analytical chemistry

Article

pubs.acs.org/ac

Fast Graphically Inspired Algorithm for Assignment of Molecular Formulae in Ultrahigh Resolution Mass Spectrometry

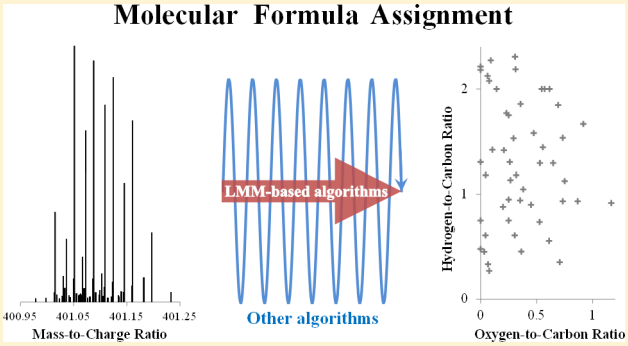
Nelson W. Green[†,‡] and E. Michael Perdue*[,‡]

[†]Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, United States

[‡]Department of Chemistry, Ball State University, Muncie, Indiana 47306, United States

Supporting Information

ABSTRACT: This study focuses on the deterministic task of assigning molecular formulae to exact masses that are generated by ultrahigh resolution mass spectrometry. A new algorithm based on low-mass moieties (LMMs) such as CH_4O_{-1} and C_4O_{-3} completely replaces conventional computational loops that explore a user-defined range of C, H, and O when searching for molecular formulae that have a given exact mass. The LMM-based algorithm has been coupled with a combinatorial algorithm that uses nested loops for N, P, S, and ^{13}C to assign molecular formulae. The resulting program is more than 1700 times faster than its brute-force counterpart that uses nested loops for all elements, and both programs yield identical output files. The new LMM-based program is 1050 times faster than the open-source program HR2, 60 times faster than Molecular Formula Calculator, and 3.6 times faster than MassCalc/FormCalc.



The characterization of natural organic matter (NOM) and other chemically complex mixtures (e.g., petroleum, metabolites) is challenging because of the many thousands of compounds that may exist within these mixtures. Using ultrahigh resolution mass spectrometry (e.g., Fourier transform ion cyclotron resonance mass spectrometry (FTICR-MS)), the masses of individuals compounds can be resolved with great accuracy[1] (approximately 0.1 mDa at 500 Da) over the range of 150−2000 Da. With accurate masses, molecular formulae can be assigned to the compounds in these mixtures. In the study of NOM, the molecular formulae yielded from FTICR-MS are a data-rich source[2] that, when interpreted, have given insightful information on photochemical reactions,[3] biogeochemical processes,[4] degradation processes,[5] and reactivity.[6]

Manual assignment of molecular formulae can be time-consuming because of the several thousand masses in a typical FTICR-MS mass list. For this reason, much effort has been directed toward the automated assignment of molecular formulae.[7−15] Computer programs developed for automated assignment of molecular formulae can be considered a combination of two tasks. The first task is deterministic and calculates all possible molecular formulae for a given mass constrained by the element limits,[10] error tolerances, and rules of chemical bonding.[9,16] Once the allowed elements are specified, molecular formulae are calculated following rules of chemical bonding. Multiple formulae may be found for a single mass within the user-specified tolerance of mass, which introduces the need for a method of selecting the most probably correct molecular formula. The second task, the heuristic task, selects the most probably correct formula from the possible formulae by following user-defined rules. For example, a simple heuristic algorithm might select the most probable formula as that formula most nearly matching the experimental mass. For some programs,[8,11,13] the deterministic and heuristic tasks are connected so interdependently that it is not easy to separate the tasks. This paper focuses on the deterministic task of assigning molecular formulae to masses from FTICR-MS.

We have theorized a new algorithm based on low-mass moieties (LMMs) for calculating molecular formulae from exact masses.[17] Conceptually, this algorithm was inspired by the van Krevelen graph of isobaric (same nominal mass) molecular formulae. It was argued qualitatively that the LMM-based algorithm should be very fast in comparison to more conventional deterministic algorithms. In this work, a program based on LMMs was developed to determine whether it was faster than other programs in assigning molecular formulae. The program extends which elements can be assigned, which were limited to C, H, and O in the original algorithm, to any element. In further discussions, with the exception of ^{13}C, the most abundant isotope for each element is considered, and ^{12}C is represented as C (unless the mass number is specifically needed for clarity).

Developing faster programs for assigning molecular formulae will increase the rate at which FTICR-MS data can be processed. With less time spent assigning formulae, more time

Received: November 7, 2014
Accepted: April 10, 2015
Published: April 10, 2015

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

may be allotted to the challenging heuristic task. The elemental limits are somewhat limited by the speed of the algorithm for assignment of formulae, which makes higher limits possible with a faster algorithm. In this paper, both the rates of assigning molecular formulae and the comprehensiveness of LMM-based programs are compared with those of other programs commonly used to assign formulae to FTICR-MS data.

DEVELOPMENT OF CHOFIT SOFTWARE FOR ASSIGNMENT OF MOLECULAR FORMULAE

The rate of the deterministic task of formula assignment is governed by the efficiency of the computer algorithm. The mathematical challenge in obtaining molecular formulae from exact masses is to minimize the square of the difference between a measured exact mass (EM) and the sum of the product of the number of moles (n_i) and exact mass (EM_i) of all the isotopes in the molecular formula:

$$(\mathrm{EM} - \sum n_i \mathrm{EM}_i)^2 \quad (1)$$

In the brute-force approach, the number of moles of each isotope is varied from its lower limit to its static upper limit in a series of nested loops. If n nested loops are used to fit n isotopes to an exact mass, the computer program will be classified as a type [n] program. Programs that use fewer than n nested loops to fit n isotopes to an exact mass will be classified as [n-1], [n-2], and [n-3] programs. All of these programs attempt to find all possible molecular formulae for a given EM, subject to the user-imposed constraints on the number of moles of each element in a valid molecular formula.

Molecular Formulae Containing Only C, H, and O. The original LMM-based algorithm[17] solves the problem of finding all the valid molecular formulae for a given exact mass when the formula contains only C, H, and O. For the purposes of this paper, a valid CHO molecular formula obeys the following constraints: $C > 0$; $2 \leq H \leq 2C + 2$; $0 \leq O \leq C + 2$.

A type [n] program will use three nested loops to solve for the number of C, H, and O atoms in a valid molecular formula.

A type [n-1] program will eliminate one loop and solve directly for the number of moles of that element by forcing eq 1 to equal zero. For example, the C loop can be eliminated by directly calculating the number of C atoms needed to force eq 1 to equal zero, after taking into account the exact mass contributions of O and H. For the tested combination of isotopes to be valid, the number of C atoms must be a positive whole number.

A type [n-2] program will eliminate two loops and solve directly for those elements by forcing eq 1 to equal zero. For example, if both the C and H loops are eliminated, then the exact mass that remains after subtraction of the mass of O should be the mass of a hydrocarbon, in which all remaining mass defect is due to H. Once H is known, all remaining mass is due to C. For the tested combination of isotopes to be valid, the number of C and H atoms must be positive whole numbers.

A type [n-3] program eliminates three loops, leaving no loops at all in the case of CHO molecular formulae. This begs the question of how a molecular formula can be determined, given only the exact mass of a molecule. The LMM-based algorithm[17] is an application of principles gleaned from a careful analysis of the information content of van Krevelen plots (see the Supporting Information for ref 17) to eliminate the C, H, and O loops. LMMs are used to search rapidly through van Krevelen plots of isobaric (same nominal mass) CHO formulae to find those molecular formulae that most nearly match the exact masses from an FTICR-MS data set.

All isobaric molecular formulae of C, H, and O can be interconverted by addition and/or subtraction of CH_4O_{-1} and C_4O_{-3} (or any other pair of CHO LMMs).[17] The search algorithm starts with the isobaric hydrocarbon having the lowest atomic H/C ratio and then either adds or subtracts these LMMs until molecular formulae having the correct exact mass are found and/or CHO compositional space is explored completely. The LMM-based algorithm explores CHO compositional space completely in significantly fewer steps than a combinatorial algorithm that uses nested loops.

Molecular Formulae Containing Non-Oxygen Heteroatoms. NOM compounds are not restricted to C, H, and O but may contain other non-oxygen heteroatoms such as ^{13}C, N, S, P, and Na. The original LMM-based algorithm[17] cannot assign correct molecular formulae to exact masses of molecules that contain heteroatoms. Either an incorrect CHO formula or no formula at all would be assigned. It is necessary to extend the LMM-based algorithm to be able to solve for the heteroatom-containing molecular formulae in NOM.

Potentially, LMMs with heteroatoms could be used in molecular formula assignment. LMMs such as $^{13}C_1C_{-10}H_1O_4N_3$ (+0.060 mDa) and $C_{-7}H_8O_{-3}N_2S_3$ (+0.217 mDa) can also be used to convert between isobaric molecular formulae. For example, $C_{24}H_{16}O_8$ (432.084517 Da) may be transformed into $C_{17}H_{24}O_5N_2S_3$ (432.084734 Da) by addition of $C_{-7}H_8O_{-3}N_2S_3$ (+0.217 mDa). However, heteroatom-containing LMMs differ from CHO LMMs. CHO LMMs such as CH_4O_{-1} and C_4O_{-3} have elemental ratios (H/C and O/C) in two dimensions, whereas heteroatom-containing LMMs are multidimensional (H/C, O/C, N/C, S/C, P/C). CHO LMMs can be simplified because they are all related (i.e., $H/C = 12 - 16(O/C)$); however, analogous simplifications for heteroatom-containing LMMs have not been found. LMMs with heteroatoms are constrained by additional elemental limits on the non-oxygen heteroatoms in addition to the constraints for valid CHO molecular formulae. For example, $C_{-7}H_8O_{-3}N_2S_3$ may only be added once to $C_{24}H_{16}O_8$ because adding it twice results in $C_{10}H_{32}O_2N_4S_6$, which contains too much hydrogen to be chemically valid.

In the deterministic process of assigning molecular formulae to exact masses, the allowed ranges of C, H, and O are typically much greater than those of other atoms. Lechtenfeld et al.,[5] for example, used limits of $^{13}C_{0-1}C_{0-50}H_{0-120}O_{0-35}N_{0-2}S_{0-1}$. Using these limits, there are 51, 121, and 36 possibilities for C, H, and O, respectively, which is a total of 222156 unique combinations of C, H, and O. In contrast, there are 2, 3, and 2 possibilities for ^{13}C, N, and S, respectively, which leads to only 12 unique combinations. When an LMM-based algorithm is used to determine C, H, and O, the remaining combinatorial problem for ^{13}C, N, P, S, etc. is actually rather small and can be managed efficiently using nested loops.

With this insight, the rapid LMM-based algorithm was used to calculate C, H, and O (the core formula), and the combinatorial algorithm (nested loops) was used for all other elements/isotopes. This approach provides both the ability to assign molecular formulae using a full set of elements and a potentially large increase in performance. Valid molecular formulae must meet the requirements of the Senior rules[16] from which it can be shown that (1) the sum of the odd-valence atoms must be an even number and (2) the difference between the maximum number and minimum number of bonds, which

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

is known as unsaturation or double bond equivalents (DBEs), must be greater than or equal to zero.

The Senior rules do not distinguish between ^{12}C and ^{13}C. The core formula must contain all C atoms if it is to be validated using the Senior rules. This can be accomplished by introduction of an exchange operator **E** that has a chemical composition of $^{13}C^{12}C_{-1}$. With this modification, all C in the core formula is ^{12}C, and **E** from the combinatorial formula converts a ^{12}C into a ^{13}C. The Senior rules also require that molecular formulae must contain an even number of odd-valence atoms. Valid core formulae must contain an even number of H atoms, and valid combinatorial formulae must contain an even number of H + N + P atoms. The latter requirement is met by introducing additional multielement components **N** and **P**, whose chemical compositions are NH and PH, respectively. After all of these adjustments, the core formula will contain C, H, and O, and the combinatorial formula will contain **E**, **N**, **P**, and S. For example, the molecular formula $^{13}CC_{20}H_{45}O_{10}N_2S_3P$ would be resolved into a core CHO formula ($C_{21}H_{42}O_{10}$) and a combinatorial formula ($\mathbf{EN_2S_3P}$).

The resulting computer program is designated as CHOFIT [*n-3*]. The overall algorithm starts with the EM of a molecule whose molecular formula is sought and subtracts all mass contributions of **E**, **N**, **P**, and S to obtain a core mass (EM_{core}) for each unique combination of **E**, **N**, **P**, and S. The core mass should be due to C, H, and O, which are determined using the LMM-based algorithm. If no combination of C, H, and O has mass that equals EM_{core}, then the combination of **E**, **N**, **P**, and S from the combinatorial algorithm is not correct. When the combination of C, H, and O has a mass that equals EM_{core}, the complete molecular formula having exact mass EM is generated by combining the values of **E**, **N**, **P**, and S from the combinatorial algorithm with the values of C, H, and O. The combinatorial algorithm (nested loops) used for **E**, **N**, **P**, and S is shown in Figure 1, and the LMM-based algorithm used for C, H, and O is shown in Figure 2d.

To better assess the performance of the LMM-based algorithm versus the combinatorial algorithm for determination of C, H, and O, three other versions of CHOFIT were created and tested in this study. CHOFIT [*n*] uses nested loops not only for **E**, **N**, **P**, and S but also for C, H, and O. CHOFIT [*n-1*] uses nested loops not only for **E**, **N**, **P**, and S but also for H and O, and C is determined by forcing mass balance on eq 1. CHOFIT [*n-2*] uses nested loops for **E**, **N**, **P**, and S and also for O, and C and H are determined by forcing mass balance on eq 1. Specifically, all unexplained mass defect is assigned to H, and C is obtained by forcing eq 1 to equal zero. The modified algorithms are shown in parts a, b, and c, respectively, of Figure 2. All four versions of CHOFIT were developed using the Free Pascal compiler (v. 2.6.2) in release mode.

OTHER SOFTWARE THAT IS USED TO ASSIGN FORMULAE

Other existing software programs have been used in numerous studies for fitting molecular formulae to exact masses from FTICR-MS. Three such programs have been used in this study to compare their rates of fitting molecular formulae to the performance of CHOFIT [*n-3*], which uses the LMM-based algorithm for determination of C, H, and O. HR2 (v. 1.02) is an open-source molecular formula calculator that is used within the Seven Golden Rules software[9] for assigning formulae in metabolomics. It was developed in C++ by Joerg Hau under

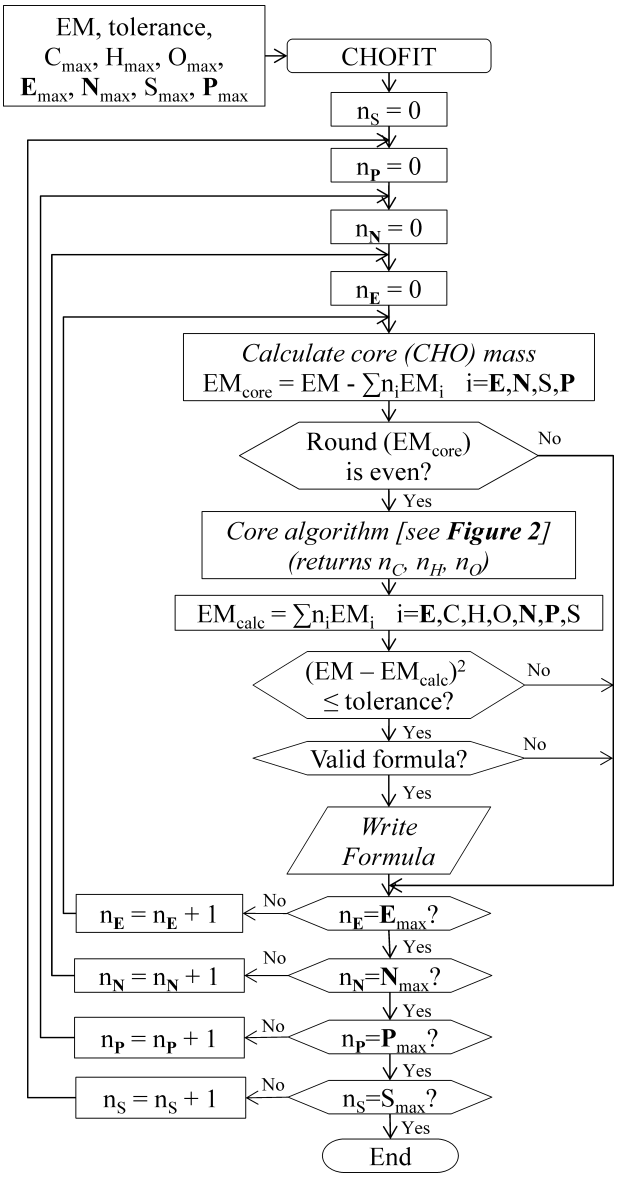


Figure 1. Generalized CHOFIT algorithm with external nested loops for **E**, **N**, **P**, and S.

the GNU general public license, and the source code was obtained from http://fiehnlab.ucdavis.edu/projects/Seven_Golden_Rules. The source code was compiled in Visual Studio 2012 Express with full compiler optimizations (/On). By inspection of the source code, it was found that the program uses fully nested loops, so it has been designated HR2 [*n*]. This program was used in its original form and after modification of the source code to replace loops for C, H, and O with the LMM-based algorithm that is used in CHOFIT [*n-3*]. The modified version of HR2 is designated as HR2 [*n-3*]. The Molecular Formula Calculator (v. 1.2.3) from the National High Magnetic Field Laboratory (NHMFL) at Florida State University has been used frequently to assign molecular formulae and was obtained from http://www.magnet.fsu.edu/usershub/scientificdivisions/icr/icr_software.html. A customized version (v. 1.2.5) of this software containing timers for precise measurement of execution times was provided in a personal correspondence by Dr. Greg Blakney at NHMFL.[14] MassCalc, which is the molecular formula finder in FormCalc,

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

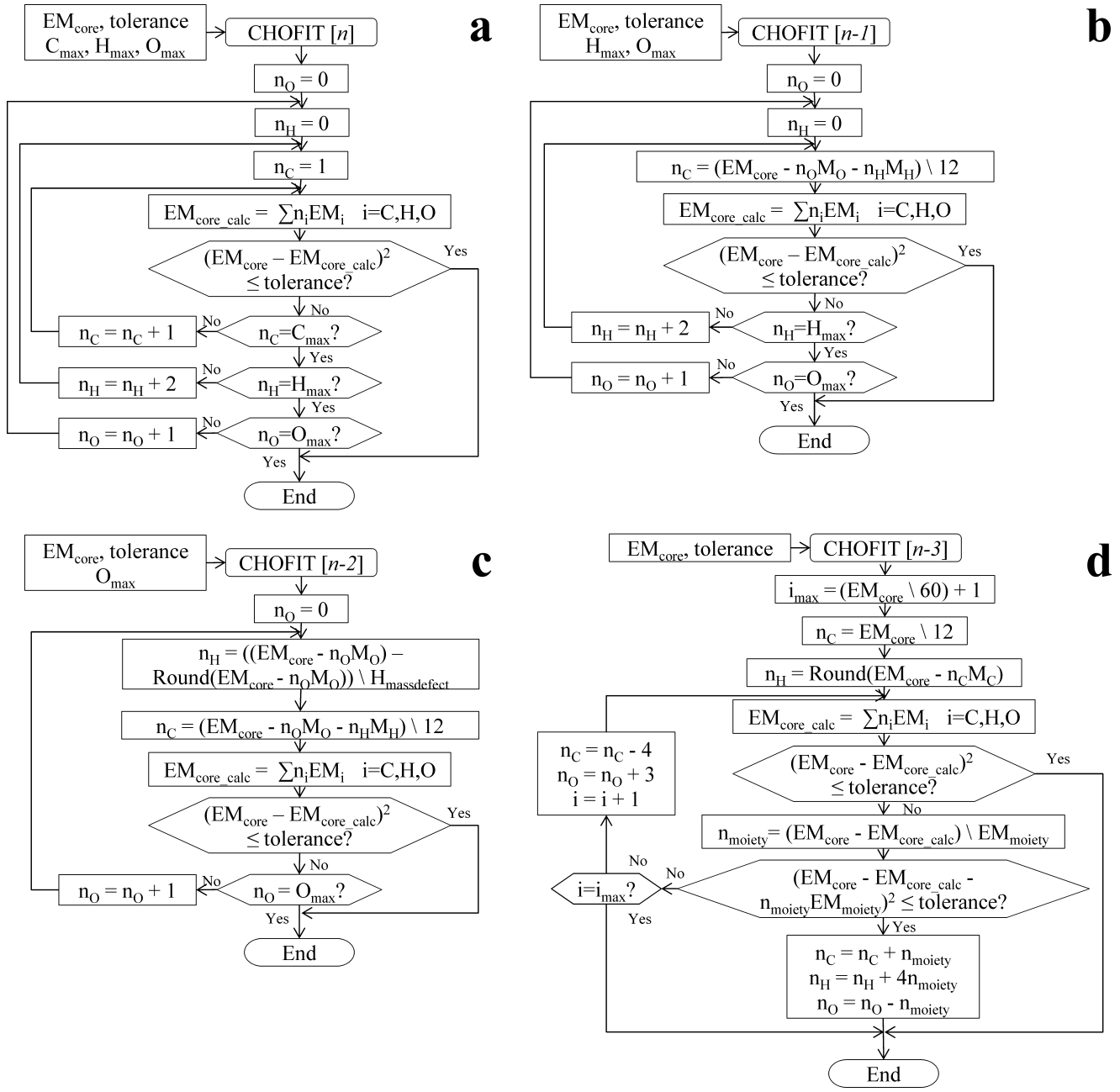


Figure 2. Four versions of the CHOFIT core (CHO) algorithm, which are used within the generalized CHOFIT algorithm [see Figure 1]. (a) The [*n*] algorithm loops CHO loops. (b) The [*n-1*] algorithm loops OH and calculates C from the remaining mass. (c) The [*n-2*] algorithm only loops O and calculates H from the remaining mass defect and then C from the remaining mass. (d) The [*n-3*] algorithm does not loop CHO, but instead searches for the core CHO formula using LMMs in the manner described in ref 17.

was obtained in a personal correspondence from Prof. Dr. Philippe Schmitt-Kopplin at Helmholtz Zentrum Muenchen.[15]

MOLECULAR FORMULA DATA SETS

The rates of assignment of molecular formulae to exact masses by the four versions of CHOFIT, the two versions of HR2, Molecular Formula Calculator, and Masscalc/Formcalc were compared using four mass lists. The four mass lists were obtained from three computer-generated data sets and one experimental mass list (see the Supporting Information). A simple computer program for generating all C, H, and O formulae in an isobaric series[17] was modified to include ^{13}C, N, S, or P, and to generate molecular formulae over any mass range. Valid molecular formulae comply with the Senior rules and are further constrained to have C > 0, H ≥ 2, and O + N + S + P ≤ C + 2.

One computer-generated data set contains all 53573 valid combinations of C, H, and O in the nominal mass range of 150−1000 Da and has an average molecular weight of 748 Da. A second data set includes all of these molecular formulae plus 48339 molecular formulae containing ^{13}C, N, S, or P (a total of 101912 molecular formulae). The additional elements/isotopes were added randomly, subject to the constraints that $0 \le {}^{13}C \le 1$, $0 \le N \le 10$, $0 \le S \le 6$, and $0 \le P \le 4$. ^{13}C, N, and P were actually added using the multielement components **E**, **N**, and **P**. The second data set has a nominal mass range of 150−1273 Da

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

Table 1. Aggregated Statistics for Analysis of Four Mass Lists Containing 198789 m/z Values

	CHO mode[a]		full mode[a]	
program	no. of assigned formulae	fitting rate (no. of masses s^{-1})	no. of assigned formulae	fitting rate (no. of masses s^{-1})
CHOFIT [*n*]	126352	395	5400522	0.8
CHOFIT [*n-1*]	126352	16057	5400522	28.9
CHOFIT [*n-2*]	125785	150802	5370792	869.1
CHOFIT [*n-3*]	126323	384527	5387589	1429.2
Molecular Formula Calculator	126336	2207	7244622	23.9
MassCalc/FormCalc	129347	80481	22427850	401.4
HR2 [*n*]	113093	539	3930238	1.4
HR2 [*n-3*]	119051	98682	3871778	4096.2

[a]CHO mode includes C, H, and O. Full mode also includes ^{13}C, N, P, and S.

and an average molecular weight of 763 Da. A third data set was generated using modulus operations that selected a nonrandom subset of 10714 molecular formulae from the first data set and further modulus operations that introduced ^{13}C, N, S, and P nonrandomly into an additional 4076 molecular formulae (a total of 14790 molecular formulae), using the same limits that were used for the second data set. The third data set has a nominal mass range of 150−1434 Da and an average molecular weight of 762 Da.

Each data set of molecular formulae was used to generate a mass list in which the exact masses of the molecular formulae were calculated using IUPAC masses[18] of all elements/isotopes and rounded to 10 decimal places. The exact masses of negative molecular ions were calculated by subtracting the exact mass of a proton (H^+). The fourth mass list was obtained directly from an FTICR mass spectrum of Suwannee River fulvic acid (SRFA) that was acquired at Helmholtz Zentrum Muenchen on a Bruker (Bremen, Germany) APEX-Q FTICR-MS instrument equipped with a 12 T superconducting magnet and an APOLLO II electrospray source that was used in negative mode. The SRFA spectrum contains 28514 peaks with signal-to-noise ratio ≥2, and was described by Koprivnjak et al.[19] The nominal mass range (assuming that ions were singly charged) is 148−2000 Da, and the average molecular weight is 504 Da.

RATE OF ASSIGNMENT OF MOLECULAR FORMULAE

Experimental Methodology. The programs for assigning molecular formulae were compared using the four mass lists that have been described. If only a single mass list were used, then the time required to process that mass list would be an acceptable metric for comparing the programs. For aggregation of results from processing of four mass lists, each having a different mass distribution and a different number of masses to assign, the rate of assignment was selected as a better metric for comparing the programs. Rates (no. of masses s^{-1}) were calculated by dividing the number of masses in each mass list by the overall time required to process that mass list.

The rate of processing a mass list should depend on the number of elements/isotopes being considered and the allowed range for each element/isotope in a combinatorial algorithm based on nested loops. Two cases were considered in this study. First, all programs were constrained to process the four mass lists using $C_{1-83}H_{0-144}O_{0-36}$, which is referred to here as the CHO mode. The upper limits for C, H, and O are based on an assumed core mass of 1000 Da. Molecular formulae containing non-oxygen heteroatoms or ^{13}C cannot be found in CHO mode. For a more comprehensive test, all programs were constrained, if possible, to process the four mass lists using elemental limits of $^{13}C_{0-1}C_{1-83}H_{0-144}O_{0-36}N_{0-10}S_{0-6}P_{0-4}$, which is referred to as full mode. The CHOFIT programs, which use multielement components (**E** = $^{13}CC_{-1}$, **N** = NH, and **P** = PH), used very nearly comparable limits of $\mathbf{E}_{0-1}C_{1-83}H_{0-144}O_{0-36}\mathbf{N}_{0-10}S_{0-6}\mathbf{P}_{0-4}$. The upper limit for H could not be constrained in MassCalc. For all programs except HR2, formulae were assigned with an error tolerance of 0.4 ppm. In HR2, an absolute tolerance of 0.2 mDa (averaging 0.33 ppm across all mass lists) was used. Rates should also vary according to the mass distribution within each mass list, because the number of valid molecular formulae per mass is known to increase drastically with increasing mass.[10] The relationship between processing rate and mass distribution was explored by processing not only the four mass lists but also eight segments of each mass list. Before creation of the segments, the mass lists were sorted by mass to ensure that each segment covered a distinctly different range of mass. The four mass lists contain 53573, 101912, 14790, and 28514 masses, respectively, so segments of different mass lists contain different numbers of masses. With the exception of the last segment for each mass list, the segments of the four mass lists contained 6700, 13000, 1900, and 3600 masses, respectively, and the last segment of each mass list contained the remainder of that mass list. The average molecular weight was calculated for each of the 32 segments. One program, MassCalc, was unable to run certain mass lists and segments of mass lists in full mode, so in those cases, the lists were further subdivided and the run time for a mass list or segment was estimated as the sum of run times for smaller subsegments.

All computer programs for deterministic assignment of molecular formulae to the four mass lists were executed on a Dell Optiplex 990 with an Intel Core i7-2600 processor at 3.4 GHz with 8.00 GB of RAM running Windows 7 (SP 1), with all nonessential processes disabled using msconfig.exe. The run time was measured from the start of reading a mass list until the completion of writing molecular formulae to an output file. With the exception of Molecular Formula Calculator, all programs were executed using batch scripts that recorded the start time, executed the program one or more times, and recorded the end time. Run times were calculated as the elapsed time divided by the number of times the program was executed within the batch file. The uncertainty in elapsed time is 10−20 ms, so programs were executed multiple times, if necessary, to increase the elapsed time to more than 500 ms. Molecular Formula Calculator could not be executed from a batch file, so Dr. Greg Blakney provided a custom version of the NHMFL software[14] that contains internal timing code with

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

a 55 ms precision, according to National Instruments. All tests of Molecular Formula Calculator were conducted using this custom version (v. 1.2.5), which is 2.9 times faster, on average, than the version of this software (v. 1.2.3) that was available on the NHMFL Web site at the time of this study.

Overview of Aggregated Results. Altogether, the four mass lists contained 198789 masses, including 117860 molecular formulae containing only C, H, and O in the first three data sets. Koprivnjak et al.[19] analyzed the fourth mass list, using charges of −1 and −2 and fitting with ^{13}C, C, H, and O. They assigned 6448 molecular formulae containing C, H, and O to singly charged ions, including half of the 7928 paired, singly charged ions (with the excluded half of singly charged ions having a single ^{13}C in their assigned molecular formulae) and another 2484 unpaired, singly charged ions (not included in Table 7 of that paper). Altogether, the four mass lists contain approximately 124308 (117860 + 6448) molecular formulae that contain only C, H, and O. If all eight programs assign molecular formulae perfectly, approximately 124308 molecular formulae should be assigned using CHO mode and all 198789 molecular formulae should be assigned in full mode. A general overview of the performance of all tested programs in CHO mode and in full mode is given in Table 1, which combines all mass lists.

In CHO mode, the number of assigned molecular formulae averaged 124000 ± 4.3% for all eight programs, which is in excellent agreement with the anticipated total number of CHO molecular formulae in the four mass lists. A closer look at Table 1 shows that both versions of HR2 assigned relatively lower numbers of molecular formulae and that all other programs assigned slightly more molecular formulae than the anticipated upper limit for CHO molecular formulae. The programs diverge sharply where rates of processing are considered, with rates varying by 3 orders of magnitude, with CHOFIT [*n*] being the slowest and CHOFIT [*n*-3] being the fastest.

In full mode, the number of assigned molecular formulae averaged 5230000 ± 22% for all programs except MassCalc, which generated more than 22000000 molecular formulae. The variation among all programs is less than a factor of 5 and is attributable to various degrees of internal filtering to eliminate chemically invalid and/or unlikely molecular formulae before an output file is written. In all cases, the number of assigned molecular formulae greatly exceeds the number of masses that were processed (198789). In full mode, the rates of processing vary by more than 3 orders of magnitude, with CHOFIT [*n*] being the slowest and HR2 [*n*-3] being the fastest. Among the three external programs in full mode, MassCalc was 17 times faster than Molecular Formula Calculator and 295 times faster than HR2 [*n*]. Even so, CHOFIT [*n*-3] was 3.6 times faster and HR2 [*n*-3] was 10.2 times faster than MassCalc for the same task.

Using elemental limits of $\mathbf{E}_{0-1}C_{1-83}H_{0-144}O_{0-36}\mathbf{N}_{0-10}S_{0-6}\mathbf{P}_{0-4}$, there are 2 × 11 × 7 × 5 = 770 combinations of **E**, **N**, **P**, and S that are attempted in nested loops in full mode but not in CHO mode. For this reason, CHO mode might be expected to be 770 times faster than full mode for all programs, solely on the basis of the number of combinatorial possibilities. The difference was actually smaller for all the programs, with the rates of processing mass lists averaging 274 times faster in CHO mode than in full mode. This suggests that the assignment of non-oxygen heteroatoms was faster than expected for all the programs, especially for HR2 [*n*-3], for which CHO mode was only 24 times faster than full mode.

Rates and Algorithms among the CHOFIT Programs. The four CHOFIT programs use different core algorithms (Figure 2), and a comparison of those programs may provide insight into how the LMM-based algorithm greatly accelerates the assignment of molecular formulae. The rate at which a program assigns molecular formulae should vary with the number of combinatorial steps that must be taken to assign molecular formulae and the number of calculations and Boolean decisions that must be used to eliminate chemically invalid and/or unlikely molecular formulae before an output file is written.

The maximum number of combinatorial steps in each of the four programs that are described by Figure 1 and the algorithms in Figure 2 is determined by the elemental limits that have been used ($\mathbf{E}_{0-1}C_{1-83}H_{0-144}O_{0-36}\mathbf{N}_{0-10}S_{0-6}\mathbf{P}_{0-4}$). In all four programs, there are 2 × 11 × 7 × 5 = 770 combinations of **E**, **N**, **P**, and S that may need to be attempted in nested loops. The difference among programs lies in the processes by which C, H, and O are assigned to a molecular formula. Both C and O may be assigned any value within their respective ranges; however, H must always be an even number in molecular formulae containing only C, H, and O. Thus, the [*n*] algorithm considers 83 × 73 × 37 = 224183 combinations of C, H, and O, so the maximum number of combinatorial steps for the [*n*] program is 224183 × 770 = 172620910. The [*n-1*] and [*n-2*] programs should have maxima of 2079770 and 28490 steps, respectively. The [*n-3*] program includes 770 combinations of **E**, **N**, **P**, and S and several steps inside the LMM-based [*n-3*] algorithm. The number of steps inside the LMM-based algorithm depends on the number of CH_4O_{-1} mixing lines that contain valid molecular formulae, which is calculated empirically as one plus the rounded integer quotient of $EM_{core}/60$. Assuming an upper limit of 1000 Da for EM_{core}, there should be a maximum of 18 CH_4O_{-1} mixing lines, and the maximum number of steps for the [*n-3*] program is thus 18 × 770 = 13860.

On the basis of the maximum number of steps in each algorithm, the [*n-3*] program is predicted to be approximately 2, 150, and 12500 times faster than the [*n-2*], [*n-1*], and [*n*] programs, respectively. In full mode, the [*n-3*] program is actually 1.6, 49, and 1742 times faster than the [*n-2*], [*n-1*], and [*n*] programs, respectively. In CHO mode, the [*n-3*] program is 2.5, 24, and 974 times faster than the [*n-2*], [*n-1*], and [*n*] programs, respectively. The anticipated trend in rates of assigning molecular formulae is observed qualitatively, but the actual increases in rate are substantially less than predicted from differences in the maximum number of combinatorial steps.

Combinatorial algorithms can be optimized far beyond a brute-force algorithm, e.g., by enabling dynamic upper limits for all elements that are adjusted according to the mass to which molecular formulae are being assigned. Such dynamic adjustments would be most effective when assigning molecular formulae to low masses and should become gradually less important for higher masses that must contain relatively large numbers of the elements that are being assigned. The algorithms in Figure 2 actually use mass-derived, dynamic upper limits (C_{max}, H_{max}, and O_{max}) on loops for C, H, and O, which limits the number of combinatorial possibilities for a given mass, especially for lower masses. For example, consider a mass of 500 Da, for which the dynamic upper limits for C, H, and O are 41, 68, and 18, respectively. Fixed upper limits, based on a mass of 1000 Da, lead to 83 × 73 × 37 = 224183 combinations of C, H, and O, but dynamic upper limits lead to only 41 × 35 × 19 = 27265 combinations of C, H, and O. The

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

search space is reduced by a factor of 8.2 (224183/27265) through the use of dynamic upper limits. In the [*n*], [*n-1*], [*n-2*], and [*n-3*] programs, the number of loops that are optimized by use of dynamic upper limits gradually decreases from three to zero. The greatest impact on rate is thus on the [*n*] program, and the least impact is on the [*n-3*] program. This is probably the main reason why the [*n-3*] program increases the rate of assignment of molecular formulae to a lesser degree than predicted from differences in the maximum number of combinatorial steps.

Dependence of Rates of Processing Mass Lists on Molecular Weight. The rates of processing mass lists are expected to be dependent on the molecular weights of individual masses and, by extension, the average molecular weight of a set of masses. This expectation is realized in the analysis of the rates of processing of mass-sorted segments of the four mass lists in CHO mode (Figure 3) and in full mode (Figure 4).

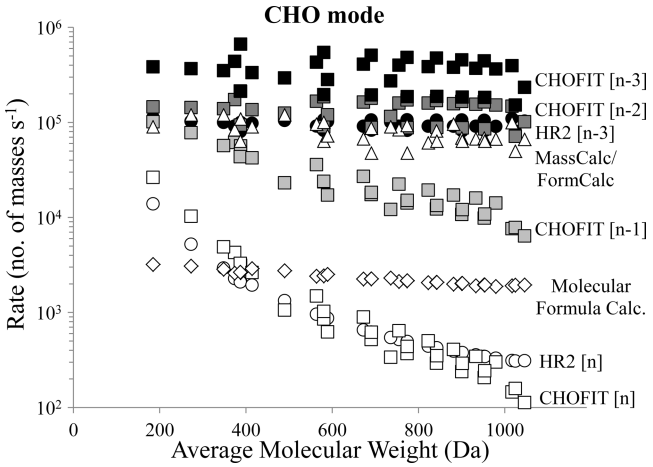


Figure 3. Rate of processing mass lists (no. of masses s^{-1}) for eight computer programs with four mass lists in eight continuous mass segments plotted against the average molecular weight of each segment. The mass lists were fit using CHO mode ($C_{1-83}H_{0-144}O_{0-36}$). External programs in order of increasing rate are HR2 [*n*] (○), NHMFL Molecular Formula Calculator (◇), and MassCalc/FormCalc (Δ). The in-house CHOFIT programs in order of increasing rate are [*n*] (□), [*n-1*] (light gray □), [*n-2*] (dark gray □), and [*n-3*] (■). HR2 [*n-3*] (●) is the enhanced version of HR2 [*n*] that uses the [*n-3*] algorithm.

In full mode, the decrease in the rate of processing mass lists with increasing mass could be described by a simple power function of the form

$$\text{rate} = A(X)^{-B} \quad (2)$$

where X is the average molecular weight for a segment of a mass list and A and B are empirical fitting parameters. For all programs except MassCalc, the best fit line had an r^2 of 0.978–0.997. For MassCalc, the best fit line had an r^2 of only 0.841, and the results were better described by a linear equation (r^2 = 0.909). The results in CHO mode vary less consistently with average molecular weight and are not well fit by eq 2 or by a linear equation.

The most obvious explanation for slower rates at higher molecular weight is the drastic increase in the number of chemically valid molecular formulae that can be assigned to a single mass at higher molecular weight. In addition, more atoms

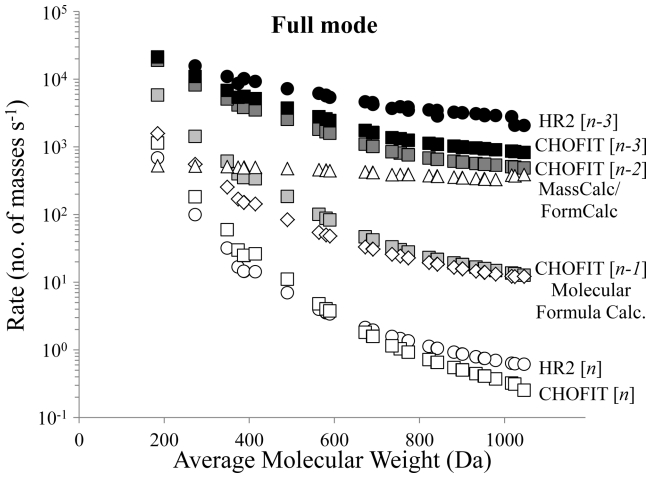


Figure 4. Rate of processing mass lists (no. of masses s^{-1}) for eight computer programs with four mass lists in eight continuous mass segments plotted against the average molecular weight of each segment. The mass lists were fit using full mode ($^{13}C_{0-1}C_{1-83}H_{0-144}O_{0-36}N_{0-10}S_{0-6}P_{0-4}$). External programs in order of increasing rate are HR2 [*n*] (○), NHMFL Molecular Formula Calculator (◇), and MassCalc/FormCalc (Δ). The in-house CHOFIT programs in order of increasing rate are [*n*] (□), [*n-1*] (light gray □), [*n-2*] (dark gray □), and [*n-3*] (■). HR2 [*n-3*] (●) is the greatly enhanced version of HR2 [*n*] that uses the [*n-3*] algorithm.

are needed to account for a large mass, so programs will benefit less from optimization of nested loops as mass increases.

The power function relationships between rate and average molecular weight in the CHOFIT and HR2 programs are probably the result of mass-based optimizations, which are expected to have greater impact at lower masses. This makes the rates higher at low masses than they would otherwise be in a brute-force calculation that explores the complete range of elements for each mass. The unique linear relationship between rate and average molecular weight for MassCalc in Figure 4 suggests that it lacks mass-based optimizations.

The results for CHOFIT and HR2 programs in Figure 4 demonstrate that programs using similar algorithms exhibit similar dependencies of rate on molecular weight. On this basis, Molecular Formula Calculator appears to be an [*n-1*] program, because it exhibits behavior nearly identical to that of CHOFIT [*n-1*].

■ COMPREHENSIVENESS OF THE LMM-BASED ALGORITHM FOR ASSIGNING MOLECULAR FORMULAE

The [*n*] algorithm is often called the brute-force method because all possible combinations of elements are attempted when assigning molecular formulae to a mass. Programs using this algorithm are thus expected to find all possible molecular formulae for a given mass, although chemically invalid and/or unlikely molecular formulae might be filtered out before an output file is written. The comprehensiveness of the [*n-3*] algorithm in assigning molecular formulae was evaluated by comparing its output to that of the [*n*] algorithm. Only the output files from tests that were conducted using full mode were used because correct molecular formulae can contain non-oxygen heteroatoms. The SRFA mass list was excluded because correct molecular formulae are only known for mass lists that were computer-generated.

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086–5094

The numbers of masses to which molecular formulae were assigned correctly, assigned incorrectly, or not assigned are useful metrics for comparing the output from the programs. For a given mass, a molecular formula is assigned correctly if it is among the molecular formulae that were assigned to that mass, and a molecular formula is assigned incorrectly if the correct molecular formula is not among the assigned molecular formulae. A molecular formula is not assigned if no molecular formulae were assigned to a given mass. The output from all programs was processed with Excel, Notepad++, and a custom Pascal program to count the number of masses that were not assigned a molecular formula, were assigned an incorrect formula, and were assigned the correct molecular formula.

As described earlier, the three computer-generated mass lists contain 53573 + 101912 + 14790 = 170275 masses. All four versions of the CHOFIT programs and Molecular Formula Calculator assigned the correct molecular formula to all 170275 masses. MassCalc/FormCalc assigned 165549 correct formulae. HR2 programs, [*n*-3] and [*n*], assigned the correct molecular formula to 154064 and 154021 masses, respectively, while assigning incorrect molecular formulae to 16006 and 15964 masses, respectively, and failing to assign molecular formulae to 205 and 290 masses, respectively. These minor differences between HR2 versions were not significant and could probably be eliminated by further refinement of the source code. This analysis confirms that the LMM-based algorithm used in CHOFIT [*n*-3] and HR2 [*n*-3] is as comprehensive as a brute-force algorithm.

The results raise the question of why fewer correct molecular formulae were assigned by some programs. Approximately 16250 masses were assigned incorrect molecular formulae or not assigned a molecular formula by the HR2 programs. Many of these formulae had values of H/C < 0.2, so they were classified as invalid molecular formulae and discarded by HR2. MassCalc/FormCalc failed to find correct molecular formulae for 4726 exact masses, including 4695 exact masses that exceed 1000 Da. It is likely that 1000 Da is a built-in upper limit for this program.

CONCLUSIONS

Interpreting the large mass lists from FTICR-MS of complex mixtures such as NOM has necessitated the development of computational methods for assigning the correct molecular formulae to an exact mass. The overall task may be partitioned into the deterministic task of finding all possible molecular formulae for a given mass and the heuristic task of selecting the most probably correct molecular formula for a given mass. This work clearly provides a much faster algorithm for the deterministic task of assigning chemical formulae, enabling the entire SRFA mass list (28514 masses) to be assigned molecular formulae in full mode in only 10.8 s. The [*n*-3] algorithm eliminates nested loops of C, H, and O, which together make the largest contributors to the combinatorial problem, and instead uses LMMs to solve for C, H, and O. For the CHOFIT programs, both the [*n*-3] and [*n*] algorithms correctly assign molecular formulae to all of the masses that were processed. For the HR2 programs, which use more conservative criteria for validity of molecular formulae than does CHOFIT, the [*n*-3] and [*n*] algorithms correctly assign molecular formulae to 90.5% of the masses that were processed. For both CHOFIT and HR2, the LMM-based [*n*-3] program was 3 orders of magnitude faster than the [*n*] program. The LMM-based [*n*-3] algorithm thus achieves a major increase in the rate of assignment of molecular formulae without any loss in comprehensiveness. For the mass lists used in this study, the [*n*-3] version of CHOFIT is 3.6 times faster than MassCalc, 60 times faster than Molecular Formula Calculator, and 1050 times faster than the open-source version of HR2.

Because the LMM-based [*n*-3] algorithm greatly reduces the computational time that is required for the deterministic task, that task can be expanded by adding nested loops for a significant number of additional elements/isotopes without adversely affecting the overall time required to process a mass list. Alternatively, more computational time can be devoted to the challenging heuristic task of selecting the best molecular formula from among many possible molecular formulae. At the very least, the much greater speed of the LMM-based [*n*-3] algorithm will enable more rapid processing of FTICR-MS mass lists, thus increasing productivity in laboratories that generate and interpret FTICR-MS data.

ASSOCIATED CONTENT

Supporting Information

CHOFIT [*n*-3] program source code, examples of CHOFIT programs (Figures 1 and 2), and Excel file giving mass lists used in formula assignment. This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: emperdue@bsu.edu. Phone: +0017652858096.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank Dr. Greg Blakney for compiling a custom version of the NHFML Molecular Formula Calculator and for his insights. We also thank Prof. Dr. Philippe Schmitt-Kopplin of Helmholtz Zentrum Muenchen, who provided MassCalc/FormCalc for our use in this study. Both he and Dr. Moritz Frommberger contributed helpful insights.

REFERENCES

(1) Kim, S.; Rodgers, R. P.; Marshall, A. G. *Int. J. Mass Spectrom.* **2006**, *251*, 260−265 DOI: 10.1016/j.ijms.2006.02.001.

(2) Mopper, K.; Stubbins, A.; Ritchie, J. D.; Bialk, H. M.; Hatcher, P. G. *Chem. Rev.* **2007**, *107*, 419−422 DOI: 10.1021/cr050359b.

(3) Stubbins, A.; Spencer, R. G. M.; Chen, H.; Hatcher, P. G.; Mopper, K.; Hernes, P. J.; Mwamba, V. L.; Mangangu, A. M.; Wabakanghanzi, J. N.; Six, J. *Limnol. Oceanogr.* **2010**, *55*, 1467−1477 DOI: 10.4319/lo.2010.55.4.1467.

(4) Chen, H.; Stubbins, A.; Perdue, E. M.; Green, N. W.; Helms, J. R.; Mopper, K.; Hatcher, P. G. *Mar. Chem.* **2014**, *164*, 48−59 DOI: 10.1016/j.marchem.2014.06.002.

(5) Lechtenfeld, O. J.; Kattner, G.; Flerus, R.; Mccallister, S. L.; Schmitt-Kopplin, P.; Koch, B. P. *Geochim. Cosmochim. Acta* **2014**, *126*, 321−337 DOI: 10.1016/j.gca.2013.11.009.

(6) Tfaily, M. M.; Hamdan, R.; Corbett, J. E.; Chanton, J. P.; Glaser, P. H.; Cooper, W. T. *Geochim. Cosmochim. Acta* **2013**, *112*, 116−129 DOI: 10.1016/j.gca.2013.03.002.

(7) Stenson, A. C.; Marshall, A. G.; Cooper, W. T. *Anal. Chem.* **2003**, *75*, 1275−1284 DOI: 10.1021/ac026106p.

(8) Kujawinski, E. B.; Behn, M. D.. *Anal. Chem.* **2006**, *78*, 4363−4373 DOI: 10.1021/ac0600306.

(9) Kind, T.; Fiehn, O. *BMC Bioinf.* **2007**, *8*, 105 DOI: 10.1186/1471-2105-8-105.

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094

(10) Koch, B. P.; Dittmar, T.; Witt, M.; Kattner, G. *Anal. Chem.* **2007**, *79*, 1758−1763 DOI: 10.1021/ac061949s.
(11) Kunenkov, E. V.; Kononikhin, A. S.; Perminova, I. V.; Hertkorn, N.; Gaspar, A.; Schmitt-Kopplin, P.; Popov, I. A.; Garmash, A. V.; Nikolaev, E. N. *Anal. Chem.* **2009**, *81*, 10106−10115 DOI: 10.1021/ac901476u.
(12) Grinhut, T.; Lansky, D.; Gaspar, A.; Hertkorn, N.; Schmitt-Kopplin, P.; Hadar, Y.; Chen, Y. N. *Rapid Commun. Mass. Spectrom.* **2010**, *24*, 2831−2837 DOI: 10.1002/Rcm.4709.
(13) Tziotis, D.; Hertkorn, N.; Schmitt-Kopplin, P. *Eur. J. Mass Spectrom.* **2011**, *17*, 415−421 DOI: 10.1255/ejms.1135.
(14) Blakney, G. (National High Magnetic Field Laboratory, NHMFL). Personal communication, 2014.
(15) Schmitt-Kopplin, P.; Frommberger, M. Personal communication, 2014.
(16) Senior, J. K. *Am. J. Math.* **1951**, *73*, 663−689.
(17) Perdue, E. M.; Green, N. W. *Anal. Chem.* **2015**, *87*, DOI: 10.1021/ac504165k.
(18) de Laeter, J. R.; Böhlke, J. K.; de Bièvre, P.; Hidaka, H.; Peiser, H. S.; Rosman, K. J. R.; Taylor, P. D. P. *Pure Appl. Chem.* **2003**, *75*, 683−800 DOI: 10.1351/pac200375060683.
(19) Koprivnjak, J. F.; Pfromm, P. H.; Ingall, E.; Vetter, T. A.; Schmitt-Kopplin, P.; Hertkorn, N.; Frommberger, M.; Knicker, H.; Perdue, E. M. *Geochim. Cosmochim. Acta* **2009**, *73*, 4215−4231 DOI: 10.1016/j.gca.2009.04.010.

DOI: 10.1021/ac504166t
Anal. Chem. 2015, 87, 5086−5094