Table of Contents

Introduction

KniMet is a workflow for the post-processing of LC-MS, GC-MS and LC-IM-MS metabolomics data. It is based on the KNIME analytical platform (Berthold *et al.*, 2007) with integrated R (R Core Team, 2014) nodes, and allows to perform missing values imputation (MVI), feature filtering, normalisation, batch correction and feature annotation.

Installation

Install R

Download and install R from your preferred CRAN mirror, where you will find both the precompiled binary distributions for different operative systems, and the instructions for installation. Please refer to the R user guide which can be found here.

Once the R installation has completed, the package 'Rserve' (Urbanek, 2013) needs to be installed to allow the interaction between R and KNIME. Please install it in R using

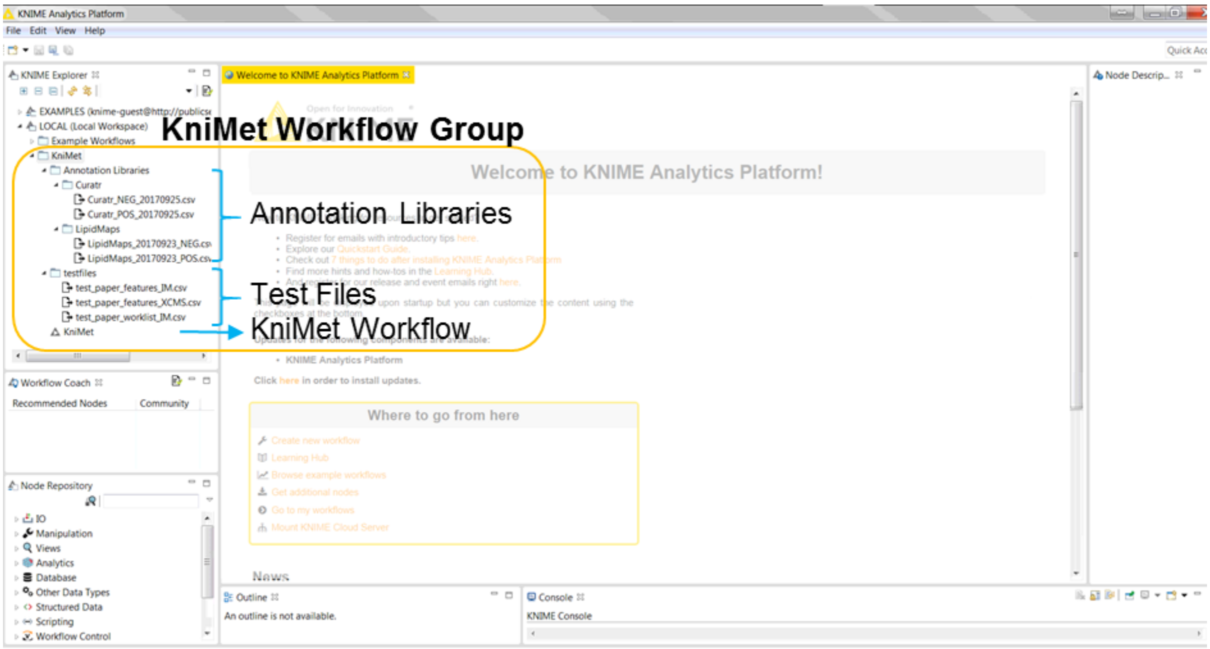
```
install.packages('Rserve')
```

Install KNIME

Download and install KNIME; please refer to the getting started page where instructions regarding the download, installation and usage are provided.

Download KniMet and import it in KNIME

Once the `KniMet.knar` file has been downloaded, it needs to be imported into KNIME: click on `File → Import KNIME workflow…`, in the window that opens tick `Select File` and browse to the directory where you saved the `KniMet.knar`, select it, and click `Finish`.
The KniMet workflow group will now be visible on the `KINME Explorer` pane, and can be expanded to see its content by clicking on the white arrow on its left. Apart from the actual KniMet workflow, it also contains the `Annotation Libraries` folder and a `test files` folder. The `Annotation Libraries` folder contains both positive and negative ionisation mode files downloaded from the Curatr - EMBL-Metabolomics CoreFacility Spectral Library (http://curatr.mcf.embl.de/MS2/export/) (Palmer *et al.*, 2017), as well as two files resulting from a merge of the outputs of the LIPID MAPS Mass Spectrometry Combinatorial Expansion Package (Fahy *et al.*, 2007) for [M+H]+ and [M-H]- adducts.



By double clicking on the KniMet workflow, you will be prompted to an error window asking for installation of the missing and required extensions: click on `Yes` and then in the `Install` window that opens `Next >`, `I accept the terms of the licence agreements` and `Finish`. You will need to restart KNIME to apply the changes.

Point KNIME to use local version of R:

Click on `File→ Preferences` and then `KNIME → R` on the left hand side of the open window, and point to your local installation of R (usually `C:\Program Files\R\R-x.x.x` where x.x.x stands for the version number of your R installation) by clicking on the browse button on the right.

Congratulations, you succesfull installed all the requirements to run the KniMet workflow!

How to use it

To start analysing the data, first open the KniMet workflow. You might get a warning saying `Warning during load Reason: KniMet loaded with warnings` which can be ignored by clicking on `OK`. The several processing steps performed by this pipeline will be explained one by one in the following section.

1. Perform data deconvolution/Import deconvoluted data**.** The user can decide whether to pre-process the data directly in the KNIME platform by using the connection with the locally installed R version provided by the **R Source (Table) - Perform pre-processing with XCMS and CAMERA** node, or instead read the data matrix obtained from deconvolution performed externally with the **CSV Reader – Read Deconvolution output** node.

   1.1. In case the user would like to perform deconvolution using XCMS (Smith *et al.*, 2006) and CAMERA (Kuhl *et al.*, 2012) inside the workflow, the **R Source (Table) - Perform pre-processing with XCMS and CAMERA** node could be configured for the scope. Double clicking on it, or right clicking and then clicking on `Configure…` will open the configuration window, where in the central pane will be shown an R script to perform deconvolution with XCMS and peak annotation with CAMERA. In the example R script provided, the data package faahKO (Saghatelian *et al.*, 2004) is used, and the results obtained from it can be used to test the pipeline. Otherwise, the user could edit the script to perform deconvolution on their data. Once the editing is completed, the node can be run by clicking `Ctrl+Enter` from the configuration window, or by first saving the changes clicking on `OK` in the configuration window and then right clicking on the node and selecting `Execute`. Please note that on the first run of this node, as well as in several other steps of this pipeline involving R nodes, you will be prompted to a window asking for permission to install some R libraries which are missing in your local R installation. Accept, select the CRAN mirror that you would like to use and wait for the installation to finish. Once the execution of the node has terminated, the user can access to the deconvoluted data matrix, as well as the R standard error and output, by right clicking on the node and selecting `View: R Std Output`, `View: R Error Output` or `Data from R` from the drop-down menu that opens.

1.2. In case the deconvolution step has been performed externally, the matrix containing the analysed samples and other information (such as m/z, RT, etc.) as columns, and the detected features as rows can be imported using the **CSV Reader – Read Deconvolution output** node. Double click on the node, or right click on the node and click on `Configure…` will open the configuration window. From the `Settings` tab click on `Browse` and select the input data that you wish to analyse. Alternatively, if you want to try analysing one of the files provided here as examples, point your cursor over it in the KNIME Explorer, right click and select `Copy Location → Absolute URL`. You can now copy the path of the data file under `Input location:`. Make sure that the right parameters are selected, such as column delimiter = \t for tab separated files, and tick `Has Column Header`. Usually MassHunter MassProfiler, which is the program used to deconvolute Ion Mobility – MS (IM-MS) data acquired with the Agilent 6560 Instrument, inserts in the file 4 initial lines that do not need to be imported. The number of rows to be read can be modified by moving to the `Limit Rows` tab, ticking on `Skip first lines` and selecting the number of lines to avoid (4 in this case). When the configuration process is finished, the settings can be saved by clicking on `OK` and the node can be run by right clicking on it and selecting `Execute`, the second voice from the drop down menu, or by clicking `Ctrl+Enter` from the configuration window.

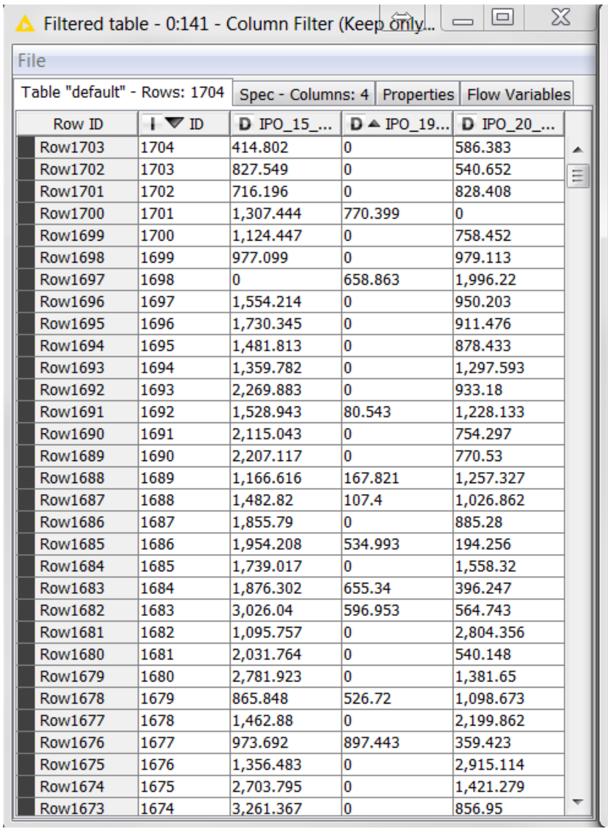
Once the chosen initial node has been executed, the imported table can be visualised by right clicking on it and selecting `File Table` at the bottom of the drop down menu. We suggest the user to always check the output of each step and make sure that the result is what was expected. The output port of the run node needs to be connected to the input port of the following node to proceed with the processing. In particular, If the output from **CSV Reader – Read Deconvolution output** node contains a column named `ID` with integers as feature identifiers, and the column containing the m/z values is named `mz`, its output can directly be connected to **Remove special characters and spaces from column names** in point 3. These columns can be missing or present different names when deconvolution is performed with either the **R Source (Table) - Perform pre-processing with XCMS and CAMERA** or when the deconvoluted data obtained, for instance, from XCMS online (Tautenhahn *et al.*, 2012) is imported with the **CSV Reader – Read Deconvolution output**. In these cases, then the output of the node needs to be connected to the input port of the following node **Add ID column, modify "mzmed" to "mz** described in point 2.

2. Modify Column Names. The node **Add ID column, modify "mzmed" to "mz"** should be executed only when the `ID` column is missing and the m/z column is not called `mz`, as it is needed to insert a feature identifier column containing unique integers and to modify the name

of the column `mzmed` to `mz`. SKIP if working with output from MassHunter: in this case the output of the previous node can be connected to the input port of the node described in point 3. Does not need to be configured, simply right click on it and click `Execute`.

3. Remove special characters and spaces from column names. This node will modify column names in order to replace spaces with underscores and delete any character different from letters, numbers and underscores. Does not need to be configured, simply right click on it and click `Execute`.

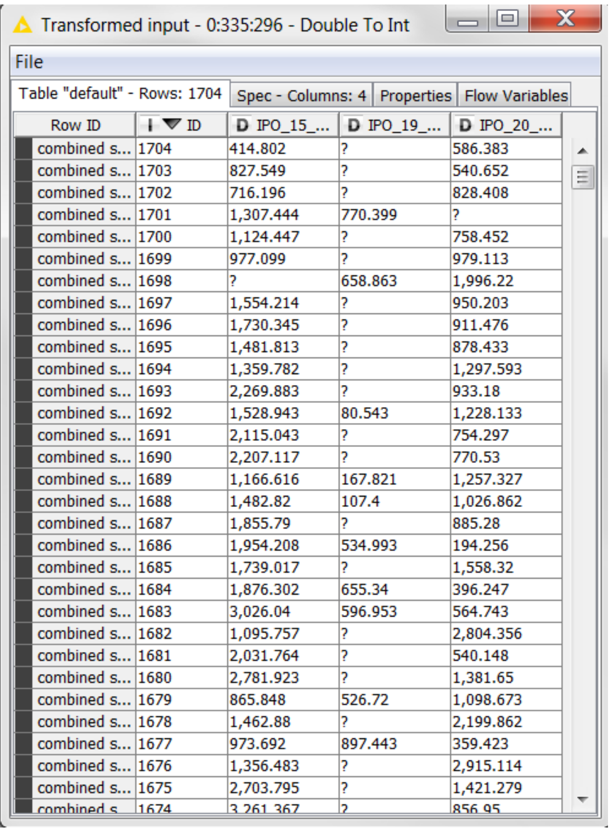
4. Keep only samples and ID**.** Like all the other column filters that will follow, the **Column Filter - Keep only samples and ID** configuration window is characterised by a left pane bordered in red (`Exclude`) where the columns to be discarded should be listed, and a right pane bordered in green (`Include`) where only the columns to be kept should be listed. Columns can be moved from one side to the other by selecting them and using the buttons in between the two panes `add >>`, `add all >>`, `<< remove` and `<< remove all`. As suggested by the description below the node, at this point only the columns containing the feature identifiers and the samples should be kept. Hence, they should be moved to the right pane whereas all the other columns containing other types of information (mz, RT, average intensity, etc) should be directed to the left hand side. You might find that in the `Exclude` panel there are some entries enclosed in a red square which do not correspond to any of the columns in your dataset. These derive from the previous selection, as the ticked `Enforce exclusion` voice under the red panel keeps memory of the columns excluded the last time that the node was run. Once the configuration has terminated, the node can be executed.
5. Insert missing values symbol. Two nodes are available depending on the source of the input file, **Replace 0.001 with missing values - Use with MassHunter output** and **Replace 0 with missing values - Use with XCMS output**. Indeed, when a feature is not found in a given sample, XCMS inserts a zero whereas MassHunter inserts 0.001. Either way, these are interpreted as values from the following missing value imputation nodes, hence they need to be replaced with the missing value symbol. Once the right node has been connected to the output from the the **Column Filter - Keep only samples and ID** node described in point 4, it can be run with no configuration needed. Make sure that the replacement has been performed correctly and check the difference between the outputs of this and the previous node.



Filtered table - 0:141 - Column Filter (Keep only...

Table "default" - Rows: 1704 | Spec - Columns: 4 | Properties | Flow Variables

Row ID	ID	IPO_15_...	IPO_19...	IPO_20_...
Row1703	1704	414.802	0	586.383
Row1702	1703	827.549	0	540.652
Row1701	1702	716.196	0	828.408
Row1700	1701	1,307.444	770.399	0
Row1699	1700	1,124.447	0	758.452
Row1698	1699	977.099	0	979.113
Row1697	1698	0	658.863	1,996.22
Row1696	1697	1,554.214	0	950.203
Row1695	1696	1,730.345	0	911.476
Row1694	1695	1,481.813	0	878.433
Row1693	1694	1,359.782	0	1,297.593
Row1692	1693	2,269.883	0	933.18
Row1691	1692	1,528.943	80.543	1,228.133
Row1690	1691	2,115.043	0	754.297
Row1689	1690	2,207.117	0	770.53
Row1688	1689	1,166.616	167.821	1,257.327
Row1687	1688	1,482.82	107.4	1,026.862
Row1686	1687	1,855.79	0	885.28
Row1685	1686	1,954.208	534.993	194.256
Row1684	1685	1,739.017	0	1,558.32
Row1683	1684	1,876.302	655.34	396.247
Row1682	1683	3,026.04	596.953	564.743
Row1681	1682	1,095.757	0	2,804.356
Row1680	1681	2,031.764	0	540.148
Row1679	1680	2,781.923	0	1,381.65
Row1678	1679	865.848	526.72	1,098.673
Row1677	1678	1,462.88	0	2,199.862
Row1676	1677	973.692	897.443	359.423
Row1675	1676	1,356.483	0	2,915.114
Row1674	1675	2,703.795	0	1,421.279
Row1673	1674	3,261.367	0	856.95



Transformed input - 0:335:296 - Double To Int

Table "default" - Rows: 1704 | Spec - Columns: 4 | Properties | Flow Variables

Row ID	ID	IPO_15_...	IPO_19_...	IPO_20_...
combined s...	1704	414.802	?	586.383
combined s...	1703	827.549	?	540.652
combined s...	1702	716.196	?	828.408
combined s...	1701	1,307.444	770.399	?
combined s...	1700	1,124.447	?	758.452
combined s...	1699	977.099	?	979.113
combined s...	1698	?	658.863	1,996.22
combined s...	1697	1,554.214	?	950.203
combined s...	1696	1,730.345	?	911.476
combined s...	1695	1,481.813	?	878.433
combined s...	1694	1,359.782	?	1,297.593
combined s...	1693	2,269.883	?	933.18
combined s...	1692	1,528.943	80.543	1,228.133
combined s...	1691	2,115.043	?	754.297
combined s...	1690	2,207.117	?	770.53
combined s...	1689	1,166.616	167.821	1,257.327
combined s...	1688	1,482.82	107.4	1,026.862
combined s...	1687	1,855.79	?	885.28
combined s...	1686	1,954.208	534.993	194.256
combined s...	1685	1,739.017	?	1,558.32
combined s...	1684	1,876.302	655.34	396.247
combined s...	1683	3,026.04	596.953	564.743
combined s...	1682	1,095.757	?	2,804.356
combined s...	1681	2,031.764	?	540.148
combined s...	1680	2,781.923	?	1,381.65
combined s...	1679	865.848	526.72	1,098.673
combined s...	1678	1,462.88	?	2,199.862
combined s...	1677	973.692	897.443	359.423
combined s...	1676	1,356.483	?	2,915.114
combined s...	1675	2,703.795	?	1,421.279
combined s	1674	3,261.367	?	856.95

6. Keep QC samples**.** The aim of the **Column Filter - Keep only QC samples** node is keeping only the columns regarding the quality controls/pooled samples (from now on named QC), which will then be used in the following step to perform features filtering. Configuration consist in moving only the QC samples to the `Include` panel, while all the other columns should be listed under `Exclude` (please refer to point 4 for details on the configuration). In case QC samples are not available, or the user does not want to perform a filtering based on them, this step can be skipped and the output of the previous node can directly be connected to the desired imputation method (point 8).

7. Features Filtering. As explained in the name and description of this step, **QC-based Features Filtering - Remove features present in less than 50% of the QCs; filter features whose RSD > 20%** removes all the features not consistently detectable across the QC runs will be deleted from the data matrix. Make sure that the top input of this node is connected to the previous column filter described in point 6, while the bottom input port should be connected to the output from the node used in step 5. Does not need to be configured, simply right click on it and click `Execute`.

8. Missing Values Imputation**.** The user should evaluate the origin of the missing values in their data matrix, consecutively decide which MVI method is the most appropriate and connect the output from the previous node (either point 5 or point 7) to the method of choice to perform MVI with either Random Forest (**Missing Values Imputation RF**), Key-nearest neighbour (**Missing Values Imputation KNN**) or Small Value (**Missing Values Imputation SV**). Does not need to be configured, simply right click on it and select `Execute`. Please note that on the first run of this step you will be prompted to the installation of the R packages required by these nodes and not already installed in your local R version.

9. Normalisation. The nodes **PQN-QC** and **PQN-ALL** are designed to perform Probabilistic Quotient Normalisation either on all samples or only on the QCs (if available).

   9.1. In case the **PQN-QC** method is chosen, the previous **Column filter - Keep only QCs** node should be executed to select only the QC samples (configuration of this node follows the same direction provided above in point 4). Once the column filter has been executed, make sure that its output is connected to the top input port of the **PQN-QC** node, whereas the output of the MVI node of choice should be connected to the bottom input port.

   9.2. In case **PQN-ALL** is chosen, no preliminary nodes need to be run, and its only input port should be connected with the output from the MVI method used.

The PQN nodes do not need to be configured and can be run by right clicking on them and clicking on `Execute`.

10. Import meta-information for batch correction. The node **CSV reader - read file containing sample names, order of injection, sample type and batch ID** is needed to import into the pipeline the information required to perform Batch correction (as described in point 11). The file to be imported should be in the following format:

SampleName	**injectionOrder**	**sampleType**	**Batch**
QC1	1	pool	1
Sample1	2	sample	1
Sample2	3	sample	2
….	….	….	…

Please note that a correct functioning of the following batch correction nodes is possible only if the file contains exactly these column names and the sample names are the same as those present in the data matrix fed to the CSV reader in point 1. Once this node has been executed, its output should be connected to the bottom input port of the Batch Correction method of choice described in point 11.

11. Batch Correction. This step is required when the samples were acquired in multiple analytical runs and a batch effect is still present after PQN normalisation. Three methods are available, namely **ALL-Loess Batch Correction**, **QC -Loess Batch Correction** and **ComBat Batch Correction**.

    11.1. **QC-Loess Batch Correction.** This method is based on the R script and R wrapper developed by the Workflow4Metabolomics team to perform the robust locally estimated scatterplot smoothing (LOESS) signal correction (RLSC) (Dunn *et al.*, 2011; Thévenot *et al.*, 2015; Giacomoni *et al.*, 2015) using a span for loess calculation equal to 1.0. Loess batch correction is performed based on the QC samples, which should be properly indicated in the metadata file imported in point 10, only if there are at least 5 QC runs per batch, otherwise linear regression will be used. This node provides 5 outputs: the first data output is the corrected data matrix, the second and the third plot both the sum of intensities for each sample and the PCA score plots of components 1-4 before and after signal correction, and finally the fourth and the fifth are respectively the standard output and error of the R calculation.

11.2. **ALL-Loess Batch Correction.** Differs from the QC-based approach only on the fact that correction is performed on all samples.

11.3. **ComBat Batch Correction.** This step is performed using the R package SVA (Leek *et al.*, 2011, 2012) using the method ComBat (Johnson *et al.*, 2007). The three output port carry the transformed data matrix, the standard output and the standard error of the R calculation

Neither of them needs to be configured, just make sure that the output from the **CSV reader - read file containing sample names, order of injection, sample type and batch ID** (point 10) is connected to the bottom input port and the output from the **PQN-QC/PQN-ALL** node to the top input port.

12. Joiner - Add features info (m/z, RT, DT, CCS)**.** This node is needed to add to the processed signals their m/z values, fundamental for the following annotation step, as well as any other information that might be useful (such as RT or CCS for IM data). The top input port of this node has to be connected to the output of the previous node (either point 9 or 11), while the bottom input table should be connected to the output from **Remove special characters and spaces from column names** (point 3). The configuration pane of this node consist of several tabs. In the `Joiner Setting` tab, make sure that `Inner join` is selected under the `Join Mode` section, while under the `Joining columns` section of the same tab the ID column has to be selected for both `Top input ('left table')` and `Bottom input ('right table')`. In this way, the values in the ID column will be used to match the two tables and join the normalised signals will the other information relative to them. The second tab `Column Selection` is the one where the columns to be joined from the input tables have to be selected. Its layout is similar to the column filter described above (point 3), with the difference that there will be a section for the top input table and another for the bottom input table. Move the columns that need to be kept in the output in the `Include` box of both sections, and make sure that `Filter duplicates` is selected under `Duplicate columns handling` and `Remove joining columns from bottom input` is ticked under `Joining columns handling`. In this way, if by accident a column relative to a given samples is included from both top (normalised) and bottom (original) inputs, only the transformed data is kept. Once the configuration is completed, the node can be run.

13. CSV Reader - Select Annotation Library File**.** This node allows to import an MS library in text format (such as .csv or .tsv) which can be used to annotate features. In particular, embedded into the workflow group we provided part of the LIPID MAPS Library and the Curatr library. To select one of these library, right click on one of them in the `KNIME Explorer` pane in the left hand side of your KNIME window, click on `Copy Location` and then on `Absolute URL`. Now that you have copied the path of the library, you can paste it into the configuration pane of the CSV Reader. Make sure also that `Has column header` and `Support short lines` are ticked, and finally execute the node.

14. Annotation. The output from the previous step need to be connected with either the **Lipid Maps Annotation** or the **Curatr - EMBL Metabolomics Core Facility Spectral Library Annotation** node depending on the annotation file loaded in the previous step (point 13). Both nodes will perform an annotation of the features based on mass accuracy, with only annotations with Δppm < 20 retained.

15. CSV Writer**.** The transformed and annotated data can be written to an output file in the desired format. Configuration of the **CSV Writer** node includes indicating location and filename of the

output file to be written by clicking on `Browse`, ticking `Write column header` and selecting whether the file should be overridden in case of name already assigned, or if the node execution should fail. Once the node has been configured, it can be run and the output is ready for any other analysis that the user wishes to perform outside KNIME.

In case the user would like to save the output in the xls format, the **XLS Writer** node can be selected form the `Node Repository` tab on the left hand side of the main KNIME window. The node can be imported into the pipeline by double clicking on it, connected to the output port of the last executed node and finally configured in a way similar to what described for the CSV reader.

References

Berthold,M.R. *et al.* (2007) KNIME: The Konstanz information miner. In, *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, pp. 319–326.

Dunn,W.B. *et al.* (2011) Integration of metabolomics in heart disease and diabetes research: current achievements and future outlook. *Bioanalysis*, **3**, 2205–2222.

Fahy,E. *et al.* (2007) LIPID MAPS online tools for lipid research. *Nucleic Acids Res.*, **35**, W606–W612.

Giacomoni,F. *et al.* (2015) Workflow4Metabolomics: A collaborative research infrastructure for computational metabolomics. *Bioinformatics*, **31**, 1493–1495.

Johnson,W.E. *et al.* (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.

Kuhl,C. *et al.* (2012) CAMERA: An integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Anal. Chem.*, **84**, 283–289.

Leek,J. *et al.* (2011) The SVA package for removing batch effects and other unwanted variation in high-throughput experiments.

Leek,J.T. *et al.* (2012) The SVA package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, **28**, 882–883.

Palmer,A. *et al.* (2017) Curatr: a web application for creating, curating, and sharing a mass spectral library. *bioRxiv*.

R Core Team (2014) R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria.

Saghatelian,A. *et al.* (2004) Assignment of Endogenous Substrates to Enzymes by Global Metabolite Profiling †. *Biochemistry*, **43**, 14332–14339.

Smith,C.A. *et al.* (2006) XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. *Anal. Chem.*, **78**, 779–787.

Tautenhahn,R. *et al.* (2012) XCMS online: A web-based platform to process untargeted metabolomic data. *Anal. Chem.*, **84**, 5035–5039.

Thévenot,E.A. *et al.* (2015) Analysis of the Human Adult Urinary Metabolome Variations with Age, Body Mass Index, and Gender by Implementing a Comprehensive Workflow for Univariate and OPLS Statistical Analyses. *J. Proteome Res.*, **14**, 3322–3335.

Urbanek,S. (2013) Rserve: Binary R server.