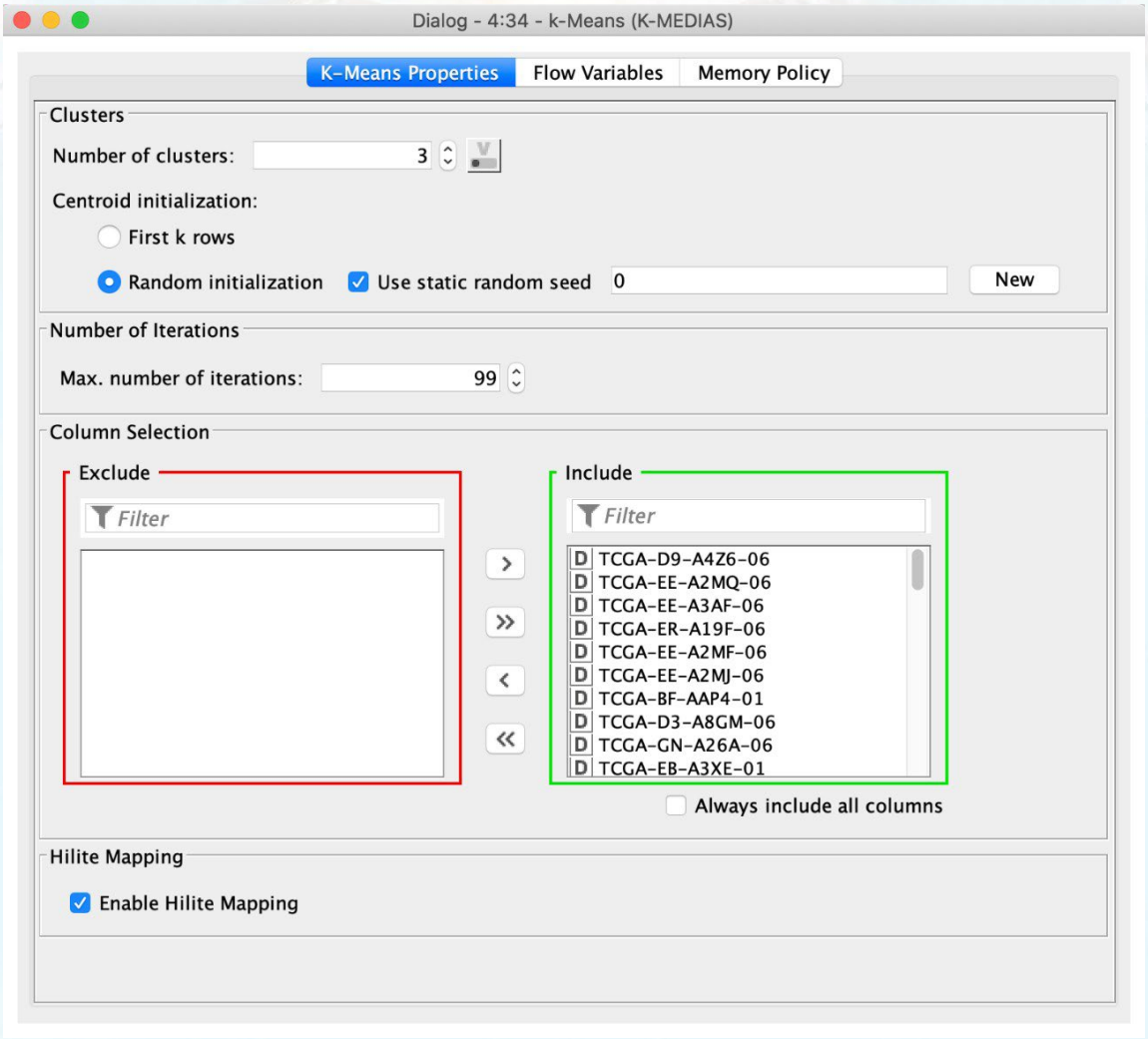

Figure 7. The options and parameters for the "k-Means" node.

1.2. Association rules

In this section we will create data flows to solve an association rules problem (figure 8) using the same data set as used in Module 6 (Capsule 3, *Association rules*), which comprises six variables:

- MUTATIONSUBTYPES
- UV-signature
- RNASEQ-CLUSTER_CONSENHIER
- MethTypes.201408
- MIRCluster
- LYMPHOCYTE.SCORE

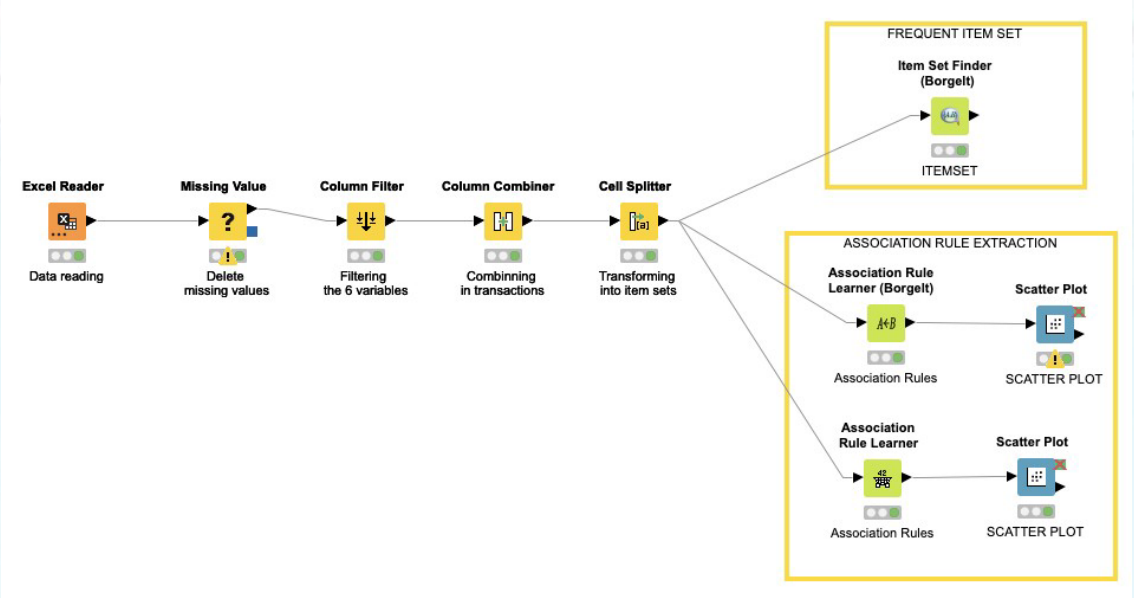

Figure 8. Data flow for association rules.

First, we use the "Excel Reader" node to read the data set file containing all the variables in the set. The "Column Filter" node is used to select 6 of the variables through the configuration options of this node, as shown in figure 9.

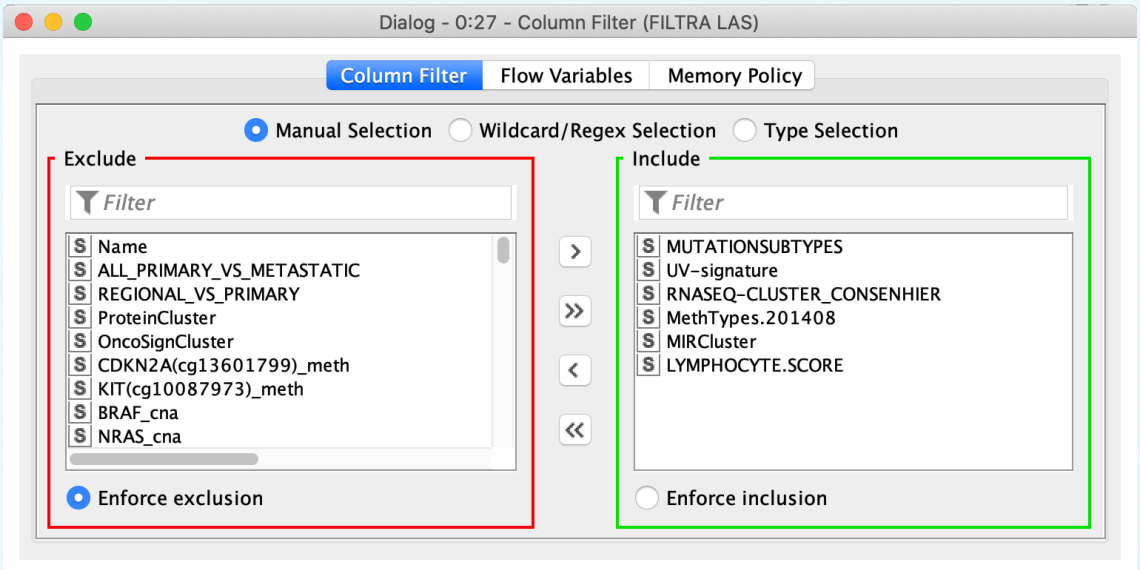

Figure 9. The "Column Filter" node configuration menu.

These two steps conclude the data preprocessing required to perform the transactions and calculate the association rules. As explained in Module 6 (*Unsupervised learning: clustering and association rules*), the first step in obtaining association rules is identifying what defines the data items and transactions. To do this we will use the "Column

Combiner" and "Cell Splitter" nodes to join the values of the variables into a single column and group them into an array. Figure 10 shows the output of the "Cell Splitter" node, showing the newly created "Transaction" column containing all the values for each of the variables in an array.

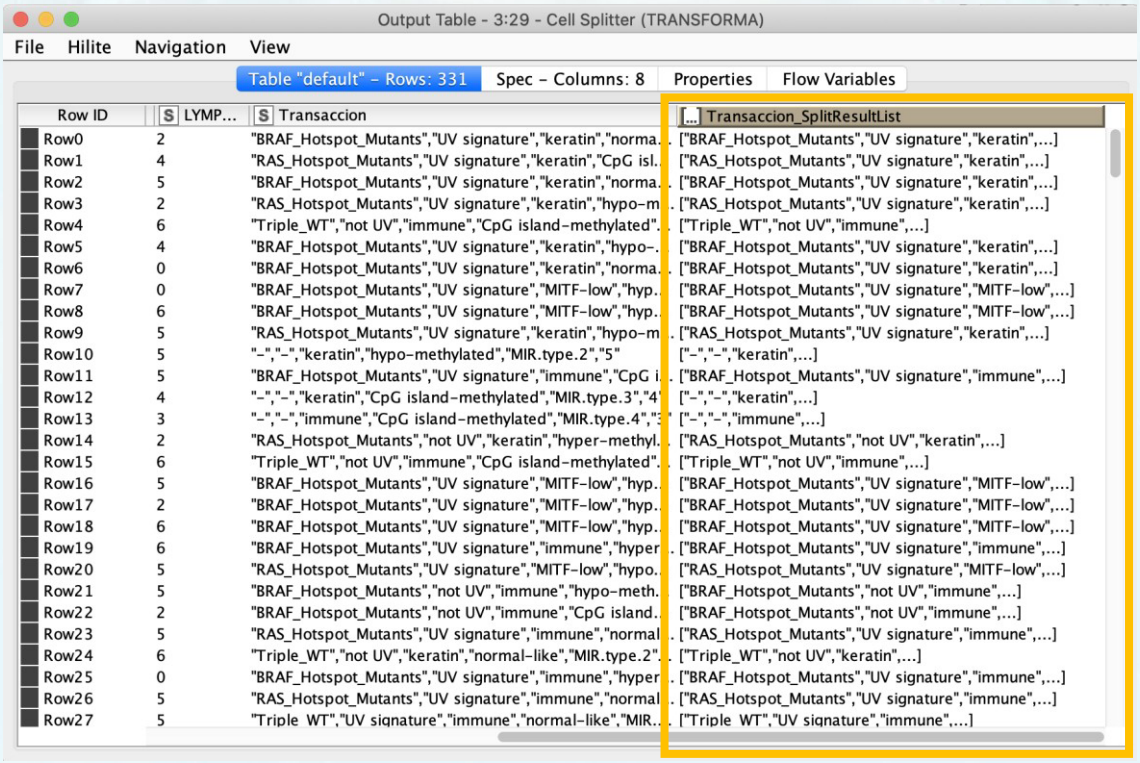

Figure 10. Output of the "Cell Splitter" node showing the transactions as an array.

These transactions can be used to determine a set of frequent items and to extract association rules. In the following, we illustrate how to represent these two examples in *KNIME*.

1. Determine a set of frequent item set

We can search for frequent items in a list of item sets with the "Item Set Finder" node which provides different algorithms for this task, including "A priori", "FP-growth", "RElim", "Sam", "JIM", "DICE", "TANIMOTO", as shown in figure 11. In addition, we can also determine the objective ("Frequent", "Closed", or "Maximal") and assign the Minimum set size and Minimum support levels for the items. In this example, we will explore the "A priori" algorithm (as detailed in Module 6, *Unsupervised learning: clustering and association rules*) which has a support value of 0.015. Figure 12 shows the item sets obtained when applying the "Item Set Finder" node with these settings, sorted by the relative percentage of support.

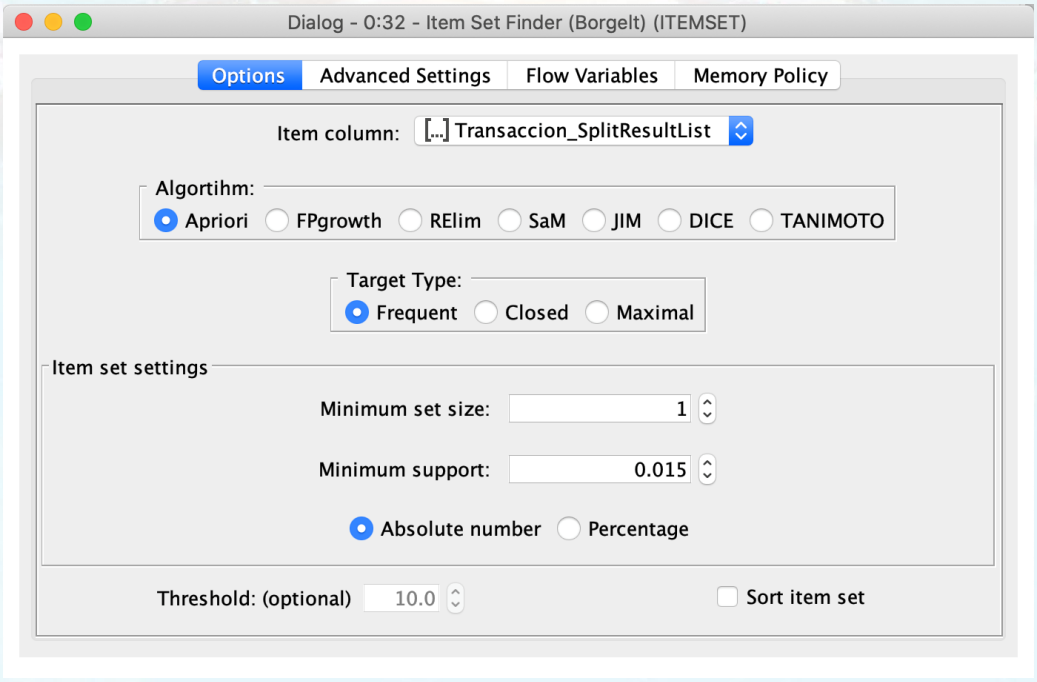

Figure 11. Options and parameters for the "Item Set Finder" node.

Item Sets - 3:32 - Item Set Finder (Borgelt) (ITEMSET)

File Hilite Navigation View

Table "default" - Rows: 4305 | Spec - Columns: 4 | Properties | Flow Variables

ItemSet	ItemSetSize	ItemSetSupport	RelativeItemSetSupport%
.. ["UV signature"]	1	265	80.06
.. ["immune"]	1	168	50.755
.. ["BRAF_Hotspot_Mutants"]	1	150	45.317
.. ["immune","UV signature"]	2	142	42.9
.. ["BRAF_Hotspot_Mutants","UV signature"]	2	136	41.088
.. ["keratin"]	1	100	30.211
.. ["RAS_Hotspot_Mutants"]	1	92	27.795
.. ["0"]	1	91	27.492
.. ["hyper-methylated"]	1	91	27.492
.. ["RAS_Hotspot_Mutants","UV signature"]	2	86	25.982
.. ["CpG island-methylated"]	1	85	25.68
.. ["hypo-methylated"]	1	84	25.378
.. ["MIR.type.2"]	1	83	25.076
.. ["MIR.type.1"]	1	82	24.773
.. ["MIR.type.3"]	1	81	24.471
.. ["2"]	1	77	23.263
.. ["BRAF_Hotspot_Mutants","immune"]	2	77	23.263
.. ["hypo-methylated","UV signature"]	2	73	22.054
.. ["0","UV signature"]	2	73	22.054
.. ["MIR.type.4"]	1	72	21.752
.. ["hyper-methylated","UV signature"]	2	72	21.752
.. ["normal-like"]	1	71	21.45
.. ["BRAF_Hotspot_Mutants","immune","UV signature"]	3	71	21.45
.. ["MIR.type.1","UV signature"]	2	70	21.148
.. ["MIR.type.2","UV signature"]	2	66	19.94
.. ["keratin","UV signature"]	2	66	19.94
.. ["CpG island-methylated","UV signature"]	2	65	19.637
.. ["MIR.type.3","UV signature"]	2	64	19.335
.. ["2","UV signature"]	2	63	19.033

Figure 22. Item sets obtained using for the "Item Set Finder" node and ordered by their relative percentage of support.

2. Extract and analyze the association rules

Association rules can be extracted with the "Association Rule Learner" node. Figure 13 shows the different parameters and configuration options available in the "Association Rule Learner" node, including the minimum number of items, minimum support, and minimum confidence levels. In this example, we set the minimum number of items to 3, a support of 0.015, and a confidence level of 80%.

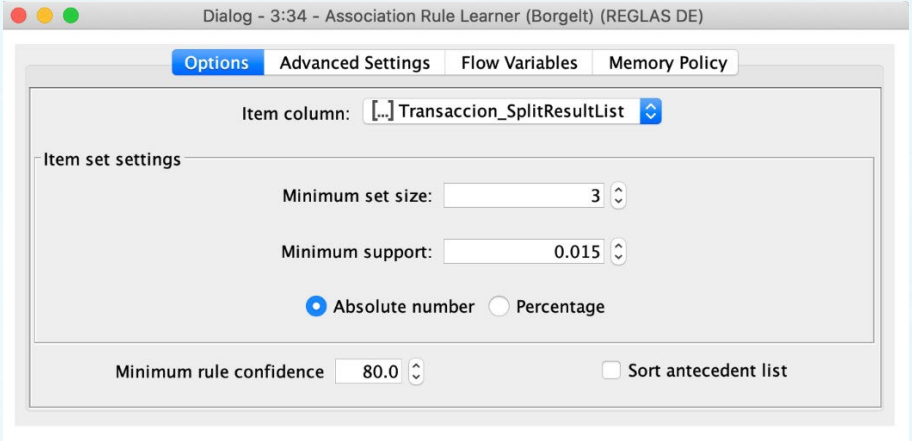

Figure 33. The "Association Rule Learner" node options and parameters.

Figure 14 shows a scatter plot indicating how the rules generated are distributed according to two metrics. In this example, the plot illustrates the percentage of confidence and support, although other metrics such as lift could also be used by selecting which ones to display in the upper right menu.

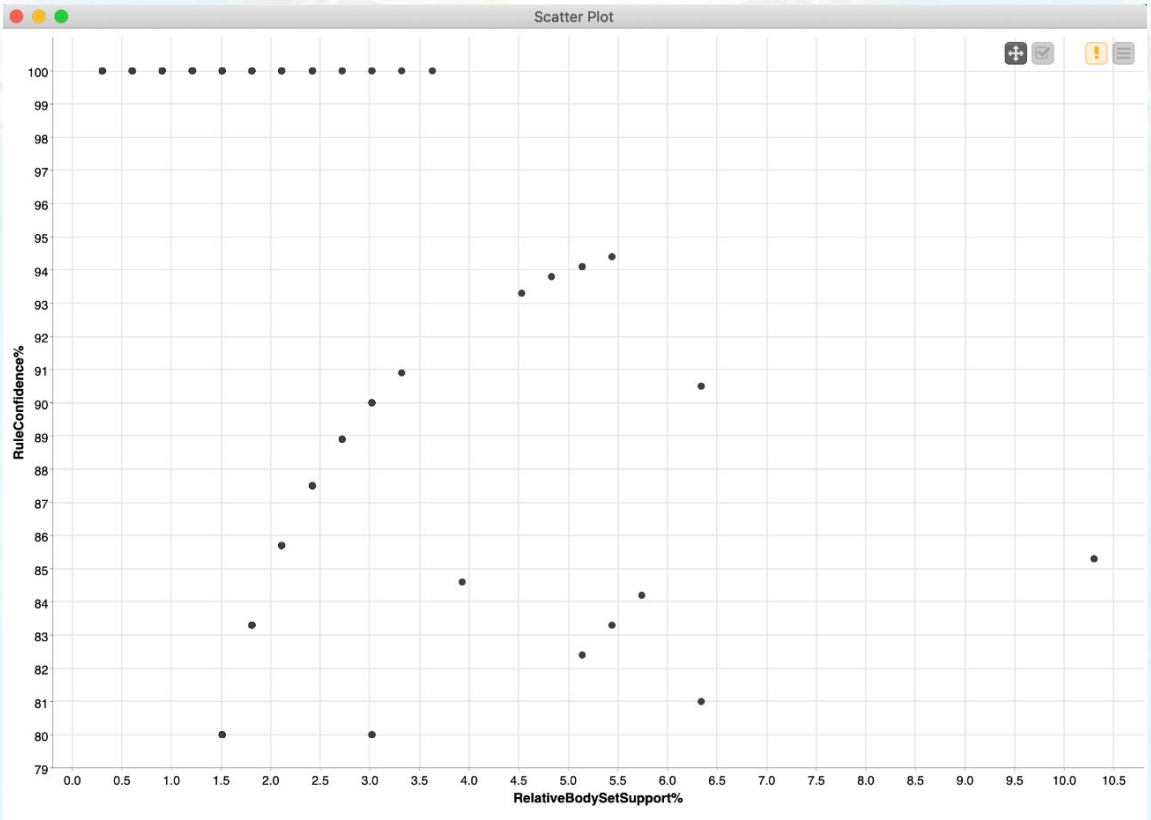

Figure 44. Scatter plot of the rules generated by applying the "Association Rule Learner" node, displayed as a function of the support and confidence metrics.

REFERENCES

- Bakos, G. (2013). KNIME essentials. Packt Publishing Ltd.
- Blokdyk, G. (2019). KNIME a Complete Guide - 2019 Edition. Emereo Pty Limited, 2019.
- McCormick, K. (2019). Introduction to Machine Learning with KNIME. linkedin.com.
- Silipo, R. (2016). Introduction to Data Analytics with KNIME: A Data Science Approach to Analytics. O'Reilly.
- Silipo, R., & Mazanetz, M. P. (2012). The KNIME cookbook. KNIME Press, Zürich, Switzerland.
- Strickland, J. (2016). Data Analytics Using Open-Source Tools. Lulu. com.