Supporting Information for the technical brief – CalibraCurve: a tool for calibration of targeted MS-based measurements

This Supporting Information covers the following topics:

- Installation of required software (both for usage of CalibraCurve from the command line and as KNIME (https://www.knime.com/) workflow)
- Usage of CalibraCurve. This section includes guidance for running the application from the command line. Alternatively, a variety of Integrated Development Environments (e.g. RStudio, https://www.rstudio.com/) can be used to start CalibraCurve.
  Information regarding the usage of the KNIME workflow is given in this section as well.
- Requirements regarding the input data files (e.g. data format)
- Description of CalibraCurve result files
- Detailed Explanation of CalibraCurve parameter settings. Special emphasis is given to parameters that control program execution (e.g. for the selection of different techniques for level-removal).
- Experimental setup of the IgG subtype quantification study. Results of this study are shown within the technical brief.
- Two short Appendices that deal with 'Adaption of Response factor plots' and with 'Application of user defined weighting factors'.

List of CalibraCurve abbreviations

C_h:	Highest concentration level of the current data set
C_l:	Lowest concentration level of the current data set
CV:	Coefficient of variation
LLOQ:	Lower limit of quantification
ULOQ:	Upper limit of quantification
$Meas_{CV}$:	Coefficient of variation value calculated for each concentration level from the measurement intensities.
PAR:	Peak area ratio
PB:	Percentage bias
PB_{av}:	Either mean or median value calculated from the PB values of a concentration level
PB_{cv}:	Coefficient of variation value calculated from the PB values of a concentration level
RF:	Response factor
RF_{mean_total}:	Mean value calculated from the complete RF value set of the final linear range
SIS:	Stable isotope-labeled standard
TPB_{av}:	Threshold for PB_{av} values
$TMeas_{CV}$:	Threshold for $Meas_{CV}$ values

1 Installation

Usage of CalibraCurve requires installation of the free software environment R. CalibraCurve has been developed with R version 3.5.1. It is recommended to install the most recent R software, which can be acquired for Linux, (Mac) OS X and Windows machines from https://www.r-project.org/. No additional installation of packages is required for utilization of CalibraCurve directly as an R script.
If you are fine with the AGPL v3 license, it is also recommended to install the open source edition of RStudio from https://www.rstudio.com/. RStudio facilitates code inspection and code adaption for inexperienced R users.

1.1 Installation required for usage of CalibraCurve as KNIME workflow

CalibraCurve is also provided as a KNIME workflow(s). This is a more convenient opportunity to utilize the software for users with less or no experiences with command line tools.
Installation of the CalibraCurve KNIME workflow requires just few additional steps. First, please download and install the KNIME Analytics platform from https://www.knime.com/downloads/download-knime in a version that fits the operating system of your local computer. The CalibraCurve KNIME workflow takes advantage from R code on a large scale. Therefore, installation of the R software is also mandatory and several steps are required to establish the communication between KNIME and the R software:

- From within the R console or RStudio type the command **install.packages("Rserve")** and press enter. This installs the Rserve package that is essential for the KNIME – R interaction.
- From the KNIME menu, choose *Help – Install New Software* in order to open the dialog depicted in figure 1.
- Within the dialog select the following update site: **KNIME Analytics Platform 4.0 Update Site** - **http://update.knime.com/analytics-platform/4.0** (highlighted with a red rectangle in figure 1). Please note that the name of the update site/the URL may vary depending on the installed KNIME version.
- In the filter text field type **R statistics** (highlighted with a green rectangle in figure 1)
- From the presented R plugins select **KNIME Interactive R Statistics Inte**gration (highlighted with a blue rectangle in figure 1) and continue with the wizard.
- After installation of the plugin and a restart of KNIME select *File – Preferences* from the menu in order to open the KNIME Preferences dialog (figure 2)
- Within the dialog specify the path to your R installation folder (highlighted with a red rectangle in figure 2). Apply and close the dialog

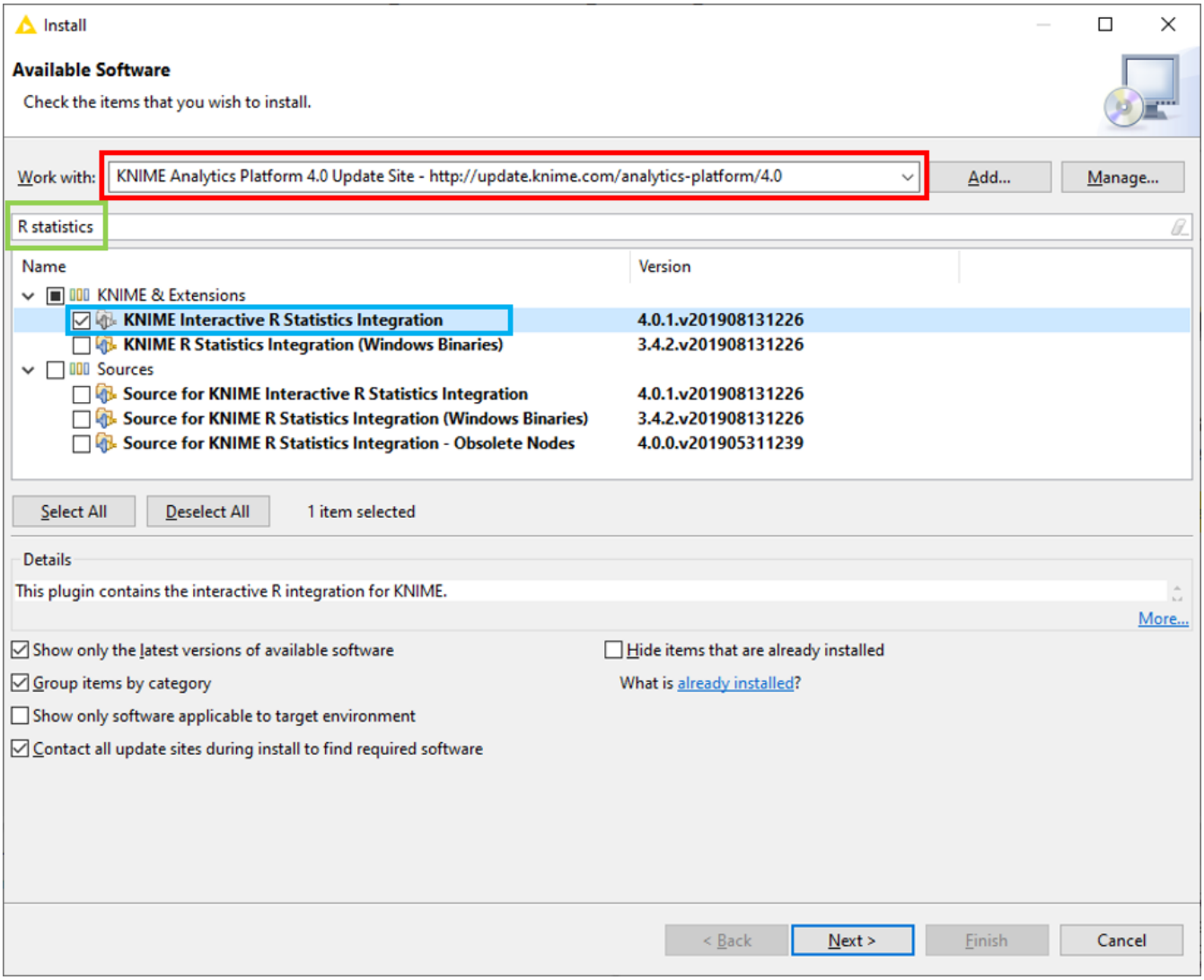

Figure 1. KNIME dialog that enables installation of plugins. Required settings for installation of the R extension are highlighted with colored rectangles, which are explained in more detailed in the main text.

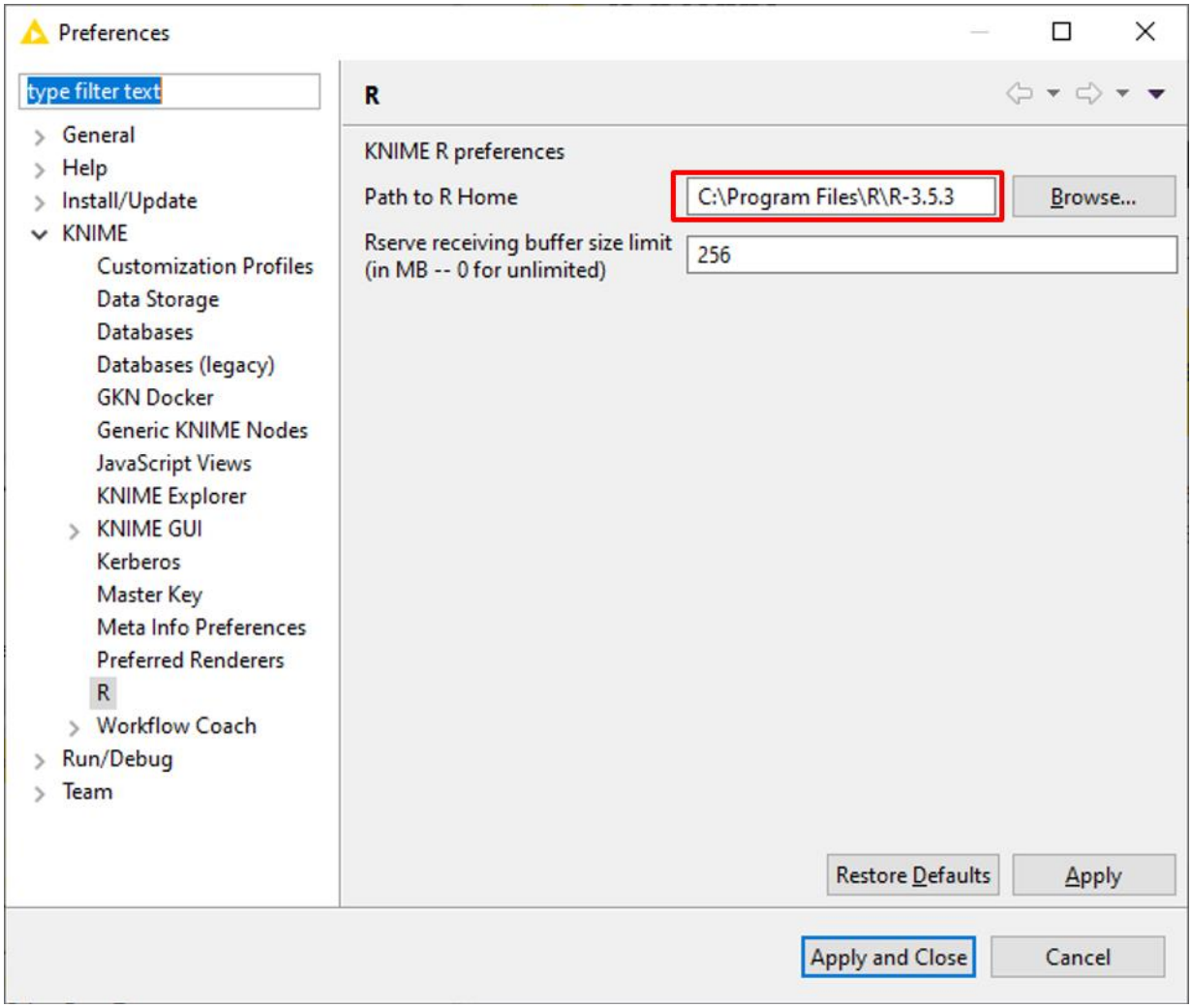

Figure 2. KNIME Preferences dialog that enables usage of R code within KNIME workflows. The required setting is highlighted with a red rectangle. The specified path must correspond to your actual R installation.

You are now ready to import CalibraCurve workflows into KNIME. To this end, please use the menu option *File – Import KNIME workflow* and navigate to the file named CalibraCurve-v.2.0.knwf, which is part of the supplement that also includes this supporting information.
If the import succeeded, the name of the workflow appears within the 'KNIME Explorer', which is located top left per default. Double-click on the workflow to open it (figure 3).
In the same way, you can also import the workflow named CalibraCurve-v.2.0_Excel.knwf into the KNIME software, which is intended for processing of xls/xlsx input files (cf. paragraph 3.1 of this manual).

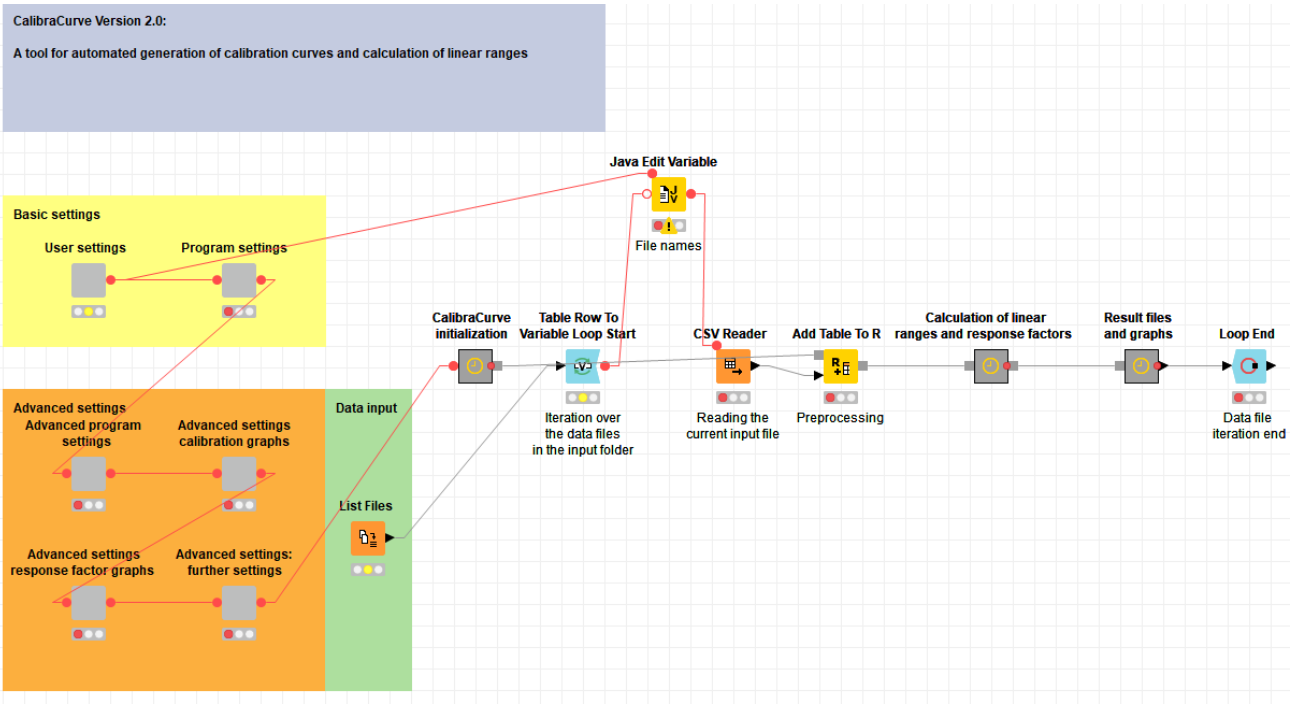

Figure 3. Screenshot of the loaded CalibraCurve-v.2.0 workflow within the KNIME Analytics platform. The workflow consists of squared elements called nodes, which represent small programming units. The grey nodes within the yellow area provide access to the most important CalibraCurve settings. In contrast the grey nodes in the orange area provide advanced settings used for fine-tuning of the CalibraCurve software execution.

2 Usage of CalibraCurve

During execution, CalibraCurve prints error messages to the console in order to assist users with problem solving if necessary. The software also checks the quality of the calibration and prints information concerning inconsistencies that may arise to the console and/or to the log file and the result files.

2.1 Usage of the CalibraCurve R script

You can execute the CalibraCurve R script using RStudio or a similar software, which is recommend for people with minor programming experience.
CalibraCurve can also be executed from the command line using the program *Rscript*:

```
Rscript --vanilla path/to/directory/CalibraCurve_v2.0.R
```

Usage of the above given command is especially interesting for experienced users, e.g. if it is required to execute CalibraCurve as a cron job. The option *vanilla* enables the non-interactive mode of R, which is beneficial if you want to integrate CalibraCurve into other scripts or software.
In order to execute Rscript, you can either navigate to the folder that contains the Rscript software (i.e. the *bin* subfolder of your R installation) or include the bin subfolder in your path variable (recommended for Windows systems). Please avoid special characters (e.g. German ‘Umlaute’) or spaces in your path specification. This holds also for the path specifications given in the CalibraCurve script file.

2.2 Usage of the CalibraCurve KNIME workflow

Once a CalibraCurve KNIME workflow has been loaded (cf. figure 3), please adapt and review at least the settings provided by the grey nodes in the yellow area of the workflow (figure 3). Detailed information for all settings are given in section 5 of this manual. Additionally, open the 'List Files' node in the green area (double-click) and navigate to the directory that store your input files. Close all setting nodes and the 'List Files' nodes. Afterwards, please select the rightmost KNIME node (i.e. the blue Loop End node named 'Data file iteration end') and press the run button  available from the KNIME toolbar.

3. Requirements for the input data

Input data must be provided as plain files given in the comma or tab separated format. An arbitrary number of columns that contain any kind of information can be given. However, the user must specify the column that contains the concentration and the column that contains the measurements in the settings. Originally, CalibraCurve was developed to support generic calibration tasks, where measurement values and amounts or concentrations are related. In targeted proteomics, calibration frequently involves consideration of peak area ratios (PAR). PAR is calculated by PAR = SIS_{area}/NAT_{area}, where NAT_{area} is the peak area of the natural peptide and SIS_{area} denotes the peak area of the (synthetic) stable isotope-labeled standard (SIS) peptide. If you want to use PAR values for calibration, such calculation can be performed easily with standard spreadsheet software (e.g. Microsoft Excel™).
It is recommended to use names for the input files that are related to the measurements therein, because these filenames (without file extensions) are used for naming of the result files, for the headers in these files and for the generated graphs.
The Supporting information contains several example input files.

3.1 Processing of Spreadsheet files with the KNIME CalibraCurve workflow

One of the KNIME CalibraCurve workflow enables analysis of files given in the Microsoft Excel™ formats (xls/xlsx). To this end, please import the CalibraCurve-v.2.0_Excel.knwf file, which is also part of the Supporting information. Within this workflow the CSV Reader node named '*Reading the current input file*' is substituted with the corresponding Excel reader (XLS) node. Usage of this workflow variant is identical with the basic CalibraCurve workflow variant, but now only .xls or .xlsx files are processed.
Please note that the structure of the data must also follow the specifications given for the .csv files (i.e. tabular data structure with headers and at least two columns (one column for concentrations and one column for the measurements). **Important: only one data sheet is allowed for each Excel input file**.

4. Description of CalibraCurve result files

For each input file, CalibraCurve generates two result files and two diagrams (i.e. a calibration graph and a response factor plot). In this section, the content of the result files is explained. The first result file, which starts with the prefix 'res_' followed by the name of the input data file, is divided in six sections:

1. A sentence that states the LLOQ and ULOQ
2. Coefficient of variation values calculated for each concentration level from the measurement intensities ($Meas_{CV}$). In this section, $Meas_{CV}$ values for the *preliminary linear range* are listed.

Note that all further sections refer solely to the *final linear range*.

3. A table containing the original data of the input data file and for each row (measurement) percentage bias values (PB) calculated from both the weighted and the unweighted linear model. A further column gives the corresponding response factor (RF).
4. Descriptive statistics for each concentration level calculated with respect to the **weighted linear model**. This includes PB_{av}, which is either the mean or the median value (user setting) calculated from the PB values of a concentration level, the standard deviation of the PBs and the corresponding CV (PB_{cv}) value.
5. The same as section four but calculated with respect to the **unweighted linear model**.
6. The last section of the result file gives the mean response factors

The second result file, which starts with the prefix 'res_regr_' followed by the name of the input data file gives summary information for the linear fit of the weighted and the unweighted linear model. The models are fitted to the data of the final linear range only.

5. Explanation of CalibraCurve settings

CalibraCurve is highly customizable. Therefore, a comprehensive set of parameters is available that allows fine-tuning of both the algorithm and the output of the software (including visualizations).
In this section, parameter names are given in *italics*. If the parameter values are limited to several possibilities, possible parameter values are highlighted with red font, important remarks and pitfalls are highlighted **bold**. Because settings related to warnings or error messages are self-explaining, they are not listed here.
The CalibraCurve KNIME workflows provide several setting nodes (grey nodes shown in the yellow and orange areas of the workflows, cf. figure 3). Within the setting nodes, the workflows use in comparison with the CalibraCurve R script more detailed labels for the description of the parameters in order to improve readability. In the following parameter listing these detailed labels are given in parentheses and indicated with blue font. Specific features and remarks that only apply to the CalibraCurve KNIME workflow are also given in a suitable position using blue font. Whenever possible, the CalibraCurve KNIME workflow uses dropdown lists for parameter selection. This approach reduces the possibility that wrong (e.g. misspelled) parameter values are applied. Usage of dropdown lists is considered self-explaining and not further explained in the following.
Please note that only a small number of settings must be adapted to run the software with default values that are reasonable for most analyses. These values (namely *inputDir, outputDir ('Output*

directory'), fileTypeOfData ('File format (input data)'), colDelimiterIn ('Column delimiter (input)'), colNumberConcentration ('Column number concentrations', colNumberMeasurements ('Column number measurements'), **and** *concUnit ('Concentration unit')***) are explained in the section 'User settings'.**

User settings

The user MUST adapt or check the settings of this section in order to enable a proper execution of CalibraCurve.

inputDir

This parameter specifies the location of the **directory** that contains the input data (.csv files). Several files with input data are allowed, because CalibraCurve supports batch-wise data processing. An absolute path name is required. Please use the slash ("/") as delimiting character for the path components/directories.
Users of Windows machines may also use the double backslash character(s) ("\\").
This parameter does not exist in the settings of the KNIME workflows, because the input files are loaded using the List Files nodes provided by KNIME (cf. figure 3). Please note that the setting 'C:\tmp' that is given as setting in the imported workflow is only a placeholder that must be substituted with a valid path specification.
Additionally, there is a KNIME workflow that supports processing of spreadsheet data (.xls/.xlsx files). Details are given in paragraph 3.1 of this manual.

outputDir ('Output directory')

This parameter specifies the location of the output directory used for CalibraCurve result files and visualizations (i.e. calibration curves and response factor plots).
An absolute path names is required. Please use the slash ("/") as delimiting character for the path components/directories. Users of Windows machines may also use the double backslash character(s) ("\\").
KNIME supports standard Windows path specifications (i.e. single backslash characters ("\"). Please **do not use double backslash character(s)** ("\\") when using the KNIME workflow.
Please note that the setting 'C:\tmp' that is given as setting in the imported workflow is only a placeholder that must be substituted with a valid path specification.

fileTypeOfData ('File format (input data)')

File extension of the files that contain the input data. Because input data should be provided in the comma-separated format, reasonable extensions are txt or csv.
Do not use dots (e.g. .txt or .csv) for this parameter.
Note, that all files with the specified extension are read from the input directory. Therefore, no other files with the specified extension are allowed in the input directory.
This parameter does not exist in the settings of the CalibraCurve KNIME workflows. Instead, the user needs to configure the 'List Files' node in the green area of the workflows. Depending on the imported workflow either csv files or xls/xlsx files are processed. In case of the workflow named CalibraCurve-v.2.0.knwf, only files with the file

extension csv are accepted as input. In case of the CalibraCurve-v.2.0_Excel.knwf workflow all files in the input directory with either the extension xls or xlsx are processed.

colDelimiterIn (‘Column delimiter (input)’)

The column delimiter used in the input file(s). Appropriate values are , or ; or \t (for tab-separated values).
When using the CalibraCurve KNIME workflow with Microsoft Excel™ formatted files (CalibraCurve-v.2.0_Excel.knwf) this parameter is not necessary/available.

colDelimiterOut (‘Column delimiter (output)’)

Column delimiter used for the result files. Appropriate values are , or ; or \t (for tab-separated values).

colNumberConcentration (‘Column number concentrations’)

The number of the column in the input files that contains the concentration values.

colNumberMeasurements (‘Column number measurements’)

The number of the column in the input files that contains the measurement values or the peak area ratios (PAR).

concUnit (‘Concentration unit’)

Unit that is used to for the concentration quantities (e.g. fmol/µl). The value of this parameter is used in the result files and in CalibraCurve plots.

Prog**ram settings**

The settings described in this section are related to data processing and software execution. Adaption of these settings allow fine-tuning of the CalibraCurve algorithm.

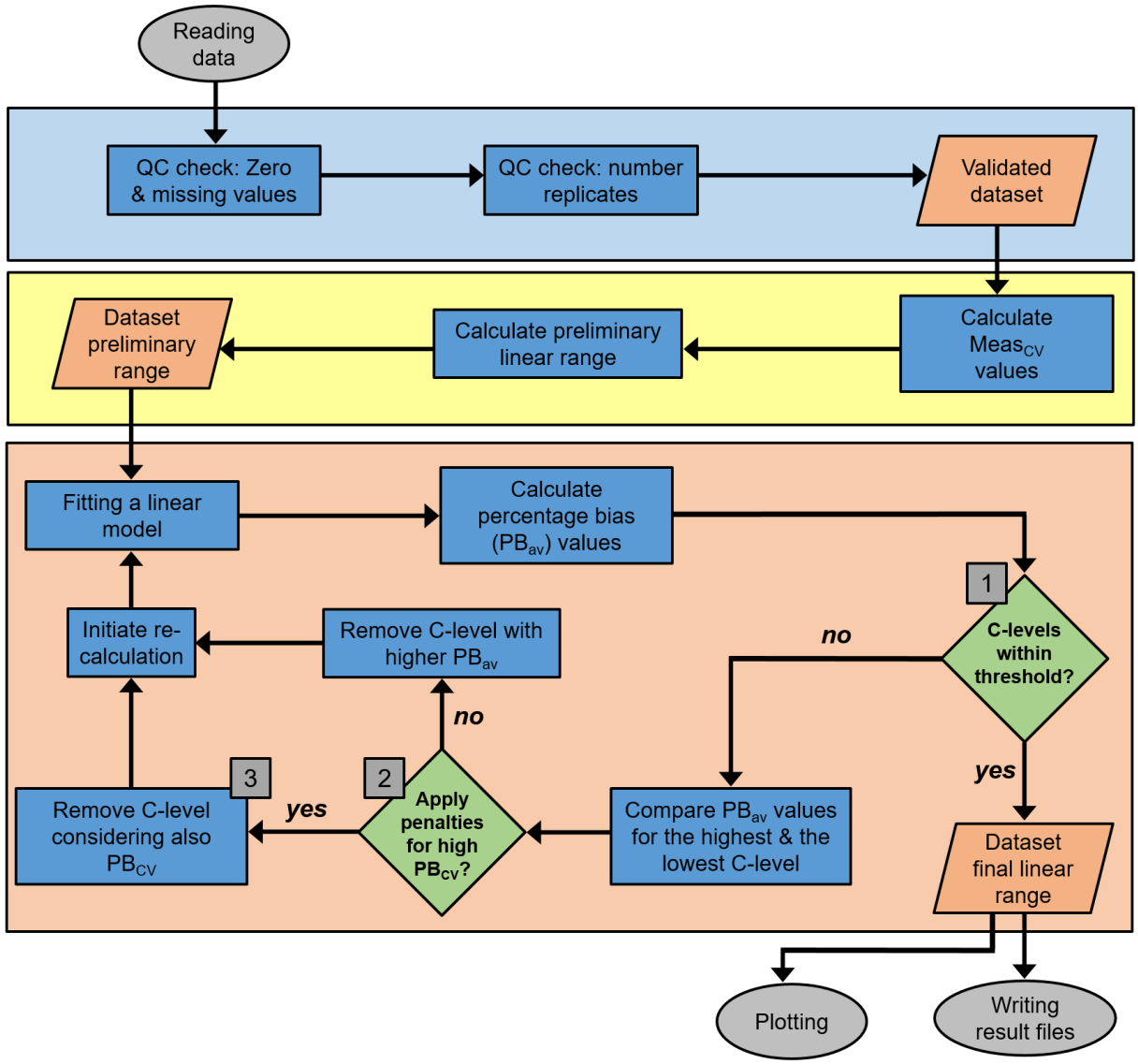

Figure 4. Flowchart depicting the workflow of the CalibraCurve software. For details, please refer to the explanations given in the technical brief. The numbered grey rectangles are referred in the descriptions of parameters that have major effect on the branches of the algorithm (indicated by the green diamonds) or on PB_{CV}-depending removal of concentration levels (grey rectangle with number 3).

<u>Settings related to quality control (Figure 4, light blue frame)</u>

minNumberReplications (‘Minimum number of replications’)

The value of this parameter is used for quality control of the input data. It specifies the minimum number of replications required for each concentration level. The value of this

parameter must be greater than 1, because CalibraCurve calculates measures of dispersion for each concentration level.

Settings related to the assessment of precision (used for calculation of the preliminary linear range, (Figure 4, yellow frame)

cv_thres (‘Meas_CV threshold’)

Threshold (given as a percentage) for $Meas_{CV}$. Common values for this threshold are 20 percent. Note that this parameter is named $TMeas_{CV}$ in the technical brief.

Settings related to the assessment of (percentage) bias (used for the calculation of the final linear range, (Figure 4, orange frame)

perBiasThres (‘Percent bias threshold’)

Threshold for the maximal allowed average percentage bias (PB_{av}) calculated for each concentration level. Percentage bias (PB) measures the deviation of the calculated concentration from the expected concentration (given as a percentage). This parameter is crucial for the decision whether the algorithm finishes with final calculations and the generation of plots or starts another round where one concentration level is selected for removal (grey rectangle number 1 in figure 4).
The algorithm finishes if both the lowest and the highest concentration level pass *perBiasThres*. Note that this parameter is named TPB_{av} in the technical brief.
Appropriate settings for this threshold may vary for different areas of application. Using drug development as an example, the Food and Drug Administration (FDA) demands not more than 20% PB_{av} deviation at LLOQ and not more than 15% PB_{av} deviation at the higher qualified concentration levels[1]. CalibraCurve uses 20% as default PB_{av} threshold value for all levels of the linear range.
However, users can adapt this threshold and review the PB_{av} values that are reported for each concentration level in the result file in order to ensure that the quality requirements of a specific assay are accomplished.

weightingMethod (‘Weighting method’)

This setting is used to select the weights that are used for fitting of the linear model. Possible values are 1/x^2 and 1/x. It is also possible to define an own weighting method (cf. Appendix 2 of this manual).
Please note, that CalibraCurve always computes both a weighted and an unweighted linear model.

centralTendencyMeasure (’Central tendency measure’)

This parameter specifies whether mean or median values are calculated as a measure for central tendency of the **percentage bias**. Hence, possible values are mean and median.

[1] U.S. Food and Drug Administration, Guidance document: Bioanalytical Method Validation Guidance for Industry, Issued by: Center for Drug Evaluation and Research, 2018. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/bioanalytical-method-validation-guidance-industry

Advanced settings

Advanced Program settings

calcContinuousPrelimRanges (‘Apply continuous preliminar range calculation’)

This setting is a switch for different methods that can be used for calculation of the preliminary linear range. Possible values are TRUE or FALSE.
If the parameter is set to FALSE, the preliminary range extends from the lowest to the highest concentration level that pass *cv*_thres. However, with this parameter setting **intermediate** concentration levels may fail to pass the threshold criterion.
If the parameter is set to TRUE (default setting), all concentration levels of the preliminary linear range must show $Meas_{CV}$ values that pass the threshold criterion (*cv*_thres).
If several such ranges exist that are separated by concentration levels with $Meas_{CV}$ > *cv*_thres, CalibraCurve selects the range with the highest number of successive concentration levels. If by chance several ranges show the **same number** of successive concentration levels, CalibraCurve again selects the range that extends from the lowest to the highest concentration level that pass the threshold criterion (similar to the behavior implemented for ‘calcContinuousPrelimRanges= FALSE’). Occurrence of several ranges separated by concentration levels with $Meas_{CV}$ values > *cv*_thres is considered as inconsistency and warnings are written to the result file.

considerPerBiasCV (’Apply advanced level selection’)

Possible values are TRUE or FALSE. This parameter is crucial for the decision (grey rectangle number 2 in figure 4) whether the algorithm applies penalties for high PB_{cv} values. If *considerPerBiasCV* == TRUE, PB_{cv} values are considered for the selection of a concentration level for removal. Otherwise (*considerPerBiasCV* == FALSE), the concentration level with the higher PB_{av} value is selected for removal. This parameter is related to the *perBiasDistThres* parameter.

perBiasDistThres (‘Distance threshold for percent bias’)

Threshold for the **distance** between PB_{av} values of different levels. This parameter allows application of penalties for high PB_{cv} values:
Depending on *perBiasDistThres,* different methods are used to select a concentration level for removal. If the difference between PB_{av} of the highest (C_h) and the lowest concentration level (C_l) undercut *perBiasDistThres*, the level that shows higher PB_{cv} is removed. Otherwise, the level with higher PB_{av} is selected for removal.
Consideration of this parameter can be disabled (c*onsiderPerTrueCV* = FALSE)

Settings related to the linear models

finalRangeCalculationMethod (‘Calculation method for the final linear range’)

Possible values are weighted_linear_model or unweighted_linear_model.
This parameter selects whether the weighted or the unweighted model should be used for the calculation of the final linear range. **It is strongly recommended to use the weighted model especially if analysis of broad concentration ranges is carried out.**

The following settings are used for fine-tuning of the CalibraCurve output (plots and result files). From our point of view most users will be satisfied with the default values. However, the parameters are especially useful for users that are not familiar with the R programming language in order to create diagrams with sufficient quality for publication (e.g. if a higher resolution of plots is required).

Advanced settings calibration graph(s)

plotOptReg (‘Presentation of calibration graphs’)

Possible values are combined and separate. *plotOptReg* selects whether both the unweighted and weighted regression lines are drawn in the same plot (*plotOptReg* == combined). If the option separate is given, separate files are generated for the unweighted and the weighted regression lines.

Scaling (‘Scaling’)

The value for *scaling* specifies whether a logarithmic or a linear scale is used for both the x- and the y- axis of calibration graph(s). Possible values are log and linear.

magnificAxis, magnificLabels, magnificTitle, magnificEquation (’Magnification of axes annotations, Magnification of axes labels, Magnification of the title, Magnification of the equations’)

Settings that define the magnification of several parts of the calibration graphs relative to 1 (default value). *magnificAxis* affects magnification of axes annotations (labels associated with a tick), m*agnificLabels* affects magnification of the axes labels, *magnificTitle* affects magnification of the title and finally *magnificEquation* affects magnification the equations displayed in the calibration graphs.

plotResolution (‘Resolution (dpi)’)

Resolution of the calibration graphs, given as dpi (dots per inch).

plotHeight (‘Plot height (cm)’)

Height of the calibration plot given in cm.

plotWidth (‘Plot width (cm)’)

Width of the calibration plot given in cm.

titleOpening (‘Title text part 1’)

First part of the calibration graph title.

titleMiddlePart (‘Title text part 2’)

Second part of the calibration graph title.

xLab (‘X-axis label part 1’)

First part of the X-axis label. The text is extended with the specified concentration unit (i.e. the value of the parameter *concUnit*).

yLab (‘Y-axis label’)

Y-axis label

legend_unweighted (‘Legend for unweighted line/equation’)

Text used in the legend of the calibration graphs in order to identify the unweighted line and equation.

legend_weighted (‘Legend for weighted line/equation’)

Text used in the legend of the calibration graphs in order to identify the weighted line and equation.

showLegend (‘Show legend’)

Switches the display of the legend in the calibration graphs on or off.

Advanced settings response factor graphs

Using a combination of several parameter values (*yLimAdaptRF, minY_lim, maxY_lim, restrictRfToFinalRange)* CalibraCurve supports adaption of the y-axis of the response factor graphs. However, finding an appropriate combination of these parameter settings is sometimes challenging. Therefore, a short instruction is given as an appendix of this manual.

yLimAdaptRF (‘RF plot: y-Axis adaption yes/no’)

Possible values are TRUE or FALSE. The parameter is a switch that enables adaption of the y-axis range with user defined settings (TRUE). For *yLimAdaptRF == FALSE* default values for the range of the y-axis are used.

minY_lim (‘Min. y-axis limit’)

Lower limit of the y-axis (**Note that this parameter value needs adaption depending on the analyzed data!!!**). Application of this parameter requires *yLimAdaptRF* == TRUE.

maxY_lim (‘Max. y-axis limit’)

Upper limit of the y-axis (**Note that this parameter value needs adaption depending on the analyzed data!!!**). Application of this parameter requires *yLimAdaptRF* == TRUE.

restrictRfToFinalRange (‘Confine RF plots to the final linear range’)

Possible values are TRUE or FALSE. The parameter is a switch that enables restriction of the response factor plots to the data calculated as final linear range (TRUE). Otherwise (*restrictRfToFinalRange* == FALSE) the response factors values calculated for the validated dataset are displayed in the plots.

scalingRFX (‘RF plot: Scaling’)

The value for *scalingRFX* specifies whether a logarithmic or a linear scale is used for the x-axis of response factor plot. Possible values are log and linear. **Note that linear scale is always used for the y-axis.**

RfThresL (‘Lower horizontal line (percent RF threshold)’)

Threshold (given as a percentage). The value of this parameter is used to draw a horizontal line in the response factor plot indicating a **downward deviation** from the mean response factor (RF_{mean_total}). RF_{mean_total} is computed from all RFs that are calculated for the final linear range.

RfThresU (‘Upper horizontal line (percent RF threshold)’)

Threshold (given as a percentage). The value of this parameter is used to draw a horizontal line in the response factor plot indicating an **upward deviation** from the mean response factor (RF_{mean_total}). RF_{mean_total} is computed from the complete RF value set of the final linear range.

magnificAxisRF, magnificLabelsRF, magnificTitleRF (‘RF plot: Magnification of axes annotations, RF plot: Magnification of axes labels, RF plot: Magnification of the title’)

Settings that define the magnification of the several parts of the response factor plots relative to 1 (default value). *magnificAxisRF* affects magnification of axes annotations (labels associated with a tick), *magnificLabelsRF* affects magnification of the axes labels and finally, *magnificTitleRF* affects magnification of the title.

plotResolutionRF (‘RF plot: Resolution (dpi)

Resolution of the response factor plots, given as dpi (dots per inch).

plotHeightRF (‘RF plot height (cm)’)

Height of the response factor plots given in cm.

plotWidthRF (‘RF plot width (cm)’)

Width of the response factor plots given in cm.

titleOpeningRF (‘First part of the RF plot title.’)

xLabRF (‘RFplot: X-axis label’)

X-axis label of the response factor plot

yLabRF (‘RFplot: Y-axis label’)

Y-axis label of the response factor plot

showLegendRF (‘Show legend for response factor plots’)

Switches the display of the legend in the response factor plots on or off.

Further settings

Settings described in this section do not influence the algorithm and have no impact on CalibraCurve calculations. Mainly, these settings are intended to adapt the output of CalibraCurve (result files, graph captions).

verbose

Possible values are TRUE or FALSE. Enables the verbose mode of CalibraCurve (TRUE). This mode is mainly useful for debugging. In the verbose mode, intermediate percentage bias calculations are written to the log file.
This parameter is of interest only for developers that want to adapt the R script.
This parameter is not available in the KNIME CalibraCurve workflows.

numberDecimals (‘Number decimal places’)

Control for the number of used decimal places in the result files.

FileTypeOfResults (‘File format (output data)’)

This setting specifies the file extension of the result files. Because the result files contain plain text and tables reasonable extensions are txt or csv.
Do not use dots (e.g. .txt or .csv) for this parameter.
Note, that existing files with the same names in the output directory are overwritten without further warnings.

The R script of CalibraCurve includes several variables (*resFHeader, resRegFHeader, resFCVHeader, resFCVPrefix, headerPerBias, headerOverviewPerBiasW, headerOverviewPerBiasUW, headerMeanRf*) that contain simply headings for different parts of the result file. They are not detailed here, because they are self-explaining.

6. Experimental setup of the IgG subtype quantification study

IgG peptide sample preparation

Native, non-labelled (light) purified immunoglobulin G (IgG) subtype protein samples from *Macaca mulatta* were kindly provided by Prof. Dr. med. Klaus Überla from the Institute of Clinical and Molecular Virology of the Universitätsklinikum Erlangen. IgG samples were reduced with DTT followed by alkylation with iodacetamide and digested with trypsin (SERVA Electrophoresis, Heidelberg, Germany) for 14 h at 37 °C. Concentration of digested IgG peptides and corresponding synthetic heavy labelled peptides (IgG1: GPSVFPLAPSSR, IgG4: GPSVFPLASSSR; INTAVIS Bioanalytical Instruments AG, Cologne, Germany) were determined by quantitative amino acid analysis (AAA) on an ACQUITY-UPLC (Waters).

LC-MS/MS analysis

LC-MS/MS analysis was performed on a Q Exactive Orbitrap mass spectrometer (Thermo Fisher Scientific) coupled to an Ultimate 3000 RSLCnano (Dionex, Thermo Fisher Scientific) HPLC system. All

peptide samples were diluted in 0.1% TFA prior to injection. Thereby, different concentrations of the heavy stable labelled peptides were spiked in with increasing concentration per sample (11 concentration steps from 0 – 1707 fmol/µl) to 140 ng native IgG (light). Samples were loaded and pre-concentrated on a trap column (Acclaim PepMap, 100, 100 µm x 2 cm, nanoViper, C18, 5 µm, 100 Å, Thermo Fisher Scientific) within 7 min at a flow rate of 30 µl/min with 0.1% TFA. Subsequent peptide separation was performed on an analytical column (Acclaim PepMap RSLC, 75 µm x 50 cm, nanoViper, C18, 2 µm, 100 Å) at a flow rate of 400 nl/min with a 12 min linear gradient from 25 to 35% solvent B (solvent A: 0.1% FA, solvent B: 0.1% FA / 84% acetonitrile). The column oven temperature was set to 60 °C. Peptides eluting from the column were ionized at 1.55 kV in positive mode in the Nanospray Flex Ion Source (Proxeon Biosystems A/S, Odense, Denmark). Data were acquired in PRM mode with survey scans from 400 to 1100 m/z at a resolution of 35,000 in profile mode with an automated gain control (AGC) of 3E6 ions, and a maximum injection time (IT) of 100 ms. An inclusion list with masses from heavy and light IgG peptides (e.g.: IgG1: GPSVFPLAPSSR, m/z mass: 607.8300 (light) and 612.8342 (heavy), IgG4: GPSVFPLASSSR, m/z mass: 602.8197 (light) and 607.8238 (heavy)) was used to schedule acquisition of MS/MS spectra within 5 minutes time windows. PRM scans were performed with a resolution of 17,500 in profile mode with an automated gain control (AGC) of 2E5 ions, and a maximum injection time (IT) of 50 ms and a maximum loop count of 12. The isolation window was set to 2.0 m/z. Higher-energy collisional dissociation with a normalized collision energy (NCE) of 27 was used for peptide fragmentation.

Data analysis

Generated *.raw files were inspected and processed in Skyline (Version 4.1.0.11796). Automated peak picking of co-eluting heavy and light-labelled peptides were adjusted manually. Peak areas of best product ions were extracted and further processed in the software R (R (v.3.5.2), R Studio (v.1.1.463)). Peak areas of most intense product ions were scaled to the maximum values of each peptide and label type.

Appendix 1: Adaption of Response factor plots

CalibraCurve enables customization of the response factor plots. The most obvious adaption is activation of the parameter *restrictRfToFinalRange (‘Confine RF plots to the final linear range’)*, i.e. *restrictRfToFinalRange* == TRUE. With this setting it is possible to review/evaluate response factor values of the final linear range in more detail. Most user are maybe satisfied with the results of this setting.

An additional opportunity is available by choosing *yLimAdaptRF (‘RF plot: y-Axis adaption yes/no’)* == TRUE. This setting enables adaption of coordinate ranges for the y-axis in order to control the scaling of this axis in more detail. However, wrong settings may result in graphics that are drawn outside of the graphics window.
In order to find appropriate settings users should review the response factor values reported in the result files. The minimum and maximum response factor values should be assigned to the *minY_lim (‘Min. y-axis limit’)* and the *maxY_lim (‘Max. y-axis limit’)* parameter, respectively. This is a good starting point in order to figure out the ideal settings.
It is also recommended to switch the display of legends off (*showLegendRF (‘Show legend for response factor plots’)* == FALSE) when adapting the y-axis.

Appendix 2: Application of user defined weighting factors

CalibraCurve enables customization of the R script in order to utilize your own weighting method.
The numbers given in the following explanation refer to the lines of the R script source code that must be adapted.

- **Line 67**: Change the value of the parameter 'weightingMethod' to a character string of your choice (e.g. 'OwnWeights'). Please note that the values '1/x^2' and '1/x' refer to the weighting methods pre-implemented in CalibraCurve.
- **Line 336**: Please define your own weighting method here.