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 (*Figure 1*). Apart from the actual KniMet workflow, it also contains the `Annotation Libraries` 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 select `Next >`, `I accept the terms of the licence agreements` and `Finish`. You will need to restart KNIME to apply the changes. When you will open again KniMet after restart, you might get a window saying `Warning during load Reason: KniMet loaded with warnings`. It derives from previous configurations, it can be ignored by clicking on `OK`.

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!

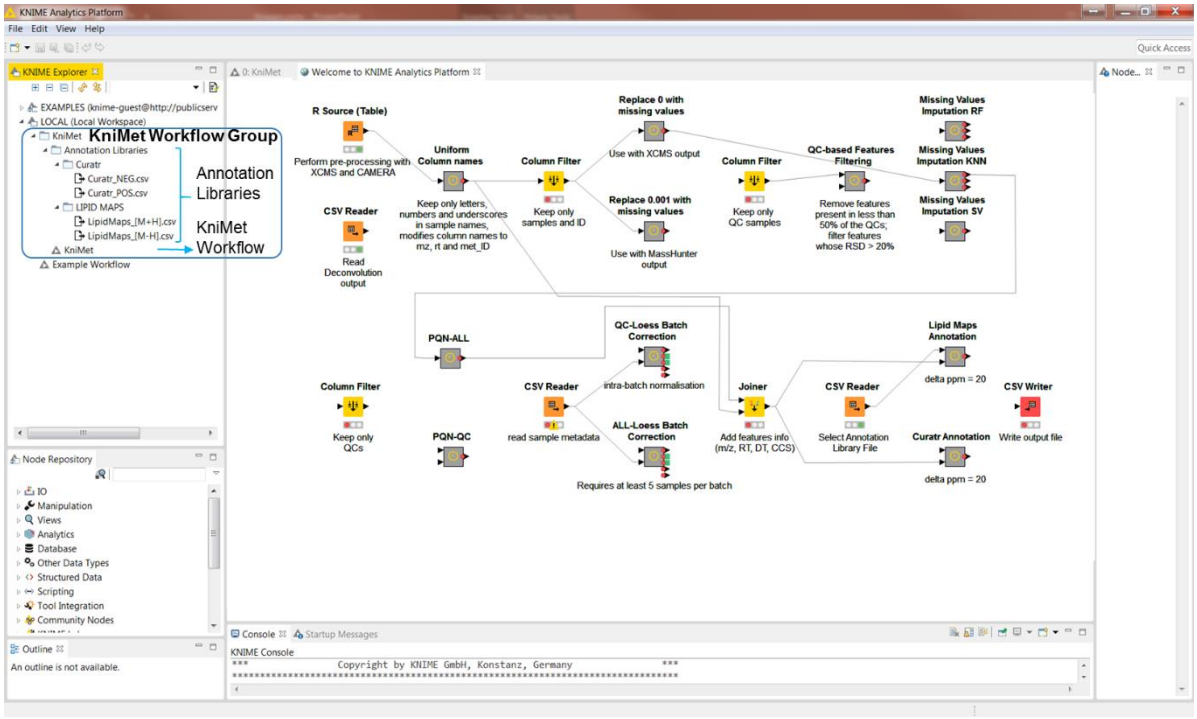

Figure 1: The KniMet workflow group is highlighted on the KNIME Explorer pane, while on the central pane the KniMet pipeline is shown

What does it do?

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 (*Figure 2*).

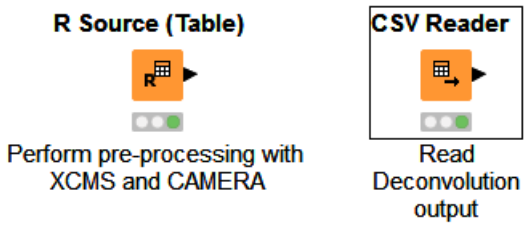

Figure 2: The R Source node on the left hand side allows data deconvolution within the KniMet platform, while the CSV Reader on the right imports a data matrix into the platform.

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 (shown in *Figure 3*), 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 stderr and stdout, by right clicking on the node and selecting `Data from R`, `View: R Std Output` or `View: R Error Output` from the drop-down menu that opens.

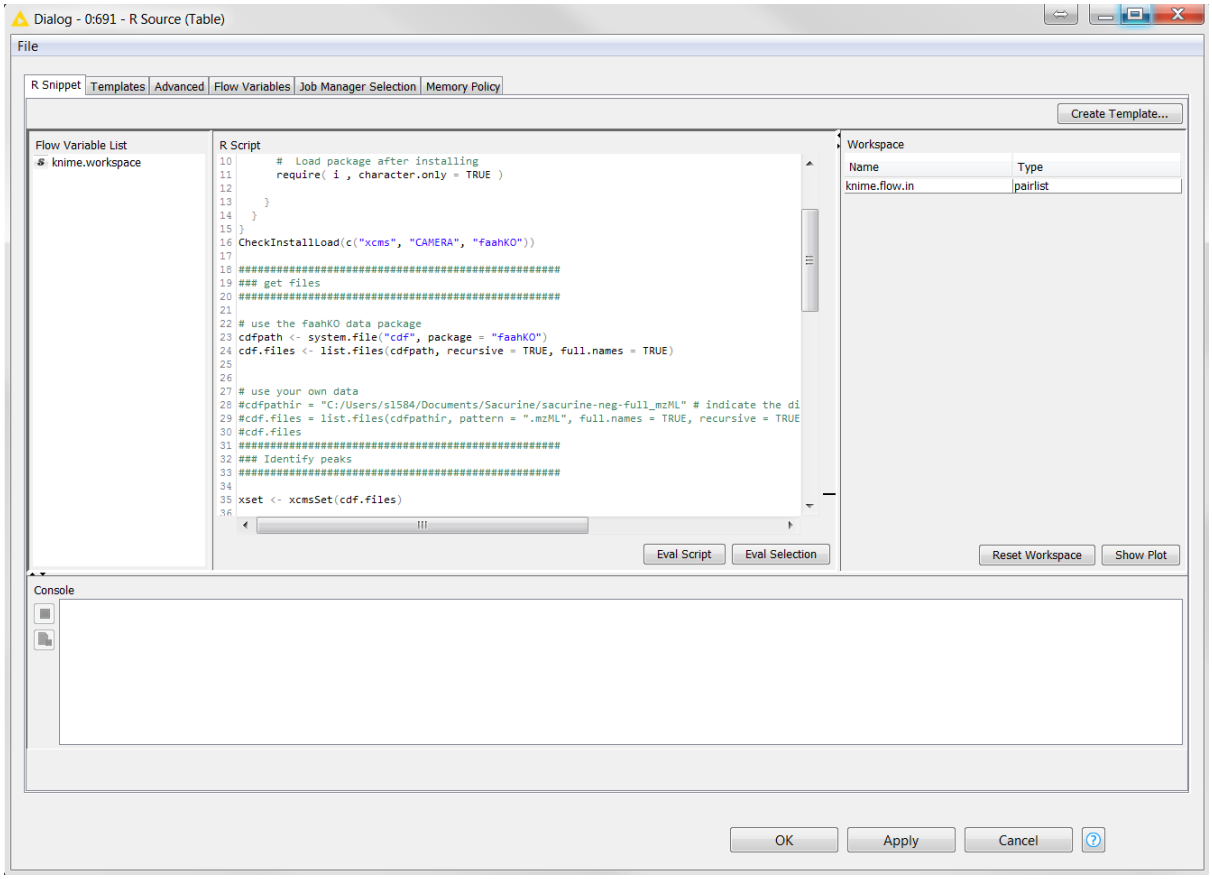

Figure 3: The R Source (Table) configuration pane

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 (*Figure 4*). 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.

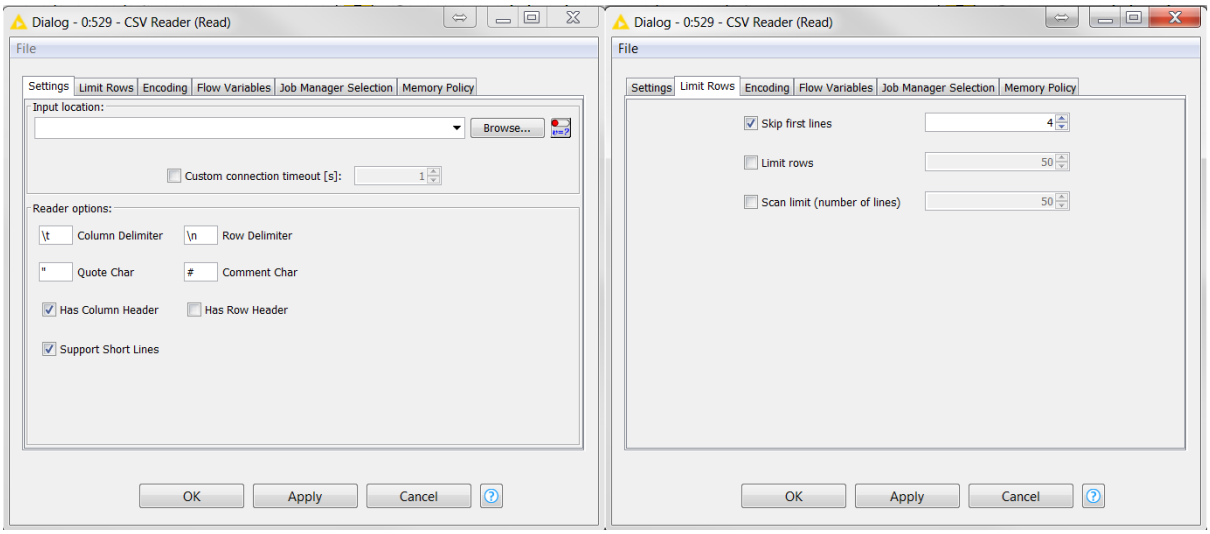

Figure 4: The Setting tab (left) and the Limit Rows tab (right) of the CSV Reader node are shown

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, i.e. rows corresponds features and columns to samples and their relative m/z, RT, etc. The column names of the output table need to be modified in order to have a standardised table to process, and this step can be done by connecting the output port of the last executed node to the input port of the following one described in point 2.

2. Uniform Column Names.

The column names of the deconvoluted data matrix need to be standardised in order to proceed smoothly with the following steps. This step is performed with the metanode (more on the definition of metanode later) **Uniform column names - remove spaces and special characters from sample names, add met_ID feature identifier** (*Figure 5*), which will modify sample names by replacing spaces with underscores and deleting any character different from letters, numbers and underscores. Moreover, it will uniform also the naming of the columns containing mass-to-charge, retention time and feature identifiers to `mz`, `rt` and `met_ID` respectively. Hence, the output will be a table where the only differences with its input will be in the name of the columns regarding mass-to-charge, retention time, feature identifiers and (possibly) samples.

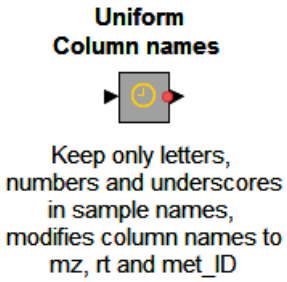

Figure 5: The Uniform Column Names metanode

As said before, this is not a node but a metanode, i.e. a node containing a sub-workflow inside. Although the sub-workflows can be visualised and modified by double-clicking on the metanodes, this as well as all the following metanodes in KniMet were designed so that do not need any kind of configurations, and can be run simply by right-clicking on them and clicking on `Execute`.

3. Keep only samples and ID.

At this point it is important to keep only the columns regarding met_ID and samples, and exclude the others which can be recovered in later stages. This step is performed with the **Column Filter - Keep only samples and ID** node (*Figure 6*).

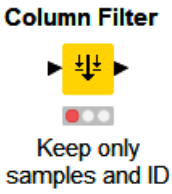

Figure 6: The column Filter node used to keep only column containing samples intensities and features identifiers

Like all the other column filter nodes that will follow, the 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 (*Figure 7*). 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`. Since only the columns containing the samples and `met_ID` should be kept, they should be moved to the right pane, whereas all the other columns containing other types of information 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. You do not need to worry and can just ignore them: they 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. The output of this node will be a table containing only the columns relative to the samples and `met_ID`.

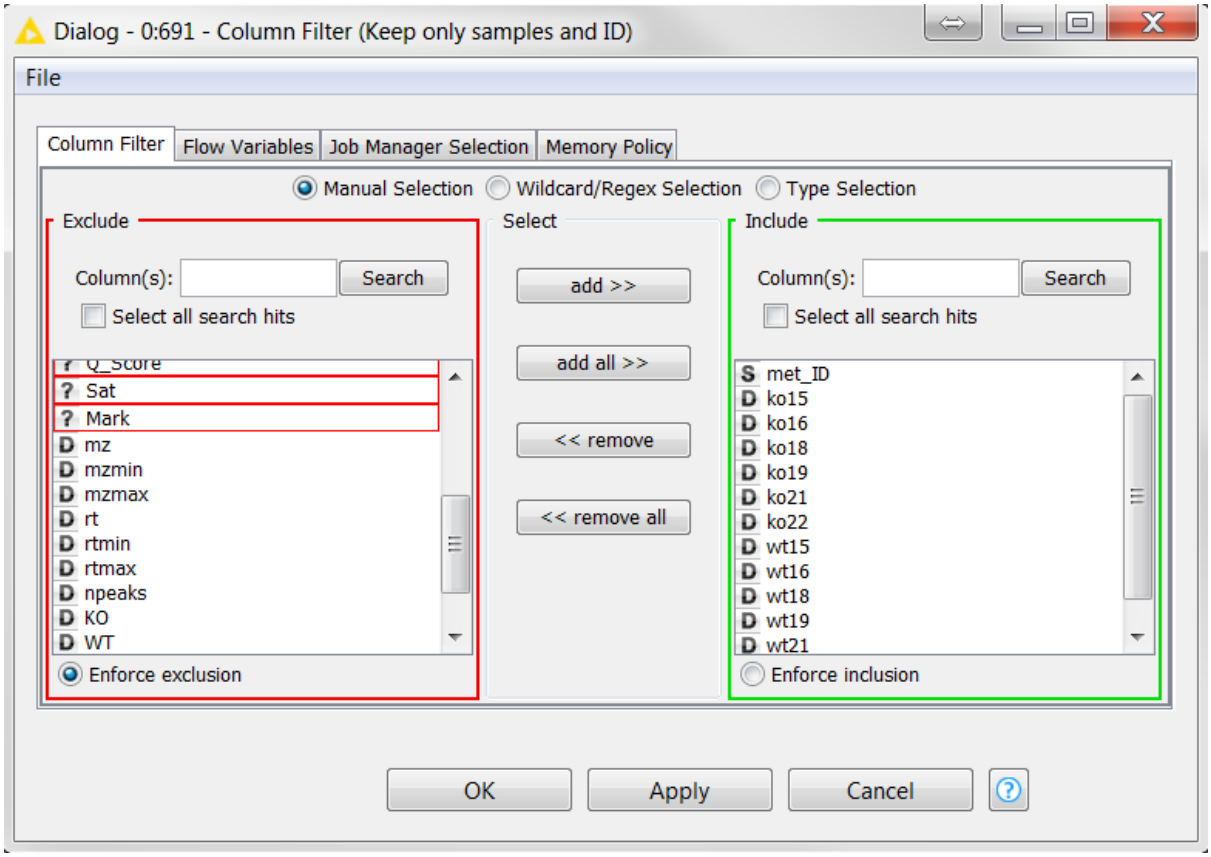

Figure 7: The Column filter node

4. Insert missing values symbol.

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 both the feature filtering based on QC and the missing value imputation steps (points 0 and 6 respectively), hence they need to be replaced with the missing value symbol. Two metanodes 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** (Figure 8).

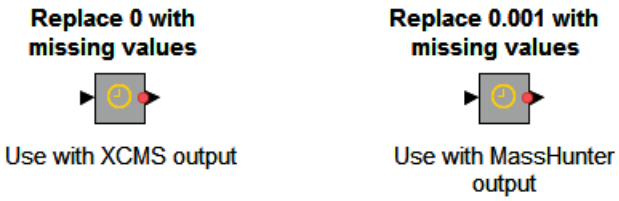

Figure 8: The metanodes to insert the correct missing value symbol in the data matrix obtained from MassHunter (right) and XCMS (left) are shown.

Once a table containing only the appropriate columns, i.e. samples and `met_ID`, the metanode can be run with no configuration needed. The output will be a table differing from the input only on the zeroes (or 0.001) which are replaced by the missing value symbol (*Figure 9*).

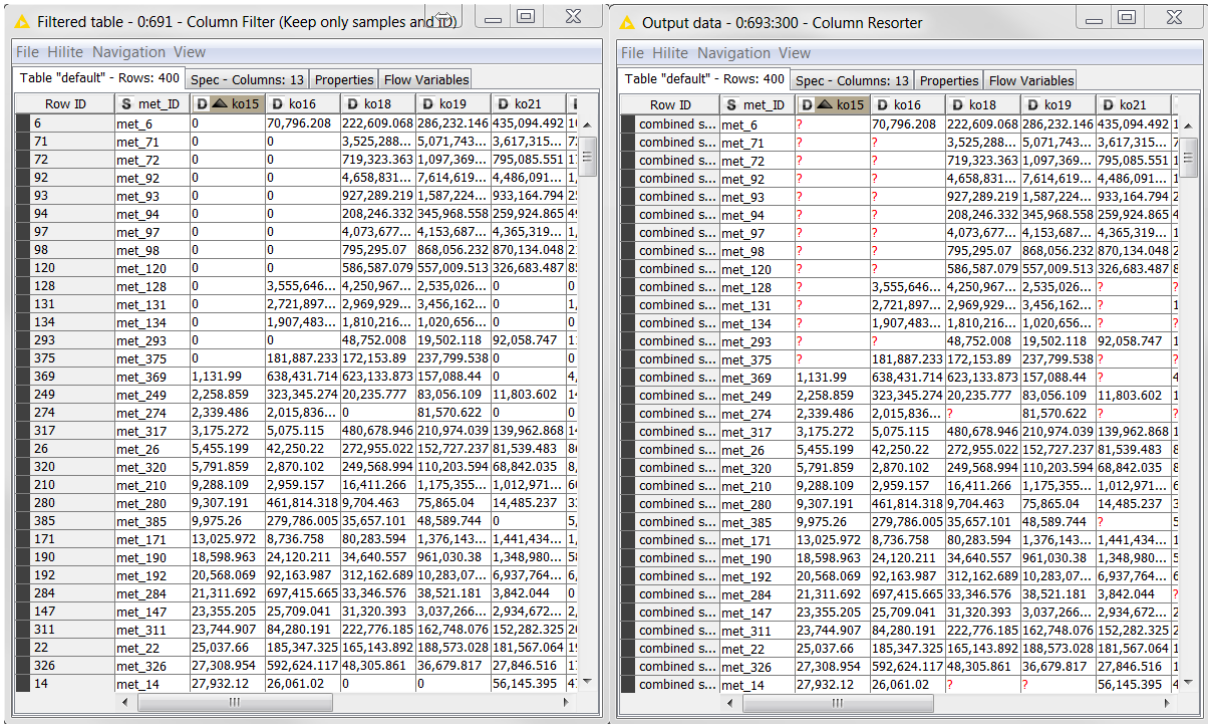

Figure 9: On the left is shown the data matrix obtained from the deconvolution of the faahKO data, containing several zeroes, while on the left table is shown the same data presenting the missing value symbol

5. Feature Filtering.

If pooled samples (from now on named QC) have been run along with the study samples, they can be used to assess the quality of the data by removing features with poor repeatability. To do this, first we need to select the columns containing the QC samples with the **Column Filter - Keep only QC samples** node (Figure 10). Configuration of the node consists in moving only the QC samples to the `Include` panel, while all the other columns should be listed under `Exclude` (please refer to point 3 for details on the configuration). Once configuration has finished, the node can be run and it will yield a table containing only the columns relative to the QCs. The output of this node can be fed to the **QC-based Features Filtering - Remove features present in less than 50% of the QCs; filter features whose RSD > 20%** metanode (Figure 10), which will delete all the features not consistently detectable across the QC runs.

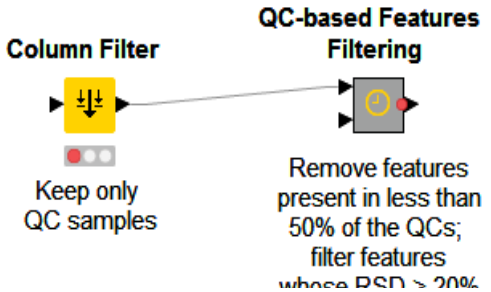

Figure 10: The column filter to select only the pooled samples (left) and the metanode to perform features filtering (right)

Make sure that the top input of this metanode is connected to the previous column filter, while the bottom input port should be connected to the output from the node inserting missing values (step 4). Once the right inputs are fed to the metanode, executing it will yield a table containing the columns `met_ID` as well as all the samples, but missing the rows corresponding to the features not repeatable across the run.

6. Missing Values Imputation.

Several methods have been implemented to impute missing values, namely Random Forest (**Missing Values Imputation RF**), Key-nearest neighbour (**Missing Values Imputation KNN**) or Small Value (**Missing Values Imputation SV**) (Figure 11).

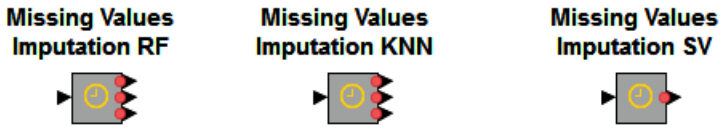

Figure 11: The metanodes to perform missing value imputation using Random Forest (left), Key-nearest neighbours (centre) and small value replacement (right).

These metanodes take as input the result of either **Replace 0/0.001 with missing values** or **QC-based Features Filtering**, and give as result the data matrix differing from the input only in the imputed missing values (*Figure 12*). In particular, the SV imputation metanode gives only this output, while KNN and RF present three output ports corresponding to the data matrix (top), R stderr (centre) and R stdout (bottom). Please note that on the first run of the KNN and RF metanodes you will be prompted to the installation of the R packages required by these nodes and not already installed in your local R version.

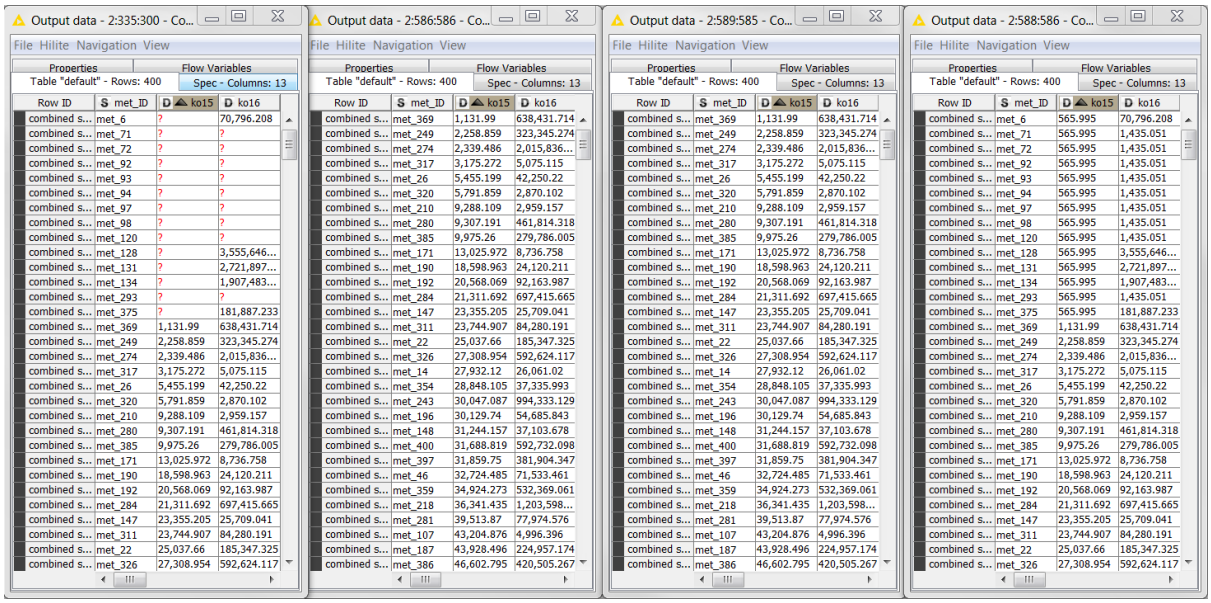

Figure 12: Comparison between the results of the Replace 0 with missing value symbol node (first table starting from left), missing value imputation RF (second), Missing value imputation KNN (third), Missing value imputation SV (last).

7. Normalisation.

Several normalisation methods are available, namely Probabilistic Quotients Normalisation (PQN), Loess Batch Corrections, both based on either all samples or the QCs.

7.1. The nodes **PQN-QC** and **PQN-ALL** (Figure 13) are designed to perform Probabilistic Quotient Normalisation either on all samples or only on the QCs (if available). In case the **PQN-QC** method is chosen, the **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 3). 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 a table with all samples and

`met_ID` as columns (such as the output of missing value imputation) should be connected to the bottom input port. In case **PQN-ALL** is chosen, no preliminary nodes need to be run, and its only input port should be connected with a table with all samples and `met_ID` as columns. The output of these metanodes will be a table containing the same rows and columns as the input, but normalised values in the cells.

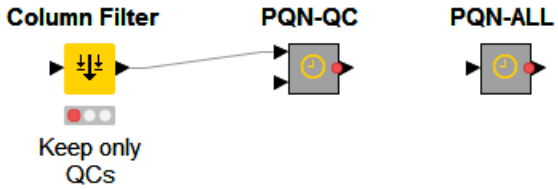

Figure 13: The metanodes to perform Probabilistic Quotient Normalisation based on either all samples (right) or QC samples (centre) are shown along with the Column Filter node (left) required for PQN-QC normalisation

7.2. In case of a batch-effect deriving from acquisition of the sample in multiple analytical blocks, a more specific normalisation method should be utilised. Both methods available, **QC-Loess Batch Correction** and **ALL-Loess Batch Correction**, need the preliminary import of sample metadata through the **CSV reader - read file containing sample metadata** node (*Figure 14*).

Table 1: Example of a sample metadata table.

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

Please note that a correct functioning of the Loess batch correction methods is possible only if the file contains exactly the column names showed in the example (*Table 1*), and the sample names are the same as those present in the data matrix fed to the CSV reader in point 1. If extra columns with other meta-information (for instance gender) are present, they will simply be ignored by this step without influencing the results. Once this node has been

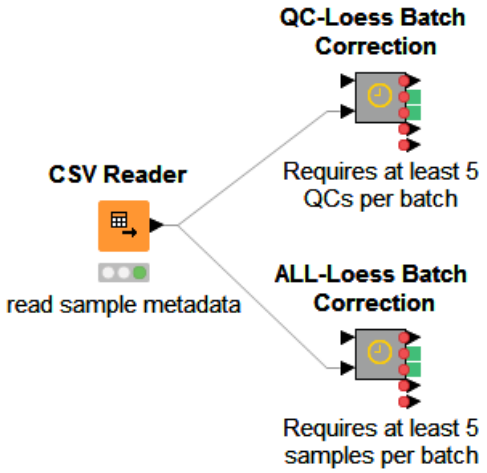

*Figure 14: The CSV Reader node imports the sample metadata, as described in **Error! Reference source not found.**, into the QC-Loess and the ALL-Loess Batch correction metanodes*

executed, its output should be connected to the bottom input port of the Batch Correction method of choice.

Both **Loess Batch Correction** methods are based on the R script and R wrapper implemented 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. If the **QC -Loess Batch Correction** metanode is selected, correction is performed based on the QC samples, which should be properly indicated in the metadata file imported with the previous **CVS Reader**. Both Batch Correction metanodes will perform Loess correction only if there are at least five QC/samples runs per batch, otherwise linear regression will be used to normalise the data. The Loess Batch Correction metanodes provide five outputs: the corrected data matrix (first), a plot depicting both the sum of intensities for each sample and the PCA score plots of components 1-4 before and after signal correction (second and third), R stdout (fourth) and R stderr (fifth). For more details on this specific step, the user is referred to the "How to" section in the Workflow4metabolomics website (http://workflow4metabolomics.org/howto).

None of the batch correction metanode needs to be configured, just make sure that the output from the **CSV reader - read file containing sample metadata** is connected to the bottom input port and the output from the **PQN-QC/PQN-ALL** node to the top input port.

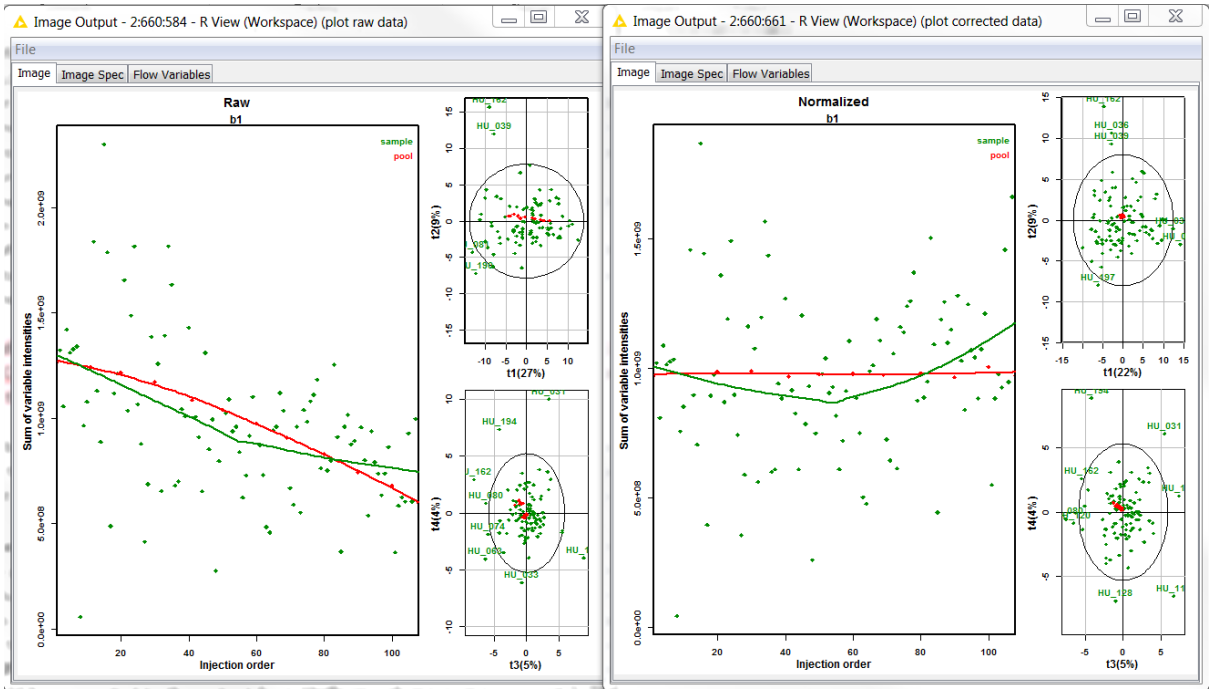

Figure 15: Plots showing the distribution of average intensities across the injection order and the PCA for the components 1-4 before (left) and after (right) QC-LOESS normalisation

8. Annotation.

Feature annotation based on accurate mass was implemented based on LIPID MAPS (Fahy et al., 2007) and Curatr (Palmer *et al.*, 2017) libraries (*Figure 16*). Please note that given the high number detected features in some experiments, and the elevated number of annotation per feature that might result, this step could be time consuming.

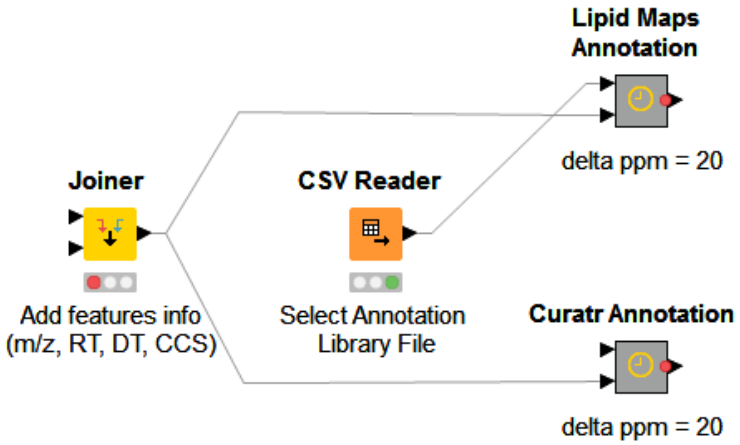

Figure 16: The set of nodes and metanodes required to perform feature annotation based on accurate mass: a joiner to add to the processed data information regarding the m/z of the features, a CSV Reader to import the library to use for annotation, and finally the two metanodes to perform feature annotation with either Lipid Maps and Curatr libraries.

The first step to perform before proceeding with annotation is making sure that the data matrix contains the `mz` column, which is missing if any of the steps from point 3 onwards was performed. The Joiner - Add features info (m/z, RT, DT, CCS) node can accomplish this task by connecting its top input port to the output of the previously run node (for instance **PQN-QC** if normalisation on the pooled samples was performed), while the bottom input table should be connected to the output from **Uniform column names** in point 2. 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 `met_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 (processed) and bottom (original) inputs, only the transformed data is kept (*Figure 17*).

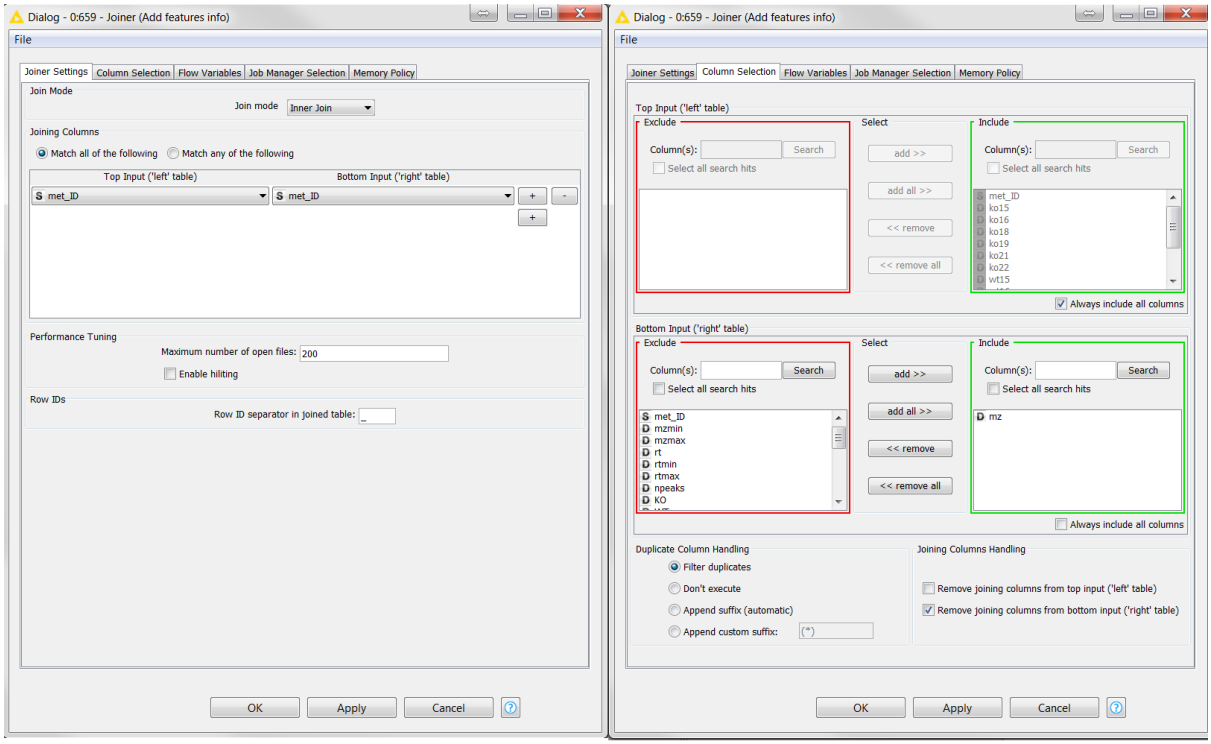

Figure 17: The Joiner setting tab (left) ad the Column Selection tab (right) of the Joiner node are shown

Once the joiner node has been properly configured and executed, we need to import the annotation library that we want to use with the **CSV Reader - Select Annotation Library File**. Embedded into the workflow group you will find part of the LIPID MAPS Library and the Curatr library (as shown in *Figure 1*). To select one of these library, right click on it 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 execute the node.

Now that the desired library has been imported into KNIME, we can proceed with the feature annotation by using either the **Lipid Maps Annotation** or the **Curatr - EMBL Metabolomics Core Facility Spectral Library Annotation**. Please note that both metanodes will perform an annotation of the features based on mass accuracy, with only annotations with Δppm < 20 retained. The output file will look identical to that fed to the annotation metanode, with the only addition of one column (either **Lipid Maps (Name, Class, Ion type, Deltappm)** or **Curatr (Name, Ion type, Deltappm)**).

9. Output.

The processed 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 **Excel Writer (XLS)** node can be selected from 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.

Guided Examples

These examples are based on the tab-separated files provided into the testfiles folder of this repository, which can be downloaded to your local machine.

1. Perform deconvolution with XCMS in KniMet.

In this example we will use the R data package faahKO (Saghatelian *et al.*, 2004), containing LC/MS data files from 12 wild-type and knockout mice collected in positive ionization mode, to perform deconvolution within the KniMet workflow by using the R source node and proceed with data processing. Please note that the dataset used does not contain injection of QC samples, hence neither feature filtering nor normalisation could be performed based on the QCs; data can be normalised only on all samples. The workflow, shown in *Figure 18*, comprises connection between nodes and metanodes to perform the following steps:

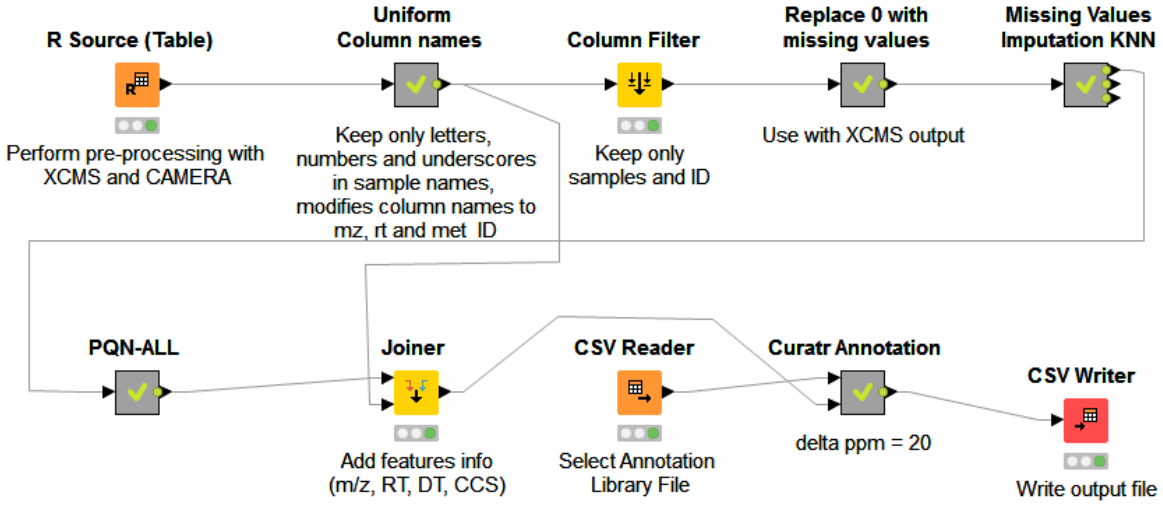

Figure 18: KniMet workflow used to process the data deconvoluted within the pipeline with the R Source node

1.1. Data Deconvolution, performed with the **R Source (Table)** node;
1.2. **Uniform column names** metanode to eliminate special characters from names;
1.3. **Column Filter** to remove all the columns but samples and features identifiers;
1.4. **Replace 0 with missing values** and **Missing Values imputation KNN** to insert the missing value symbol and replace its value;
1.5. **PQN-ALL** normalisation;
1.6. **Curatr Annotation** preceded by the **Joiner** node to add column containing m/z values;
1.7. **CSV Writer** to write to an external file the output of the processing.

2. Perform Processing of Drift tube Ion Mobility data.

The data acquired with the Agilent 6560 Ion Mobility Q-TOF LC-MS cannot be pre-processed with open access software, but the result from the deconvolution performed with the vendor's software can be imported into KniMet and subjected to post-processing. The file "Test_LCIMMS_features.tsv" in the testfiles folder of this repository was obtained performing feature extraction with MassHunter on mouse liver samples and QCs run on the Agilent 6560 Ion Mobility Q-TOF LC-MS in negative ion mode. This file can be downloaded and used to follow this example. In *Figure 19* the connection between nodes and metanodes to process this matrix are shown, namely:

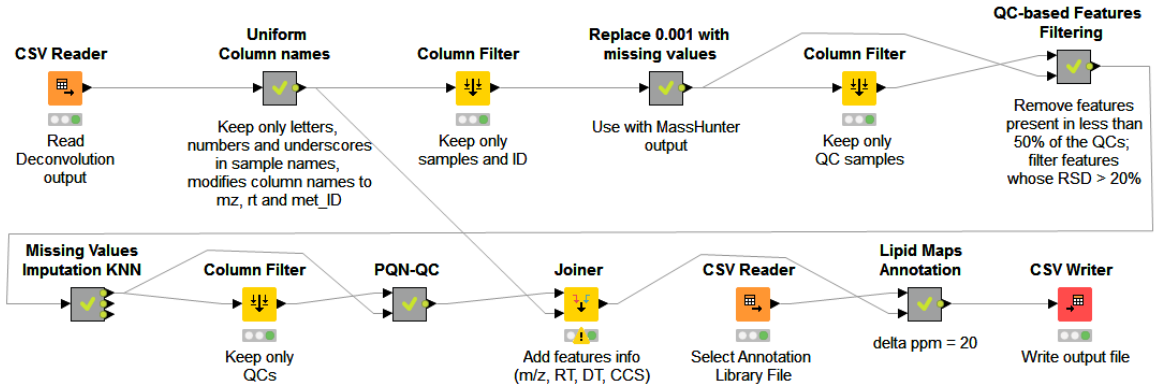

Figure 19: The complete workflow for the analysis of LC-IM-MS data containing pooled samples and pre-processed outside the KniMet workflow

2.1. Import of the deconvoluted data matrix using the **CVS Reader** node;
2.2. **Uniform column names** metanode to eliminate special characters from names;
2.3. **Column Filter** to remove all the columns but samples and features identifiers;
2.4. **Replace 0.001 with missing values** to insert the missing value symbol;
2.5. Selection of the QC samples and feature filtering based on them with the **Column Filter** node and the **QC-based feature filtering** metanode.
2.6. **Missing Values imputation KNN** to impute missing values;
2.7. Selection of the QC samples with the **Column Filter** node and normalisation **with the PQN-QC** metanode;
2.8. **Lipid Maps Annotation** preceded by the **Joiner** node to add column containing m/z values;
2.9. **CSV Writer** to write to an external file the output of the processing.

3. Perform Batch Correction

The testfiles folder contains the tab-separated file Test_Batchfeatures.tsv, which was obtained by processing human plasma acquired on a LC-MS instrument on positive ionisation mode in two different analytical blocks. In order to correct for batch effect information regarding samples' injection order and batch is needed, which can be found in the Test_Batch_samplemetadata.tsv file in the testfiles folder. Please note that this dataset does not contain QCs injection, hence the Loess method based on all samples will be used instead. The workflow to process this dataset is shown in *Figure 20* and contains:

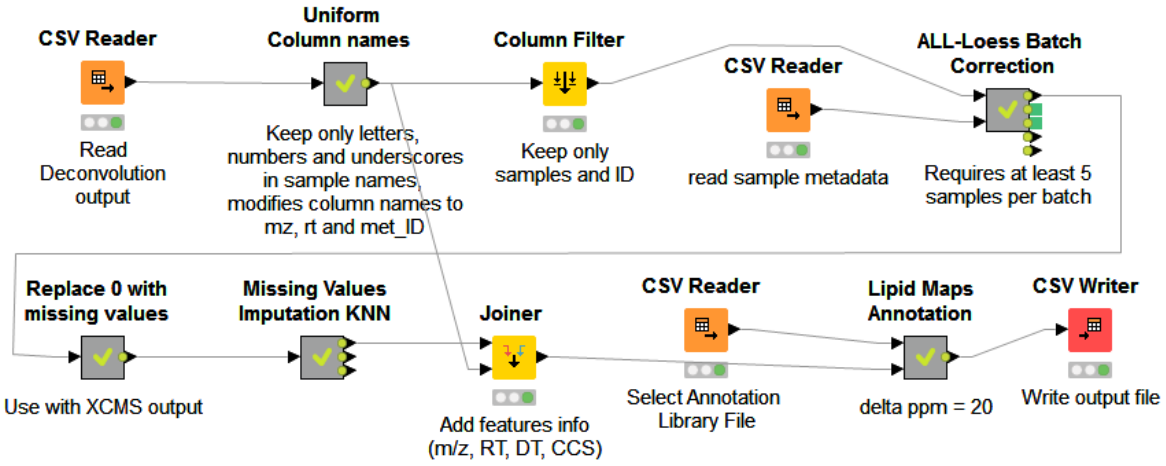

Figure 20: The KniMet workflow adapted to perform Loess batch correction based on all samples

3.1. Import of the deconvoluted data matrix using the **CVS Reader** node;

3.2. **Uniform column names** metanode to eliminate special characters from names;
3.3. **Column Filter** node to remove all the columns but samples and features identifiers;
3.4. **CVS Reader** node to import the sample metadata;
3.5. **ALL-Loess Batch Correction** metanode to perform batch correction on all samples;
3.6. **Replace 0 with missing values** to insert the missing value symbol followed by **Missing Values imputation KNN** to impute missing values;
3.7. **Lipid Maps Annotation** preceded by the **Joiner** node to add column containing m/z values;
3.8. **CSV Writer** to write to an external file the output of the processing.

The effective results of batch correction can be evaluated visualising the two plots outputted by the **All-Loess Batch Correction** metanode (*Figure 21*), which allow the evaluation of the correction by comparing variable intensities as well as the separation of samples in PCA space before and after normalisation.

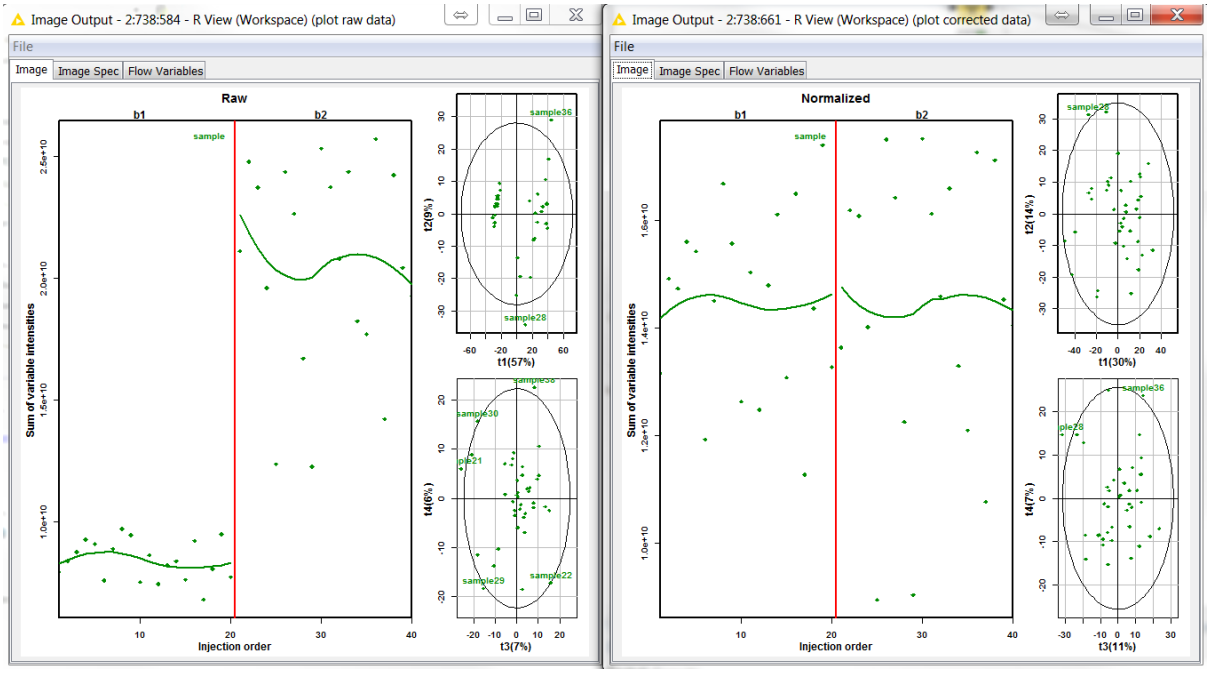

*Figure 21: The image outputs of the **All-Loess Batch Correction** metanode for raw (left) and normalised (right) data present on the left hand side the sum of intensity for all variables in each given sample as a function of the order of injection, along with the fitted loess curve. On the right hand side of each output image there are two PCA plots for the first and second components (top) and for the third and fourth (bottom).*

Bibliography

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.

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.

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.

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.