Anal Bioanal Chem
DOI 10.1007/s00216-014-8210-0

NOTE

Distributed computing strategies for processing of FT-ICR MS imaging datasets for continuous mode data visualization

Donald F. Smith · Carl Schulz · Marco Konijnenburg · Mehmet Kilic · Ron M. A. Heeren

Received: 1 August 2014 / Revised: 12 September 2014 / Accepted: 19 September 2014

Abstract High-resolution Fourier transform ion cyclotron resonance (FT-ICR) mass spectrometry imaging enables the spatial mapping and identification of biomolecules from complex surfaces. The need for long time-domain transients, and thus large raw file sizes, results in a large amount of raw data ("big data") that must be processed efficiently and rapidly. This can be compounded by large-area imaging and/or high spatial resolution imaging. For FT-ICR, data processing and data reduction must not compromise the high mass resolution afforded by the mass spectrometer. The continuous mode "Mosaic Datacube" approach allows high mass resolution visualization (0.001 Da) of mass spectrometry imaging data, but requires additional processing as compared to feature-based processing. We describe the use of distributed computing for processing of FT-ICR MS imaging datasets with generation of continuous mode Mosaic Datacubes for high mass resolution visualization. An eight-fold improvement in processing time is demonstrated using a Dutch nationally available cloud service.

Published in the topical collection *Mass Spectrometry Imaging* with guest editors Andreas Römpp and Uwe Karst.

Electronic supplementary material The online version of this article (doi:10.1007/s00216-014-8210-0) contains supplementary material, which is available to authorized users.

D. F. Smith · C. Schulz · M. Konijnenburg · M. Kilic · R. M. A. Heeren (✉)
FOM Institute AMOLF, Science Park 104, 1098 XG Amsterdam, The Netherlands
e-mail: heeren@amolf.nl

D. F. Smith (✉)
National High Magnetic Field Laboratory, Florida State University, 1800 East Paul Dirac Drive, Tallahassee, FL 32310-4005, USA
e-mail: donsmith@magnet.fsu.edu

Keywords Imaging mass spectrometry · FTMS · Supercomputing · Cloud computing · Parallel processing · MALDI

Introduction

Mass spectrometry imaging (MSI) enables high-specificity spatial mapping of biomolecules from complex surfaces [1, 2]. Fourier transform ion cyclotron resonance mass spectrometry (FT-ICR MS) [3] provides the highest mass measurement accuracy (lowest mass error) and mass resolving power for MSI experiments. The high mass accuracy allows confident identification of molecules by just their mass-to-charge ratio (m/z) and the high mass resolving power resolves many closely spaced ions that cannot be resolved on lower performance mass spectrometers. FT-ICR MS requires long time-domain data to be collected for ultimate analytical performance, which can result in large datasets that must be processed efficiently and quickly. This problem is compounded when high spatial resolution is used and/or when large samples are analyzed (i.e., a large number of mass spectra/pixels), where raw data sizes can easily range from 10 to 100 GB per MSI experiment.

One solution is to process the raw data to mass spectral data on-the-fly, which is common on commercial MSI platforms (e.g., Bruker flexImaging and Thermo Fisher RAW files). This eliminates the need for data post-processing but also limits parameter selection. Such parameters include apodization functions, number of zero-fills, baseline correction, spectral normalization, mass recalibration, peak-picking algorithms (for feature-based analysis) [4] and m/z bin width for continuous mode data formats (e.g., BioMap and the AMOLF Datacube Explorer [5]). Unlike feature-based methods, where the raw data is peak picked and peaks are

Published online: 02 October 2014

stored simply as *m*/z and intensity, the continuous mode retains the mass spectral character of the dataset in discrete mass window bins. For example, at *m*/*z* 800, if a mass bin width of 0.001 Da is chosen, all spectral intensity data from 800 to 800.001 will be summed together and represented as a data point at 800.001 Da. The generation of such high mass resolution continuous mode datasets requires additional processing as compared to feature-based methods, thus more time is needed to produce these large datasets. We have recently shown that the continuous mode data format requires bin sizes of 0.00075–0.001 *m*/*z* in order to faithfully reflect the complexity of biological tissue samples, as well as retain the high mass resolving power inherent to the FT-ICR experiment [6, 7]. Further, phase correction of FT-ICR MS data requires the raw time-domain transient [7–9], which can also add additional time to the data processing pipeline.

Distributed computing is one approach to deal with the "big data" challenge associated with large FT-ICR MS imaging datasets. Such approaches are already used to deal with the large amounts of data generated from modern day mass spectrometry-based proteomics [10]. One example, that employs the same cloud service used in this manuscript, used a parallel approach for identification of tandem mass spectra using data decomposition and resulted in a 36-fold improvement over a four-core local machine [11]. For MSI, the OpenMSI project uses supercomputing resources at National Energy Research Scientific Computing Center to improve data management, storage, visualization, and statistical analysis by way of a web-based platform [12]. Jones et al. have reported up to 13-fold improvement in data processing speeds for multivariate analysis of MSI datasets by means of graphical processing units (GPU) [13]. Further, many cloud-based services exist for processing and analysis of mass spectrometry data (proteomics and metabolomics), which includes those based on the Galaxy Platform [14–17], such as Galaxy-P (https://usegalaxyp.org/. Accessed July 27, 2014), NBICGalaxy@Cloud (http://galaxy.nbic.nl/. Accessed July 27, 2014), and Galaxy WUR (http://galaxy.wur.nl/. Accessed July 27, 2014), and Taverna-based workflows [11, 18, 19]. Further, peptide search engines for bottom-up proteomics have been parallelized, such as X!Tandem on a Linux cluster [20] and using Amazon Web Services [21], as well as Hydra on a Hadoop environment [22] and SpectraST using graphical processing units [23]. However, none of these services have provisions for MSI data or continuous mode data generation.

In this manuscript, we describe distributed computing approaches for processing of FT-ICR MS imaging data for the generation of continuous mode "Mosaic Datacubes" for high mass resolution visualization. The AMOLF developed MSI data processing software "Chameleon" was deployed on a "self-service" cloud infrastructure and on a desktop personal computer. The architecture is easily scalable and is based on Microsoft Windows for compatibility with existing software tools. An eight-fold improvement in processing speed is demonstrated for a FT-ICR MS imaging experiment of a rat heart, over a local desktop personal computer (PC) running a single processing instance.

Experimental

Samples and mass spectrometry

Rat heart from adult rats (type WU) was sectioned on a cryomicrotome (Microm International, Waldorf, Germany) to 12 μm thick and thaw-mounted on an indium-tin-oxide coated glass slide (ITO, 4–8 Ω resistance; Delta Technologies, Stillwater, MN). The heart section was coated with a 20 mg/mL solution of 2,5-dihydroxybenzoic acid (DHB; 1:1 methanol/water (0.2 % trifluoroacetic acid)) with a Bruker ImagePrep. Mass spectrometry imaging experiments were performed with a solariX FT-ICR MS equipped with a 15-T super-conducting magnet (Bruker Daltonics, Billerica, MA). The laser was operated at 200 Hz and 60 laser shots were accumulated per pixel at a raster step size of 150 μm, for a total of 8531 pixels (mass spectra). The raw data was 67 GB and was compressed (ZIP file format) to a size of 25 GB. Time-domain transients of 2.3 s were collected for a Fourier limited mass resolving power, $m/\Delta m_{50\%}$=330,000 at *m*/*z* 800, where $\Delta m_{50\%}$ is the magnitude-mode peak width at half-maximum peak height. Figure S1 in the Electronic Supplementary Material (ESM) shows an example of the MSI data acquired from the rat heart, with overlay of three ion selected images.

Desktop computer processing

A desktop workstation with Windows 7 Enterprise was used for comparison with cloud computing results. The computer has an Intel® Core™ i7-2600 K processor (4 CPU @ 3.4 GHz) and 16 GB RAM. A batch file was created to run multiple instances of Chameleon simultaneously, each with a designated set of spectra for processing. The local hard drive was used for data read/write.

Cloud architecture

The SURFsara high-performance computing (HPC) cloud (https://www.surfsara.nl/systems/hpc-cloud. Accessed March 7, 2014), run on OpenNebula middleware (http://www.opennebula.org. Accessed March 7, 2014) was used. A Microsoft Windows-based cloud environment was developed for flexible implementation of existing mass spectrometry imaging data processing software. A schematic representation of the cloud architecture is shown in Fig. 1. The head node

runs on Windows Server 2012 (64 bit, 4 CPU, 64 GB RAM) and is a dedicated, persistent system that serves as the user gateway and administrative hub. The system is accessed over a secure internet connection by way of Windows Remote Desktop or a Virtual Network Computing (VNC) connection from the SURFsara OpenNebula web interface. In the present work, the compressed raw data (25 GB total) was uploaded to the cloud via Windows Remote Desktop over a 1-Gb/s network connection at a speed of ~160 Mb/s (approximately 20 min to transfer). Following the experiments presented herein, a File Transfer Protocol connection has been implemented, as well as a 10-Gb/s network connection to the SURFsara location for improved upload and download speed.

The worker nodes are non-persistent systems that run on Windows Core Server 2008 R2 (64 bit) and each worker has four CPU cores with 32 GB RAM (both 2.13 GHz Intel® Xeon® E7 "Westmere-EX" and 2.70 GHz Intel® Xeon® CPU E5-4650 CPUs are available and are assigned as available). Worker nodes can be accessed by Windows Remote Desktop from the head node or a VNC connection from the SURFsara OpenNebula web interface. The architecture shares 500 GB of hard drive space and the head node has a 55-GB capacity RAM disk. Initial tests that wrote output data to the shared storage resulted in slower than expected processing times. However, writing to the RAM disk significantly reduced processing times, which suggests the write procedure contains a large overhead in the overall processing scheme. Thus, all processing runs reported herein use the RAM disk for data output write and the raw data for processing is located on the shared storage.

The SURFsara HPC cloud uses the ToPoS token pool (token pool server) as a pilot job server for administration of token (job) requests and status (https://grid.sara.nl/wiki/index.php/Using_the_Grid/ToPoS. Accessed March 7, 2014). The head node and worker nodes run programs to administer and check for new jobs from the token pool, respectively. A front-end program was developed (with graphical user interface, GUI) for submission of cloud processing runs and definition of all pertinent cloud parameters (see ESM, Fig. S2). This includes job name and description, number of worker nodes, number of processes per worker-node, total number of tokens (tasks) and token lock time (time until the task is re-submitted). The GUI also displays a list of jobs with progress monitor, which can be used to stop running jobs. The program also sends an electronic mail, with time-stamp, to a designated recipient that indicates when a job has been queued, has started running, and has finished. In addition to cloud processing parameters, the front-end program also contains inputs to define the data processing in the in-house-developed Chameleon software package. This includes the path to the Chameleon executable, the path to an XML script file that defines all processing parameters, and optional wildcard processing parameters (that take priority over those in the XML script file).

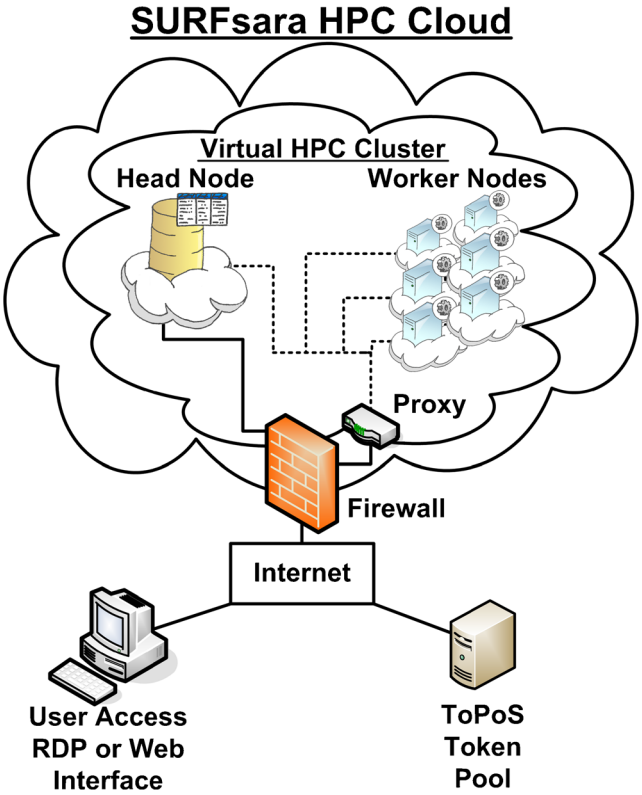

Fig. 1 Schematic of the HPC cloud environment. A head node controls administrative tasks and serves as the user gateway to the cloud. Worker nodes perform the data processing and the number of workers is easily scaled to fit processing demands

Data processing and analysis

The AMOLF developed Chameleon software package [24] was used for data processing, using a workflow similar to that described previously [6]. For cloud processing runs, all processing steps were done on the cloud environment. Briefly, the raw time-domain transient is read into Chameleon, apodized using an exponential function, and zero filled once before Fast Fourier transformation [25]. The spectra were converted from the frequency to mass domain by the method described by Francl et al. [26]. Two mass spectral peak-picking strategies were tested, a signal-to-noise (S/N) baseline peak picker and a set threshold peak picker, where both methods use a three-point apex peak-picking algorithm. The baseline peak-picking algorithm scans the local baseline over a window of 0.2 Da and saves any peaks with abundance greater than or equal to five times the standard deviation of the noise level inside that window (i.e., a $S/N \geq 5$). For threshold peak picking, a threshold

that corresponds to five times the standard deviation of the baseline noise at *m*/*z* 800 was used (*S*/*N*≥5; *m*/*z* 800 is approximately the center of the lipid mass distribution observed), where any peak with abundance over that value was saved as a peak. Peaklists were saved in XML format.

In addition to peak picking, a high mass resolution Mosaic Datacube was created. The datacube data format uses three dimensions to store and visualize two-dimensional MSI data. The first two dimensions are the *X* and *Y* coordinates from the imaging experiment and the third dimension is the *m*/*z* dimension which stores the mass spectral abundance data over a desired mass bin width (here, this abundance value is stored as a 32-bit float). A mass bin width of 0.001 Da was used to facilitate high mass resolution visualization of the datatset. The resultant dataset is large; a mass range of *m*/z 550–2,000 (1,450 Da), a mass bin of 0.001 Da, 8,531 pixels, and abundance values stored as 32-bit float results in an expected datacube size of 49.6 gigabytes (GB), as given by:

$$\frac{1450\ \text{Da}}{\text{Spectrum}} \times \frac{1000\ \text{points}}{1\ \text{Da}} \times \frac{4\ \text{bytes}}{\text{point}} \times 8,531\ \text{spectra} = 49.5\ \text{GB} \quad (1)$$

The Mosaic Datacube architecture has been developed to ease memory demands of such a large dataset. The dataset is split into a matrix of adjacent datacubes, which when combined represent the entire dataset. Here, the dataset was split into a pre-determined set of 60 individual datacubes, though this set of datacubes could be calculated at the beginning of the processing run based on the number of pixels, the area of interest, and the mass range of interest. Any volume of interest, consisting of a mass range of interest and/or a spatial area of interest, can then be called on-demand from the in-house-developed Datacube Explorer software [5]. The generation of these Mosaic Datacubes takes additional processing time, as compared to peak-picking only (see "Results and discussion" and Table 1), due to additional storage of the three-dimensional arrays in memory and the writing of these arrays to data files. However, since each cube is independent, they can be created in parallel, which yields a decrease in processing time, as discussed below.

Datacube Explorer [5, 27] was used for visualization of the Mosaic Datacubes and in-house-developed Matlab routines (MATLAB version 7.13.0.564 (64 bit), Mathworks, Natick, MA) were used for analysis of peaklists [6]. Unless noted otherwise, cloud processing runs had 65 tokens (tasks) with a lock time of 1,200 s each (25 min) and a Mosaic Datacube that divided that data into 60 cubes (60 cubes vertically that span the entire horizontal pixel range). Triplicate processing runs were done while varying the number of parallel Chameleon instances (each with five Chameleon instances per worker node). For desktop PC processing, three different Mosaic Datacube structures were tested; totaling 64, 100, and 225 individual cubes, all in triplicate.

Results and discussion

The AMOLF FT-ICR MS imaging data processing pipeline produces a continuous data format (Mosaic Datacube) and feature-based data (i.e., peak picked). The Mosaic Datacube continuous format allows high mass resolution exploration of datasets, whereas the feature-based approach enables multivariate data analysis (e.g., principal component analysis) and reduces the data load [4]. Table 1 shows the data-processing times of different processing approaches in the HPC cloud environment. All runs were performed in triplicate with 5 worker nodes, each with 5 instances of Chameleon running simultaneously (25 total parallel instances). Generation of a Mosaic Datacube (0.001 Da mass bin size) with *S*/*N* baseline peak picking (of each spectrum) results in an average processing time of 50.3 min. The Mosaic Datacube alone takes 39.7 min and *S*/*N* baseline peak picking alone takes 32.9 min. This suggests the creation and writing of the Mosaic Datacube is more time-intensive than the *S*/*N* baseline peak-picking algorithm and writing of the XML peak lists.

A reduction of processing time is observed when the *S*/*N* baseline peak picker is replaced with a simple signal magnitude threshold peak-picking algorithm, from 50.3 to 42.5 min. This reflects the time required for the calculation of local baseline noise in the *S*/*N* peak picker. When only peak picking, the processing time is again reduced, from 32.9 min for the *S*/*N* baseline peak picking to just 20.2 min for the threshold peak picking (an average of 7 mass spectra processed per second). However, the reduction in processing time is accompanied by an increase in the number of potential noise peaks in the peak lists. This is observed in the output dataload of the threshold peak picking versus the baseline *S*/*N* peak picking, where an increase from 0.42 to 1.13 GB is observed. This is due to a non-uniform baseline, where the noise level is higher at higher *m*/z, which results in unwanted (potential) noise peaks to be chosen by the threshold peak-picking algorithm (see ESM, Fig. S3 for an example from a single mass spectrum) [4]. The other consequence of the non-uniform baseline on threshold peak picking is the loss of peaks at low *m*/*z* that have intensity lower than the set threshold. For these reasons, and that the *S*/*N* baseline peak picker represents a more compute intensive operation, all further processing runs described in this manuscript produced a Mosaic Datacube and used a *S*/*N* baseline peak picker.

The worker nodes are capable of running six simultaneous instances of the Chameleon software. Errors occur if greater than six Chameleon instances are run on a single worker node,

Table 1 Comparison of average processing time, output dataload and number of files based on data processing parameters for cloud-based processing of a MALDI FT-ICR MS imaging experiment of a rat heart

	Average processing time (min)	Output dataload (GB)	Number of files
Datacube, baseline peak picking[a]	50.3	47.2	8638
Datacube only	39.7	46.8	107
Baseline peak-picking only	32.9	0.42	8532
Datacube, threshold peak picking	42.5	47.9	8638
Threshold peak-picking only	20.2	1.13	8532

The number of parallel Chameleon instances was fixed at 25 (5 worker nodes, each running 5 Chameleon instances)

[a] The parameters used throughout this manuscript

where the ToPoS token system fails. Thus, a conservative maximum value of five Chameleon instances per worker node was chosen for all cloud processing tests. The number of worker nodes was increased from one to six to determine the dependence of processing time on the number of Chameleon instances running in parallel. Figure 2 shows the data-processing time as a function of Chameleon instances. Processing runs were done in triplicate and the percent relative standard deviation of the run times varies from 3.8 to 17.5 %. The processing time decreases on an exponential curve as the number of Chameleon instances is increased (R^2=0.9953), as expected. There is deviation from the theoretical exponential decay function (as calculated from the processing time using on a single Chameleon instance) that can be due to read/write issues over the internal network and/or the overhead required by the ToPoS token pool to assign tasks to the workers. The processing time begins to level off around ~25 Chameleon instances, with a processing time of ~50 min. The shortest processing time achieved with these tests was 41.4 min (30 parallel Chameleon instances), which is a 24-fold increase over the (average) processing time with a single Chameleon instance.

The same dataset was also processed on a desktop PC, where multiple instances of Chameleon were run at the same time, each with an equal number of mass spectra to process. The inset in Fig. 2 shows this in detail, where the number of Chameleon instances was increased for three different Mosaic Datacube configurations. The RAM demands of the processing are apparent, where the PC can only run a maximum of four Chameleon instances for a Mosaic Datacube with a total of 64 cubes (8×8). The PC can run six simultaneous instances with a Mosaic Datacube with 100 cubes (10×10) and eight instances when the number of cubes is increased to 225 (15×15). As the number of cubes in the Mosaic is increased, the size of individual datacubes decreases and thus the amount of data that must be stored in RAM (for each cube) is decreased, which allows more instances of Chameleon to run simultaneously. The overload to the RAM on this desktop is apparent by the slight increase of the processing time at the end of each curve in the inset of Fig. 2, where the processes have demanded more RAM than is available and performance is degraded (likely due to virtual memory swapping). Note that the PC used here has half the RAM of the cloud worker nodes, thus more RAM would ease memory demands. Under non-overload conditions,

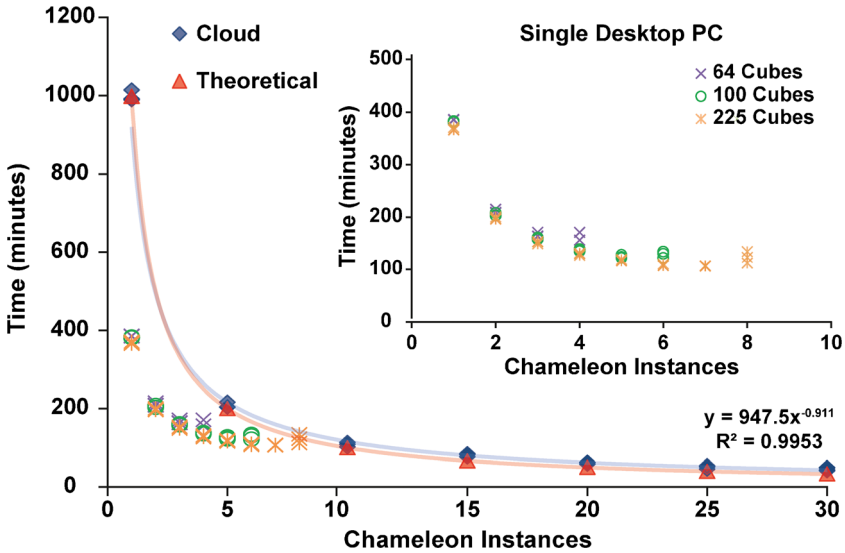

Fig. 2 Processing time versus number of Chameleon instances for distributed computing by means of cloud-based data processing (*blue*) and on a single PC (*inset*) of a MALDI FT-ICR MS imaging experiment of a rat heart (8,531 spectra/pixels). The output was a Mosaic Datacube with a *m/z* bin size of 0.001 and XML peaklists for each mass spectrum

the processing time for this dataset can be improved by a factor of 3.5 if a Mosaic of 225 cubes is created.

The processing time with a single Chameleon instance on the desktop PC is 2.6× faster than with a single instance on the cloud. Data read/write is most likely the main cause of this difference, as well as a small contribution from the ToPoS token system employed on the cloud. The desktop PC uses the local hard disk for read/write, while the cloud reads the raw data from the shared storage and writes the output data to the RAM disk of the head node, both over the local network. However, if only two worker nodes are used (each with 5 Chameleon instances, 10 total), the processing time is already comparable to the desktop PC. Cloud processing with 30 parallel Chameleon instances is 8.2× faster than a single instance on the four-core PC and 2.4× faster than the distributed processing on the four-core PC (7 parallel instances of Chameleon). More importantly, Fig. 2 shows that four worker nodes on the cloud (here, each with 5 Chameleon instances) can process two datasets (of the same size) in the same amount of time that the single desktop PC can process one dataset. Herein lays the ultimate power of the cloud processing approach, where multiple *different* datasets can be processed rapidly at the same time. Further, the ability to easily scale the cloud based on processing demands eases desktop workstation usage for extended data processing tasks.

Conclusions

The large amount of raw data associated with a FT-ICR MS imaging experiment can present a challenge for rapid processing. Distributed computing works well for these datasets; here, it is shown that an eight-fold improvement in processing speed is obtained when using a cloud-based architecture (over a single instance on a desktop PC, although with less RAM). The deployment of the cloud architecture on a "self-service" system allows high flexibility for choice of operating system, node configuration, and allows dynamic scaling of the system based on processing needs. Data read and write were found to be a limiting factor in the initial implementation of our system. Current work is focused on the use of worker nodes with local storage capability to eliminate the use of the RAM disk, fast data transfer to and from the cloud, automated startup and shut down of worker nodes, and inclusion of feature-based multivariate data analysis into the workflow. These improvements should improve processing speeds further, which will allow full exploitation of cloud resources for processing of FT-ICR MS and other mass spectrometer data types.

Acknowledgments This work is part of the research program of the Foundation for Fundamental Research on Matter (FOM), which is part of the Netherlands Organization for Scientific Research (NWO). This work was carried out on the Dutch national e-infrastructure with the support of SURF Foundation. This publication was supported by the Dutch national program COMMIT. A portion of the research was performed using EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory. The authors thank Markus van Dijk and Jhon Masschelein for technical support with the SURFsara HPC cloud.

References

1. McDonnell LA, Heeren RMA (2007) Imaging mass spectrometry. Mass Spectrom Rev 26:606–643
2. Chughtai K, Heeren RMA (2010) Mass spectrometric imaging for biomedical tissue analysis. Chem Rev 110:3237–3277
3. Marshall AG, Hendrickson CL, Jackson GS (1998) Fourier transform ion cyclotron resonance mass spectrometry: a primer. Mass Spectrom Rev 17:1–35
4. McDonnell LA, van Remoortere A, de Velde N, van Zeijl RJM, Deelder AM (2010) Imaging mass spectrometry data reduction: automated feature identification and extraction. J Am Soc Mass Spectrom 21:1969–1978
5. Klinkert I, Chughtai K, Ellis SR, Heeren RMA (2013) Methods for full resolution data exploration and visualization for large 2D and 3D mass spectrometry imaging datasets. Int J Mass Spectrom 362:40–47
6. Smith DF, Kharchenko A, Konijnenburg M, Klinkert I, Paša-Tolić L, Heeren RMA (2012) Advanced mass calibration and visualization for FT-ICR mass spectrometry imaging. J Am Soc Mass Spectrom 23:1865–1872
7. Smith DF, Kilgour DPA, Konijnenburg M, O'Connor PB, Heeren RMA (2013) Absorption mode FTICR mass spectrometry imaging. Anal Chem 85:11180–11184
8. Xian F, Hendrickson CL, Blakney GT, Beu SC, Marshall AG (2010) Automated broadband phase correction of Fourier transform ion cyclotron resonance mass spectra. Anal Chem 82:8807–8812
9. Kilgour DPA, Wills R, Qi Y, O'Connor PB (2013) Autophaser: an algorithm for automated generation of absorption mode spectra for FT-ICR MS. Anal Chem 85:3903–3911
10. Verheggen K, Barsnes H, Martens L (2014) Distributed computing and data storage in proteomics: many hands make light work, and a stronger memory. Proteomics 14:367–377
11. Mohammed Y, Mostovenko E, Henneman AA, Marissen RJ, Deelder AM, Palmblad M (2012) Cloud parallel processing of tandem mass spectrometry based proteomics data. J Proteome Res 11:5101–5108
12. Rübel O, Greiner A, Cholia S, Louie K, Bethel EW, Northen TR et al (2013) OpenMSI: a high-performance web-based platform for mass spectrometry imaging. Anal Chem 85:10354–10361
13. Jones EA, Zeijl RJM, Andrén PE, Deelder AM, Wolters L, McDonnell LA (2012) High speed data processing for imaging MS-based molecular histology using graphical processing units. J Am Soc Mass Spectrom 23:745–752
14. Blankenberg D, Kuster GV, Coraor N, Ananda G, Lazarus R, Mangan M et al (2001) Galaxy: A Web-Based Genome Analysis Tool for Experimentalists. In Current Protocols in Molecular Biology. Vol. pp. Wiley
15. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P et al (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15:1451–1455
16. Goecks J, Nekrutenko A, Taylor J, and Galaxy T (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biology 11
17. Afgan E, Chapman B, Taylor J (2012) CloudMan as a platform for tool, data, and analysis distribution. BMC Bioinformatics 13:315

18. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M et al (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20:3045–3054
19. Wolstencroft K, Haines R, Fellows D, Williams A, Withers D, Owen S et al (2013) The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res 41:W557–W561
20. Duncan DT, Craig R, Link AJ (2005) Parallel tandem: a program for parallel processing of tandem mass spectra using PVM or MPI and X!Tandem. J Proteome Res 4:1842–1847
21. Pratt B, Howbert JJ, Tasman NI, Nilsson EJ (2012) MR-Tandem: parallel X!Tandem using Hadoop MapReduce on Amazon Web Services. Bioinformatics 28:136–137
22. Lewis S, Csordas A, Killcoyne S, Hermjakob H, Hoopmann MR, Moritz RL et al (2012) Hydra: a scalable proteomic search engine which utilizes the Hadoop distributed computing framework. BMC Bioinf 13
23. Baumgardner LA, Shanmugam AK, Lam H, Eng JK, Martin DB (2011) Fast parallel tandem mass spectral library searching using GPU hardware acceleration. J Proteome Res 10:2882–2888
24. Chameleon is available upon request: contact m.konijnenburg@amolf.nl (M.K.) or heeren@amolf.nl (R.M.A.H.)
25. Frigo M, Johnson SG (2005) The design and implementation of FFTW3. Proc IEEE 93:216–231
26. Francl TJ, Sherman MG, Hunter RL, Locke MJ, Bowers WD, McIver RT (1983) Experimental-determination of the effects of space-charge on ion-cyclotron resonance frequencies. Int J Mass Spectrom Ion Process 54:189–199
27. Datacube Explorer is available at: http://www.amolf.nl/download/datacubeexplorer/