IsoSpec2: ultrafast fine structure calculator

Technical Note

IsoSpec2: Ultrafast Fine Structure Calculator

Mateusz Krzysztof Lacki, Dirk Valkenborg, and Micha# Piotr Startek

Anal. Chem., **Just Accepted Manuscript** • DOI: 10.1021/acs.analchem.0c00959 • Publication Date (Web): 05 Jun 2020

IsoSpec2: Ultrafast Fine Structure Calculator

Mateusz K.Łącki,*,† Dirk Valkenborg,‡,¶,§,‖ and Michał Startek *,⊥

†*Institute of Immunology, University Medical Center of the Johannes-Gutenberg University Mainz, Mainz 55131, Germany*
‡*Data Science Institute, Hasselt University, BE3500 Hasselt, Belgium*
¶*Interuniversity Institute of Biostatistics and Statistical Bioinformatics, Hasselt University, BE3500 Hasselt, Belgium*
§*Center for Proteomics, University of Antwerp, 2000 Antwerp, Belgium*
‖*Applied Bio and Molecular Systems, Flemish Institute for Technological Research (VITO), 2400 Mol, Belgium*
⊥*Department of Mathematics, Informatics, and Mechanics, University of Warsaw, 02-097 Warsaw, Poland.*

E-mail: matlacki@uni-mainz.de; mist@duch.mimuw.edu.pl

Abstract

High-resolution mass spectrometry becomes increasingly available with its ability to resolve the fine isotopic structure of measured analytes. It allows for high-sensitivity spectral deconvolution, leading to less false-positive identifications. Analytes can be identified by comparing their theoretical isotopic signal with the observed peaks. Necessary calculations are, however, computationally demanding and lead to long processing times. For wheat (*trictum æstivum*) alone, Uniprot holds more than 142,000 candidate protein sequences. This is doubled upon sequence reversal for identification FDR estimation, and further multiplied by performing *in silico* digestion into peptides. The same peptide might originate from more than one protein, which reduces the overall number of sequences to be calculated. However, it is still huge. IsoSpec2 can perform these calculations fast. Compared to IsoSpec1, the algorithm is simpler, orders of magnitude faster, and offers more flexibility for the developers of algorithms for raw data analysis. It is freely available under a 2-clause BSD license, with bindings for C++, C, R, and Python programming languages.

Introduction

The ability to separate molecules by their mass-to-charge ratio and to record a signal reflecting their quantity proved crucial for the success of mass spectrometry across the wide span of its applications in modern sciences. Progress in high resolution mass spectrometry[1–7] sparked interest in its potential uses in metabolomics,[8] proteomics,[9] and in specific clinical contexts.[10,11]

The adoption of these method hinges upon the availability of software able to handle complex isotopic patterns that are present in the observed signal. The theoretical description of these patterns is known for more than 60 years[12] and found repeated use.[13–15] A theoretical pattern can be calculated based on a chemical formula, masses, and natural frequencies of elements it the formula. Unfortunately, the bigger the compound, the more complicated such pattern becomes. For example, $C_{1000}H_{2002}$ results in 2,005,003 configurations – the *isotopologues*. The most probable is $^{12}C_{990}\,^{13}C_{10}\,^{1}H_{2002}$, with mass equal to 14027.7 u and probability of roughly 9.68%. Next in line is $^{12}C_{989}\,^{13}C_{11}\,^{1}H_{2002}$ with mass 14028.7 u and probability circa 9.5%. These two already consume 19% of probability,

and with 68 more one already covers 99.9% of the pattern. High-resolution calculators[16–18] exploit probability concentration[19,20] to efficiently represent most of the fine isotopic distribution. Typically, these calculators ask the user for a lower bound on the stick height to limit the calculations. `IsoSpec2` can do that too, but works also when told how much of the pattern to reveal using most probable isotopologues alone. If that fraction is P, we call the result an optimal P-set. For example, $\{^{12}C_{990}{}^{13}C_{10}{}^{1}H_{2002}, {}^{12}C_{989}{}^{13}C_{11}{}^{1}H_{2002}\}$ is the optimal 19%-set of $C_{1000}H_{2002}$. Masses and probabilities of the reported isotopologues form a stick spectrum, like in Figure 1 (top), and can be used to assess if a signal originates from a given input analyte.[21–23]

Focussing on top probable isotopologues is not the only way to solve the *isotopic distribution conundrum*.[13] Instead of filtering isotopologues, one can also aggregate them. Most notably, the problem simplifies under nominal mass approximation that assumes that differences in masses between consecutive isotopes (from lightest to heaviest) are the same across elements. Two isotopologues like $^{12}C_{990}{}^{13}C_{10}{}^{1}H_{2002}$ and $^{12}C_{991}{}^{13}C_{9}{}^{1}H_{2002}{}^{2}H$ cannot be told apart in low-resolution mass spectrometry. This opens the possibility to use either Fourier Transform methods,[21,24,25] such as Mercury,[26] or plunge into the combinatorics of generating functions;[27–29] check Supporting Information (SI) for a detailed comparison with `IsoSpec2`. Intermediate levels of resolution between nominal approximation and raw isotopologues were considered.[30] Finally, Poisson approximation to the basic model was also investigated[31] which reduces the dimension of the problem at a cost of less faithful representation thereof.

Our approach proposes an exact solution at an optimal asymptotic runtime. Though operating at infinitely resolved isotopic peaks, `IsoSpec2` also offers the possibility to aggregate these automatically to match given resolution.

In what follows, we describe the modifications that `IsoSpec2` offers with respect to its predecessor, `IsoSpec1`, followed by runtime tests.

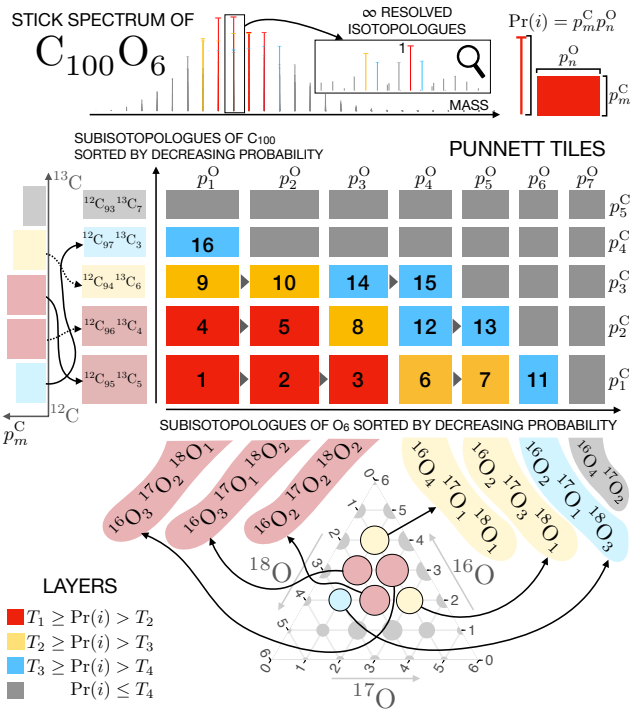

Figure 1: Getting optimal P-set stick spectrum of $C_{100}O_6$. Stick heights/isotopologue probabilities, Pr(i), are mapped to areas of *Punnett tiles* (top-right). Each area is a product of subisotopologue probabilities, $\Pr(i) = p_m^C p_n^O$. Tiles form layers in the product space (middle) and are reported in order indicated by numbers 1-16. Layers are defined by cut-offs $1 = T_1 > T_2 > T_3 > T_4$. Subisotopologues of C_{100} and O_6 (left and bottom) are generated upon creating a layer of isotopologues, also in layers, in decreasing probability. From top probable isotopologue 1 we go right, reporting 2 and 3. $\Pr(3) \geq T_1$ and $\Pr(6) < T_1$, so from 3 we jump to 4 and the layer with 5. $\Pr(1,2,3,4,5) < P$, so a new cut-off is selected. We restart from 6 and continue till 7. $\Pr(11) < T_2$, so is not reported yet. We jump back to 6, mount up to 12, and go left until we find an isotopologue more probable than the previous cut-off. This way, we recognize 5 as already reported, and report 8. We would continue right from 8, but there are no more isotopologues above T_2 there. So, we go from 8 to 14 and proceed the same way we did from 6. Layer finishes at 10, from which we jump back to 11, and continue to unveil isotopologues more probable than T_3, as before, until we reach 16.

Methods

We describe the algorithm on the example of a dummy molecule $C_{100}O_6$. We modify their isotope frequencies, so that chances of meeting ^{13}C, ^{17}O and ^{18}O are much higher than in nature. This way, they are more spread out and easier to visualize in Figure 1. The more concentrated istopic frequencies are, the smaller the resulting optimal P-set is, as shown in SI.

Each isotopologue of $C_{100}O_6$ consists of two *sub*-isotopologues of mono-atomic formulas C_{100} and O_6. Its mass is the sum of their masses and probability is the product of their probabilities. Because of the latter, we represent each isotopologue as a *Punnett tile*. Each tile in Figure 1 has area equal to that isotopologue's probability, and side lengths equal to probabilities of its subisotopologues (top-right). For example, for isotopologue 1 $\Pr(1) = p_1^C p_1^O$, where p_1^C and p_1^O are the probabilities of $^{12}C_{95}{}^{13}C_5$ and $^{16}O_3{}^{17}O_2{}^{18}O_1$. Isotopologue 2 consists of $^{12}C_{95}{}^{13}C_5$ and $^{16}O_3{}^{17}O_1{}^{18}O_2$, and so $\Pr(2) = p_1^C p_2^O$, as $^{12}C_{95}{}^{13}C_5$ is repeated.

Subisotopologues in Figure 1 are ordered with decreasing probability, $p_1^C \geq p_2^C \geq \ldots$, $p_1^O \geq p_2^O \geq \ldots$. This way, larger tiles concentrate in the corner of the quadrant. Tiles with areas above some cut-off always form one cluster. The optimal P-set can be gradually obtained with layers of less and less probable isotopologues. These correspond to tiles with areas between consecutive, ever smaller cut-offs T_i, obtained the same way `IsoSpec1` does.[18] Order of tile visits is described in caption of Figure 1.

Suppose we have a new cut-off T_2, lower than the previous one $T_1 = 1$, like in Figure 1. We now need all subisotopologues of C_{100} and of O_6 necessary to reconstruct isotopologues 1-5 (in Figure 1, in pale red). For C_{100} these are all those more probable than $t_2^C = T_2 p_1^C / \Pr(1)$, and for O_6 all those above $t_2^O = T_2 p_1^O / \Pr(1)$. Indeed, only then isotopologues 1, 2 and 3 are more probable than $p_1^C t_2^O$, but 6 is not. Also, isotopologues 4 and 9 are more probable than $p_1^O t_2^C$, but 9 is not. One can generate all subisotopologues above a given cut-off with the breadth-first-search algorithm. One starts from the top probable one and pauses after finding all subisotopologues above the current cut-off. This works, because subisotopologues follow a multinomial distribution. There, as in case of the tiles, the top probable configurations cluster around the mode and decline the further away one gets from it.[32] For instance, we can see, that in Figure 1 (bottom) the red top three probable subisotopologues are next to each other. Next two (in yellow) are next to the red ones, and the sixth most probable one (in blue) again neighours the two previous layers. The same follows for C_{100} (Figure 1, left).

`IsoSpec2` proceeds in steps. It iteratively generates new cut-offs, finds additional subisotopologues, and combines them into isotopologues. It stops, when joint probability of all reported isotopologues exceeds P. Also, note that operations required for visiting isotopologues (caption of Figure 1) require storing only subisotopologues and two isotopologues at a time. Decoupling isotopologue generation from their storage or postprocessing is a new feature of IsoSpec2. The *data flow* of isotopologues can be piped into user-defined post-processing. `IsoSpec2` offers two such procedures. The simpler one just dumps all results into a vector. In that case, the user can additionally require the trimming of the last layer to obtain a precise P-set. The other option is to bin isotopologues to a given resolution level and report mass bins and their probabilities. Finally, the user can directly provide a cut-off T, i.e. a lower bound on reported probabilities. In that case, `IsoSpec2` calculates only the first layer using T directly, which is how most other algorithms operate.

In SI we show, that `IsoSpec2` performs asymptotically a number of operations proportional to the number of reported isotopologues, which is optimal. Also, `IsoSpec2` is capable of modelling the ion statistics by directly sampling random mass spectra for a given substance with a custom sampling algorithm.[33]

Results and Discussion

We compare `IsoSpec2` runtime with that of `IsoSpec1` and the `enviPat` algorithm[34] `enviPat` was shown to be faster than the

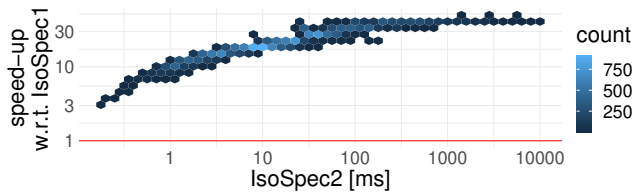

Figure 2: Run-time of `IsoSpec2` plotted against the speed-up with respect to `IsoSpec1`.

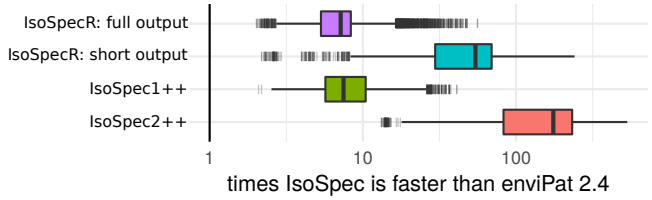

Figure 3: How much faster are variants of the `IsoSpec` than `enviPat` 2.4?

`ecipex` algorithm,[17] so we skip the direct comparison here. `IsoSpec1` outperforms[18] `enviPat` v.2.0. A faster version (v.2.4) was since released, and we compare directly to that one. `IsoSpec1` outperforms the `isoVector` algorithm[30] while calculating infinitely resolved isotopologues. As `IsoSpec2` is faster than `IsoSpec1`, we also skip that comparison.

The comparison is performed on a collection of 19,817 human protein sequences from Uniprot.[35] `IsoSpec2` is written in `C++`, and has bindings to `R` and `Python` that we call `IsoSpecR` and `IsoSpecPy`. `R` run-times are measured with the `microbenchmark` package, and `C++` code is timed using the `chrono` library. Reported results correspond to medians of 13 runs to minimize the impact of varying CPU load. The principal task: get all isotopologues more probable than one-millionth of the probability of the most probable isotopologue. We use `IsoSpec2` in the generator mode, i.e. only iterate over isotopologues without any additional form of post-processing. Our computer specifications are available in the SI.

In Figure 2 we compare the `C++` runtimes of the current and previous versions of `IsoSpec`. `IsoSpec2` is at least three times faster for smaller formulas and took about a tenth of a millisecond. Biggest compounds took between 1 to 10 seconds, which is more than 30 times faster than `IsoSpec1`.

Next, we compare `IsoSpecR` to `enviPat` v2.4. `enviPat` is written in `C++`, with bindings only to `R`. Due to the molecule size limits, only 14,101 out of all 19,817 proteins could be processed with it. `IsoSpecR` failed only in case of titin that exceeded `R`'s vector allocation limits. The `C++` generator of `IsoSpec2` obtained results for that formula in a few seconds. Figure 3 reports run-times of `enviPat` divided by that of `IsoSpec2` (v2.1, in salmon), `IsoSpec1` (v1.0.7, in green), and two modes of the `IsoSpecR`. In the short output mode, (in cyan) `IsoSpecR` reports only masses and probabilities. In the full output mode (in violet), it reports isotopologue counts too. `enviPat` provides but the full output; yet, minimizing the use of R structures brings significant run-time speedups. `IsoSpecR` is between 3 to 23 times faster then `enviPat` (first and 99th percentiles) to get the same output. This grows to anywhere between 10 to 135 times if one does not want to obtain isotope counts. If the user is willing to use C++, he can expect 31 to 329 times speedup.

What follows from the above cross-language comparison is that `IsoSpec2` is a versatile tool that can be used both in research scripts and in professional software development.

In conclusion, `IsoSpec2` is a significant improvement in isotopic calculations. It is notably faster than other approaches and can be used both for scripting and in development of professional software for raw data analysis. The `IsoSpec` algorithm has found application in a variety of projects.[23,36–42] It is open-source, can be used on a variety of operating systems (Linux, MacOS, Windows) and is available for download from github, PyPI, and CRAN.

Acknowledgement We thank Dr Błażej Miasojedow. This work was supported by Polish NCN grants 2015/17/N/ST6/03565, 2017/26/D/ST6/00304, 2018/29/B/ST6/00681, and partially by Flemish SBO grant *InSPECtor*, 120025, IWT. Plots were made with `ggplot2`,[43] Keynote, and Inkscape.

Supporting Information Available

- isospec_2_SI.pdf: mathematical details

References

(1) Senko, M. W.; Hendrickson, C. L.; Pašа-Tolić, L.; Marto, J. A.; White, F. M.; Guan, S.; Marshall, A. G. Electrospray ionization Fourier transform ion cyclotron resonance at 9.4 T. *Rapid Commun. Mass Spectrom.* **1996**, *10*, 1824–1828.

(2) Shi, S. D.-H.; Hendrickson, C. L.; Marshall, A. G. Counting individual sulfur atoms in a protein by ultrahighresolution Fourier transform ion cyclotron resonance mass spectrometry: experimental resolution of isotopic fine structure in proteins. *Proc. Natl. Acad. Sci. U. S. A.* **1998**, *95*, 11532–11537.

(3) Gorshkov, M. V.; Tolic, P.; Tolić, L. P.; Udseth, H. R.; Anderson, G. A.; Huang, B. M.; Bruce, J. E.; Prior, D. C.; Hofstadler, S. A.; Tang, L., et al. Electrospray ionization—Fourier transform ion cyclotron resonance mass spectrometry at 11.5 tesla: Instrument design and initial results. *J. Am. Soc. Mass Spectrom.* **1998**, *9*, 692–700.

(4) Nikolaev, E. N.; Jertz, R.; Grigoryev, A.; Baykut, G. Fine structure in isotopic peak distributions measured using a dynamically harmonized fourier transform ion cyclotron resonance cell at 7 T. *Anal. Chem.* **2012**, *84*, 2275–2283.

(5) G. Marshall, A.; T. Blakney, G.; Chen, T.; K. Kaiser, N.; M. McKenna, A.; P. Rodgers, R.; M. Ruddy, B.; Xian, F. Mass Resolution and Mass Accuracy: How Much Is Enough? *Mass Spectrom.* **2013**, *2*, S0009.

(6) Michalski, A.; Damoc, E.; Lange, O.; Denisov, E.; Nolting, D.; Muller, M.; Viner, R.; Schwartz, J.; Remes, P.; Belford, M.; Dunyach, J.-J.; Cox, J.; Horning, S.; Mann, M.; Makarov, a. Ultra High Resolution Linear Ion Trap Orbitrap Mass Spectrometer (Orbitrap Elite) Facilitates Top Down LC MS/MS and Versatile Peptide Fragmentation Modes. *Mol. Cell. Proteomics* **2012**, *11*, O111.013698–O111.013698.

(7) Hendrickson, C. L.; Quinn, J. P.; Kaiser, N. K.; Smith, D. F.; Blakney, G. T.; Chen, T.; Marshall, A. G.; Weisbrod, C. R.; Beu, S. C. 21 Tesla Fourier Transform Ion Cyclotron Resonance Mass Spectrometer: A National Resource for Ultrahigh Resolution Mass Analysis. *J. Am. Soc. Mass Spectrom.* **2015**, *26*, 1626–1632.

(8) Nagao, T.; Yukihira, D.; Fujimura, Y.; Saito, K.; Takahashi, K.; Miura, D.; Warishi, H. Power of isotopic fine structure for unambiguous determination of metabolite elemental compositions: In silico evaluation and metabolomic application. *Anal. Chim. Acta* **2014**, *813*, 70–76.

(9) Schwudke, D.; Schuhmann, K.; Herzog, R.; Bornstein, S. R.; Shevchenko, A. *Cold Spring Harbor Perspect. Biol.*

(10) Purcell, A. W.; Ramarathinam, S. H.; Ternette, N. Mass spectrometry–based identification of MHC-bound peptides for immunopeptidomics. *Nat. Protoc.* **2019**, *14*, 1687.

(11) Maes, E.; Oeyen, E.; Boonen, K.; Schildermans, K.; Mertens, I.; Pauwels, P.; Valkenborg, D.; Baggerman, G. The challenges of peptidomics in complementing proteomics in a clinical context. *Mass Spectrom. Rev.* **2019**, *38*, 253–264.

(12) Kienitz, H. Mass Spectrometry and its Applications to Organic Chemistry. *Angew. Chem.* **1961**, *73*, 634.

(13) Valkenborg, D.; Mertens, I.; Lemière, F.; Witters, E.; Burzykowski, T. The isotopic distribution conundrum. *Mass Spectrom. Rev.* **2012**, *31*, 96–109.

(14) Böcker, S.; Letzel, M. C.; Lipták, Z.; Pervukhin, A. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics* **2009**, *25*, 218–224.

(15) Böcker, S.; Dührkop, K. Fragmentation trees reloaded. *J. Cheminf.* **2016**, *8*, 5.

(16) Snider, R. K. NIH Public Access. *J. Am. Soc. Mass Spectrom.* **2007**, *18*, 1511–1515.

(17) Ipsen, A. Efficient Calculation of Exact Fine Structure Isotope Patterns via the Multidimensional Fourier Transform. *Anal. Chem.* **2014**, *86*, 5316–5322.

(18) Łącki, M. K.; Startek, M.; Valkenborg, D.; Gambin, A. IsoSpec: hyperfast fine structure calculator. *Anal. Chem.* **2017**, *89*, 3272–3277.

(19) Talagrand, M. A new look at independence. *Ann. Probab.* **1996**, *24*, 1–34.

(20) Ledoux, M. *The concentration of measure phenomenon*; American Mathematical Soc., 2001.

(21) Sperling, E.; Bunner, A. E.; Sykes, M. T.; Williamson, J. R. Quantitative analysis of isotope distributions in proteomic mass spectrometry using least-squares Fourier transform convolution. *Anal. Chem.* **2008**, *80*, 4906–4917.

(22) Slawski, M.; Hussong, R.; Tholey, A.; Jakoby, T.; Gregorius, B.; Hildebrandt, A.; Hein, M. Isotope pattern deconvolution for peptide mass spectrometry by non-negative least squares/least absolute deviation template matching. *BMC Bioinf.* **2012**, *13*, 291.

(23) Majewski, S.; Ciach, M. A.; Startek, M.; Niemyska, W.; Miasojedow, B.; Gambin, A. The Wasserstein Distance as a Dissimilarity Measure for Mass Spectra with Application to Spectral Deconvolution. 18th International Workshop on Algorithms in Bioinformatics (WABI 2018). 2018.

(24) Rockwood, A. L. Relationship of Fourier transforms to isotope distribution calculations. *Rapid Commun. Mass Spectrom.* **1995**, *9*, 103–105.

(25) Fernandez-de Cossio Diaz, J.; Fernandez-de Cossio, J. Computation of isotopic peak center-mass distribution by Fourier transform. *Anal. Chem.* **2012**, *84*, 7052–7056.

(26) Rockwood, A. L.; Van Orden, S. L. Ultrahigh-speed calculation of isotope distributions. *Anal. Chem.* **1996**, *68*, 2027–2030.

(27) Kubinyi, H. Calculation of isotope distributions in mass spectrometry. A trivial solution for a non-trivial problem. *Anal. Chim. Acta* **1991**, *247*, 107–119.

(28) Dittwald, P.; Claesen, J.; Burzykowski, T.; Valkenborg, D.; Gambin, A. BRAIN: a universal tool for high-throughput calculations of the isotopic distribution for mass spectrometry. *Anal. Chem.* **2013**, *85*, 1991–1994.

(29) Dittwald, P.; Valkenborg, D. BRAIN 2.0: time and memory complexity improvements in the algorithm for calculating the isotope distribution. *J. Am. Soc. Mass Spectrom.* **2014**, *25*, 588–594.

(30) Wang, Z.; Chen, X.; Ren, J.; Hu, G. Efficient simulation of isotope aggregated and fine structure by vector manipulation and change-making strategy. *Int. J. Mass Spectrom.* **2019**, *443*, 70–76.

(31) Sadygov, R. G. Poisson Model To Generate Isotope Distribution for Biomolecules. *J. Proteome Res.* **2017**, *17*, 751–758.

(32) Finucan, H. M. The Mode of a Multinomial Distribution. *Biometrika* **1964**, *51*, 513–517.

(33) Startek, M. An asymptotically optimal, online algorithm for weighted random sampling with replacement. *arXiv preprint arXiv:1611.00532* **2016**,

(34) Loos, M.; Gerber, C.; Corona, F.; Hollender, J.; Singer, H. Accelerated Isotope Fine Structure Calculation Using Pruned Transition Trees. *Anal. Chem.* **2015**, *87*, 5738–5744.

(35) Bateman, A. et al. UniProt: a hub for protein information. *Nucleic Acids Res.* **2015**, *43*, 204–212.

(36) Łącki, M. K.; Lermyte, F.; Miasojedow, B.; Startek, M. P.; Sobott, F.; Valkenborg, D.; Gambin, A. masstodon: a tool for assigning peaks and modeling electron transfer reactions in top-down mass spectrometry. *Anal. Chem.* **2019**, *91*, 1801–1807.

(37) Yunker, L. P.; Donnecke, S.; Ting, M.; Yeung, D.; McIndoe, J. S. PythoMS: A Python framework to simplify and assist in the processing and interpretation of mass spectrometric data. *J. Chem. Inf. Model.* **2019**, *59*, 1295–1300.

(38) De Bruycker, K.; Krappitz, T.; Barner-Kowollik, C. High Performance Quantification of Complex High Resolution Polymer Mass Spectra. *ACS Macro Lett.* **2018**, *7*, 1443–1447.

(39) Rusconi, F. mineXpert: Biological Mass Spectrometry Data Visualization and Mining with Full JavaScript Ability. *J. Proteome Res.* **2019**, *18*, 2254–2259.

(40) Chung, N. C.; Miasojedow, B.; Startek, M.; Gambin, A. Jaccard/Tanimoto similarity test and estimation methods for biological presence-absence data. *BMC Bioinf.* **2019**, *20*, 644.

(41) Röst, H. L.; Sachsenberg, T.; Aiche, S.; Bielow, C.; Weisser, H.; Aicheler, F.; Andreotti, S.; Ehrlich, H.-C.; Gutenbrunner, P.; Kenar, E., et al. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat. Methods* **2016**, *13*, 741.

(42) Ji, H.; Xu, Y.; Lu, H.; Zhang, Z. Deep MS/MS-Aided Structural-similarity Scoring for Unknown Metabolites Identification. *Anal. Chem.* **2019**,

(43) Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*; Springer-Verlag New York, 2016.

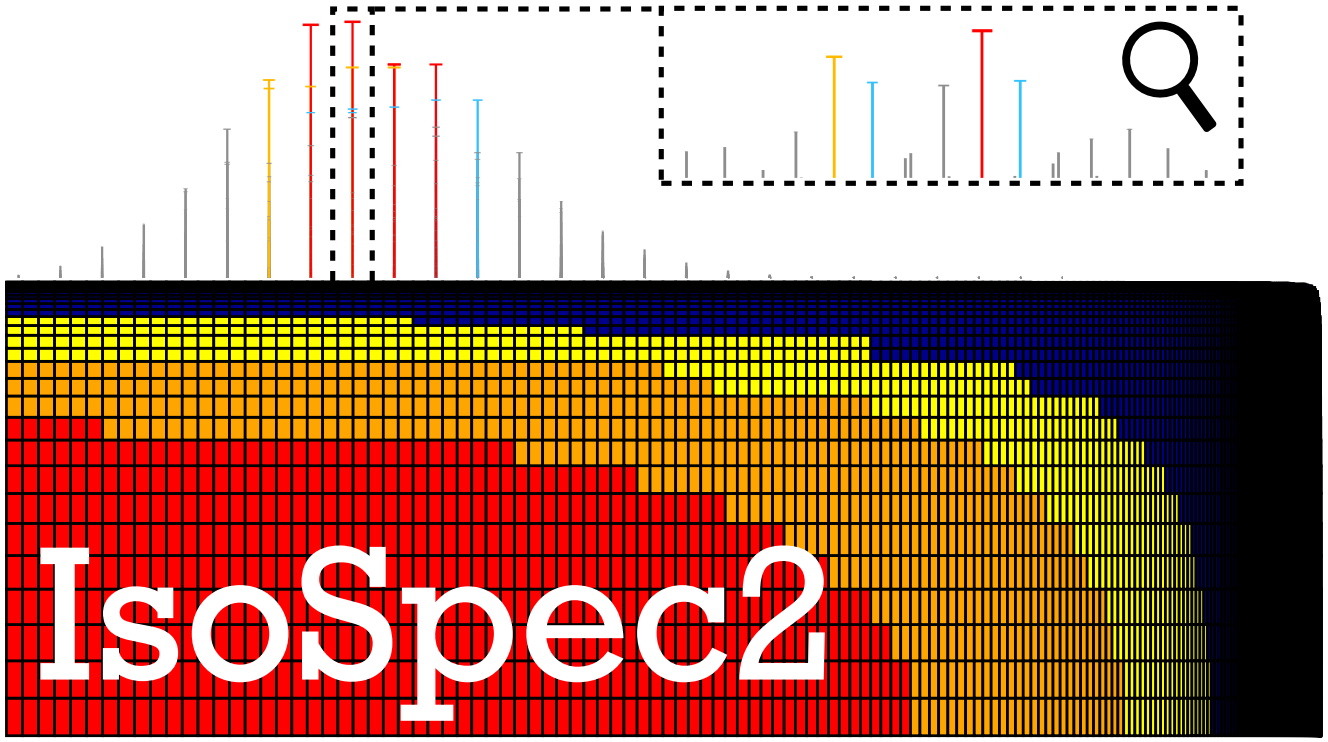

Figure 4: For TOC only.