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

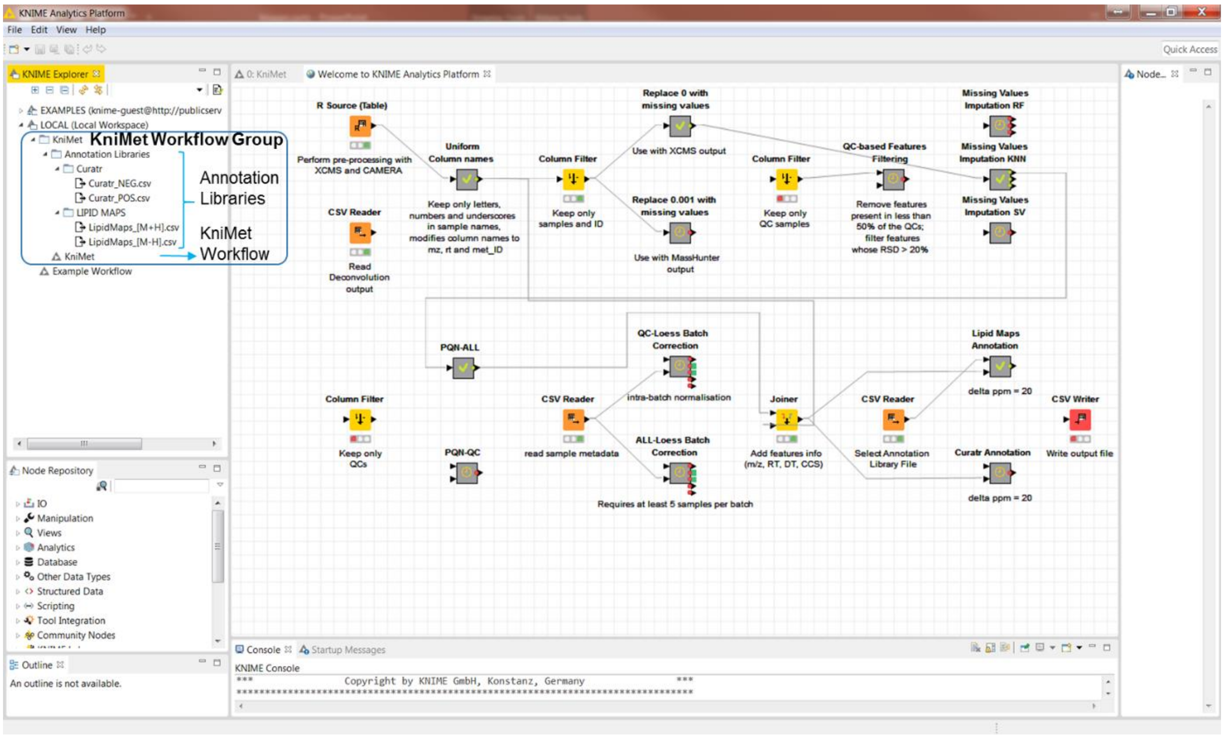
```
install.packages('Rserve')
```

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.



```
***          Copyright by KNIME GmbH, Konstanz, Germany          ***
```

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

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:

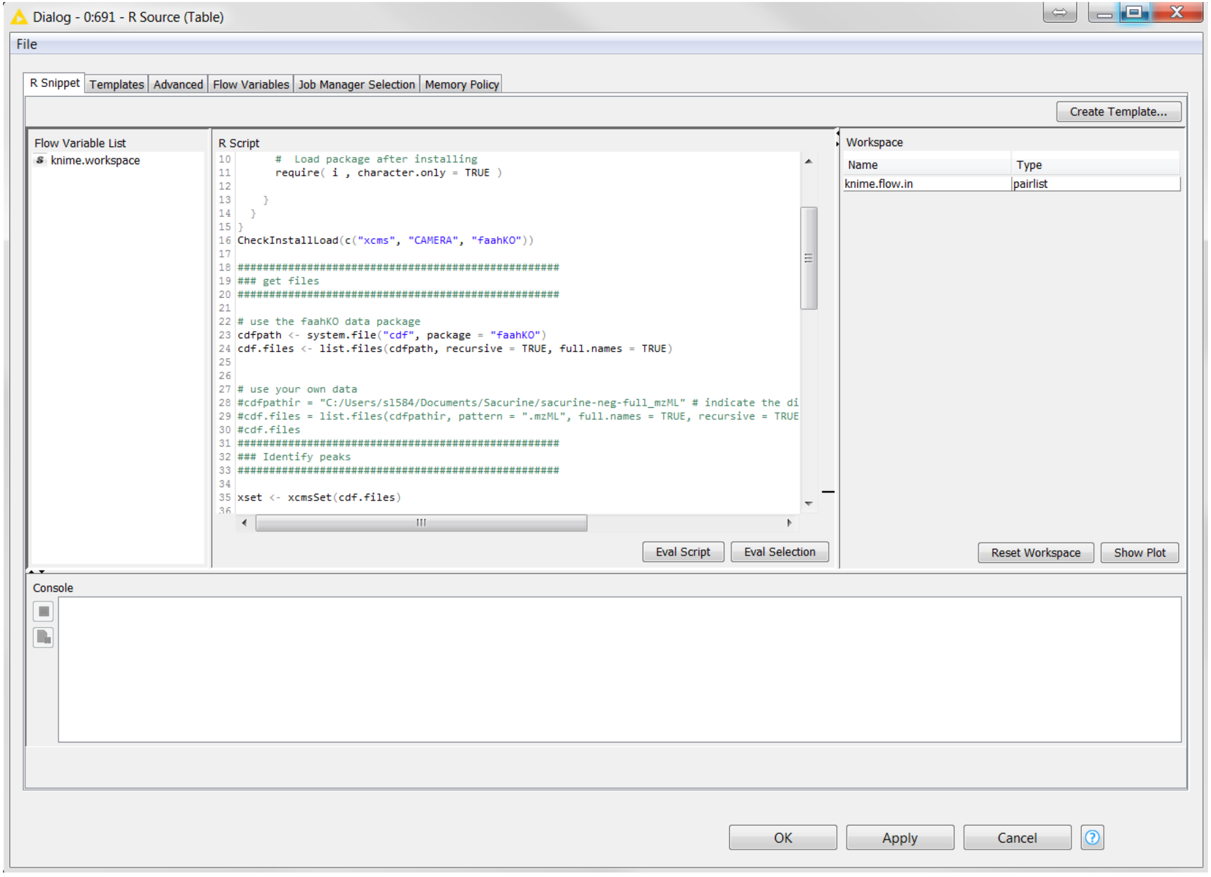
Click on `File→ Preferences` and then `KNIME → R` on the left hand side of the open window, and point to your local installation of R (usually `C:\Program Files\R\R-x.x.x` where x.x.x stands for the version number of your R installation) by clicking on the browse button on the right.

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

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.

   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 2*), 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.

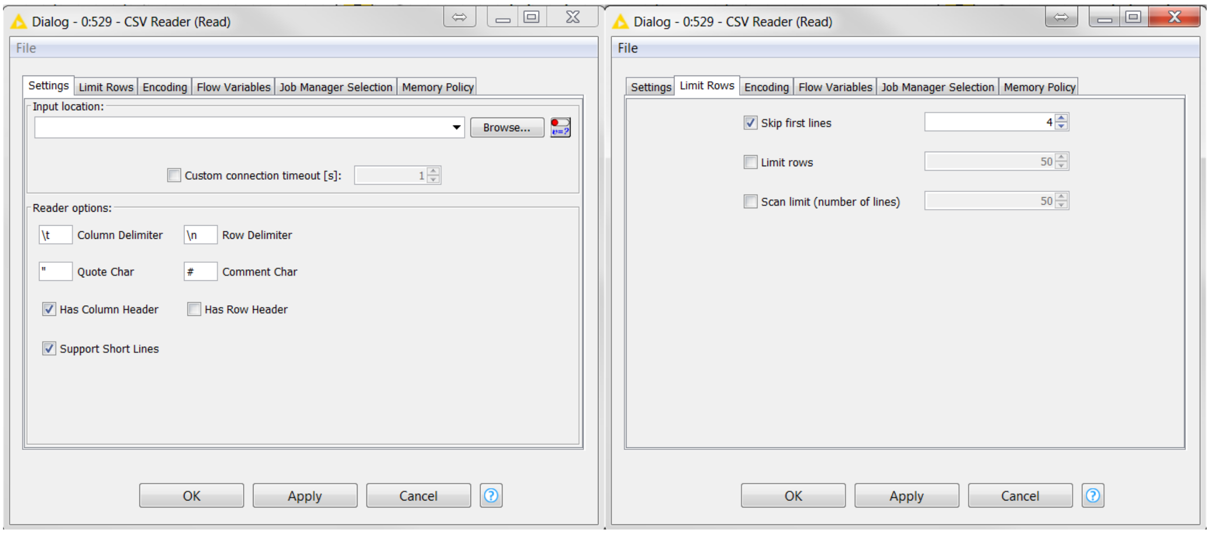


Name	Type
knime.flow.in	pairlist

```
      #  Load package after installing
      require( i , character.only = TRUE )

    }
  }
}
CheckInstallLoad(c("xcms", "CAMERA", "faahKO"))

###################################################
### get files
###################################################

# use the faahKO data package
cdfpath <- system.file("cdf", package = "faahKO")
cdf.files <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)


# use your own data
#cdfpathir = "C:/Users/s1584/Documents/Sacurine/sacurine-neg-full_mzML" # indicate the di
#cdf.files = list.files(cdfpathir, pattern = ".mzML", full.names = TRUE, recursive = TRUE
#cdf.files
###################################################
### Identify peaks
###################################################

xset <- xcmsSet(cdf.files)
```

Figure 2: 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 3*). 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.



Input location:

□ Custom connection timeout [s]: 1

Reader options:

\t Column Delimiter \n Row Delimiter

" Quote Char # Comment Char

☑ Has Column Header □ Has Row Header

☑ Support Short Lines

☑ Skip first lines 4

□ Limit rows 50

□ Scan limit (number of lines) 50

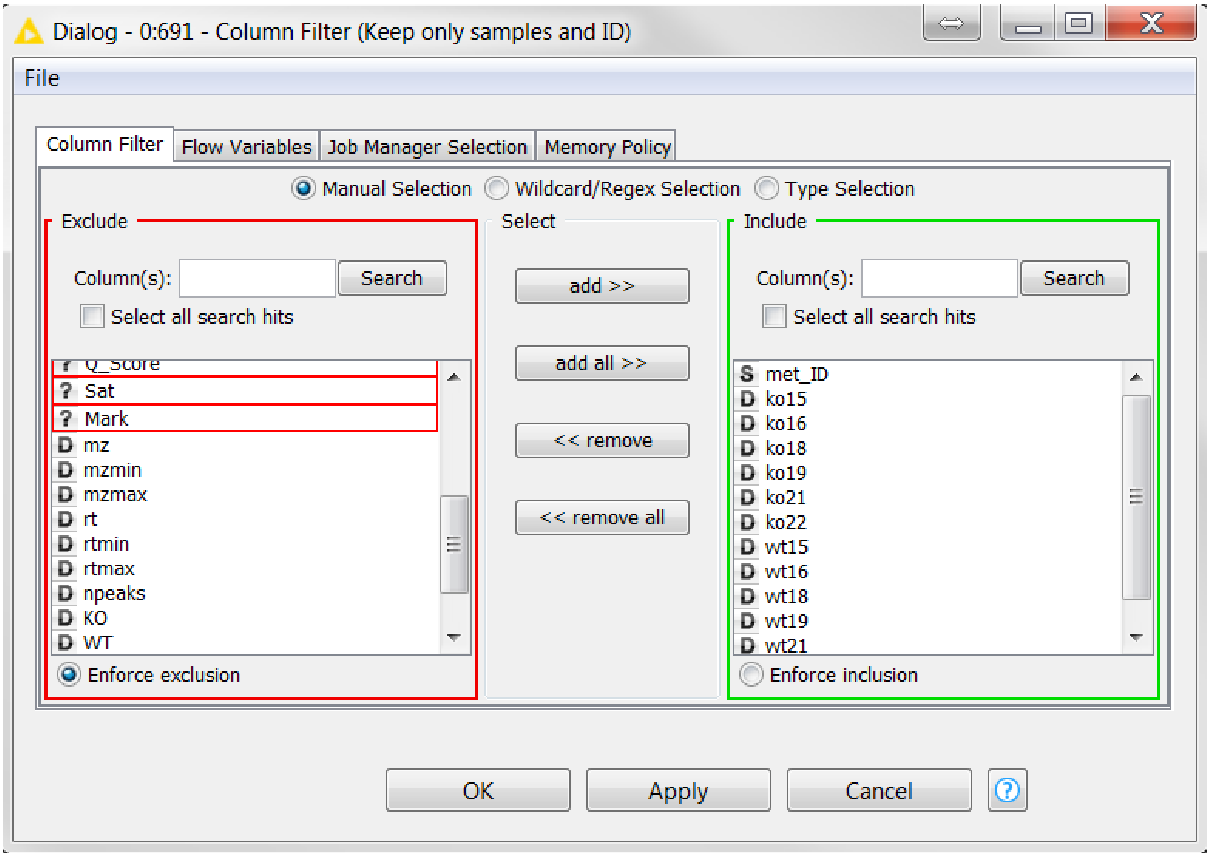
Figure 3: The Setting tab (left) and the Limit Rows tab (right) of the CSV Reader node are shown

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**, 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.

   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. 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 4*). 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`.



● Manual Selection ○ Wildcard/Regex Selection ○ Type Selection

Exclude

? Q_Score
? Sat
? Mark
D mz
D mzmin
D mzmax
D rt
D rtmin
D rtmax
D npeaks
D KO
D WT

● Enforce exclusion

Include

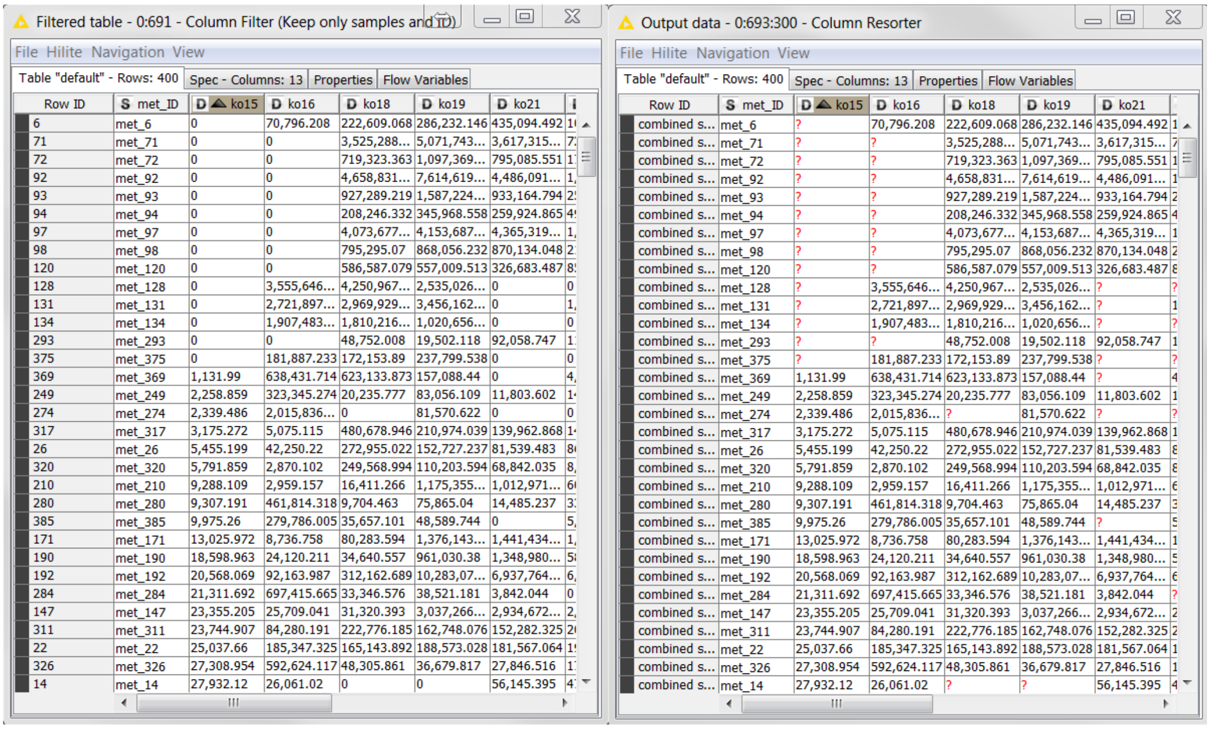
S met_ID
D ko15
D ko16
D ko18
D ko19
D ko21
D ko22
D wt15
D wt16
D wt18
D wt19
D wt21

○ Enforce inclusion

Figure 4: 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 5 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**. 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 5*) .



Filtered table - 0:691 - Column Filter (Keep only samples and ID)

Row ID	S met_ID	D ko15	D ko16	D ko18	D ko19	D ko21	
6	met_6	0	70,796.208	222,609.068	286,232.146	435,094.492	1
71	met_71	0	0	3,525,288...	5,071,743...	3,617,315...	7
72	met_72	0	0	719,323.363	1,097,369...	795,085.551	1
92	met_92	0	0	4,658,831...	7,614,619...	4,486,091...	1,
93	met_93	0	0	927,289.219	1,587,224...	933,164.794	2
94	met_94	0	0	208,246.332	345,968.558	259,924.865	4
97	met_97	0	0	4,073,677...	4,153,687...	4,365,319...	1,
98	met_98	0	0	795,295.07	868,056.232	870,134.048	2
120	met_120	0	0	586,587.079	557,009.513	326,683.487	8
128	met_128	0	3,555,646...	4,250,967...	2,535,026...	0	0
131	met_131	0	2,721,897...	2,969,929...	3,456,162...	0	1,
134	met_134	0	1,907,483...	1,810,216...	1,020,656...	0	0
293	met_293	0	0	48,752.008	19,502.118	92,058.747	1
375	met_375	0	181,887.233	172,153.89	237,799.538	0	0
369	met_369	1,131.99	638,431.714	623,133.873	157,088.44	0	4,
249	met_249	2,258.859	323,345.274	20,235.777	83,056.109	11,803.602	1
274	met_274	2,339.486	2,015,836...	0	81,570.622	0	0
317	met_317	3,175.272	5,075.115	480,678.946	210,974.039	139,962.868	1
26	met_26	5,455.199	42,250.22	272,955.022	152,727.237	81,539.483	8
320	met_320	5,791.859	2,870.102	249,568.994	110,203.594	68,842.035	8,
210	met_210	9,288.109	2,959.157	16,411.266	1,175,355...	1,012,971...	6
280	met_280	9,307.191	461,814.318	9,704.463	75,865.04	14,485.237	3
385	met_385	9,975.26	279,786.005	35,657.101	48,589.744	0	5,
171	met_171	13,025.972	8,736.758	80,283.594	1,376,143...	1,441,434...	1,
190	met_190	18,598.963	24,120.211	34,640.557	961,030.38	1,348,980...	5
192	met_192	20,568.069	92,163.987	312,162.689	10,283,07...	6,937,764...	6,
284	met_284	21,311.692	697,415.665	33,346.576	38,521.181	3,842.044	0
147	met_147	23,355.205	25,709.041	31,320.393	3,037,266...	2,934,672...	2,
311	met_311	23,744.907	84,280.191	222,776.185	162,748.076	152,282.325	2
22	met_22	25,037.66	185,347.325	165,143.892	188,573.028	181,567.064	1
326	met_326	27,308.954	592,624.117	48,305.861	36,679.817	27,846.516	1
14	met_14	27,932.12	26,061.02	0	0	56,145.395	4

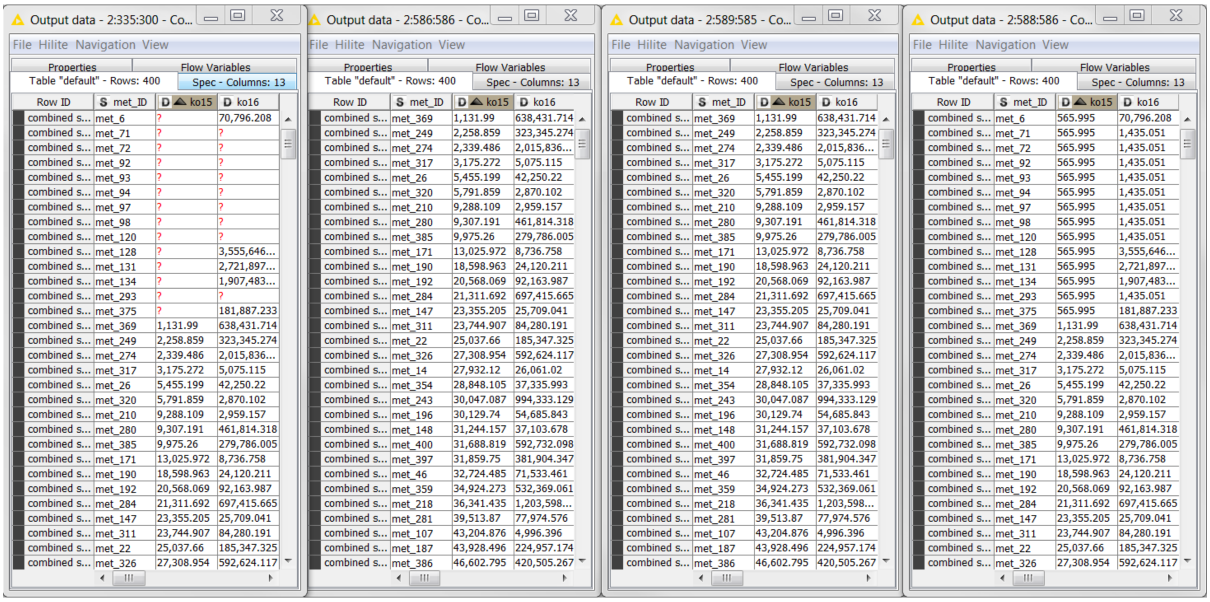
Output data - 0:693:300 - Column Resorter

Row ID	S met_ID	D ko15	D ko16	D ko18	D ko19	D ko21	
combined s...	met_6	?	70,796.208	222,609.068	286,232.146	435,094.492	1
combined s...	met_71	?	?	3,525,288...	5,071,743...	3,617,315...	7
combined s...	met_72	?	?	719,323.363	1,097,369...	795,085.551	1
combined s...	met_92	?	?	4,658,831...	7,614,619...	4,486,091...	1
combined s...	met_93	?	?	927,289.219	1,587,224...	933,164.794	2
combined s...	met_94	?	?	208,246.332	345,968.558	259,924.865	4
combined s...	met_97	?	?	4,073,677...	4,153,687...	4,365,319...	1
combined s...	met_98	?	?	795,295.07	868,056.232	870,134.048	2
combined s...	met_120	?	?	586,587.079	557,009.513	326,683.487	8
combined s...	met_128	?	3,555,646...	4,250,967...	2,535,026...	?	?
combined s...	met_131	?	2,721,897...	2,969,929...	3,456,162...	?	1
combined s...	met_134	?	1,907,483...	1,810,216...	1,020,656...	?	?
combined s...	met_293	?	?	48,752.008	19,502.118	92,058.747	1
combined s...	met_375	?	181,887.233	172,153.89	237,799.538	?	?
combined s...	met_369	1,131.99	638,431.714	623,133.873	157,088.44	?	4
combined s...	met_249	2,258.859	323,345.274	20,235.777	83,056.109	11,803.602	1
combined s...	met_274	2,339.486	2,015,836...	?	81,570.622	?	?
combined s...	met_317	3,175.272	5,075.115	480,678.946	210,974.039	139,962.868	1
combined s...	met_26	5,455.199	42,250.22	272,955.022	152,727.237	81,539.483	8
combined s...	met_320	5,791.859	2,870.102	249,568.994	110,203.594	68,842.035	8
combined s...	met_210	9,288.109	2,959.157	16,411.266	1,175,355...	1,012,971...	6
combined s...	met_280	9,307.191	461,814.318	9,704.463	75,865.04	14,485.237	3
combined s...	met_385	9,975.26	279,786.005	35,657.101	48,589.744	?	5
combined s...	met_171	13,025.972	8,736.758	80,283.594	1,376,143...	1,441,434...	1
combined s...	met_190	18,598.963	24,120.211	34,640.557	961,030.38	1,348,980...	5
combined s...	met_192	20,568.069	92,163.987	312,162.689	10,283,07...	6,937,764...	6
combined s...	met_284	21,311.692	697,415.665	33,346.576	38,521.181	3,842.044	?
combined s...	met_147	23,355.205	25,709.041	31,320.393	3,037,266...	2,934,672...	2
combined s...	met_311	23,744.907	84,280.191	222,776.185	162,748.076	152,282.325	2
combined s...	met_22	25,037.66	185,347.325	165,143.892	188,573.028	181,567.064	1
combined s...	met_326	27,308.954	592,624.117	48,305.861	36,679.817	27,846.516	1
combined s...	met_14	27,932.12	26,061.02	?	?	56,145.395	4

Figure 5: On the left is shown the data matrix obtained from the deconvolution of the faacko data, containing several zeroes, while on the left table is shown the same data presenting the missing value symbol

5. Feature Filtering. The quality of the data can be assessed based on the pooled samples (from now on named QC) by removing features with poor repeatability to avoid a bias in the following results. To do this, first we need to select the columns containing the QC samples with the **Column Filter - Keep only QC samples** node. 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, which will delete all the features not consistently detectable across the QC runs. 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**). 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 6*). 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.



Output data - 2:335:300 - Co...

Table "default" - Rows: 400 | Spec - Columns: 13

Row ID	S met_ID	D ko15	D ko16
combined s...	met_6	?	70,796.208
combined s...	met_71	?	?
combined s...	met_72	?	?
combined s...	met_92	?	?
combined s...	met_93	?	?
combined s...	met_94	?	?
combined s...	met_97	?	?
combined s...	met_98	?	?
combined s...	met_120	?	?
combined s...	met_128	?	3,555,646...
combined s...	met_131	?	2,721,897...
combined s...	met_134	?	1,907,483...
combined s...	met_293	?	?
combined s...	met_375	?	181,887.233
combined s...	met_369	1,131.99	638,431.714
combined s...	met_249	2,258.859	323,345.274
combined s...	met_274	2,339.486	2,015,836...
combined s...	met_317	3,175.272	5,075.115
combined s...	met_26	5,455.199	42,250.22
combined s...	met_320	5,791.859	2,870.102
combined s...	met_210	9,288.109	2,959.157
combined s...	met_280	9,307.191	461,814.318
combined s...	met_385	9,975.26	279,786.005
combined s...	met_171	13,025.972	8,736.758
combined s...	met_190	18,598.963	24,120.211
combined s...	met_192	20,568.069	92,163.987
combined s...	met_284	21,311.692	697,415.665
combined s...	met_147	23,355.205	25,709.041
combined s...	met_311	23,744.907	84,280.191
combined s...	met_22	25,037.66	185,347.325
combined s...	met_326	27,308.954	592,624.117

Output data - 2:586:586 - Co...

Table "default" - Rows: 400 | Spec - Columns: 13

Row ID	S met_ID	D ko15	D ko16
combined s...	met_369	1,131.99	638,431.714
combined s...	met_249	2,258.859	323,345.274
combined s...	met_274	2,339.486	2,015,836...
combined s...	met_317	3,175.272	5,075.115
combined s...	met_26	5,455.199	42,250.22
combined s...	met_320	5,791.859	2,870.102
combined s...	met_210	9,288.109	2,959.157
combined s...	met_280	9,307.191	461,814.318
combined s...	met_385	9,975.26	279,786.005
combined s...	met_171	13,025.972	8,736.758
combined s...	met_190	18,598.963	24,120.211
combined s...	met_192	20,568.069	92,163.987
combined s...	met_284	21,311.692	697,415.665
combined s...	met_147	23,355.205	25,709.041
combined s...	met_311	23,744.907	84,280.191
combined s...	met_22	25,037.66	185,347.325
combined s...	met_326	27,308.954	592,624.117
combined s...	met_14	27,932.12	26,061.02
combined s...	met_354	28,848.105	37,335.993
combined s...	met_243	30,047.087	994,333.129
combined s...	met_196	30,129.74	54,685.843
combined s...	met_148	31,244.157	37,103.678
combined s...	met_400	31,688.819	592,732.098
combined s...	met_397	31,859.75	381,904.347
combined s...	met_46	32,724.485	71,533.461
combined s...	met_359	34,924.273	532,369.061
combined s...	met_218	36,341.435	1,203,598...
combined s...	met_281	39,513.87	77,974.576
combined s...	met_107	43,204.876	4,996.396
combined s...	met_187	43,928.496	224,957.174
combined s...	met_386	46,602.795	420,505.267

Output data - 2:589:585 - Co...

Table "default" - Rows: 400 | Spec - Columns: 13

Row ID	S met_ID	D ko15	D ko16
combined s...	met_369	1,131.99	638,431.714
combined s...	met_249	2,258.859	323,345.274
combined s...	met_274	2,339.486	2,015,836...
combined s...	met_317	3,175.272	5,075.115
combined s...	met_26	5,455.199	42,250.22
combined s...	met_320	5,791.859	2,870.102
combined s...	met_210	9,288.109	2,959.157
combined s...	met_280	9,307.191	461,814.318
combined s...	met_385	9,975.26	279,786.005
combined s...	met_171	13,025.972	8,736.758
combined s...	met_190	18,598.963	24,120.211
combined s...	met_192	20,568.069	92,163.987
combined s...	met_284	21,311.692	697,415.665
combined s...	met_147	23,355.205	25,709.041
combined s...	met_311	23,744.907	84,280.191
combined s...	met_22	25,037.66	185,347.325
combined s...	met_326	27,308.954	592,624.117
combined s...	met_14	27,932.12	26,061.02
combined s...	met_354	28,848.105	37,335.993
combined s...	met_243	30,047.087	994,333.129
combined s...	met_196	30,129.74	54,685.843
combined s...	met_148	31,244.157	37,103.678
combined s...	met_400	31,688.819	592,732.098
combined s...	met_397	31,859.75	381,904.347
combined s...	met_46	32,724.485	71,533.461
combined s...	met_359	34,924.273	532,369.061
combined s...	met_218	36,341.435	1,203,598...
combined s...	met_281	39,513.87	77,974.576
combined s...	met_107	43,204.876	4,996.396
combined s...	met_187	43,928.496	224,957.174
combined s...	met_386	46,602.795	420,505.267

Output data - 2:588:586 - Co...

Table "default" - Rows: 400 | Spec - Columns: 13

Row ID	S met_ID	D ko15	D ko16
combined s...	met_6	565.995	70,796.208
combined s...	met_71	565.995	1,435.051
combined s...	met_72	565.995	1,435.051
combined s...	met_92	565.995	1,435.051
combined s...	met_93	565.995	1,435.051
combined s...	met_94	565.995	1,435.051
combined s...	met_97	565.995	1,435.051
combined s...	met_98	565.995	1,435.051
combined s...	met_120	565.995	1,435.051
combined s...	met_128	565.995	3,555,646...
combined s...	met_131	565.995	2,721,897...
combined s...	met_134	565.995	1,907,483...
combined s...	met_293	565.995	1,435.051
combined s...	met_375	565.995	181,887.233
combined s...	met_369	1,131.99	638,431.714
combined s...	met_249	2,258.859	323,345.274
combined s...	met_274	2,339.486	2,015,836...
combined s...	met_317	3,175.272	5,075.115
combined s...	met_26	5,455.199	42,250.22
combined s...	met_320	5,791.859	2,870.102
combined s...	met_210	9,288.109	2,959.157
combined s...	met_280	9,307.191	461,814.318
combined s...	met_385	9,975.26	279,786.005
combined s...	met_171	13,025.972	8,736.758
combined s...	met_190	18,598.963	24,120.211
combined s...	met_192	20,568.069	92,163.987
combined s...	met_284	21,311.692	697,415.665
combined s...	met_147	23,355.205	25,709.041
combined s...	met_311	23,744.907	84,280.191
combined s...	met_22	25,037.66	185,347.325
combined s...	met_326	27,308.954	592,624.117

Figure 6: 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** 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.

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. 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 batch correction methods 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. If extra columns with other metainformation (for instance gender) are present, they will simply be ignored by this step without influencing the results. Once this node has been 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 in point 7.2, only if there are at least 5 QC runs per batch, otherwise linear regression will be used. The Loess Batch Correction metanodes provide 5 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).

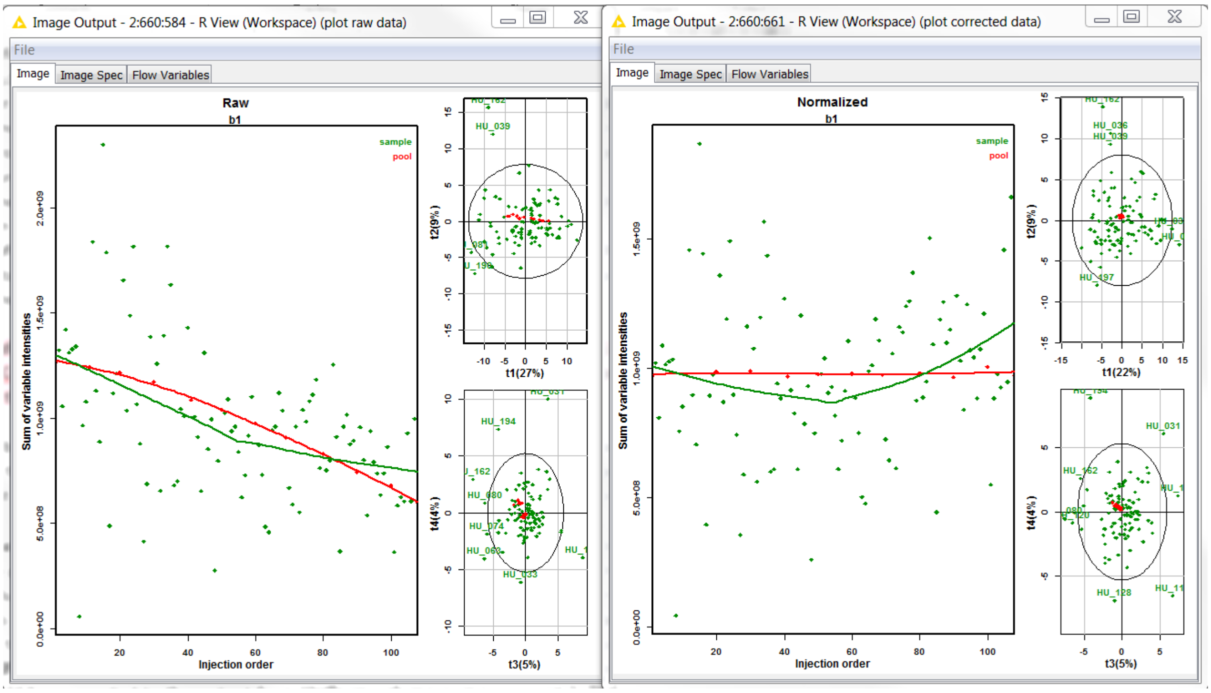
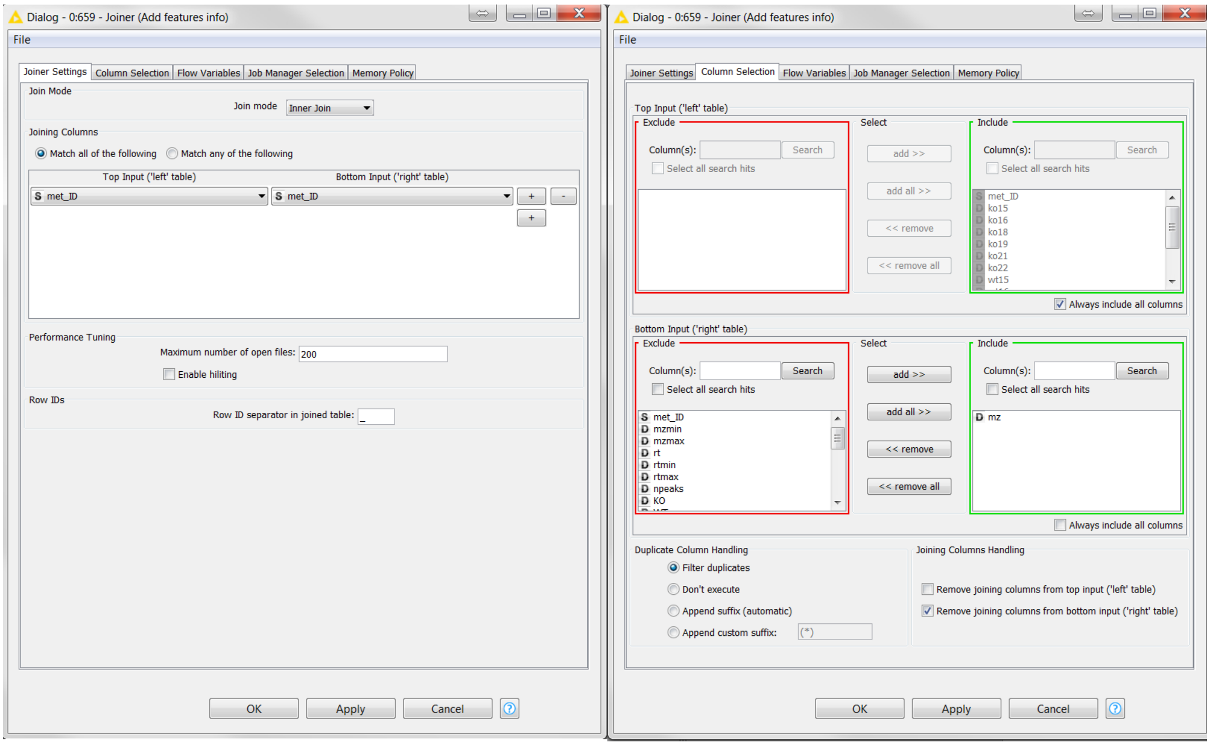


Figure 7: Plots showing the distribution of average intensities across the injection order and the PCA for the components 1-4 before (left) and after (right) QC-LOESS normalisation

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.

8. Annotation. Feature annotation was implemented based on 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 8*).



Dialog - 0:659 - Joiner (Add features info)

Joiner Settings

Join Mode
Join mode: Inner Join

Joining Columns
● Match all of the following ○ Match any of the following

Top Input ('left' table)	Bottom Input ('right' table)
S met_ID	S met_ID

Performance Tuning
Maximum number of open files: 200
□ Enable hiliting

Row IDs
Row ID separator in joined table: _

Dialog - 0:659 - Joiner (Add features info)

Column Selection

Top Input ('left' table)
Exclude: (empty)
Include: S met_ID, D ko15, D ko16, D ko18, D ko19, D ko21, D ko22, D wt15
☑ Always include all columns

Bottom Input ('right' table)
Exclude: S met_ID, D mzmin, D mzmax, D rt, D rtmin, D rtmax, D npeaks, D KO
Include: D mz
□ Always include all columns

Duplicate Column Handling
● Filter duplicates
○ Don't execute
○ Append suffix (automatic)
○ Append custom suffix: (*)

Joining Columns Handling
□ Remove joining columns from top input ('left' table)
☑ Remove joining columns from bottom input ('right' table)

Figure 8: 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. 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 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)). Please note that this step can be relatively long depending on the number of features to annotate.

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.

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.