Full native timsTOF PASEF-enabled quantitative proteomics with the i2MassChroQ software package

Olivier Langella, Thomas Renne, Thierry Balliau, Marlène Davanture, Sven Brehmer, Michel Zivy, Mélisande Blein-Nicolas, Filippo Rusconi

Full native timsTOF PASEF-enabled quantitative proteomics with the *i2MassChroQ* software package

Olivier Langella,[*,†] Thomas Renne,[†] Thierry Balliau,[†] Marlène Davanture,[†] Sven Brehmer,[‡] Michel Zivy,[†] Mélisande Blein-Nicolas,[†] and Filippo Rusconi[*,†,¶]

†*GQE-Le Moulon, Université Paris-Saclay, INRAE, CNRS, AgroParisTech, IDEEV — 12, route 128 — Gif-sur-Yvette — F-91272 — France*

‡*Bruker software development — Bruker Daltonics GmbH & Co. KG — Bremen — D-28359 — Germany*

¶*INSERM, UMR-S 1138 — Centre de Recherche des Cordeliers — Paris — F-75005 — France*

E-mail: olivier.langella@universite-paris-saclay.fr; filippo.rusconi@universite-paris-saclay.fr

Abstract

Ion mobility mass spectrometry has become popular in proteomics lately, in particular because the Bruker timsTOF instruments have found significant adoption in proteomics facilities. The Bruker's implementation of the ion mobility dimension generates massive amounts of mass spectrometric data that require carefully-designed software both to extract meaningful information and to perform processing tasks at reasonable speed. In a historical move, the Bruker company decided to harness the skills of the scientific software development community by releasing to the public the timsTOF data file format specification. As a proteomics facility that has been developing Free

Open Source Software (FOSS) solutions since decades, we took advantage of this opportunity to implement the very first FOSS proteomics complete solution to natively read the timsTOF data, low-level process them, and explore them in an integrated quantitative proteomics software environment. We dubbed our software *i2MassChroQ* because it implements a (peptide)identification-(protein)inference-mass-chromatogram-quantification processing workflow. The software benchmarking results reported in this paper show that *i2MassChroQ* performed better than competing software on two critical characteristics: (1) feature extraction capability and (2) protein quantitative dynamic range. Altogether, *i2MassChroQ* yielded better quantified protein numbers, both in a technical replicate MS runs setting and in a differential protein abundance analysis setting.

Keywords

Mass spectrometry, DDA bottom-up proteomics, quantitative proteomics, Free Software

Introduction

Ion mobility mass spectrometry (IM-MS) has rather recently become popular in biology, with instruments providing a new orthogonal dimension to the separation of analytes that is independent of both their retention time and m/z ratio. Instruments capable of IM-MS in the field of structural biology have been commercially available since about 15 years. The different vendors have implemented ion mobility in their products in very different ways. The Synapt high definition mass spectrometer (HDMS) features a traveling wave-based IM-MS technology (TWIMS)[1,2]. This instrument proved highly valuable for the separation of medium-to-high mass biopolymers but failed to efficiently resolve peptides having homogeneous masses like the peptides obtained from tryptic digestion of proteins[3]. The IM-QTOF MS system from Agilent features a DC-only true uniform drift field tube interfaced with a

TOF analyzer[4] that makes it the best choice for scientists looking to perform direct collisional cross-section measurements for polypeptide or oligopeptide structural characterizations[5–7]. High-field asymmetric waveform ion mobility spectrometry (FAIMS) has long been described[8–10] and the Thermo Fisher Scientific vendor has implemented it as an ion source-based filtering scanning device in the hybrid LTQ-Orbitrap mass spectrometers[11] and more recently as an improved version in their Orbitrap Fusion Tribrid instruments[12]. The peptide ions that enter the device are subjected to low and high electric fields of different amplitudes and durations, which separates the ions according to their mobilities.

More recently, quantitative proteomics has benefited from the trapped ion mobility spectrometry (TIMS) implementation of IM-MS in the timsTOF line of instruments from Bruker. This implementation of IM-MS produces an inverted mobility separation where ions of greater collisional cross-section are released first, contrary to the conventional drift tube-based mobility separation. The trapped ions are released and subsequently fragmented to produce MS/MS spectra with a supplementary dimension to the retention time and m/z ratio of the precursor ion. A number of engineering feats allow the data acquisition to encompass almost the totality of the ions entering the instrument, making LC-MS data acquisitions particularly efficient in the context of complex protein hydrolysate samples separated within ultra-high resolution chromatography settings. The timsTOF Pro instrument from Bruker has gained important traction in the field of high-throughput proteomics[13]. A review describing the inner workings of the TIMS cell, instrument configurations and analytical applications was published recently[14].

During the acquisition, mass spectrometers convert the detected analog signals into processed digitized mass spectrometric data. The instrument vendors design their own mass data file formats in such a way that they can accomodate, at the maximum speed possible, specific signal data acquired on their hardware. The absence of an interoperative mass spectrometric data format has led the community of mass spectrometry scientists to ring an alarm that the failure to effectively exchange data would prevent data review, verifica-

tion and reuse across laboratories. A number of open mass data storage file formats have been designed over the years (see[15] for a minireview) and the currently most used one is mzML[16]. One portable widely used application software that can convert the mass data files from the proprietary formats to mzML is msConvert, from the ProteoWizard project[17]. This software uses the vendor-provided dynamic link libraries (files with the dll extension) to read the proprietary-format data and to convert them into the mzML format. In the case of the timsTOF instruments, the high-rate data acquisition and the high number of ion mobility slots (on average 600 slots, or bins) for each retention time acquisition unit produce a huge amount of mass spectra that are difficult to process. As a matter of fact, conversion of Bruker `.d` raw format data files into the very verbose mzML standard format using msConvert produces huge files that are almost unusable for proteomics projects (one 600 Mb binary file is converted into a file that is more than 55 Gb in size). In the case of the Bruker timsTOF instrument, conversion might not be necessary because, in a historic move, Bruker provided developers with a detailed specification of their data format so as to let them develop software to natively handle mass data acquired on these instruments. Another algorithmic-intensive evolution in the field of quantitative proteomics has been the paradigm shift in the last fifteen years involving the switch from spectral counting to area-under-the-curve determination as the basis for quantitation assessment. This shift deeply modified the quantitative proteomics algorithmic requirements, that evolved from the relatively simple spectral counting algorithms to much more sophisticated mass data signal processing and extraction and mass data integration algorithms. Getting unrestricted access to the mass spectrometric data is, in this respect, particularly useful to craft highly performant software solutions for quantitative proteomics.

The `.d` raw format mainly consists of two files. The first file is a SQLite3 relational database that contains a number of tables storing a large amount of mass spectrometric acquisition metadata describing the actual mass data located in the second file. The database file can be scrutinized with a software program like *DB Browser for SQLite* (`https://`

`sqlitebrowser.org/`). The second file is a binary non-perusable file that contains only the densely packed mass spectrometric data according to a structure indeed described in the Bruker specification. Being able to access the actual mass data in the binary file requires the previous interrogation of the database file for the binary address of the desired mass data in the binary data file.

To the best of our knowledge, most proteomics software programs supporting the timsTOF data format do so using the closed-source proprietary dynamic link library provided by the Bruker company[18,19]. The OpenTIMS software package provides a C++ library for accessing the data and two R and Python modules to perform data visualization[20]. While OpenTIMS will be of use to explore timsTOF data, it is not geared towards proteomics projects. Indeed, we found that essential functionality in this respect is missing from the C++ library, in particular the extraction and the merge of ion mobility scans, potentially spanning multiple frames, that correspond to precursor ions that have been fragmented and that yielded MS/MS spectra used for the identification of the peptides. That functionality appears to be available in the GNU-R and Python modules of the OpenTims project, but there are speed concerns related to the fact that these languages are interpreted.

We set out to actually make use of the detailed mass data format specification from Bruker to write a new native mass spectrometric data importer—without need for the Bruker proprietary library—that is tailored specifically for high performance in proteomics research projects.

We present *i2MassChroQ*, a full-featured data dependent analysis (DDA) quantitative proteomics desktop software environment that is not only fully timsTOF-PASEF data-capable but that also handles any MS run data set previously converted to either the mzML or the mzXML standard format. *i2MassChroQ* is a full rewrite in the C++ language of the *X!TandemPipeline*[21] software project originally written in the Java language. In addition to the new Bruker timsTOF native data support, the rewrite brought enormously enhanced performance and the implementation of numerous new features. Some of the new features in-

clude a number of quality assessment procedures, very much improved protein quantification results by harnessing the statistical *MSstats* GNU R package[22]. *i2MassChroQ* comprises two main graphical user interface modules that interoperate, (1) one module reads the proteomics LC-MS/MS data and performs both the peptide identification and the protein inference and (2) one module performs protein quantification.

Experimental Section

Software

Operating system compatibility, software license, and availability

The software is written in portable C++17 using the highly regarded cross-platform Qt Free Software libraries[1]. The software is thus fully cross-platform and is compatible with MS Windows (tested versions: 7 and 10) and Linux (development platform: Debian GNU/Linux version 12). All the software and the documentation provided with it are licensed under the GNU GPLv3+ Free Software license. The software is thus entirely free to use, modify and redistribute, without any restriction whatsoever, provided the license is conserved.

All the development is performed fully in the open with the source code freely available in the Git repository located at `https://forgemia.inra.fr/pappso/i2masschroq`. The *libpappsomspp* library required for building *i2MassChroQ* is available at `https://forgemia.inra.fr/pappso/pappsomspp/`. Binary Debian/Ubuntu GNU Linux packages and MS Windows setup executables are prepared and made available at `http://pappso.inrae.fr/en/bioinfo/i2masschroq/download/`.

Of note is the thorough *i2MassChroQ* documentation, provided as a heavily-illustrated user manual, either in the package, on-line (HTML and PDF formats) or as Supplementary Material S2 (PDF format).

[1] `https://www.qt.io/product/qt6`.

System memory management

While *i2MassChroQ* is fully parallelized thanks to the high-performance QThread and Qt-Concurrent application programming interfaces from the Qt threading libraries, the user is free to configure the number of threads to be used for the calculations. Whenever the temporary data generated during the calculations overrun the system's random access memory (RAM), the user can set a configuration bit to store these data on the hard disk at a defined location. This kind of situation is typically encountered in metaproteomics projects where the amount of peak data to be handled is so large that it does not fit in the RAM.

Software package architecture and requirements

i2MassChroQ is a multi-component software solution. At its basis is one shared library (GUI and non-GUI shared code). This library's features are used by two modules: the main component (formerly called *X!TandemPipeline*[21]) handles all the user-facing processes like data loading, peptide identification, and protein inference; the secondary component (formerly called *MassChroq*[23]) reads the output from the first component and derives area-under-the-curve quantifications for the identified peptides. Of note is the fact that all the code that handles the reading of the Bruker timsTOF native data is in the library so that it is freely available to any developer wishing to develop software specifically for this kind of data.

Performance benchmarking environment, data sets and statistical data processing

Software versions The software performance comparisons were carried over using the latest *i2MassChroQ* version and the latest *MSFragger* software package available (*MSFragger* 4.0, *Philosopher* 5.1.0, *IonQuant* 1.10.12, *FragPipe* 21.1). The configuration of the *MSFragger*, *FragPipe* and *X!Tandem* software programs is provided in Supplementary Material S3.

Mass spectrometry data sets Two data sets were used to perform the software performance comparisons, both obtained by running timsTOF-based LC-MS/MS acquisitions. The first data set (ProteomeXchange identification PXD010012) comprises four technical replicate MS runs of a protein extract from cultured HeLa cells[24]. A previous work did use this data set to describe the *MaxQuant* software metrics in relation to timsTOF data acquisitions[18]. The authors of the *MSFragger* software[19] did the same for software performance comparisons between their software and *MaxQuant*. The second data set (ProteomeXchange identification PXD014777) was used to perform comparisons of the software capabilities along the outline set in Yu et al. (2020). Briefly, the data set comprises MS data obtained by performing three technical replicate LC-MS/MS runs of two different samples A and B, obtained by mixing different proportions of individual single-organism protein extracts (*Saccharomyces cerevisiae*, *Homo sapiens*, or *Escherichia coli*). Sample A contained a protein mix made of the following relative proportions of protein extract amounts from these organisms (same order, 2:1:1) while sample B contained relative proportions 1:1:4. Even if the absolute amounts of individual proteins in both samples are not known, the relative protein amount differences in the samples provide a defined "ground truth" for quantitative proteomics software performance assessment.[25] In this report, we used the same two datasets described above to characterize the features of the *i2MassChroQ* software in relation to the *MSFragger* competing software.

Protein databases, search engines and results data availability The protein database searches were performed using Human, *Escherichia coli* and *Saccharomyces cerevisiae* protein databases generated according to instructions published[19] using the Uniprot proteome IDs UP000005640, UP000000625 and UP000002311, respectively. The databases were downloaded on the fourth of february 2023.

The number of protein sequences in these databases were 4400, 6060, 20389 for the *E.coli*, *S. cerevisiae* and *H. sapiens*, respectively. The contaminants database (46 sequences) was

created alongside these databases. *i2MassChroQ* uses X!Tandem[26] as the default mass search engine. The protein quantifications were performed using the *MSstats* package for GNU R[22]. This software is supported by *MSFragger* and *i2MassChroQ*. In all the protein quantification results provided in this report, the contaminant proteins were removed. The ion abundance data that *MSFragger* and *i2MassChroQ* produce as a first step of the quantitative proteomics data analysis workflow thus need to be of the same format to be fed to *MSstats*. We used *MSstats* as a protein quantification software program so as to be able to carry out reliable and unbiased comparisons of the protein quantification performance of the *i2MassChroQ* software with that of *MSFragger*. The *MSFragger*/*IonQuant* software was configured in the exact same way as described in Yu et al. (2020). The configuration files are provided for convenience as Supplementary Material S3. The *i2MassChroQ* software was configured as described in the *X!Tandem* preset file also provided in Supplementary Material S3.

All the data output by both the *MSFragger* and *i2MassChroQ* software programs in the course of the experiments described in this report are available in a repository accessible via the DOI:10.57745/PWLP4R, as described in detail in section "MSFragger and i2MassChroQ output data availability for the community" in Supplementary Material S1.

Protein identification-specific settings In any peptide or protein identification and quantification computation, two stringency factors might be configured. The first factor is the false-discovery-rate (FDR) that is applied as a metric to filter the identification either of peptide-spectrum matches (PSM FDR) or of proteins (protein FDR). In this report, we do test the effects of different FDR settings on the quantified peptide or protein yields and our setting reads thus like so: either FDR $< 0.5\,\%$ (more stringent) or FDR $< 1\,\%$ (less stringent), each time with the associated peptide or protein qualification.

In order to evaluate the false positive matches in target identifications, we set up an entrapment decoy experiment using a protein database (Uniprot proteome ID UP000001013) containing 2044 protein sequences from the *Pyrococcus furiosus* (Pfu) ATCC 43587 strain's

genome[27].

The second factor is the minimum number of MS1 precursor ions required to quantify a protein. We will refer to this factor as the Ions = 1 or the Ions = 2 setting, meaning that a protein is considered quantified if at least 1 ion (less stringent) or 2 ions (more stringent) were used for its quantification.

Results and discussion

i2MassChroQ features a native timsTOF data importer

Decoding timsTOF data The file format specification provided by Bruker for their timsTOF mass data file format has allowed us to implement a native data reader from scratch. Each mass spectrometric run data acquisition yields data that come separated into two files located in the same `.d` directory. The `analysis.tdf` file is an easily explorable standard SQLite3 relational database file that holds, in a number of related tables, all the metadata describing the digitized mass data that are packed in the `analysis.tdf_bin` file for which the file format specification by Bruker was most required because it is a non-perusable binary file. The decoding of the data in that file and the elaboration of a specific data model allowing fast access to the four dimensions—retention time, $(\mathrm{m/z}, \mathrm{intensity})$, $1/\mathrm{K_0}$ inverse mobility and charge—were the main aspects of our development of the timsTOF data reading and management software, as described below.

During a LC-MS/MS run acquisition, the mass spectrometer-associated software produces a set of mass spectra for each retention time (RT). At any given RT, the mass spectrometer first accumulates ions from the source and then resolves them into an ion mobility cell that can accomodate as many as one thousand mobility slots (or bins). The ions in each one of these ion mobility slots are sequentially released as ion packets that are either analyzed in the time-of-flight (TOF) analyzer as a full scan analysis (MS1 spectra) or fragmented; in the latter case, the product ions are then TOF-analyzed and written to MS2

spectra. Both these MS1 and MS2 spectra are ion mobility–mass spectrometry spectra (IM-MS spectra). The full set of IM-MS spectra acquired at any given RT (either for an MS or an MS/MS analysis) is called a frame and each individual IM-MS spectrum is called a scan. In a full LC-MS/MS proteomics data acquisition, there are thus as many frames as there are retention time points and the frames are either of the MS1 (no fragmentation) or of the MS2 (MS/MS analysis) kind. All of the frames are identified in the mass spectral data by an ordinal index (from 0 to n-1 along the LC-MS/MS run data acquisition). Each frame is stored in the binary file as a succession of mass spectral data elements corresponding to the IM-MS scans it is made of. The IM-MS scans are packed in the file in the order they were acquired during the IM-MS analysis.

During the mass data analysis, the data processing software needs to arbitrarily access mass data in the form of either frames or scans. In order to access any given frame, it is necessary to know its binary offset in the binary file, that is, the position of the header that documents it relative to the beginning of the file itself. The binary offset of each data frame in the binary data file is provided by the SQLite3 `analysis.tdf` database. As mentioned earlier, each frame, at any given RT, is actually made of a sequence of IM-MS scans. For each frame, the number of IM-MS scans is described in the database file, because that number is not necessarily constant over the whole LC-MS/MS run data acquisition. One distinct characteristic of the way mass spectral data are packed in the binary file is that these data are first encoded and then compressed before being written to that file. In order to access any IM-MS scan inside a given frame at a given index, it is thus necessary to first access the frame at the binary offset stored in the database for that specific index, then decompress the data and decode them into a series of IM-MS scans. The position of each scan in any given frame is provided in a header section at the beginning of that frame. The binary structure described above ensures that the huge amount of mass spectrometric data acquired during a LC-MS/MS run is densely packed in the binary data file. Handling the decompression and decoding of the data along with the processing of the raw mass spectral data into meaningful

proteomics data is thus the responsibility of the software that uses these data.

For each frame, the database file provides, amongst numerous other metadata, the m/z calibration and the IM calibration. The inverse mobility ($1/K_0$) is calculated using the scan number and the IM calibration. The IM calibration is actually described using a set of eleven parameters that are documented in the database file and that might change from frame to frame.

Accessing the actual $(\mathrm{m/z}, \mathrm{intensity})$ pairs is the most challenging task performed by the data processing software, from both the mathematical and speed points of view. Indeed, each mass spectrum in the binary file, after having undergone decompression and decoding of its data, is represented by two vectors of integer values: one vector describes the TOF index value (one sort of arrival time index) of each ion in the spectrum and the other vector describes the corresponding intensity of the ion signal. Converting the integer TOF index value of each ion into a corresponding floating point m/z value requires solving a third-order polynomial equation. This computation is resource- and time-intensive, which makes the handling of timsTOF data highly challenging. We have written our own solver in order to gain in performance. Further, a number of optimizations in the data processing have considerably shortened the processing time; in particular, we try to defer the conversion of the TOF index values to the corresponding m/z values to the very last step of the data processing, after having dealt only with integer values for as long as possible. This design choice is justified by the fact that computer processors do handle integers in a vastly more efficient manner than floating point values.

Of note is the fact that *i2MassChroQ* does seamlessly accomodate two different compression formats in the Bruker timsTOF data files, one that was used in earlier versions of the instrument and a more recent one.

Fast ion current extractions for quantitative proteomics The typical LC-MS/MS data processing in the context of shotgun (bottom-up) proteomics data acquisitions relies

on the association between MS/MS spectra and the (m/z, intensity) pair of the precursor ions that were fragmented. In the absence of any ion mobility separation, that relation is straightforward: at a given RT, an ion of a given m/z value is fragmented and for that same RT, the MS/MS spectrum is related to the precursor ion's m/z value. In the case of timsTOF proteomics data, the situation is considerably more complex because of two main factors: 1) the Bruker's ion selection algorithm might select a precursor ion for fragmentation more than once if the signal obtained for the first fragmentation is considered too low and that ion may thus be found in different frames; 2) a given precursor ion is typically found in as many as 20 IM-MS scans, in which case the scans need to be combined to account for that specific precusor ion's intensity. These two factors impose a specific processing of the IM-MS scans where each precursor is found and the application of data filters to the reconstructed MS/MS spectra. In particular, when multiple scans need to be combined (that is, when intensities for a given m/z value found in different scans need to be summed), we elected to perform that intensity summation using the TOF index value instead of converting that value into a corresponding m/z value. The conversion of the TOF index value into a corresponding m/z value is only performed on the final spectrum obtained by combination of all the relevant spectra containinig the precursor ion.

The algorithmic developments aimed at making the most out of the binary data files were in great part empirical (in the absence of specific recommendations from Bruker), and their effectiveness was thoroughly tested on a set of proteomics LC-MS/MS run acquisition files. However, in order for the developer user to maintain a total control on the data processing, these signal post-processing steps are configurable at the application programming interface level. The post-processing is based on the sequential application of filters (see section "Filter-based processing of the MS/MS spectra" in Supplementary Material S1).

The area-under-the-curve quantitative proteomics paradigm is based on the idea that the amount of a given identified protein is a reflection of the intensities of all the identified peptidic precursor ions that led to its identification. The algorithmic process that leads

to that quantification is based on extracted ion currents (XICs) for these precursor ions in the MS1 data. In this case also, our XIC extraction code performs all the possible processing steps using TOF index values instead of corresponding m/z values. Indeed, for a XIC computation, that conversion is not required, since a XIC is a relation between an ion current intensity, a retention time and the corresponding ion mobility. XIC extractions are thus particularly fast in our implementation.

Two-factor MS data alignment enhances the match-between-run processing One of the recurring problems in DDA bottom-up proteomics is the so-called "missing value" problem that is encountered when a peptide that has successfully been identified in a given replicate MS run acquisition is missing from another. This absence is due to the fact that the program that drives the mass spectrometer selects, at each full scan, a configurable number of most intense ions for subsequent fragmentation (so-called top N concept). If, in another replicate MS run, the same peptide ion is of lower intensity, it might be skipped from the fragmentation step and will therefore be missing from the MS run data set. One data rescue strategy that has been devised in bottom-up proteomics software to mitigate this difficulty is known as "match-between-run" (MBR). MBR consists, for a given MS run in which the ion was not fragmented, in the extraction of the ion current for that specific ion at the RT, (m/z , z) and ion mobility values observed in at least another MS run. If a XIC chromatogram peak is detected at the right RT and if the ion current extraction is successful, the corresponding cell in the matrix is filled with the found area-under-the-curve intensity value. Therefore, the MBR rescue process imposes a reliable alignment of retention time values and, in the case of timsTOF data, also of the ion mobility values. Indeed, we had to deal with a timsTOF data set in which a large number of peptide (m/z,z) pairs from distinct MS run replicate acquisitions were detected in significantly different ion mobility scans. In order to be able to find in all the MS runs a given peptide (m/z,z) pair in the right ion mobility scan, we elaborated a specific algorithm for the alignment of IM-MS spectra on

the ion mobility dimension. The algorithm first creates, for each MS run acquisition pair, a vector listing, for each encountered (m/z,z) pair, the ion mobility scan number difference. After having created one such vector for each one of the combinations of MS run pairs, the corresponding sorted median value is stored in a MS run correspondence matrix which is then used to look up the right scan number for any given XIC operation involving a given peptide (m/z,z) pair that was not fragmented in a specific MS run.

Converting timsTOF data to mzXML While using timsTOF data in *i2MassChroQ* is a totally transparent process from the user perspective and works exactly the same as when using mzXML- or mzML-formatted mass spectrometry data files, we elected to provide the user with the possibility to convert the timsTOF-formatted data into mzXML-formatted data files for use with other database search engines. mzXML data files are accepted in almost all proteomics software. When using timsTOF data in *i2MassChroQ*, that conversion to mzXML is performed under the hood because the *X!Tandem*[28] database search engine used by *i2MassChroQ* relies on this format (see below).

Overall bottom-up proteomics data processing logic

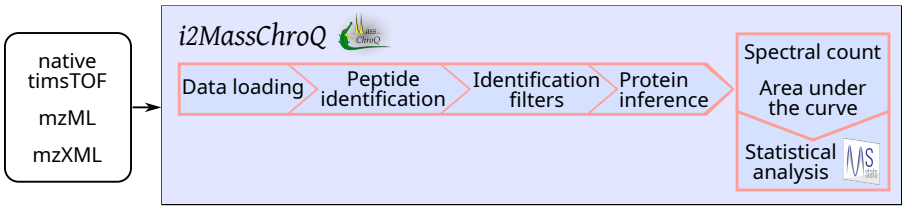


Figure 1: **Overview of the *i2MassChroQ* software program's operation.** The data are fed to the program in a number of formats. The *X!Tandem* database search engine performs the peptide identification and *i2MassChroQ* applies user dynamically defined filters to tune the protein inference process. Statistical analysis of identified and quantified protein data is performed using the *MSstats* software[22].

Loading of the data As in any proteomics software piece, the workflow is rooted in the mass spectrometry data that are fed to the program (Figure 1). *i2MassChroQ* is compatible

with the timsTOF mass spectrometry data format (using our native data reader described earlier) and with the MGF, mzXML and mzML data formats that are supported by the ProteoWizard[17] project (in these latter cases, the data loading step is actually performed by that project's libpwiz library).

Peptide identification Peptide identification is performed by the *X!Tandem* database search engine with second-stage refinement search enabled[28]. *i2MassChroQ* features an *X!Tandem* configuration graphical user (GUI) interface to help the user with the configuration intricacies (fully described in the user manual). We list the refinement parameters, provide explanations on the refinement process and detail the reasons we need refinement in the section "X!Tandem refinement" of Supplementary Material S1. The software is shipped with preset configuration files for the most commonly used instruments (Orbitrap and timsTOF). *i2MassChroQ* can load identification data produced by other database search engines like *Mascot*, *MSFragger*, *SEQUEST* or any other software that outputs identification data in the following formats: mzIdentML, pepXML, and Mascot dat.

Once the peptide identification process has completed, or the identification data have been loaded, the results might be filtered according to one of these two criteria: the E-value or the false discovery rate (FDR). In *i2MassChroQ*, the FDR is computed as the ratio between the number of PSMs matching the decoy database over the number of PSMs matching the target database: $FDR = \frac{\#decoy}{\#target}$[29]. The configuration of the data filtering is detailed in the user manual.

Protein inference Protein inference is the concept that drives the identification of a protein on the basis of a given subset of identified peptides. The implementation of that concept in *i2MassChroQ* is based on the Occam's razor, as thoroughly described by Langella et al. (2017) and assessed by Bossche et al. (2021). Here also, the protein identification list might be filtered using the two criteria mentioned above for the filtering of the peptide identification results.

Protein quantification The state-of-the-art peptide and protein quantification method nowadays is the area-under-the-curve integration method that is based on the extraction of precursor ions' current intensities (XIC) for all the peptide ions having provided a usable sequence by undergoing a successful gas phase fragmentation. The XIC-based peptide quantification method is implemented in the secondary component of *i2MassChroQ* that receives its input from the main component in a seamless manner via a file that contains a detailed description of both the validated peptide-spectrum matches (PSMs) and the protein identities, along with metadata relating the precursor ions to the data acquisition files in which their fragmentation occurred. This web of informational metadata is required so as to be able to compute XIC chromatograms for all the precursor ions by looking into the original acquisition data files. The match-between-run (MBR) concept is implemented and is configurable. In particular, the most important parameter is the retention time alignment of the MS run datasets, as described in Valot et al. (2011).

The area-under-the-curve quantification method takes as input both the peptidic data having led to the identification of any given protein and the MBR-obtained data. The protein quantification is implemented in *i2MassChroQ* using the *MSstats* GNU R-based software program[22] that processes the peptide/intensity list, ultimately yielding the protein quantification data.

Prior to providing these data to *MSstats*, however, *i2MassChroQ* pre-processes them according to the following. First, all of the ions that are found matching multiple proteins are discarded. Second, a filtering operation of the peptide/intensity list involves the removal of any ion that 1) appears to be too distant between runs in the RT dimension, 2) is found under too large an RT peak.

Assessment of the *i2MassChroQ* software performance

In order to be able to perform unbiased performance comparisons of the *i2MassChroQ* software with that of other proteomics software solutions, we elected to refer to the article by Yu

et al. (2020). In that work, the authors described their *MSFragger*/*IonQuant* software package and benchmarked it against two other commonly used software packages, *MaxQuant*[18] and *PEAKS*[31]. In the present report, we will compare the *i2MassChroQ* software characteristics to those of the *MSFragger*/*IonQuant* package because the *MaxQuant* software package was outperformed by *MSFragger*/*IonQuant* and we have no *PEAKS* copy to run (*PEAKS* is commercial non freely available software). To this end we used the same proteomics acquisition data sets from ProteomeXchange[32] with entry identifications PXD010012[13] and PXD014777[18] (see Experimental Section).

i2MassChroQ increases data completeness across technical replicates

The PXD010012 data set (four replicate MS run acquisitions performed on the same HeLa protein extract tryptic digest sample) was used as in Yu et al. (2020) to perform a standard quality assessment of both *MSFragger* and *i2MassChroQ*. Performing detailed and fair comparisons of the features and performance of different software pieces is a very difficult task because of imperfection of common metrics. One such metric is the FDR, which is set hereinafter to 1 % (or to 0.5 % occasionally). One question that often arises about FDR in the literature is that it only is an estimation value. Strategies have been devised to measure the accuracy of proteomics workflows. One such strategy is the so-called entrapment-decoy method that consists in the addition to the initial target protein sequences of a new set of protein sequences from an organism that is evolutionarily divergent from the target organism(s) of interest[27]. This addition creates a new composite target database that is used as usual in the remaining proteomics process. The final results thus contain protein identification and quantification data stemming from matches that occurred either in the sequences from the organism(s) of interest or in the sequences from the new divergent one.

We ran the entrapment-decoy experiment using a composite protein sequence database containing the 20389 human protein sequences, 48 contaminants sequences and 2044 *Pyrococcus furiosus* organism sequences (Pfu, see Experimental Section). The size of this composite

database was then doubled by the addition of computationally-generated reversed sequences. By setting the FDR to 1 % for both *MSFragger* and *i2MassChroQ*, we determined the number of detected features in the human protein sequences (respectively 504312 and 518864) and in the Pfu protein sequences (respectively 20 and 44). Likewise, the quantified proteins were determined (6880 human, 5 for Pfu for *MSFragger* and 6731 human, 2 for Pfu for *i2MassChroQ*) thus showing that both programs quantify only a few Pfu proteins, which confirms that the FDR estimation is performed reliably in both programs.

***i2MassChroQ* has the best feature extraction capabilities** We focused on the capabilities of both software programs in terms of 1) the number of quantified peptides and proteins; 2) the ability to extract the maximum amount of features as measured by the data completeness percentage; 3) the reproducibility of the feature extraction processes as standard coefficient of variation (CV) metrics.

Table 1: **Overview of *MSFragger* and *i2MassChroQ* performance.** The performance of both software programs is detailed using quantified peptides and proteins numbers, as these data are the usual endpoint of quantitative proteomics data analyses (FDR $< 1\,\%$). The data are provided for four replicate MS runs of the same sample[19]. Ions: minimal number of peptidic ions required for the quantification of a protein; ICX: number of ion current extractions; Data comp. (%): data completeness computed as the percentage of the features (ion current extractions or quantified proteins) that are effectively found in all of the four replicates; All: quantified peptides or quantified proteins that could be found in all of the four replicate MS run data files; CV: median values of the CV distributions of the protein abundance between the four replicates. The data in the table are extracted from the matrices that are fed by both tested programs to the *MSstats* software package[22].

Software	Ions	Features		Peptides		Proteins			CV
		ICX	Data comp. (%)	Quantified	All	Quantified	Data comp. (%)	All	
MSFragger	1	476045	94.0	101508	88167	6778	99.6	6665	4.00
	2	473741	94.0	100856	87686	6217	99.9	6184	3.71
i2MassChroQ	1	515773	98.4	111695	106730	6733	100	6718	3.89
	2	514765	98.4	111434	106493	6483	100	6481	3.76

The results of the standard quality assessment are displayed in Table 1, in which we elected to report only data for quantified peptides and proteins. These quantification data are yielded by the *MSstats* software package on the basis of input data matrices provided by both the *MSFragger* and *i2MassChroQ* programs. These input matrices are thus the

best data to be used for reliable unbiased comparisons of the feature extraction capabilities of these two programs. The table shows that *i2MassChroQ* carries out ≈8 % more ion current extractions (ICX header under Features) than *MSFragger* (ion current extraction is the basis of peptide area-under-the-curve quantification and is a hard requirement for subsequent protein quantification). This result is confirmed by the excess quantified peptides by *i2MassChroQ* with respect to *MSFragger*. As expected, if two peptidic ions (Ions = 2 setting) are required for the quantification of a protein, the numbers are slightly inferior to the corresponding ones obtained for the Ions = 1 setting.

The data completeness ratios (presented as a percentage in the table) computed both for ion current extractions and for quantified proteins are seldomly found in proteomics software benchmarks. We do report these values here because of their peculiar interest: the data completeness values are a direct reflection of the feature extraction capabilities of the software. The table shows that the data completeness value for ion current extractions is 94 % for *MSFragger* while it reaches more than 98 % for *i2MassChroQ*. This observation highlights better overall MBR-related capabilities for the *i2MassChroQ* software. This difference reflects on the number of peptides that were quantified in any or in all of the replicate MS runs. For *MSFragger*, the number of peptides quantified in all of the replicate MS runs is 13 % lower than that of the peptides quantified in any of the replicates (Ions = 2; 87686 vs 100856). For *i2MassChroQ*, that difference is only of 4 % (106493 vs 111434). These observations are not transposable to the protein quantification data, with both programs performing almost identically in terms of protein data completeness, which indicates both that *MSstats* does a good job at completing the missing data from *MSFragger* within incomplete features and that, overall, the features are eventually included in the quantification because the missing values are not preponderant (specific setting in *MSstats*).

Another interesting observation is that, at the protein level, the number of quantified proteins that are lost upon switching conditions from Ions = 1 to Ions = 2 in the case of *MSFragger* is more than two times greater than for *i2MassChroQ* (loss of 561 and 250 proteins

respectively). This observation is explained by the fact that *MSFragger* quantifies more proteins than *i2MassChroQ* on the basis of a single peptide ion, which corroborates the observation made above about the greater ability of *i2MassChroQ* to extract a more complete feature set.

The tabular data described above are represented in Figure 2, panel A. The Venn diagrams show the overlap of the peptides (left) and proteins (right) quantified using either *i2MassChroQ* or *MSFragger* (peptide and protein FDR $< 1\,\%$). The proportion of common peptides (quantified by both programs) is significantly lower than the proportion of common proteins ($62\,\%$ vs $87\,\%$). This reflects the fact that, although both programs quantify a different set of peptides, the protein inference process yields essentially the same set of quantified proteins, with only $4\,\%$ and $8\,\%$ of proteins quantified specifically by *MSFragger* and *i2MassChroQ*, respectively. Figure 2, panel B shows the distribution of the CV of the protein quantification in the four technical replicates with two different protein quantification settings (Ions $= 1$ and Ions $= 2$). In both cases, both software programs did perform equivalently with very similar CV distribution profiles, as shown on the graphs.

Overall, these results show that both programs do reliably quantify proteins from multiple technical replicate MS run data sets, with a slightly better performance of *i2MassChroQ* in terms of numbers of quantified proteins in all the samples (All header under Proteins), thanks to both better ICX numbers and greater feature completeness.

***i2MassChroQ* performs highly sensitive reproducible protein quantifications** In order to further characterize the behavior of *i2MassChroQ* in the protein quantification reproducibility over the four technical replicate MS runs, we performed a pairwise correlation analysis between these four MS run data sets.

The reproducibility of the protein quantification process in *i2MassChroQ* has been scrutinized using the same four technical replicates described above and the results were compared to those obtained by running *MSFragger* on the same data set. The plots shown in Figure 3

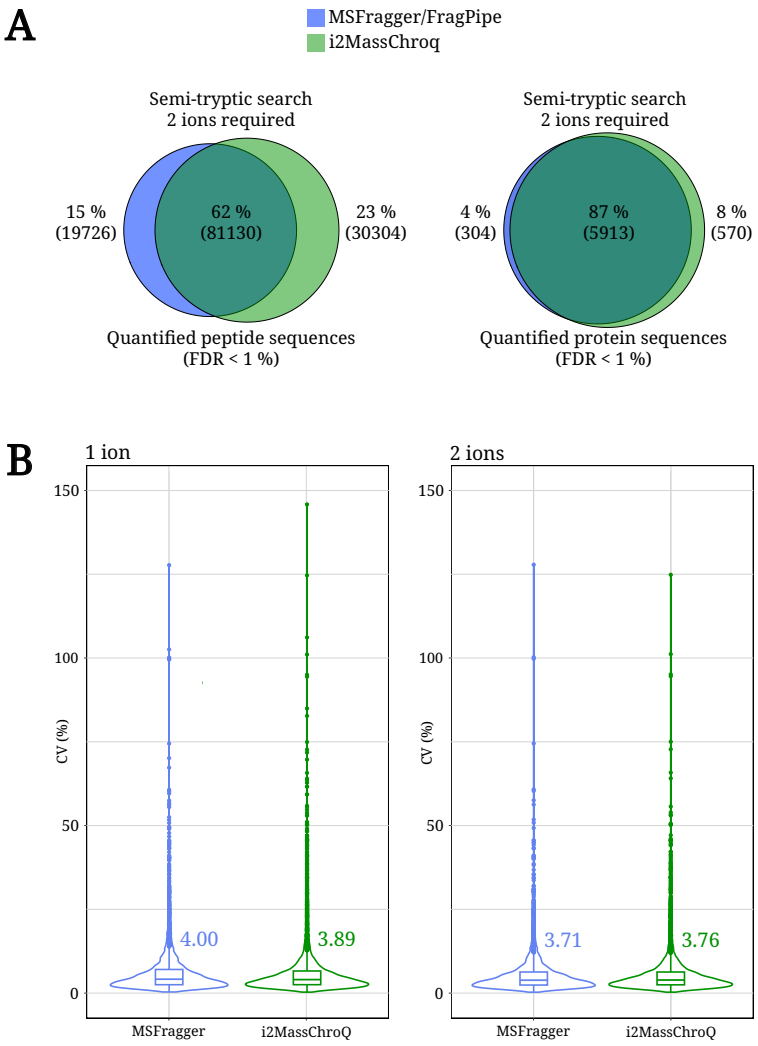


Figure 2: **Overall peptide and protein quantification performance of the *i2MassChroQ* and *MSFragger* software packages.** Four data sets acquired with four technical replicates (see Materials and Methods) were analysed using either *MSFragger* (blue color) or *i2MassChroQ* (green color). (A) The number of total quantified peptides (left) and proteins (right) found using either *i2MassChroQ* or *MSFragger* are reported in overlapping circles. The overlap area is proportional to the quantified features found by both programs. (B) Plots showing the distribution of CV for the protein quantification by both *i2MassChroQ* and *MSFragger* programs. Data are shown for two protein quantification stringency conditions: a protein might be quantified using at least one or two quantified ions (left, right, respectively).

demonstrate that both programs did perform almost identically, with inter-replicate protein quantification Pearson correlations greater or equal to 0.97. This observation is consistent with the fact that both programs reproducibly quantify an almost identical number of pro-

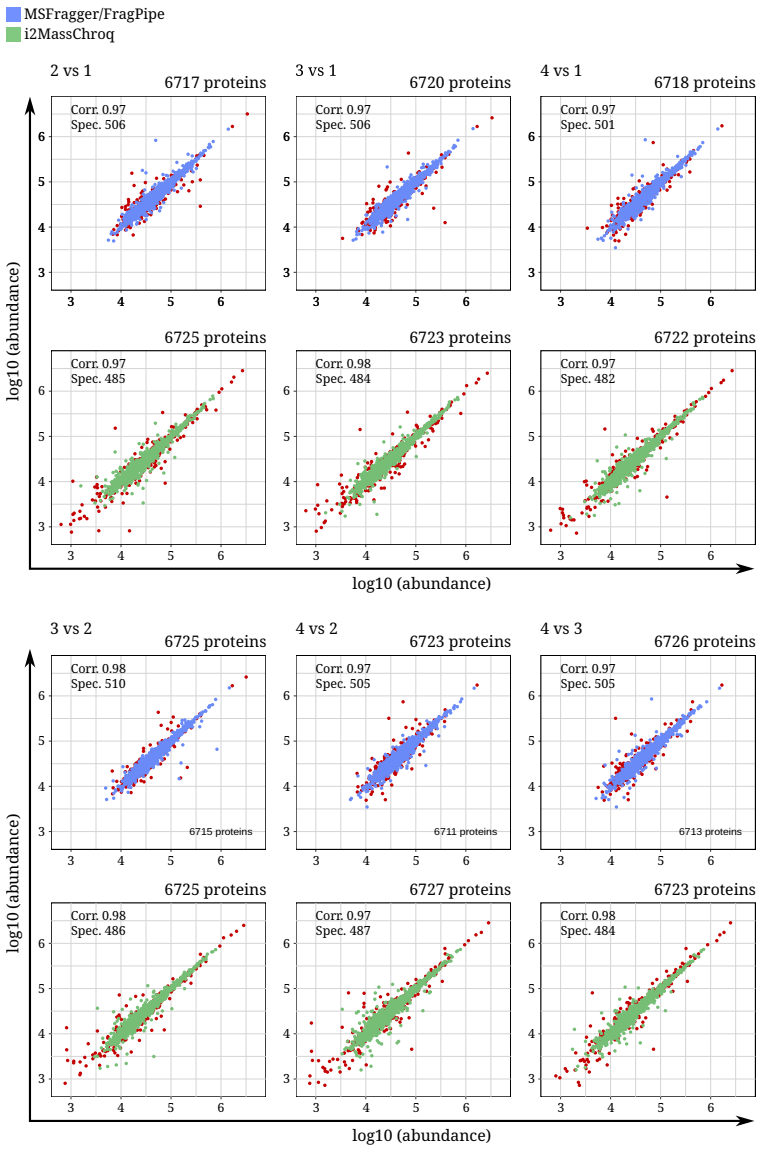


Figure 3: **Assessment of the protein quantification reproducibility.** LC-MS/MS data from four technical replicate MS runs of an HeLa protein extract sample [numbered 1 to 4; (Ions = 1), see Experimental Section] were analysed using either *MSFragger* or *i2MassChroQ*. The six vertically paired plots (top, blue for *MSFragger*; bottom, green for *i2MassChroQ*) display the correlation of quantified proteins between the four technical replicates. The replicates involved in each one of the six paired comparisons are labelled at the top left corner of each plot pair. The correlation value was determined using the Pearson correlation test. For each graph, the number of proteins specifically quantified by each software is indicated. These proteins are shown in red.

teins in the pairwise comparisons, as visible on the number of quantified proteins reported at the top of each plot. It is worthwhile noting that *i2MassChroQ* appears to explore the

protein space with a larger dynamic range than *MSFragger* (one log10 range of the quantification space), as visible on the plots for less abundant proteins, which might be explained by the fact that the *i2MassChroQ* protein quantification process is based on better data completeness in the four technical replicates compared to *MSFragger* (see Table 1; both Ions = 1 and Ions = 2 stringency settings; ICX data completeness 98.4 % vs 94 % respectively).

i2MassChroQ performs well in a differential protein abundance analysis setting

We used mass spectrometric data obtained on two different samples based on tryptic digests of protein extracts from three different sources: *Saccharomyces cerevisiae*, *Homo sapiens* (HeLa cells), and *Escherichia coli* (as described by Prianichnikov et al., 2020). Sample A contained digested proteins from these three sources at amount ratios of 2, 1 and 1, respectively while sample B contained digested proteins at amount ratios of 1, 1 and 4, respectively. We used the data in that publication (ProteomeXchange identification PXD014777) to assess the capabilities of *i2MassChroQ* and to reliably compare them to those of *MSFragger*, along the lines delineated in Yu et al. (2020).

Table 2: **Quantitative proteomics capabilities of *MSFragger* and *i2MassChroQ*.** The data are provided for two main conditions, the FDR threshold and the minimal number of ions required for a protein quantification (Ions). The Features header comprises data for the number of ion current extractions (ICX) and for the data completeness measurement over the six MS run data sets. Data under the Peptides header are the total number of quantified peptide sequences and the number of quantified peptides sequences found in all six MS run data sets. Data under the Proteins header are the total number of quantified proteins, the data completeness measurement over the six MS run data sets and the number of quantified proteins found in all six MS run data sets. The Sample header comprises data for the number of proteins quantified both in sample A and in sample B, and the median value of the distributions of the CVs of these samples restricted to proteins common to both software output.

Software	FDR	Ions	Features		Peptides		Proteins			Sample (A *vs* B)		
			ICX	Data comp. (%)	Quantified	All	Quantified	Data comp. (%)	All	Quantified A + B	CV A	CV B
MSFragger	< 1 %	1	513282	85.8	88218	59489	8430	95.1	7140	8200	4.4	4.0
		2	507791	85.1	86768	59086	7388	97.5	6737	7307	4.1	3.8
i2MassChroQ	< 1 %	1	655543	96.6	103336	94656	8345	99.9	8308	8343	4.4	4.4
		2	652862	96.6	102873	94234	7893	100	7888	7893	4.2	4.2
	< 0.5 %	1	564126	97.8	86718	81672	8028	99.9	7983	8024	4.4	4.5
		2	560769	97.8	86137	81147	7461	100	7458	7460	4.1	4.3

***i2MassChroQ* achieves full data completeness at the protein level** Table 2 reports the metrics of both the *i2MassChroQ* and *MSFragger* software programs in the context of quantitative proteomics. The data show that *i2MassChroQ* is more capable than *MSFragger* at extracting features from the MS run data sets, which results in a sizable increase of the number of ion current extractions (ICX under Features) that are, in turn, the basis of the whole bottom-up proteomics data analysis process leading to lists of reliably quantified proteins. This observation correlates well with the fact that *i2MassChroQ* does provide a fuller matrix of features (amongst which the precursor ion's m/z, z, and intensity values, for example) to the *MSstats* software package, as evidenced by the data completeness values under Features (97 % for *i2MassChroQ vs* 86 % for *MSFragger*). The data completeness metrics show that *i2MassChroQ* performs particularly well in the match-between-run process, which might be explained, on the one hand, by the fact that our Bruker timsTOF native data loader performs better than the vendor's DLL and, on the other hand, by the fact that we, as mentioned earlier, investigated deeply the problematics of the alignment of both the retention time values and the ion mobility values.

Overall, the number of quantified proteins in all the MS runs (All under Proteins) was measurably greater using *i2MassChroQ* than *MSFragger*, with *i2MassChroQ* quantifying 1168 or 843 more proteins than *MSFragger* at FDR $< 1\,\%$ or $0.5\,\%$, respectively.

***i2MassChroQ* provides *MSstats* with a dramatically fuller feature data matrix** The tabular data described above are represented in greater detail in Figure 4. Three observations can be made: (1) that *MSFragger* seems to provide quantification results with a less ample dynamic range, as evidenced on the scatter plots that do extend to a smaller range of x-axis values, when compared to *i2MassChroQ*, (2) that increasing the protein quantification stringency from setting Ions $= 1$ to Ions $= 2$ (figure rows) reduced the number of quantified proteins more considerably for *MSFragger* (loss of 893 proteins) than for *i2MassChroQ* (loss of only 450 or 564 proteins, at FDR $< 1\,\%$ or $0.5\,\%$, respectively), and (3) that the box plots

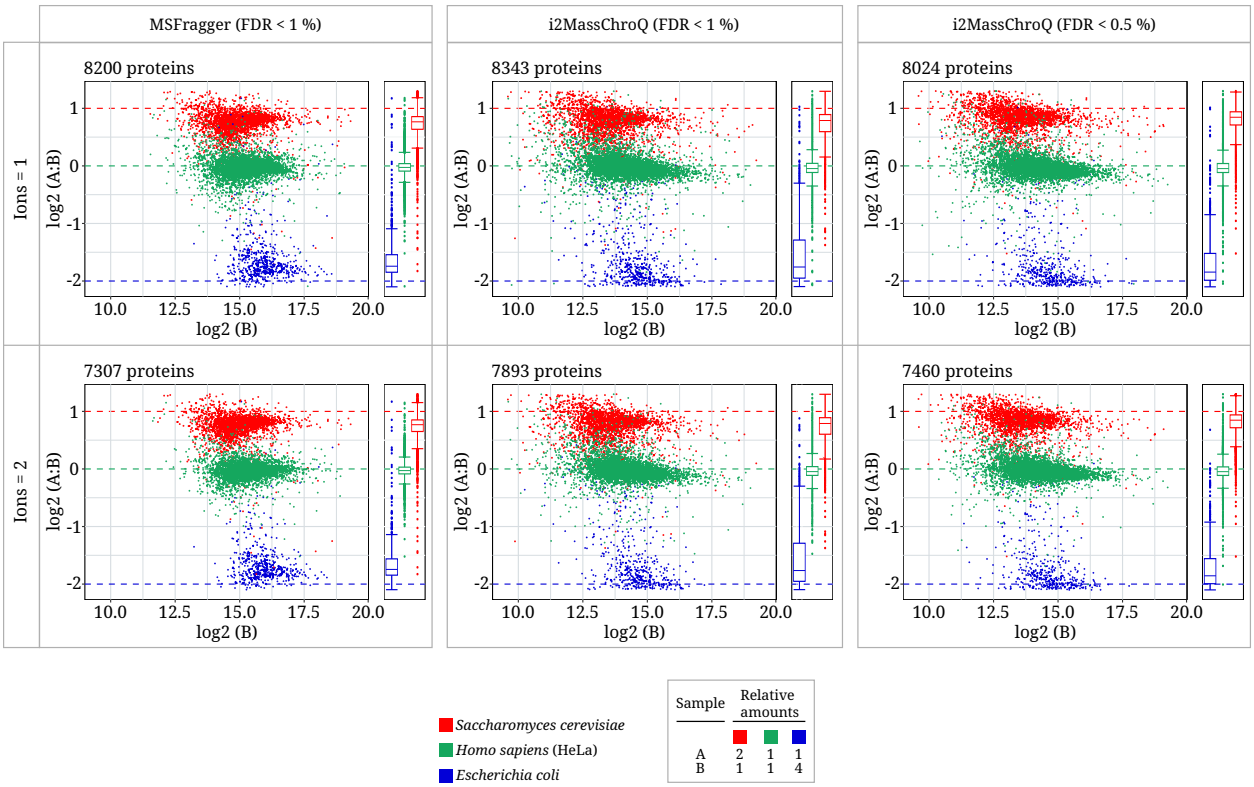


Figure 4: **Comparison of the quantitative proteomics capabilities of *i2MassChroQ* and *MSFragger***

The data sets obtained for the two samples A and B (three replicates each) were analysed by both *MSFragger* (first column; FDR < 1 %) and *i2MassChroQ* (last two columns; FDR < 1 % and FDR < 0.5 %). The plots are arranged in two rows (Ions = 1, top row and Ions = 2, bottom row). The ordinates correspond to the log2 of the protein's ratio of intensities in the sample A vs the sample B. The protein quantifications were obtained using the *MSstats* GNU R package with the Group quantification setting. The colored dashed lines indicate the expected log2(A:B) value for the different organisms. The box plots on the right of each scatter plot show the distribution of the intensities for each organism. The plot colors correspond to the three organisms: red for *Saccharomyces cerevisiae*, green for *Homo sapiens* and blue for *Escherichia coli*.

right of each scatter plot show that *MSFragger* produces log2(A:B) ratio values less dispersed along the y axis than *i2MassChroQ*.

The third observation above might indicate that *MSFragger* better assesses the protein abundance ratios for samples A and B. Nonetheless, when running *i2MassChroQ* with raised FDR stringency (PSM and protein FDR < 0.5 % instead of 1 %; respectively last and middle columns of the figure), the abundance ratios were brought closer to those of *MSFragger* albeit

at the cost of reducing the number of quantified proteins. Increasing the FDR stringency implies reducing the number of retained features at the identification step, with the immediate effect of increasing data completeness (as visible on Table 2, column Features/Data comp (%)), thus leading to the quantification of less proteins but with better reliability (reduced value dispersion). Note that the greatest discrepancy between *MSFragger* and *i2MassChroQ* with respect to the quantification values dispersion, is observed for the *E. coli* proteins which are present in the samples at the most challenging ratio (1:4, A:B). This quantitation-oriented challenge is clearly visible on the plots as a shift to the right of the blue scatter plots, with respect to the green and red ones, because the quantitation ratios could only be computed for the most abundant proteins in sample B (greater log2(B) values, x axis), correlating with a similarly greater abundance of their counterpart in sample A.

The observations above prompted us to compare the incompleteness of the feature data matrix that is provided by both the *MSFragger* and *i2MassChroQ* programs to the *MSstats* GNU R software package for it to fulfill its protein quantification task. Figure 5, panel A shows that the software programs have starkly different capabilities to perform ion current extractions. Indeed, *i2MassChroQ* has a consistent proportion of missing values (around 2-4 %, irrespective of the sample), while *MSFragger* has proportions of missing values ranging from 9 % to as much as 45 % depending on the sample (samples A and B differ in the relative amount of proteins from different organisms). A detailed look at the results shows that *MSFragger* is less capable than *i2MassChroQ* in the complete filling of the feature data matrix when the proteins in the sample are of lesser abundance. Sample A (left column) contains a larger proportion of yeast proteins compared to both the human and bacterial ones. For this sample, one can observe that *MSFragger* fills the data matrix much better for the features identified as the relatively abundant yeast proteins (9 % of missing values) than for the relatively less abundant bacterial ones (45 % of missing values). Conversely, sample B (right column) is made of much more abundant bacterial proteins, with respect to the proteins from the other two organisms. In this case, the results somehow mirror those

obtained for sample A, with *MSFragger* failing to fill a third of the feature data matrix for yeast proteins (missing values raise to 30 % for sample B from 9 % for sample A) while the proportion of missing values for the bacterial proteins is much lower (missing values drop from 45 % for sample A to 8 % for sample B).

The whole set of ICX values obtained for all the identified proteins in all the MS runs (three replicates of samples A and B) are gathered as so-called features in a feature data matrix that is fed to the *MSstats* software. *MSstats* uses that matrix to perform the protein quantification, and tries to make the best possible use of all the available features (`featureSubset` configuration bit set to "all"). We therefore asked how the missing ICX values in the feature data matrix did impact the ability of *MSstats* to perform reliable protein quantifications.

Figure 5, panel B shows that *MSstats* performed 28122 protein quantification events in the whole set of six MS run acquisitions (58 %) on the basis of a complete feature set (green stack bar; 100 % protein feature completeness) for data originating from *MSFragger* while that number reaches 40745 (81 %) for data originating in *i2MassChroQ*. Remarkably, *MSstats* had almost no need to recourse to a less-than-half-complete feature set to quantify proteins in the case of *i2MassChroQ* data (0.4 %, 216 proteins) while it had to recourse to such an incomplete feature set for 6 % of all the protein quantifications in the case of *MSFragger* data (red stack bar, <50 % protein feature completeness; 2737 proteins).

We next asked how the missing features observed in the feature data matrices output by *i2MassChroQ* and *MSFragger* and fed to *MSstats* (red and orange stacks in panel B of Figure 5) do translate into the distribution shape of the feature-complete quantified proteins as a function of the number of features that led to their quantification. Figure 5, panel C shows this distribution. One striking observation is that the number of proteins quantified with one feature only is more than twice in the *MSFragger* results than those in the *i2MassChroQ* results (1370 vs 577, respectively). In the remainder of the histogram, the excess proteins quantified with 100 % feature completeness in the *i2MassChroQ* output is distributed along

the x axis with systematic greater numbers than those of *MSFragger*.

Overall, these results show that, while *i2MassChroQ* can reliably quantify proteins in highly contrasted samples [protein amount-wise, there is a factor of 8 between the yeast:bacterial proteins ratios in samples A (2:1) and B (1:4)], *MSFragger* struggles to fill in the feature data matrix for the least abundant proteins in each sample.

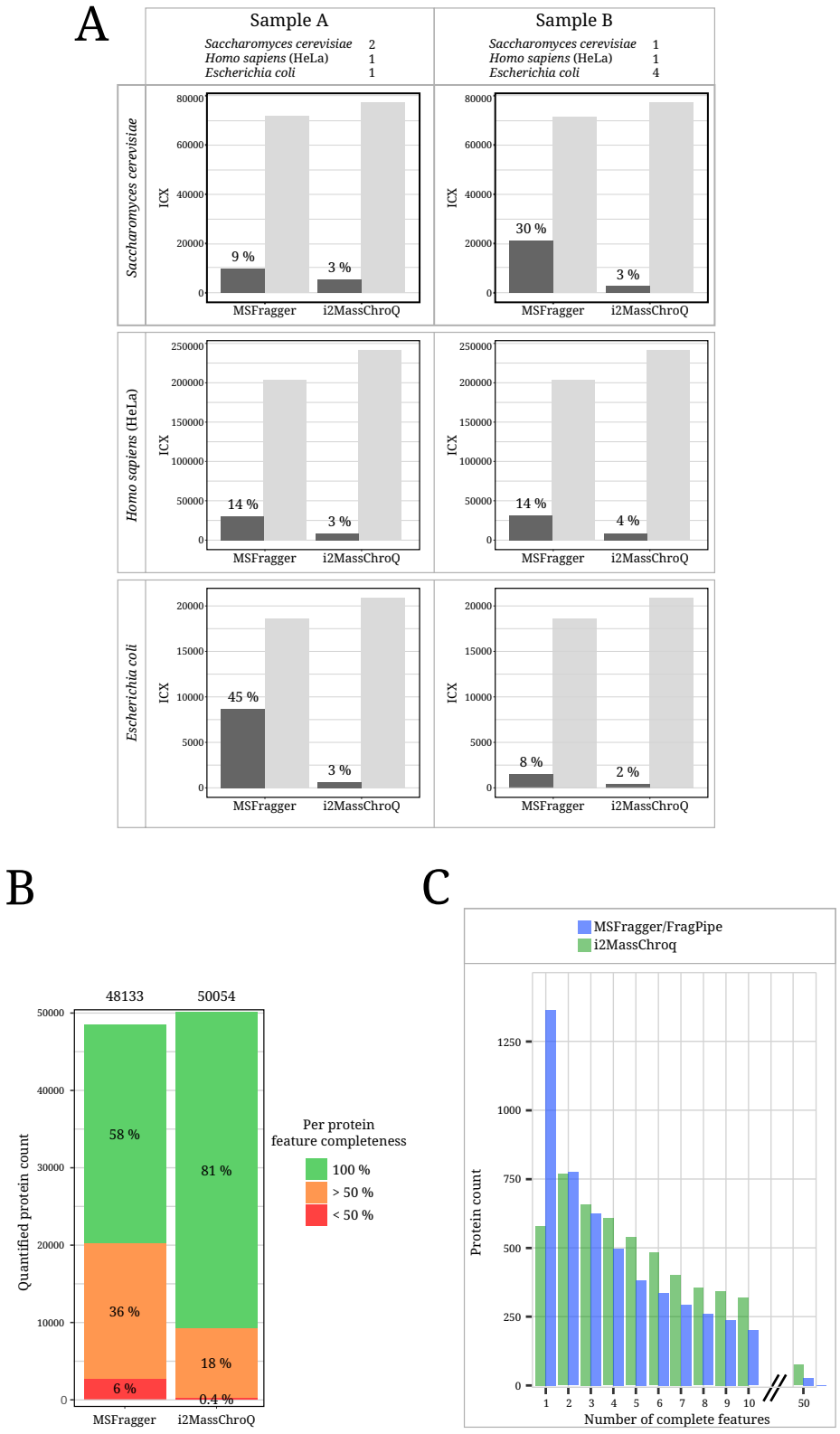


Figure 5: **Comparison of the proportion of missing values in the feature matrices produced by *MSFragger* and *i2MassChroQ* (followed)**

Caption for Figure 5 (follows) For the protein quantification, the *MSstats* software takes as input a matrix of features that contains all the data available from the three replicates of both samples A and B (six MS runs). Upon failure to quantify an ion signal, the corresponding matrix row reports a missing value for that feature's area-under-the-curve intensity. (A) Each histogram shows, for each software program, the total number of tried ion current extractions (light grey, i.e. 100 %) and the number of failed extractions (dark grey, i.e. missing values) in the whole set of MS runs. The proportion of failed extractions is indicated as a percentage. The histograms are grouped in columns (sample A: left, sample B: right; relative protein amounts from the various organisms are shown for each sample) and the data are shown separately for each set of identified proteins (yeast: top row, human: middle row, bacteria: bottom row). (B) Stacked bar graph of the number of protein quantification events occurring in the whole set of MS runs. The bars show the number of proteins quantified according to the percentage of feature completeness in the feature data matrix. The colors represent the proportion of proteins quantified with feature sets of varying completeness (red: complete feature set, orange: more than 50%-complete feature set, red: less than 50%-complete feature set). (C) Histogram showing the distribution of the number of proteins as a function of the number of complete features that led to their quantification. All the data shown above are for the Ions = 1 and FDR < 1 % stringency parameters.

Software execution speed

From an execution speed standpoint, *MSFragger* performs twice as fast as *i2MassChroQ* for the protein identifications while both programs perform equivalently for the peptide quantifications. This observation might be explained by the fact that the *FragPipe* search engine used in *MSFragger* has a powerful indexing method that speeds up considerably the database searches. *MSFragger* being closed source proprietary software, we cannot dig deeper to provide any better explanation from this perspective. Another explanatory element is the fact that the *X!Tandem* search engine used by *i2MassChroQ* imposes that the input data be in the mzXML format. *i2MassChroQ* thus has to make the conversion from the Bruker .d format to mzXML, which, although occurring seamlessly, takes time. Note that once the identifications have been performed, *i2MassChroQ* can store them in a project. In that case, reopening the project, refining the data processing and running quantifications anew does no more incur that computing time overhead. Indeed, in this case, *i2MassChroQ* performs faster than *MSFragger* (34 min vs 53 min for the PXD014777 sample data set and 29 min vs 41 min for the PXD010012 data set).

Conclusions

We report the clean-slate development of a new native reader for mass spectrometric data acquired on the Bruker timsTOF line of instruments. The raw data are low-level processed so as to feed the *i2MassChroQ* software package with the most suitable quantitative proteomics data. The *i2MassChroQ* software package is itself a full rewrite in C++ of our data dependent acquisition (DDA) *X!TandemPipeline* Java-based software, for enormously improved performance and numerous added features.

In this report we compared *i2MassChroQ* to *MSFragger*, the best software package currently available for DDA proteomics. We did focus our attention on the numbers of quantified proteins in *all* the datasets of each experiment because quantitative proteomics should strive for just this: allow the comparison of protein amounts across all the datasets. Our results show that *i2MassChroQ* consistently quantifies more proteins than *MSFragger*, in particular with a better capability to quantify peptides and proteins of lesser abundance, which is of utmost importance when searching proteins involved in cell signaling or transcription regulation. In the benchmarks, *i2MassChroQ* produced a remarkably dense feature data matrix for use by *MSstats*, which helps in the generation of highly reliable protein quantification data.

While data independent aquisition (DIA) proteomics might become a valid method for very large proteomics data sets, it is not mature as of yet because it still struggles with too large processing times and computer memory requirements. Zhao et al. (2023) reported recently about DIA metaproteomics-based protein quantifications by searching a database of 468096 entries. We experience that DIA proteomics cannot be implemented yet in full scale metaproteomics projects like those we are involved in, which require searching a database of almost ten millions protein sequences[30,34]. Equally challenging are peptidomics and multiple-post-translational modifications proteomics projects that involve searching large combinatorial sets of sequences. In this report, the development of the native timsTOF data importer, combined to the full rewrite of our quantitative proteomics software package, provides a solution to the challenging use cases above.

It is worthwhile noting that all the software described in this report is released as Free Open Source Software. In particular, the data reader has been implemented as part of a library so that interested developers can freely make use of it in their own DDA or DIA proteomics projects.

Acknowledgement

The authors thank Dr Sergey Bochkanov (ALGLIB Ltd) for interesting discussions on the algorithmic implementation of the polynomial equation solver and Sascha Winter (Bruker) for useful clarifications on file specification details.

Supporting Information Available

This article contains Supporting Information. The following files are available free of charge.

- Supplementary Material S1: supporting information for the article main text
- Supplementary Material S2: detailed illustrated User Manual
- Supplementary Material S3: configuration of proteomics workflows; contains the following files:
  - `msfragger-params-PXD010012` and `msfragger-params-PXD014777`: parameters used for the *MSFragger* software
  - `fragpipe-workflow-PXD010012` and `fragpipe-workflow-PXD014777`: parameters used for the *FragPipe* software
  - `xtandem-preset-timstof.xml`: parameters used for the *X!Tandem* database search engine used by *i2MassChroQ*.

References

(1) Giles, K.; Pringle, S. D.; Worthington, K. R.; Little, D.; Wildgoose, J. L.; Bateman, R. H. Applications of a travelling wave-based radio-frequency-only stacked ring ion guide. *Rapid Commun. Mass Spectrom.* **2004**, *18*, 2401 – 2414.

(2) Shvartsburg, A. A.; Smith, R. D. Fundamentals of traveling wave ion mobility spectrometry. *Anal. Chem.* **2008**, *80*, 9689 – 9699.

(3) Pukala, T. Importance of collision cross section measurements by ion mobility mass spectrometry in structural biology. *Rapid Commun. Mass Spectrom.* **2019**, *33 - Suppl - 3*, 72 – 82.

(4) May, J. C.; Goodwin, C. R.; Lareau, N. M.; Leaptrot, K. L.; Morris, C. B.; Kurulugama, R. T.; Mordehai, A.; Klein, C.; Barry, W.; Darland, E.; Overney, G.; Imatani, K.; Stafford, G. C.; Fjeldsted, J. C.; McLean, J. A. Conformational ordering of biomolecules in the gas phase: Nitrogen collision cross sections measured on a prototype high resolution drift tube ion mobility-mass spectrometer. *Anal. Chem.* **2014**, *86*, 2107 – 2116.

(5) May, J. C.; McLean, J. A. A uniform field ion mobility study of melittin and implications of low-field mobility for resolving fine cross-sectional detail in peptide and protein experiments. *Proteomics* **2015**, *15*, 2862 – 2871.

(6) Gabelica, V.; Livet, S.; Rosu, F. Optimizing Native Ion Mobility Q-TOF in Helium and Nitrogen for Very Fragile Noncovalent Structures. *J. Am. Soc. Mass Spectrom.* **2018**, *29*, 2189 – 2198.

(7) Harrison, J. A.; Kelso, C.; Pukala, T. L.; Beck, J. L. Conditions for Analysis of Native Protein Structures Using Uniform Field Drift Tube Ion Mobility Mass Spectrometry and Characterization of Stable Calibrants for. *J. Am. Soc. Mass Spectrom.* **2019**, *30*, 256 – 267.

(8) Buryakov, I. A.; Krylov, E. V.; Nazarov, E. G.; Rasulev, U. K. *Int. J. Mass Spectrom.* **1993**, *128*, 143 – 148.

(9) Purves, R. W.; Guevremont, R. Electrospray ionization high-field asymmetric waveform ion mobility spectrometry-mass spectrometry. *Anal. Chem.* **1999**, *71*, 2346 – 2357.

(10) Guevremont, R. High-field asymmetric waveform ion mobility spectrometry: A new tool for mass spectrometry. *J. Chromatogr. A* **2004**, *1058*, 3 – 19.

(11) Saba, J.; Bonneil, E.; Pomiès, C.; Eng, K.; Thibault, P. Enhanced sensitivity in proteomics experiments using FAIMS coupled with a hybrid linear ion trap/Orbitrap mass spectrometer. *J. Proteome Res.* **2009**, *8*, 3355 – 3366.

(12) Pfammatter, S.; Bonneil, E.; McManus, F. P.; Prasad, S.; Bailey, D. J.; Belford, M.; Dunyach, J.-J.; Thibault, P. A Novel Differential Ion Mobility Device Expands the Depth of Proteome Coverage and the Sensitivity of Multiplex Proteomic Measurements. *Mol. Cell. Proteomics* **2018**, *17*, 2051 – 2067.

(13) Meier, F.; Park, M. A.; Mann, M. Trapped Ion Mobility Spectrometry and Parallel Accumulation-Serial Fragmentation in Proteomics. *Mol. Cell. Proteomics* **2021**, *20*, 100138.

(14) Ridgeway, M. E.; Bleiholder, C.; Mann, M.; Park, M. A. Trends in trapped ion mobility - Mass spectrometry instrumentation. *TrAC Trends Anal. Chem.* **2019**, *116*, 324 – 331.

(15) Rusconi, F. Free Open Source Software for Protein and Peptide Mass Spectrometry-based Science. *Curr. Protein Pept. Sci.* **2021**, *22*, 134 – 147.

(16) Deutsch, E. mzML: A single, unifying data format for mass spectrometer output. *Proteomics* **2008**, *8*, 2776 – 2777.

(17) Kessner, D.; Chambers, M.; Burke, R.; Agus, D.; Mallick, P. ProteoWizard: Open source software for rapid proteomics tools development. *Bioinformatics* **2008**, *24*, 2534 – 2536.

(18) Prianichnikov, N.; Koch, H.; Koch, S.; Lubeck, M.; Heilig, R.; Brehmer, S.; Fischer, R.; Cox, J. MaxQuant Software for Ion Mobility Enhanced Shotgun Proteomics. *Mol. Cell. Proteomics* **2020**, *19*, 1058 – 1069.

(19) Yu, F.; Haynes, S. E.; Teo, G. C.; Avtonomov, D. M.; Polasky, D. A.; Nesvizhskii, A. I. Fast Quantitative Analysis of timsTOF PASEF Data with MSFragger and IonQuant. *Mol. Cell. Proteomics* **2020**, *19*, 1575 – 1585.

(20) Łącki, M. K.; Startek, M. P.; Brehmer, S.; Distler, U.; Tenzer, S. OpenTIMS, TimsPy, and TimsR: Open and Easy Access to timsTOF Raw Data. *J. Proteome Res.* **2021**, *20*, 2122 – 2129.

(21) Langella, O.; Valot, B.; Balliau, T.; Blein-Nicolas, M.; Bonhomme, L.; Zivy, M. X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *J. Proteome Res.* **2017**, *16*, 494 – 503.

(22) Choi, M.; Chang, C.-Y.; Clough, T.; Broudy, D.; Killeen, T.; MacLean, B.; Vitek, O. MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments. *Bioinformatics* **2014**, *30*, 2524 – 2526.

(23) Valot, B.; Langella, O.; Nano, E.; Zivy, M. MassChroQ: A versatile tool for mass spectrometry quantification. *Proteomics* **2011**, *11*, 3572 – 3577.

(24) Meier, F.; Brunner, A.-D.; Koch, S.; Koch, H.; Lubeck, M.; Krause, M.; Goedecke, N.; Decker, J.; Kosinski, T.; Park, M. A.; Bache, N.; Hoerning, O.; Cox, J.; Räther, O.; Mann, M. Online Parallel Accumulation-Serial Fragmentation (PASEF) with a Novel Trapped Ion Mobility Mass Spectrometer. *Mol. Cell. Proteomics* **2018**, *17*, 2534 – 2545.

(25) Navarro, P.; Kuharev, J.; Gillet, L. C.; Bernhardt, O. M.; MacLean, B.; Röst, H. L.; Tate, S. A.; Tsou, C.-C.; Reiter, L.; Distler, U.; Rosenberger, G.; Perez-Riverol, Y.; Nesvizhskii, A. I.; Aebersold, R.; Tenzer, S. A multicenter study benchmarks software tools for label-free proteome quantification. *Nat. Biotechnol.* **2016**, *34*, 1130 – 1136.

(26) Fenyö, D.; Beavis, R. C. A Method for Assessing the Statistical Significance of Mass Spectrometry-Based Protein Identifications Using General Scoring Schemes. *Anal. Chem.* **2003**, *75*, 768 – 774, Pmid: 12622365.

(27) Vaudel, M.; Burkhart, J. M.; Breiter, D.; Zahedi, R. P.; Sickmann, A.; Martens, L. A complex standard for protein identification, designed by evolution. *J. Proteome Res.* **2012**, *11*, 5065 – 5071.

(28) Craig, R.; Beavis, R. C. TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics* **2004**, *20*, 1466 – 1467.

(29) Nesvizhskii, A. I. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *J. Proteomics* **2010**, *73*, 2092 – 2123.

(30) Bossche, T. V. D. et al. Critical Assessment of MetaProteome Investigation (CAMPI): A multi-laboratory comparison of established workflows. *Nat. Commun.* **2021**, *12*, 7305.

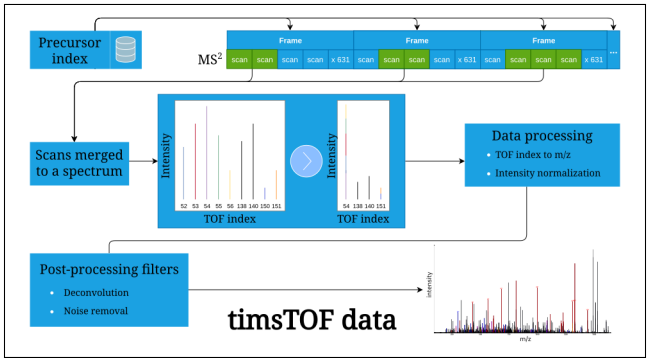
(31) Zhang, J.; Xin, L.; Shan, B.; Chen, W.; Xie, M.; Yuen, D.; Zhang, W.; Zhang, Z.; Lajoie, G. A.; Ma, B. PEAKS DB: De novo sequencing assisted database search for sensitive and accurate peptide identification. *Mol. Cell. Proteomics* **2012**, *11*, M111 – 010587.

(32) Vizcaíno, J. A. et al. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* **2014**, *32*, 223 – 226.

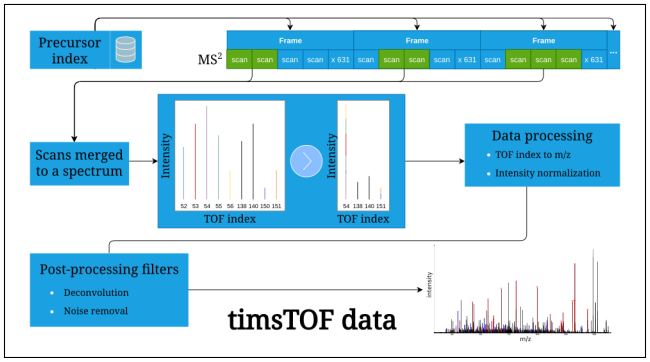
(33) Zhao, J.; Yang, Y.; Xu, H.; Zheng, J.; Shen, C.; Chen, T.; Wang, T.; Wang, B.; Yi, J.; Zhao, D.; Wu, E.; Qin, Q.; Xia, L.; Qiao, L. Data-independent acquisition boosts quantitative metaproteomics for deep characterization of gut microbiota. *NPJ Biofilms Microbiomes* **2023**, *9*, 4.

(34) Bassignani, A.; Plancade, S.; Berland, M.; Blein-Nicolas, M.; Guillot, A.; Chevret, D.; Moritz, C.; Huet, S.; Rizkalla, S.; Clément, K.; Doré, J.; Langella, O.; Juste, C. Benefits of Iterative Searches of Large Databases to Interpret Large Human Gut Metaproteomic Data Sets. *J. Proteome Res.* **2021**, *20*, 1522 – 1534, Pmid: 33528260.

TOC Graphic


Precursor index
MS2
Frame
Frame
Frame
scan
x 631
Scans merged to a spectrum
Intensity
TOF index
Data processing
TOF index to m/z
Intensity normalization
Post-processing filters
Deconvolution
Noise removal
timsTOF data
Intensity
m/z

TOC Graphic


Precursor index
MS2
Frame
Frame
Frame
scan
x 631
Scans merged to a spectrum
Intensity
TOF index
Data processing
TOF index to m/z
Intensity normalization
Post-processing filters
Deconvolution
Noise removal
timsTOF data
Intensity
m/z