

# MARMoSET – Extracting Publication-ready Mass Spectrometry Metadata from RAW Files

## Authors

Marina Kiweler, Mario Looso, and Johannes Graumann

## Correspondence

johannes.graumann@mpi-bn.mpg.de; mario.looso@mpi-bn.mpg.de

## In Brief

MARMoSET facilitates the extraction of meta data from Thermo Fischer Scientific RAW files and in the context of data sets often comprising hundreds of files reduces it to the smallest common set, suitable for quality control, reporting and publication.

## Highlights

- Automated metadata extraction from potentially large sets of mass spectrometric raw data.
- Reduction of extracted metadata into groups of shared parameter sets.
- Tabular representation for quality control, reporting and publication.

## Graphical Abstract





# MARMoSET – Extracting Publication-ready Mass Spectrometry Metadata from RAW Files

Marina Kiweler‡,  Mario Loosolal¶¶, and  Johannes Graumann§¶¶

**In the context of publishing data sets acquired by mass spectrometry or works based on such molecular screens, metadata documenting the instrument settings are of central importance to the evaluation and reproduction of results. A single experiment may be linked to hundreds of data acquisitions, which are frequently stored in proprietary file formats. Together with community-, repository-, as well as publisher-specific reporting standards, this state of affairs frequently leads to manual – and thus error prone – metadata extraction and formatting. Data extracted from a single file also often stand in for an entire file set, implying a risk for unreported parameter divergence. To support quality control and data reporting, the C# application MARMoSET extracts and reduces publication relevant metadata from Thermo Fischer Scientific RAW files. It is integrated with an R package for easy reporting. The tool is expected to be particularly useful to high throughput environments such as service facilities with large project numbers and/or sizes. *Molecular & Cellular Proteomics* 18: 1700–1702, 2019. DOI: 10.1074/mcp.TIR119.001505.**

Aiming for evaluability and reproducibility of mass spectrometry based research and in parallel to the maturation of the field toward the acquisition of ever larger data sets, initiatives to standardize reporting of instrument settings and other relevant metadata have arisen in the community itself (1, 2), from the deposition requirements of public data repositories (3, 4), as well as have been launched by publishers and editors involved in the dissemination of mass spectrometric experiments (5, 6). The extraction and reporting of the metadata required remains a tedious process, especially given the facts that OMICS experiments frequently involve hundreds of data files and that data often resides in binary and/or proprietary file formats offering an excellent information to storage space ratio yet limiting ease of access.

One such example is the RAW file format produced by Thermo Fischer Scientific's (Bremen, Germany) mass spectrometers. Beyond the acquired spectral data, RAW files

also contain instrument settings as metadata, which are required to evaluate and reproduce the results.

An obvious and common approach to extract this data is to manually open individual RAW files using the vendor-specific Xcalibur software and copy the required information. The tediousness and error prone nature of manual interaction with the individual file, however, frequently leads to the extraction of metadata describing an entire data set from a single file. When the data set encompasses hundreds of files acquired over a potentially long period, this implies a potential for undetected parameter drift with implications for laboratory-internal quality control, reporting and publication. In this context core facility laboratories carry a particularly large burden, as the sheer number of projects they handle further compounds the data access problem. In combination with publication requiring metadata reporting often years removed from data delivery to customers, the difficulty to extract parameters frequently implies “data archeology” from deep archive.

To the best of our knowledge no software exists to date to address the need for both simple reporting from large numbers of RAW files and metadata reduction to a consensus set of parameters. Using the vendor-provided application programming interface (API)<sup>1</sup> RawFileReader (7) we create such a tool along with R (8) based infrastructure for the generation of tabular representations suitable for intra-laboratory quality control, data reporting, as well as supplemental material in publications.

## MATERIALS AND METHODS

**System Requirements**—The C# (9) command line tool MARMoSET is running as 64-bit code on Microsoft Windows only. It was compiled in Visual Studio Community (Version 15.8.7, .net 4.7.03056) (10) for the .NET Framework 4.6.1. The accompanying R package is agnostic with respect to operating system and only requires a functional R installation, as well as the package dependences assertive (11), jsonlite (12), pathological (13), Rlist (14), stringi (15), and magrittr (16).

**Implementation**—The C# (9) application MARMoSET ([https://github.molgen.mpg.de/loosolab/MARMoSET\\_C](https://github.molgen.mpg.de/loosolab/MARMoSET_C)) extracts publication relevant metadata from Thermo Fischer Scientific RAW files as a JSON (17) file.

From the ‡Bioinformatics Core Unit; §Scientific Service Group Biomolecular Mass Spectrometry, Max Planck Institute for Heart and Lung Research, W.G. Kerckhoff Institute, Ludwigstr. 43, Bad Nauheim, Germany; ¶¶The German Center for Cardiovascular Research (DZHK), Partner Site Rhine-Main

Received April 16, 2019, and in revised form, May 7, 2019

✂ Author's Choice—Final version open access under the terms of the Creative Commons CC-BY license.

Published, MCP Papers in Press, May 16, 2019, DOI 10.1074/mcp.TIR119.001505

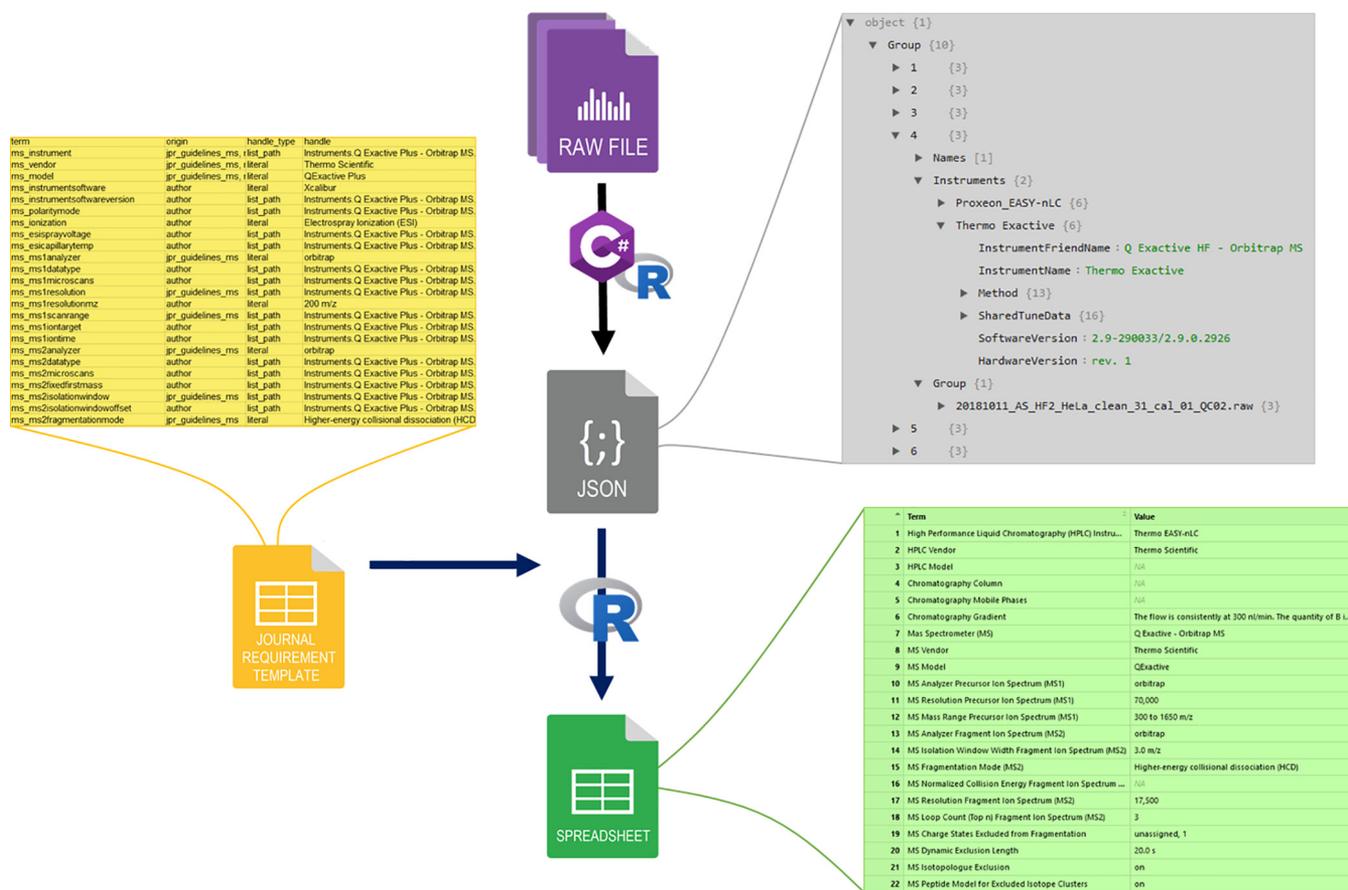


FIG. 1. Schematic representation of data processing by MARMoSET including excerpts of the data representations involved.

The RAW file format as accessed through the RawFileReader API provides multiple levels of metadata dependent on the system it was acquired on. A fixed header contains information like date, original filename, and sample information. The header is followed by a list containing instrument modules used and their respective methods as strings. The API additionally provides separate entry points for detector-associated data (such as ultraviolet spectrophotometry or mass spectrometry). MARMoSET currently only implements access to MS data, which beyond the acquired spectral data includes further instrument parameters and logs. Structure, string formatting, as well as location in the data structure of relevant metadata are specific for instrument classes. Using the “IRaw DataPlus” interface from the RawFileReader API, MARMoSET collects all relevant metadata from the divergent data structure.

In the context of liquid chromatography/mass spectrometry (LC/MS) using EASY-nLC ultra high-pressure liquid chromatography instruments (Thermo Fisher Scientific), LC parameters are available in the method strings and extracted and parsed by MARMoSET. Where chromatography instrument parameters beyond the EASY-nLC series are retrievable from the RAW file, future inclusion into the reporting infrastructure is expected to be straight forward and will be added as encountered.

Depending on whether it is provided with the path to a single RAW file or a directory, MARMoSET either acts on a single file or iterates over a collection of RAW files in a directory making use of parallelization dependent on the hardware resources available. In a first step, the metadata is gathered for each RAW file individually. In order to

reduce data from many files into a minimal set of parameters describing the entire collection, the resulting data structures are hash code evaluated and sorted in a dictionary. This information is then used to sort RAW files into groups that share all relevant parameters. Finally, a JSON file is written, comprising a minimal set of parameter groups linked to the corresponding RAW file names (see Fig. 1).

To provide easy and intuitive handling of the structured data in the JSON file, we provide an R (8) package also named MARMoSET (<https://github.molgen.mpg.de/loosolab/MARMoSET>). It is designed to create compact and clear tables following predefined journal requirements such as MIAPE or JPR. Moreover, it supports individual selections of parameter sets in a few easy steps.

On Microsoft Windows, the included function “generate\_json()” directly runs the C# command line tool from within R and captures its content into memory. Alternatively, externally generated JSON files may be read as well. In-memory data is reformatted by list flattening into a “data.frame” (function “flatten\_json()”). Based on a term-matching Table included in the package, the function “match\_terms()” extracts and arranges a subset of parameters from the total metadata set in the flattened JSON file for all given parameter groups and generates a table reduced to journal specific requirements. These tables may then be exported ready to upload in the form of tab delimited txt files and MS excel tables by using the “save\_all\_groups()” function. The included term-matching table can easily be adapted by the user to include further metadata entities or to design individual reporting styles. It is noteworthy that the same toolkit employed here has also been chosen by Trachsel et al (2018) (17) to facilitate analysis of spectrum-level metadata.

<sup>1</sup> The abbreviation used is: API, application programming interface.

*Exemplary Workflow (Windows)*—In a first step and from within a functional R environment (see <https://cloud.r-project.org/> for installation instructions), the MARMoSET R package is installed from the github repository using the package remotes (18), which may be achieved by calling “install.packages(“remotes”); remotes::install\_github(“loosolab/MARMoSET,” host = “github.molgen.mpg.de/api/v3”).” After loading MARMoSET with “library(MARMoSET),” a JSON object containing the metadata of grouped RAW files may be created by executing “json <- generate\_json(“<PATH-TO-RAW-FILES-DIRECTORY>”)” and prepared for downstream processing using “flat\_json <- flatten\_json(json),” A reporting guideline- (“MIAPE” in this example) and instrument-specific filter is generated by calling “term\_matching\_table <- create\_term\_matching\_table(instrument\_list = c(“Thermo EASY-nLC”, “Q Exactive - Orbitrap\_MS”), origin\_key = “miape”)” and applied to the JSON object using “vector\_of\_group\_tables <- match\_terms(flat\_json, term\_matching\_table).” Finally, a tab delimited text file representation may be saved using “save\_all\_groups(vector\_of\_group\_tables, output-path).”

Further use cases and more detailed instructions may be found on the top-level README page of the github repository (<https://github.com/molgen.mpg.de/loosolab/MARMoSET>), as well as through R’s help system (e.g. by calling “?generate\_json”) (19).

#### RESULTS AND DISCUSSION

A combination of an ever-increasing number of raw data files per mass spectrometry based experiment and a strong push for the standardized reporting of metadata for evaluation and reproducibility has rendered metadata extraction and its reduction into the smallest common parameter set a common need within the community. To the best of our knowledge, however, no tool simplifying metadata extraction currently exists. With the MARMoSET suite of tools presented here we fill that gap for mass spectrometric data acquired by Thermo Fisher Scientific’s instruments, combining outputs geared toward machine readability (JSON), as well as human consumption (tab delimited text and MS excel). The resulting information is suited for documentation and reporting, publication, as well as operations oversight and expected to be particularly helpful in the context of environments implying high throughput acquisition of mass spectrometric data, such as core facility laboratories.

In conclusion, MARMoSET offers tools for the simple extraction of metadata from RAW files. With the intend to particularly serve high throughput data acquisition environments, the tool enables the straightforward generation of small and clear tables containing just the metadata or parameter information needed. MARMoSET is designed for flexible adaption on individual laboratory’s needs.

|| To whom correspondence may be addressed. E-mail: johannes.graumann@mpi-bn.mpg.de.

‡‡ To whom correspondence may be addressed. E-mail: mario.looso@mpi-bn.mpg.de.

Author contributions: M.K. performed research; M.K., M.L., and J.G. wrote the paper; M.L. and J.G. designed research.

#### REFERENCES

1. Taylor, C. F., Binz, P.-A., Aebersold, R., Affolter, M., Barkovich, R., Deutsch, E. W., Horn, D. M., Hühmer, A., Kussmann, M., Lilley, K., Macht, M., Mann, M., Müller, D., Neubert, T. A., Nickson, J., Patterson, S. D., Raso, R., Resing, K., Seymour, S. L., Tsugita, A., Xenarios, I., Zeng, R., and Julian, Jr, R. K. (2008) Guidelines for reporting the use of mass spectrometry in proteomics. *Nat. Biotechnol.* **26**, 860
2. Taylor, C. F., Paton, N. W., Lilley, K. S., Binz, P.-A., Julian Jr, Jones, R. K. A. R., Zhu, W., Apweiler, R., Aebersold, R., Deutsch, E. W., Dunn, M. J., Heck, A. J. R., Leitner, A., Macht, M., Mann, M., Martens, L., Neubert, T. A., Patterson, S. D., Ping, P., Seymour, S. L., Souda, P., Tsugita, A., Vandekerckhove, J., Vondriska, T. M., Whitelegge, J. P., Wilkins, M. R., Xenarios, I., Yates, III, J. R., and Hermjakob, H. (2007) The minimum information about a proteomics experiment (MIAPE). *Nat. Biotechnol.* **25**, 887
3. Vizcaíno, J. A., Deutsch, E. W., Wang, R., Csordas, A., Reisinger, F., Ríos, D., Dianes, J. A., Sun, Z., Farrah, T., Bandeira, N., Binz, P.-A., Xenarios, I., Eisenacher, M., Mayer, G., Gatto, L., Campos, A., Chalkley, R. J., Kraus, H.-J., Albar, J. P., Martinez-Bartolomé, S., Apweiler, R., Omenn, G. S., Martens, L., Jones, A. R., and Hermjakob, H. (2014) ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* **32**, 223
4. Jones, P., and Côté, R. (2008) in *Functional Proteomics: Methods and Protocols*, Methods in Molecular Biology (Thompson, J. D., Ueffing, M., Schaeffer-Reiss, C., eds) pp. 287–303, Humana Press, Totowa, NJ
5. Bradshaw, R. A., Burlingame, A. L., Carr, S., and Aebersold, R. (2006) Reporting protein identification data: the next generation of guidelines. *Mol. Cell. Proteomics* **5**, 787–788
6. American Chemical Society (2019), *Guidelines for Reporting Proteomic Experiments Using Mass Spectrometry*, American Chemical Society, Washington, D.C.
7. Thermo Fisher :: Orbitrap :: RawFileReader <http://planetorbitrap.com/rawfilereader#.XEcJKCB7nmE> (accessed Jan 22, 2019).
8. RDevelopment Core Team. (2008) *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria
9. Microsoft Corporation ed. (2001) *Microsoft C# language specifications*, Microsoft Press, Redmond, WA
10. Visual Studio IDE *Visual Studio*, <https://visualstudio.microsoft.com/vs/> (accessed Jan 31, 2019).
11. Cotton, R. (2016) *assertive: Readable Check Functions to Ensure Code Integrity*
12. Ooms, J., Lang, D. T., and Hilaiel, L. (2018) *jsonlite: A Robust, High Performance JSON Parser and Generator for R*
13. Cotton, R., and Thyson, J. (2017) *pathological: Path Manipulation Utilities*
14. Ren, K. (2016) *rlist: A Toolbox for Non-Tabular Data Manipulation*
15. Gagolewski, M., and Tartanus, B. (2018) *stringi: Character String Processing Facilities*
16. Bache, S. M., and Wickham, H. (2014) *magrittr: A Forward-Pipe Operator for R* (R package version 1.5). <https://CRAN.R-project.org/package=magrittr>
17. Trachsel, C., Panse, C., Kockmann, T., Wolski, W. E., Grossmann, J., and Schlapbach, R. (2018) *rawDiag: An R Package Supporting Rational LC-MS Method Optimization for Bottom-up Proteomics*. *J. Proteome Res.* **17**, 2908–2914
18. Bray, T. (2017) *The JavaScript Object Notation (JSON) Data Interchange Format*
19. Hester, J., Csárdi, G., Wickham, H., Chang, W., Morgan, M., and Tenenbaum, D. (2019) *remotes: R Package Installation from Remote Repositories, Including ‘GitHub.’*