Supporting information S6: Installation, usage and description of the KNIME workflows used for AMS filtering by retention time

Installation:

Our AMS filtering pipeline was developed, tested and evaluated using KNIME 2.11.1 (64bit) with OpenMS 2.0 nodes (Version 2.0.0.201502031718). For simplicity, we will describe the installation using the KNIME full installer of this version, which contains all other nodes used in our workflows by default.

- Download the installer for the 'KNIME Analytics Platform version 2.11.1 + all free extensions' from http://www.knime.org/downloads/previous

- Follow the KNIME installation instructions

- On KNIME startup, select 'Get additional nodes' (see Fig. 1)

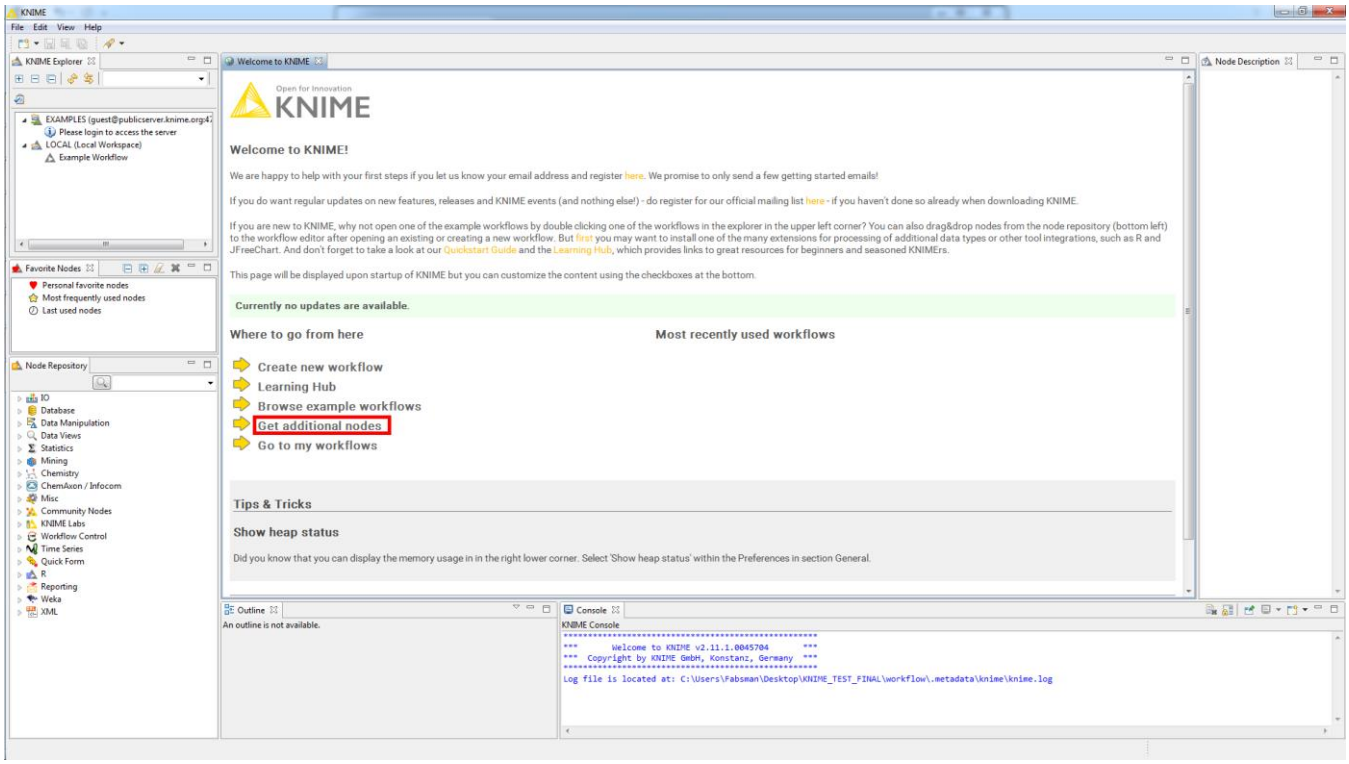

Figure 1: KNIME environment upon startup.

- From the supplied bioinformatics nodes, check OpenMS (see Fig. 2) and follow the dialog through the installation

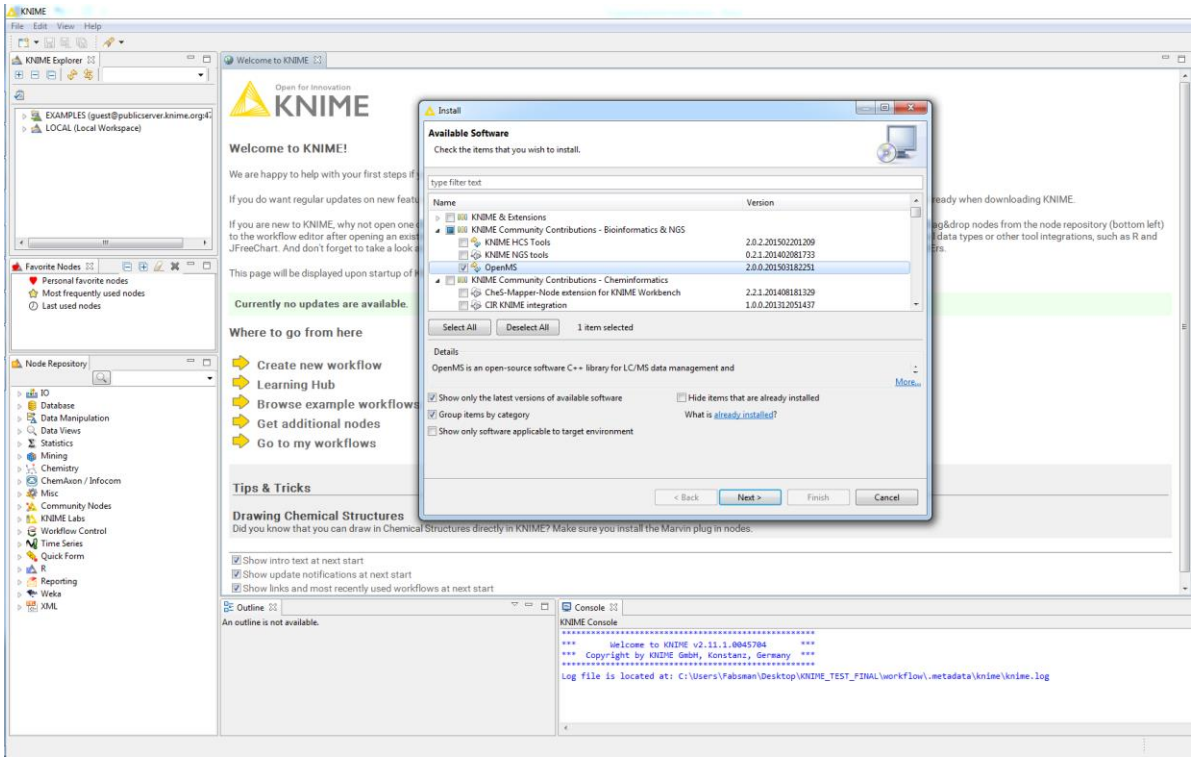

Figure 2: Node installation dialog.

- As suggested by KNIME, restart KNIME after installation of the OpenMS node

- After restarting, right click on your workspace and import the AMS filtering workflows (see Fig. 3, 4)

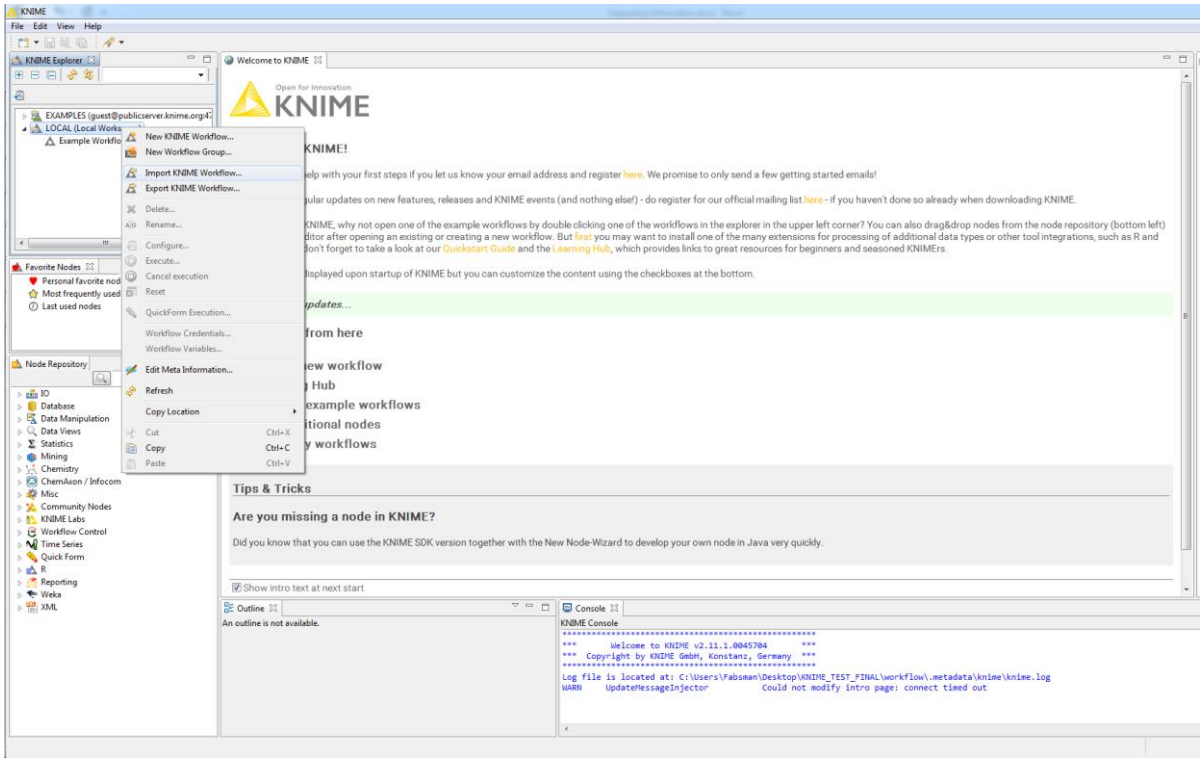

Figure 3: Import of the AMS pipeline supplied in the supporting materials.

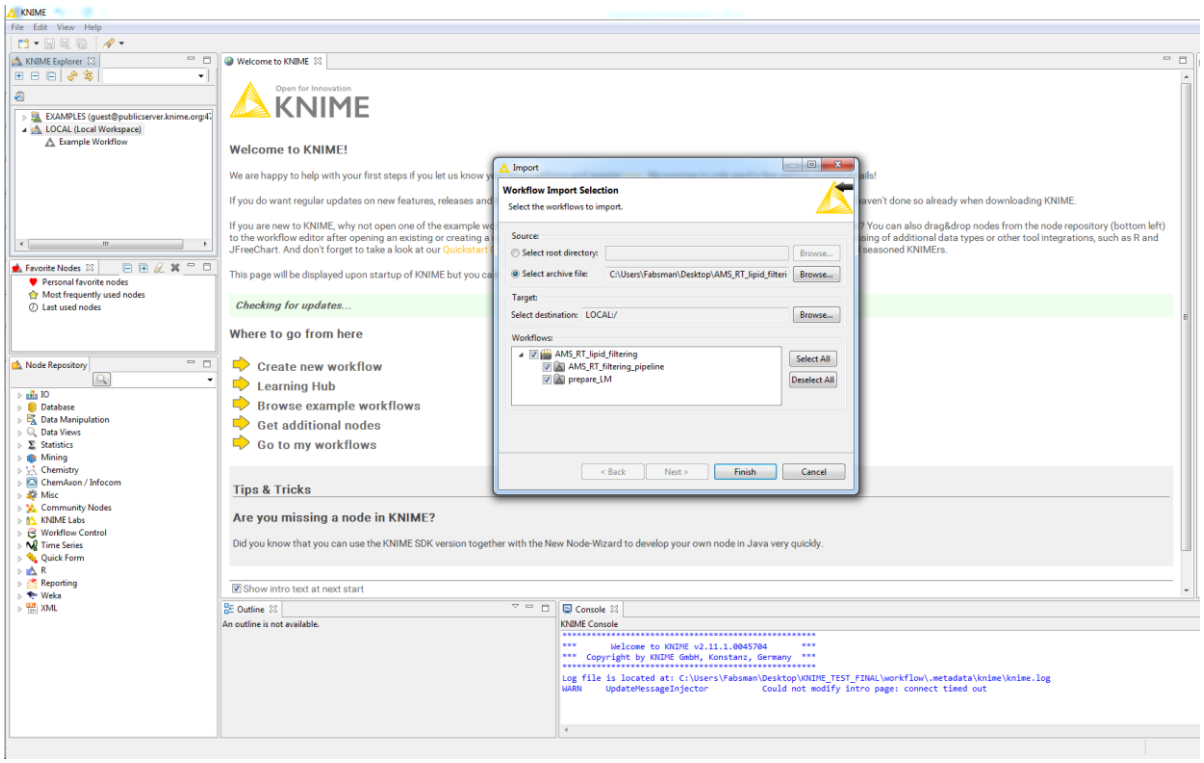

Figure 4: Import of the supplied filtering workflows, second step.

The imported .zip archive contains a workflow for feature computation and preparation of the LIPID MAPS database and the workflow for AMS followed by retention time based filtering. In addition, files necessary to use both workflows are supplied as well. Both workflows are imported in an already executed state, containing the filtering results as described in the study (see Fig. 5).

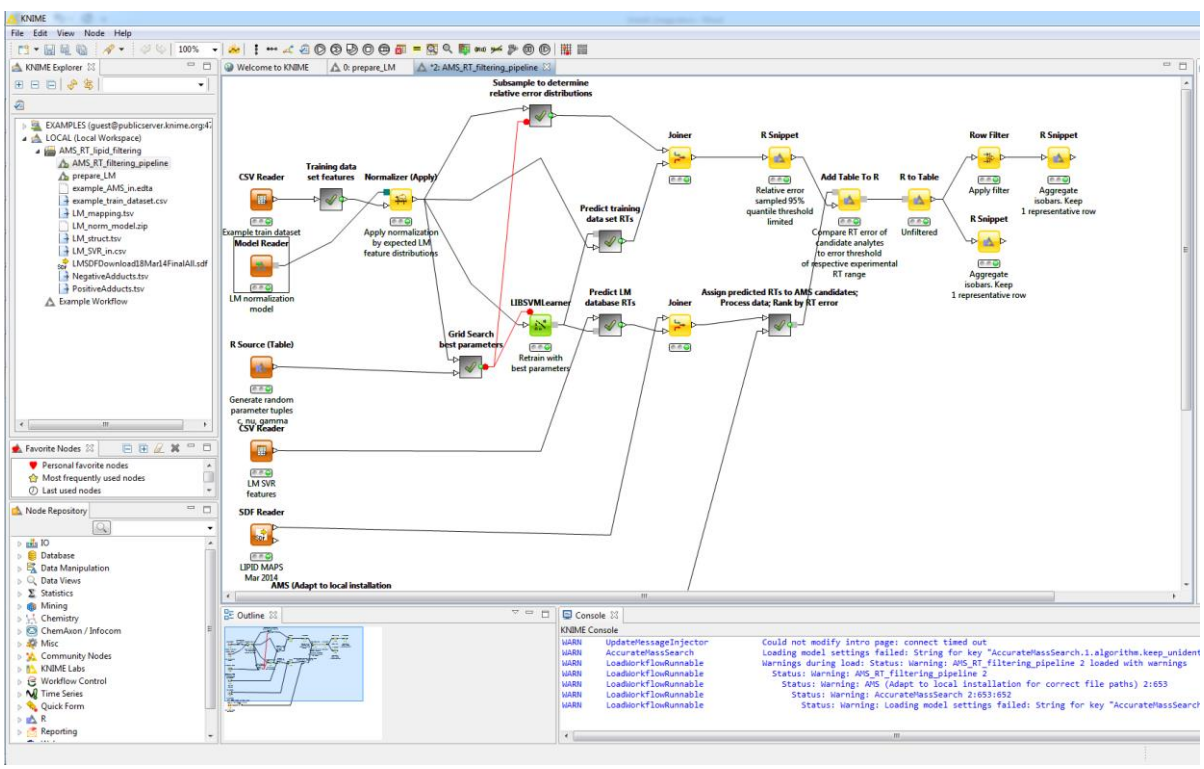

Figure 5: Overview of the imported archive. On the left, imported KNIME workflows and supplied input files are shown.

Usage:

The workflows are already pre-computed. Stored parameters represent the settings used in the study. All node settings are accessible through the configuration dialogs of the individual nodes. In the case of nodes using the R programming language for statistical analysis, the code of the internal processing steps can be seen when opening the respective configuration dialog. Nodes are partly labelled with additional information of their use. For visual reasons, some related nodes were aggregated into grey meta-nodes. Meta-nodes can be opened by double clicking on them.

To edit nodes (for example to change paths for file input or output), KNIME users can configure the nodes by right clicking on them (see Fig. 6). Changed workflows can be re-executed using the green buttons with arrows at the top of KNIME.

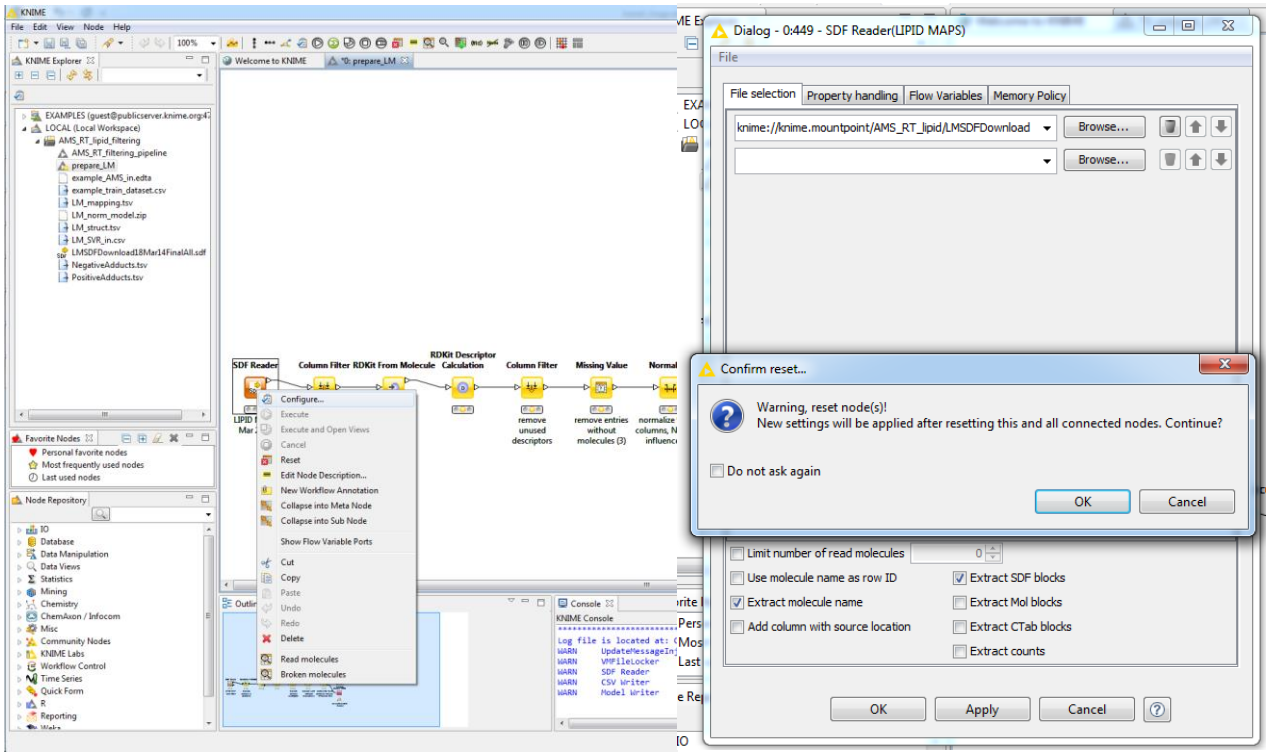

Figure 6: Configuration dialog of the SDF Reader node. Changing parameters resets the node and all nodes dependent on the node output.

Two workflows are included in the supplied KNIME workflow archive: The first workflow (prepare_LM, see Fig. 7) is used to separate the computation of physicochemical descriptors of the large LIPID MAPS database from the AMS filtering pipeline itself. It includes nodes to create structures for the RDKit nodes from SDF structures, a node to compute descriptors on these structures and a normalization node that performs a z-score normalization of the observed descriptor distributions on the database lipids. It also supplies the determined normalization model for use with new lipids in the AMS filtering pipeline.

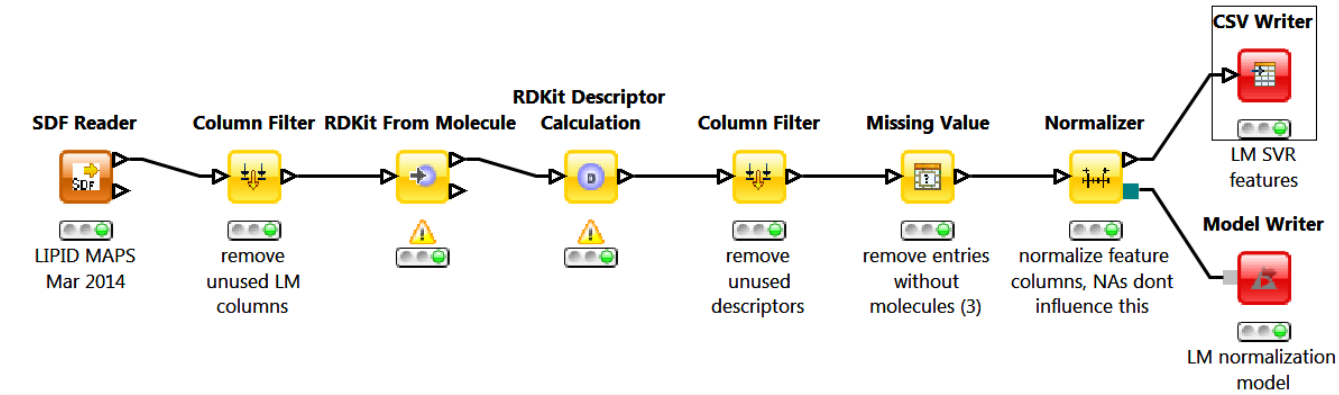

Figure 7: prepareLM workflow.

The second included workflow performs AMS and filtering by retention time as described in the study (Fig. 8).

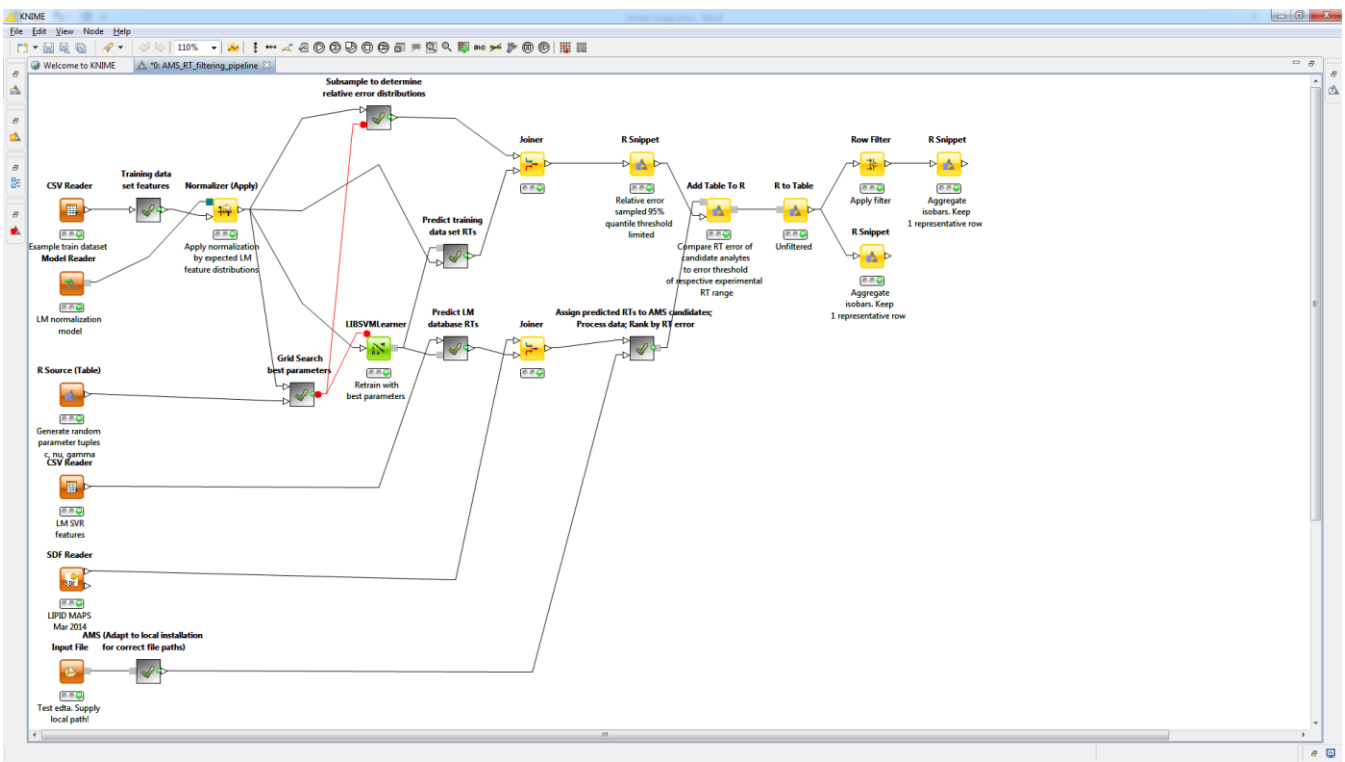

Figure 8: AMS followed by retention time based filtering.

In the top left of the workflow, training data containing retention time in seconds and SMILES structures are read in. Physicochemical descriptors of these structures are computed in the adjacent grey meta-node and normalized with a model based on descriptor distributions of LIPID MAPS lipids. The normalized training features are first used in a grid search to determine optimal parameter settings for the SVR predictor model (meta-node 'Grid Search best parameters'): For each of 100 parameter tuples, a model is trained and evaluated using the same ten data folds. Respective performances are summed across all ten folds and the parameter tuple with the lowest overall error is chosen. Optimal parameters are forwarded

(signified by the red connections) to be used for training of the retention time model as well as to determine 95% thresholds of the relative error. These thresholds are then used with candidates from the AMS to filter candidates with discrepancies between observed and predicted retention times above the threshold. The AMS itself is located in the lower left part of the workflow. It uses as input measurements in .edta format (for correct formatting of your input, consult example_AMS_in.edta among the imported files). Inside the grey AMS meta-node (Fig. 9), the input is converted to an OpenMS internal format and given as input to the OpenMS AccurateMassSearch node.

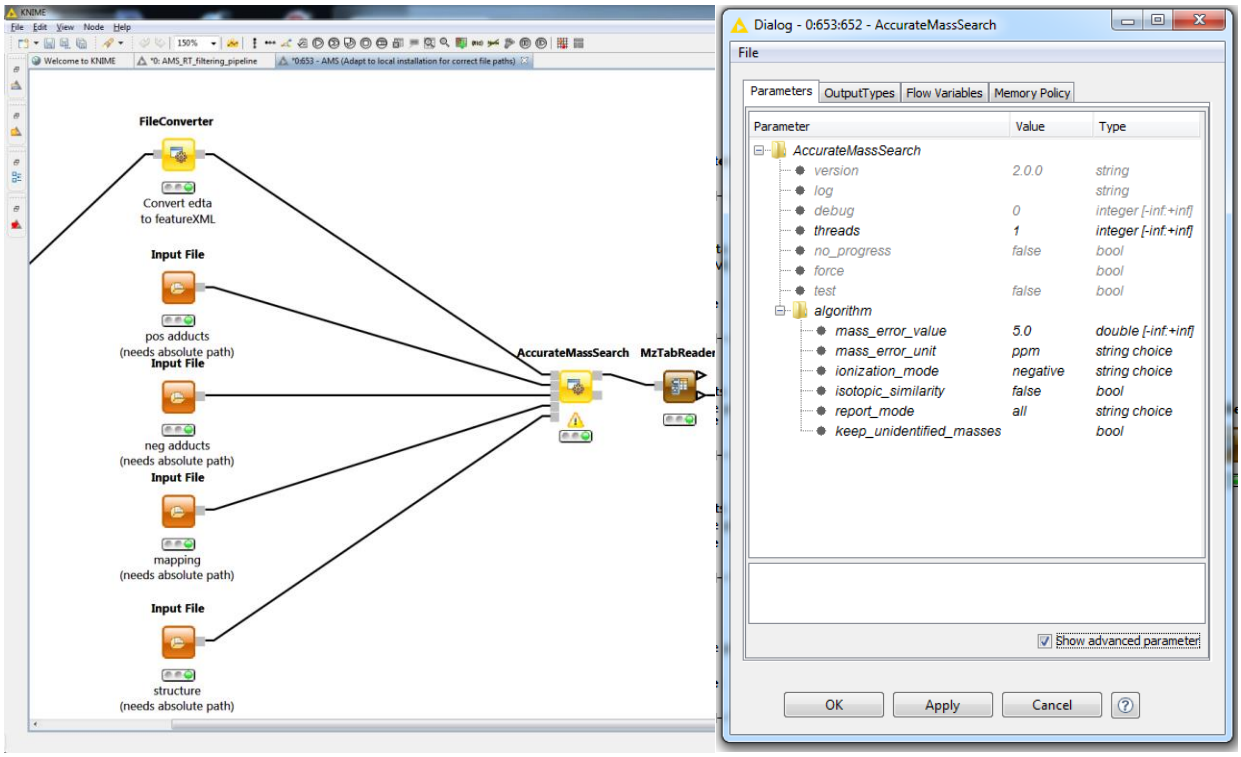

Figure 9: Left: AMS meta-node. Shown are the FileConverter node which in this case changes .edta to .featureXML, the other necessary inputs, the AMS node itself and a node which imports the OpenMS AMS results into KNIME. Right: Parameter settings used in the study for the AMS node.

This node additionally requires files with positive and negatives adducts as well as a mapping and structure file. All of these files are supplied in the workflow archive. However, all five input nodes for the AMS require absolute file paths. If users want to rerun the AMS node, they will have to configure the input nodes to adapt the file paths to their local ones. In contrast, the other input nodes of the filtering pipeline use relative paths to the files of the imported archive and are re-executable without adaption. The AMS itself was run for negative-mode MS data, with an assumed mass error of five ppm.

Output of most nodes including output of the filtering pipeline can be shown in KNIME by right clicking on the respective node and selecting the correct option (Fig. 10). Permanent storage can easily be done by adding fitting writer nodes to the output of interest.

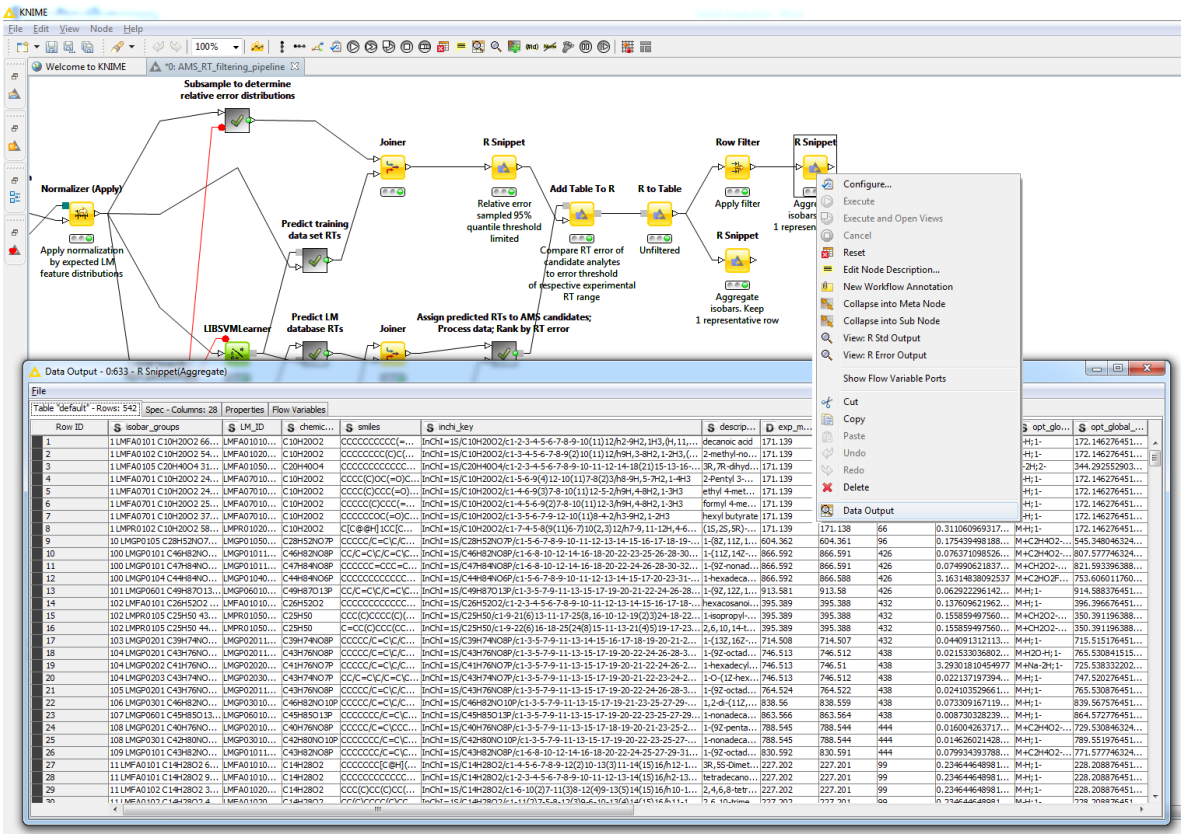

Figure 10: Example output of AMS candidates after filtering and aggregation of isobars.