

## Tutorial for Continuous Quality Monitoring for Non-target MS-based Analysis (QualiMon)

*Olle Hartvigsson<sup>1</sup>, Anton Ribbenstedt<sup>1</sup>, Marina Armeni<sup>1,2</sup>, Kristian Pirttilä<sup>3</sup>, Rui Zheng<sup>4</sup>, Otto Savolainen<sup>2,5</sup> & Carl Brunius<sup>1</sup>*

*<sup>1</sup> Division of Food and Nutrition Science, Department of Biology and Biological Engineering, Chalmers University of Technology, 412 96 Göteborg, Sweden*

*<sup>2</sup> Chalmers Mass Spectrometry Infrastructure, Chalmers University of Technology, 412 96 Göteborg, Sweden*

*<sup>3</sup> Department of Medicinal Chemistry, Uppsala University, SE-75123 Uppsala, Sweden*

*<sup>4</sup> Department of Surgical Sciences, Medical Epidemiology, Uppsala University*

*<sup>5</sup> Faculty of Health Sciences, Institute of Public Health and Clinical Nutrition, University of Eastern Finland, Finland*

Correspondence : [anton.ribbenstedt@chalmers.se](mailto:anton.ribbenstedt@chalmers.se)

## Introduction

*QualiMon* is an R package with a Shiny user interface, developed to facilitate quick, easy, yet comprehensive quality monitoring of injections. In this tutorial you will learn how to use *QualiMon* to monitor the quality of your LC-MS data in (near) real-time.

*QualiMon* is largely based on so called landmark features (LaMas), which are features present in every injection specific to instrumentation, polarity, chromatography and sample matrix. Once an injection is finished, *QualiMon* automatically processes and evaluates the injection based on several metrics. These metrics come in two forms: injection-based and LaMa-based metrics.

Injection-based metrics:

- Total number of peaks
- IPO-score (A scoring function based on features with isotopic patterns from the IPO algorithm used to optimize peak picking parameters in XCMS)
- Total ion count
- Number of LaMas found

LaMa-based metrics (recorded for each landmark):

- Total intensity (peak area)
- Full-width half maximum (FWHM)
- Tailing factor
- Peak height
- Number of scans per peak
- Noise
- Signal-to-noise ratio
- Retention time deviation from reference LaMa value

While the injection-based metrics represent the injection as a whole, LaMa-based metrics are recorded and checked for each LaMa found within the injection. The total number of outliers for each metric is then used for the quality assessment.

Based on how much an injection deviates from other samples within the same experiment, in turn based on all the user-selected metrics (default is to use all), a final score is then calculated, and the user is presented with a quality score. If an injection is marked as of potentially of poor quality, there is also an option to get notified via Slack.

## Installation

To install *QualiMon*, use the code below. The installation depends on the *remotes* R package, which if not already present among your available packages is installed in the first line of code.

```
if (!require("remotes", quietly = TRUE)) install.packages("remotes")  
remotes::install_git("MetaboComp/QualiMon ")
```

Furthermore, you have to install the LC-MS file conversion software Proteowizard on your computer (which is free to download [here](#)).

To launch the *QualiMon* Shiny app, use the following code:

```
library(QualiMon)  
launchQualiMon()
```

## Set-up

Once setup, using *QualiMon* is fairly straight-forward to use, but in order to get started, some information from the user is needed. If you have already completed the set-up, you may want to jump the following section describing the set-up wizard and proceed to the tutorial section “*Day-to-day usage*” for monitoring instructions or the “*Review Old Data*” section for instructions on how to review old data.

### Determining Landmark Features

To start the set-up wizard, press the ‘**Setup**’ button in the left-hand menu followed by pressing ‘**find LaMas & New DB**’, which will create an SQL database file. Enter the desired name of the file as well as file path where you want it to be created. Then press the button ‘**Create new DB file**’ to generate the .db file.

**IMPORTANT:** You should only set up one .db file for *QualiMon*. This db will cover all combinations of chromatography and polarity (what we in *QualiMon* refer to as chromPol).

The next step is to identify LaMas in already existing data generated with the same chromatography and polarity as you want to monitor: the more injections processed the better! We recommend around 500-1000 injections, but if your system is new a lower number is also possible.

To do this you will need to use XCMS. As missingness is a very important parameter for the LaMa identification, a dataset before peak filling is required. You will thus have to supply a peak table from an xcms object after peak correspondence (xcms function groupChromPeaks), but prior to peakfilling, saved as an .rds object ("saveRDS(object\_name, "file-name.rds)"). An example code is available at <https://github.com/MetaboComp/QualiMon/workflowexamples/xcmspreprocessing.R>. To obtain a peak table from an XCMS object, *QualiMon* comes with the function getTable, that creates a peak table, correctly formatted for the input of *QualiMon*. If you are unfamiliar with XCMS preprocessing, please see the XCMS vignette for instructions on this part (available at [the xcms R package vignette](#)). For demonstration purposes, you will also find an example script for how to generate such an xcms object as well as an example rds file at the *QualiMon* repository {address}. You also need to supply the symbol used as a delimiter between the m/z and retention time in the feature names (e.g., if you have your feature identifiers as m/z@rt, the delimiter needs to be set to “@”, without the quotation marks). In addition, the algorithm also requires the minimum intensity setting for a peak to be considered a peak (i.e., the second value of the prefilter argument in XCMS function findChromPeaks()), a retention time value from where the algorithm should start looking for LaMas (i.e., usually the dead volume, in seconds), and which chromatography and polarity the data set was generated from (i.e., RP (Reverse

phase Positive), RN (Reverse phase Negative), HP (HILIC positive) or HN (HILIC Negative). Figure 1 shows an example of what a filled-out arrangement can look like.

Choose / Create database file

---

Choose name for new db file

Choose the folder where DB file will be created

C:/temp/MyAwesomeDB.db

---

Input parameters for finding LaMas

---

File: PT\_NoFill\_0.4\_RPPOS\_1.5\_1.5.rds

Input delimiter for mz & RT (e.g. '@')

The RT (s) from where to start looking for LaMas

Prefilter intensity used for XCMS processing

Select the chrom. and pol. of the data

Optimization takes up to 2 hours

Figure 1. Example of a filled-out arrangement of Find LaMas & New DB

After the required information is filled in, the algorithm normally takes a few hours to compute (depending on sample size of input data and computer processing speed) and provides the user with 10 suggested parameter settings, together with plots of the LaMa distribution in the m/z and rt domains. Here the user needs to select which set of LaMas appear to be the best. To do this, there are a few different things to consider (in order of importance):

1. Maximize the distribution of LaMas in the rt and m/z domains (by visual inspection)
2. Maximize the minimum intensity of LaMas
3. Minimize the allowed missingness of LaMas
4. Maximize the rt coverage
5. Maximize the distance between LaMas (rt and m/z diff parameters)

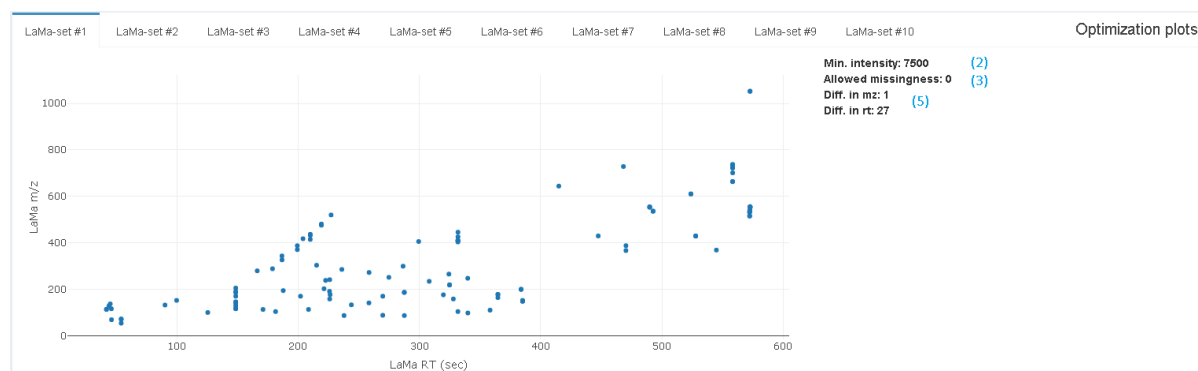


Figure 2. Plot of one of the 10 suggestions for LaMa sets obtained from optimization.

Once the optimal parameter settings have been manually selected, your LaMas can now be saved in the SQLite database by selecting which LaMa-set to use and pressing 'Choose .db file to submit LaMas to'. The user also has the option of saving the optimization data for future inspection / use by filling in a name for the save file under 'Name for opt. output save file (.rds)' and pressing 'Save opt. data'. To load old optimization data just click the 'Load opt. data' button and navigate to your saved file.

Remember: If you have already set up a .db file for another chromPol, you do not need to create a new .db file when new landmarks are wanted, simply skip the step '**Choose / create database file**' and start directly at '**Input parameters for finding LaMas**'.

### Creating a configuration file

After identification of LaMas, a configuration file needs to be created. The set-up wizard helps with this through the sidebar buttons '**Setup**' -> '**Set-up wizard**'. Here you will go through 13 steps to set up the configuration file.

**Step 1.** Select the folder you wish to monitor. This is the top level folder on your instrument computer, where original instrument data are generated. This top level folder can (and normally does) contain subfolders where the raw data files are generated in experiments and batches so that data can be seamlessly monitored without having to restart the monitoring every time a new experiment or batch is started. *QualiMon* also supports hardware set up that separates data generation from processing and monitoring on different computers: Quite simply, you will just have to specify a network folder location.

**Step 2.** Select the folder where back-up files will be copied to (optional, press the check-button '**Skip backup**' if this is not wanted). Automated file backup from an instrument computer is especially useful for data security purposes.

Please note that the folder selection during this step may malfunction due to a yet unidentified bug. If you encounter this phenomenon, you can circumvent it by pressing cancel and then re-opening the select folder menu.

**Step 3.** Select the folder where *QualiMon* will store mzML files converted from the original vendor file format that are used for the quality analysis

**Step 4.** Select the file path to the MSconvert.exe file from the ProteoWizard software (usually found in 'C:/Users/YOURUSERNAME/AppData/Local/Apps/ProteoWizard3.0.21128.7376ae98864-bit/msconvert.exe' (or similar) on computers running windows; also note that "AppData" is a hidden folder in Windows, so you might need to make it visible).

**Step 5.** Select the .db file generated during the "*Determining landmark features*" step.

**Step 6.** Enter identifiers of files that you don't want to be evaluated for quality. These are specified as character strings present in their filenames, separated by a space (e.g., "blank sst conditioning MS2"). If you later find out you need to specify more such identifiers of files to exclude, you can easily add them manually to the config file using e.g., wordpad (ctrl F in the config file for "check").

**Step 7.** Specify instrument (Currently only QTOF and Orbitrap supported). This information is for documentation purposes only and if you have another type of system you can choose either option as a "dummy".

**Step 8.** Specify the sample matrix analyzed, since LaMas are specific to the sample matrix

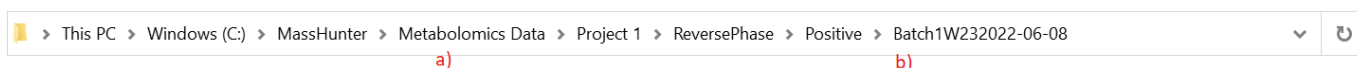
IMPORTANT: At present, only one sample matrix is allowed per config file. Thus, if you want to analyze samples from multiple sample matrices you will have to manually switch config files between experiments. This will be updated for smoother operation.

**Step 9.** Enter acceptable limits of RT and ppm deviations for LaMas (For our instrumentation, a ppm error of 10-20 and 20-30 s for RT work well, although instrument specific knowledge is recommended).

**Step 10.** Enter an alpha value for the statistical tests (recommended not to go higher than 0.01 due to the amount of tests performed during monitoring), a ‘sleep time’ (defines how often the software will check if the file size of a raw file has changed; defaults to X s). Moreover, a minimum file size is required so that the software does not start monitoring unfinished files (also, when a raw data file is created, it might take some time before the file size starts to increase). This parameter must be smaller than the expected minimum final file size of any file generated by the instrument.

**Step 11.** Specify the raw data file suffix (e.g., ‘.d’ for Agilent), and a character string that specifies long term quality control samples (If you don’t use long term QCs, any random string will do) and the folder depth. Folder depth is defined as the number of folder levels below the top level (Step 1) where raw instrument files will actually be generated.

E.g., if you monitor the top level “Data” folder, you may want to specify folder depth 4, corresponding to experiment, chromatography, polarity and batch, where the raw data are actually stored on the instrument computer (Figure 2). However, the folder structure may differ between softwares and labs.



*Figure 2. Shows an example of a folder structure where folder depth should be equal to 4, where a) shows the folder that should be selected in Step 1, and b) demonstrates 4 folders downstream where the raw data files will be generated.*

**Step 12.** (Optional). Configure slack channels. *QualiMon* can be set up for delivering live updates of automated quality monitoring to your phone or other device. Please see the separate “Setting up slack channel” section.

**Step 13.** Select a folder and a file name for the config file and finally click the create file button to create your configuration file.

2022-08-31

Version 1.0

QualiMon

Tutorial

Setup

Find LaMas & New DB

Set-up Wizard

Find S&H limits

Monitor data

Review old data

Run batch

Step 1/13

Choose the folder where the instrument will generate raw files:

Choose folder

C:/Users/antonri/Desktop/ShinyTest/Input

Step 2/13

If wanted, a folder where raw files will be backed up:

Choose folder

☐ Skip backup

C:/Users/antonri/Desktop/ShinyTest/Backup

Step 3/13

3. A folder in which mzML files will be created:

Choose folder

C:/Users/antonri/Desktop/ShinyTest/mzML

Step 4/13

Select msconvert.exe:

Choose msconvert.exe

C:/Users/antonri/AppData/Local/Apps/ProteoWizard 3.0.21128.7376ae988 64-bit/msconvert.exe

Step 5/13

The DB to use for the analysis:

Choose DB file

C:/Users/antonri/Desktop/ShinyTest/data/TestDB.db

Step 6/13

6. Type in parts of filenames of files to be excluded from analysis. Make sure to separate every word by spaces (e.g. "blank cond"):

blank cond meoh

Step 7/13

Specify instrument type:

QTOF

Step 8/13

8. Specify sample matrix (e.g. "blood"):

plasma

Step 9/13

9. Input accepted PPM and RT deviation for landmarks:

Set dPPM

Set RT window (s)

5

30

Step 10/13

Input stat alpha value, interval of file size increase and min. file size check:

Set alpha value

Set sleep time

0,01

30

Set min. file size (mb)

20

Step 11/13

Folder depth for monitoring projects and minimum file size:

Set folder depth

File format

2

.d

Name of ITOC

Step 12/13

Slack settings:

☒

Channel 1 name (bad sample notif.):

#zoidberg-hardlimit

Format for channel input: #channelName

Channel 2 name (optional, full log):

#zoidberg-log

Slack token:

yourToken

Step 13/13

Choose name and folder for config file:

Name of config file

ConfigFile

Choose folder

Create file

C:/Users/antonri/Desktop/ShinyTest/data/ConfigFile.txt

Figure 3. Example of what the configuration file setup might look like once completed.

### Determination of soft and hard limits (mostly optional, but recommended)

Only one mandatory field is required to fill during this step (**Full status limit**), however, using at least some limits is highly recommended.

Once a configuration file has been generated, there is also the possibility to determine soft and hard limits. *QualiMon* uses a scoring system to summarize the results of all tests carried out on each sample to the user. For every metric a sample which differs significantly from previous samples will receive +1 for its status score. If soft and hard limits are setup and broken for a metric the soft limit will contribute +1 while the hard limit will contribute +2. The total status score is ultimately divided by the total potential score to get a scale ranging from 0 to 1, where a higher score means the sample is worse than a lower score. Soft and hard limits are also useful to avoid situations where relative quality tests will no longer detect a bad sample because too many bad samples are influencing the distributions used for the tests. These limits monitor either percentages of LaMas which have been determined to be outliers or absolute values of sample level metrics (e.g. TIC). If a sample would surpass these limits the status score would be increased a little, for the soft limit, or a lot, for the hard limit. It's also possible to setup a hard-limit slack channel which will inform the user over notification that a sample has broken a hard limit. Setting up soft and hard limits requires data processed through *QualiMon*. Please note that it is possible to use the same data that was used to determine the LaMas for this purpose: *QualiMon* incorporates the functionality to assess injections that have already been run (so-called batch-jobs under the tab '**Review old data**'; see the corresponding section in this tutorial).

To set up soft and hard limits, we need to extract data from the SQL database. This can easily be done within the R programming environment using the *QualiMon* function `FetchLMFeatureTable()`. An example on how to perform this for samples injected in reverse phase positive is shown below. If you instead wish to fetch a corresponding feature table for QCs, you can simply change the `sampleType` from 'sample' to 'QC' in the code below.

```
QualityInfo <- FetchLMFeatureTable("Path/To/SQLdbFile.db", mode="RP",
sampleType="sample")
```

The `QualityInfo` object will now contain information on all metrics and can be accessed using the '\$' operator.

Table 1. List of information retrieved using the `FetchLMFeatureTable` function.

Metric name	Content
\$intensity	Matrix. Intensities of every LaMa for every injection
\$RT	Matrix. Retention time of every LaMa for every injection
\$height	Matrix. Peak height of every LaMa for every injection
\$fwhm	Matrix. Full-width at half maximum of every LaMa for every injection
\$tf	Matrix. Tailing factor of every LaMa for every injection
\$sn	Matrix. Signal to noise ratio of every LaMa for every injections
\$noise	Matrix. Noise level of every LaMa for every injection
\$dataPoints	Matrix. Number of scans per LaMa for every injection
\$nLMMatch	Matrix. Number of potential matches for each LaMa in each injection (Used for functionality testing and to check LaMa stability over time)
\$IPO	Vector. IPO-score of every injection
\$n	Vector. Number of LaMas found per injection
\$nPeaks	Vector. Total number of peaks found per injection
\$TIC	Vector. Total Ion current per injection



\$name                      Vector. Name of each injection

For visualisations of the injection-based metrics, a sorted scatter plot and/or an empirical distribution function (eCDF) is recommended (examples shown in Fig 4).

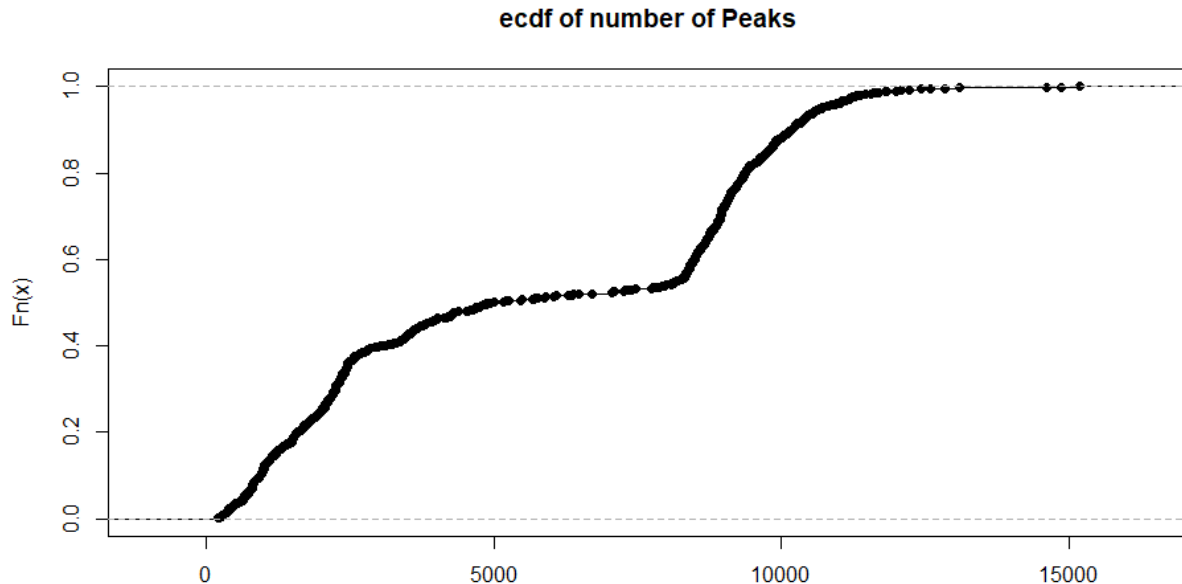


Figure 4. Empirical distribution function for the metric number of peaks.

From fig 4, we concluded that 2500 and 1000 were suitable soft and hard limits for this metric as they should represent outliers and severe outliers respectively based on a visual inspection of ‘knees’ at reasonably low proportions (in practice, one does not want a majority of injections to be below these limits). For the LaMa-based metrics, the number of outliers first needs to be calculated. This can be achieved via the `calcOutliers()` function available in the *QualiMon* R package.

```
RToutliers <- calcOutliers(QualityInfo$RT)
```

`plot(ecdf(RToutliers))` After calculation of number of outliers, a similar ecdf or scatter plot is recommended to find a suitable limit for each metric. Now that soft and hard limits are found, we can input them into the config file using the “Setup -> Find Soft & Hard Limits” button on the left-hand side menu in the shiny interface.

Please note that these limits do not have to be set-up using the data-driven approach described above, but can also be set based on user experience regarding the metrics in question. For instance, as the limits for the LaMa-based metrics are based on proportion of outliers, these can be set to e.g., 0.1 and 0.2 (10 and 20 %) respectively. If you want to use the relative tests, but not the soft limits for a metric, input 0 in the limit for this metric and the limit will not be part of the final scoring evaluation.

Now the set-up is complete and you are ready to start using *QualiMon* to monitor your injections!

## Day-to-day usage

To run *QualiMon*, you will need to name your injection according to a uniform naming strategy, so that *QualiMon* can scrape all the necessary meta-data about the file itself to process it. The file name structure looks like this:

“2017-10-05\_B1W40\_RP\_NEG\_CP01\_002.d”

“Date\_batch\_chromatography\_ionizationPolarity\_filename\_sequenceInjection”

*QualiMon* uses “\_” to split up the file name and obtain the metadata it needs to process and store the data from the file in the DB file. No fields are optional to include but are necessary for *QualiMon* to function properly. Below follows a comment of each section of the filename:

- Date: Should be formatted as above (YYYY-MM-DD)
- Batch: This information is not crucial for processing operations performed by *QualiMon*
- Chromatography: “RP” for reversed phase or “HILIC” for HILIC chromatography
- Ionization polarity: “NEG” for negative and “POS” for positive ionization
- File name: The sample identifier
- Sequence injection: The injection number of the sample within the sequence

The “.d” above is the raw file format produced by Agilent systems and may vary depending on manufacturer of your instrument (e.g. “.raw” for Thermo). Example of a filename is: 2022-08-22\_B18W34\_RP\_POS\_ID3128\_40.d.

*Sidenote: There are plans to implement a function for setting up your own file-naming strategy but at present only the one designed through our testing is available.*

To start monitoring, you should start the software the same way as above (launchQualiMon) and select the ‘**Monitoring**’ button on the left hand side menu. Load the configuration file for the relevant sample matrix and press ‘**Start monitoring**’. *QualiMon* is now actively searching for any new raw data files created and will process any such file automatically. As *QualiMon* is running and new injections are processed, the plots shown in the interface will be continuously updated.

On the left-hand side, you will see a box containing “triangle plots” (see Fig. 5 below). Apart from the status-plot, these plots represent LaMa-level data, displaying the proportion of LaMas that are considered outliers for the different metrics. The y-axis shows the injections, from the first (top) to the latest (bottom). The x-axis shows how injections are re-evaluated with the addition of new injections to the experiment: For each new injection the outlier status of every old sample is reevaluated against the new distribution which stems from the introduction of the latest injection. This is reflected in how the color of the injection changes to the right of the diagonal; the more yellow the color the more problematic the sample is. Since the distributions are constantly shifting a sample can start out being problematic but become completely normal after the introduction of more injections, or vice versa.

The various triangle plots available for evaluation are:

- **Status plot:** A sum score of all measured transgressions of a sample (+1 for each relative measurements in which the injection is an outlier, +1 for breaking the soft limit and +2 for breaking the hard limit) normalized against the number of tests performed (scale from 0-1).
- **Int(ensity) outlier plot:** The proportion of LaMas which have outlying peak areas

- **RT outlier plot:** The proportion of LaMas which have outlying RTs
- **Height outlier plot:** The proportion of LaMas which have outlying heights
- **FWHM outlier plot:** The proportion of LaMas which have outlying full-width-half-maximums
- **TF outlier plot:** The proportion of LaMas which have outlying tailing-factors
- **SN outlier plot:** The proportion of LaMas which have outlying signal-to-noise ratios
- **DataPoints outlier plot:** The proportion of LaMas which have outlying data points / peak
- **Noise outlier plot:** The proportion of LaMas which have outlying noise values

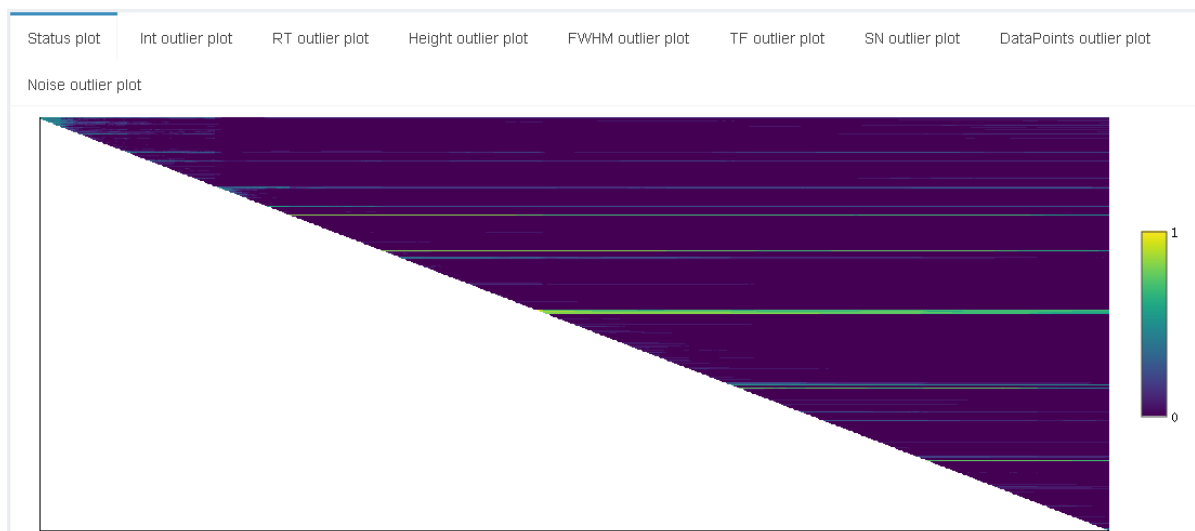


Figure 5. Example of a status plot where some of the samples (big greenish line in the middle) appear to systematically be of worse quality than the rest due to their retained color despite introduction of new injections for comparisons. There are also indications of samples being re-evaluated as more normal (lighter line turning darker towards the right).

On the right-hand side of you will see a box containing line plots (see Fig. 6 below). These plots represent injection-level data, evaluating top-level information from each injection and displaying them together with the respective soft and hard limits (if added; see above). The x-axis shows the injection sequence with the metric value on the y-axis. The plot thus shows how the injection-level metric changes over the batch injection sequences in the experiment. If an injection is below the orange line the sample has transgressed the soft limit, and if it's below the red line as well it has transgressed the hard limit as well.

The various line plots available for evaluation are:

- **n Peaks plot:** The number of peaks in the samples
- **IPO plot:** The IPO score of the samples
- **n Landmarks plot:** The number of LaMas found in the samples
- **TIC plot:** The TIC intensity of the samples

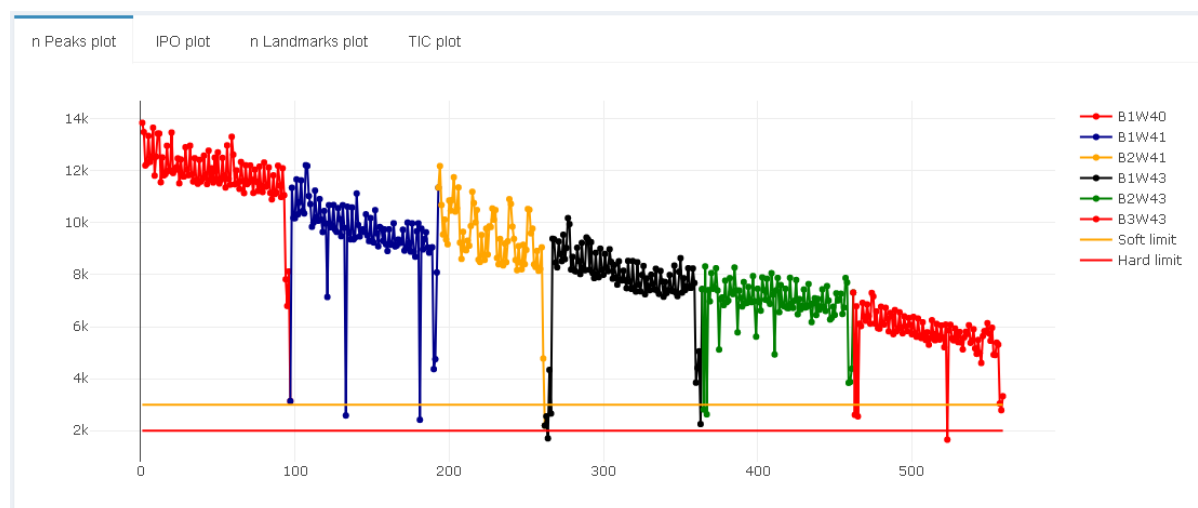


Figure 6. Example of *n Peaks* plot showing how many peaks were found in each sample during this injection sequence.

## Submit old data to DB

Reviewing old data is a functionality of *QualiMon* that allows the user to assess the quality of previous injections. This functionality can also be used to determine soft and hard limits (described in section “Determination of soft and hard limits”). To submit old data to *QualiMon*, injections from an experiment need to be submitted in a batch: Press the ‘**Batch job**’ button on the left side menu, then specify folders for mzML files and the config file and click the submit job button. This procedure can several hours, depending on how many samples are being checked and the processing power of the computer. Once this batch job is done, information about these injections will be submitted to the database file and plots can be reviewed through the ‘**Review old data**’-tab in the menu.

## Interacting with the DB file

Situations might arise where the user would like to interact with the DB, e.g. to update / change the LaMas saved for a specific chromPol mode or to remove a large chunk of bad samples which might have a negative effect on the evaluation of new injections. However, direct DB interactions are not yet incorporated into *QualiMon*, but there are ways of interacting with your DB files through third party software. [DBeaver](#) is an excellent tool which allows you to review and alter the content of an SQLite DB file. To interact with a file in DBeaver, click the “New database connection”-button (power plug) in the upper left corner. Choose “SQLite” and browse to the file you want to connect to. Once connected, the DB file will appear in the leftmost panel and can be expanded to view and alter the data within. Below are some examples of interactions a user might be interested in:

- **Removing / adding LaMas to your DB-file without running the LaMa optimization in *QualiMon***  
Open “Tables” and find the table called “landmarks”. In here you can add / remove any number of LaMas as long as you adhere to the naming and data formats used.
- **Removing all data pertaining to an injection which caused an unforeseen error**

First off determine the injection ID (injID) of the faulty injection. This is crucial information to locate the faulty / half-finished submitted data. The data to remove to assure that *QualiMon* starts functioning again are:

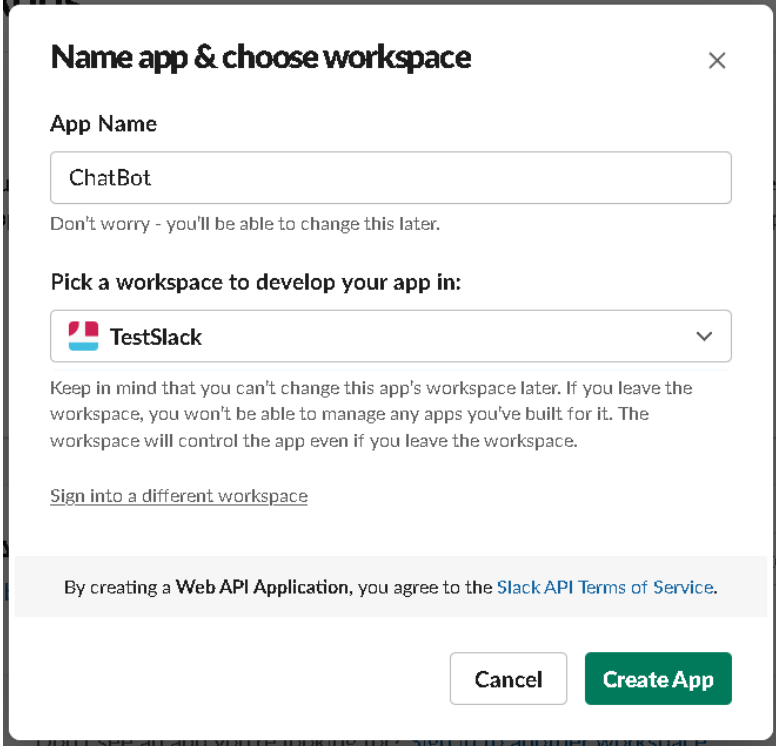
- The ImQuality table rows connected to the faulty injection
- The ImPeaks table rows connected to the faulty injection
- The injection row connected to the faulty injection

Interacting with any of the following tables is strongly discouraged (but might be necessary in case an error has occurred mid-submission of data): injections, ImPeaks, ImQuality, ms2spectra, peakMS2link, peaks, samples, projects. We encourage users to contact the authors if you suspect that there is something erroneous in these tables.

### Setting up a slack channel

How to setup slack with *QualiMon-LaMa*

1. Create a slack channel (workspace) (<https://slack.com/>)
2. Go to [api.slack.com](https://api.slack.com)
3. Your apps (Upper right corner)
4. Create new (From scratch)
5. Put the name of the bot which will send messages to your slack channel



The screenshot shows a modal dialog titled "Name app & choose workspace" with a close button (X) in the top right corner. Inside the dialog, there is a section "App Name" with a text input field containing "ChatBot". Below this input is a note: "Don't worry - you'll be able to change this later." The next section is "Pick a workspace to develop your app in:" followed by a dropdown menu showing "TestSlack" with a downward arrow. Below the dropdown is a warning: "Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace." There is a link "Sign into a different workspace" below the warning. At the bottom of the dialog, there is a grey bar with the text "By creating a Web API Application, you agree to the [Slack API Terms of Service](#)." and two buttons: "Cancel" and "Create App".

6. Click on your bot

**Your Apps** [Create New App](#)

**Warning:** If your app is (or will be) listed in the Slack App Directory, please review our [Slack App Directory Agreement](#). These terms are in addition to the existing Developer Policy, API TOS, and Brand Guidelines.

By keeping your app in the App Directory or review process, you're confirming your agreement to the Slack App Directory Agreement and to providing additional information for security review, if requested. If you don't agree with this Agreement, please send an email to [feedback@slack.com](mailto:feedback@slack.com), and we'll remove your app from the App Directory or the review process.

[I Agree](#)

Filter apps by name or workspace

App Name	Workspace	Distribution Status
ChatBot	TestSlack	Not distributed

7. Scroll-down to “App-Level Tokens” and click “Generate Token and Scopes”

## App-Level Tokens

App-level tokens allow your app to use platform features that apply to multiple (or all) installations—for example, the [API to list event authorizations](#). Features have distinct scopes, so request only the scopes for the features you need. Each app can have a maximum of 10 app-level tokens at one time.

Tokens	
Token Name	Scope
<a href="#">AccessToken</a>	connections:write

[Generate Token and Scopes](#)

8. Input name for your new token and choose the “connections:write” permission.

## Generate an app-level token ×

**Token Name**

TestToken

**Scopes to be accessed by this token**

Scope	Description
You haven't added any scopes for your app-level token.	
<div>Add permission by Scope or API method... <span>×</span></div> <div><div>connections:write Route your app's interactions and event payloads over WebSockets</div><div>authorizations:read View information about your app's authorizations on installed teams</div></div>	

Create

9. Click on your new token and you will find the token needed for the set-up wizard in *QualiMon LaMas*

## AccessToken ×

Properties of an app level token

<b>Generated By</b> parasitetwin	<b>Date Generated</b> June 14th, 2022
-------------------------------------	--

**Token**

xapp-1-A03KCDUBP7G-3662171524933-52d1 

Copy Revoke

**Scope**

connections:write	Route your app's interactions and event payloads over WebSockets
-------------------	--

Done

10. In the normal slack interface, create up to two channels which the bot can write to:
  - Log channel: Summary statistics for every injection will be posted here
  - Injection alert channel: Samples violating the user-specified threshold will be posted hereYou can choose any names you find suitable for these channels!  
These channel names are the names to put into the set-up wizard of *QualiMon*.
11. Once created, right click the channel to set its notification status. We recommend setting the injection alert channel notifications on while leaving the notifications for the log channel off.
12. Once input into *QualiMon*, any user invited to your slack channels will now be able to receive a notification on their smart-phones whenever a sample has broken the threshold specified in the config-file currently running in *QualiMon*.

## Basic Information

### Building Apps for Slack

Create an app that's just for your workspace (or build one that can be used by any workspace) by following the steps below.

#### Add features and functionality

Choose and configure the tools you'll need to create your app (or review all [our documentation](#)).

##### Building an internal app locally or behind a firewall?

To receive your app's payloads over a WebSockets connection, enable [Socket Mode](#) for your app.

##### Incoming Webhooks

Post messages from external sources into Slack.

##### Interactive Components

Add components like buttons and select menus to your app's interface, and create an interactive experience for users.

##### Slash Commands

Allow users to perform app actions by typing commands in Slack.

##### Event Subscriptions

Make it easy for your app to respond to activity in Slack.

##### Bots

Allow users to interact with your app through channels and conversations. 🤖

##### Permissions

Configure permissions to allow your app to interact with the Slack API.

13. Set up “incoming webhooks”  
Choose all channels where the bot is supposed to be able to write. They will each be assigned a specific “incoming\_webhook\_url”



## Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text": "Hello, World!"}'  
https://hooks.slack.com/services/T02HGPPBWL4/B02HWKGBCUS/H5G6N8Hp3NDcVQ3oDjAFS  
jph
```

[Copy](#)

Webhook URL	Channel	Added By
<a href="https://hooks.slack.com/services/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">https://hooks.slack.com/services/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</a> <a href="#">Copy</a>	#zoidberg-hardlimit	<a href="#">anton.ribbenstedt</a> Oct 14, 2021 
<a href="https://hooks.slack.com/services/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">https://hooks.slack.com/services/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</a> <a href="#">Copy</a>	#zoidberg-softlimit	<a href="#">anton.ribbenstedt</a> Oct 15, 2021 
<a href="https://hooks.slack.com/services/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">https://hooks.slack.com/services/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</a> <a href="#">Copy</a>	#fry-hardlimit	<a href="#">anton.ribbenstedt</a> Oct 15, 2021 
<a href="https://hooks.slack.com/services/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">https://hooks.slack.com/services/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</a> <a href="#">Copy</a>	#fry-softlimit	<a href="#">anton.ribbenstedt</a> Oct 15, 2021 

[Add New Webhook to Workspace](#)

## Known issues

Folder selection during Step 2-4 sometimes causes a bug where one must cancel and re-select a folder.

Currently, only the modes “RP” and “RN” are supported.

Currently not possible to update/change LaMas through QualiMon

Monitoring cannot be turned off if not the whole app is turned off

Soft/hard limits cannot be determined in software

Review old data (ExamineData) description

LaMa chromatograms description

Exchange Tutorial text with link to tutorial pdf