analytical chemistry

Article

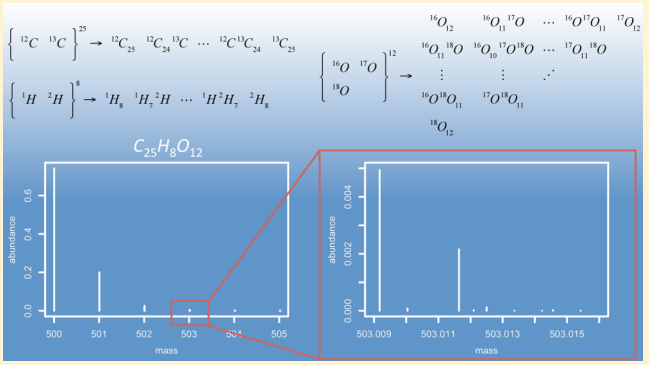
pubs.acs.org/ac

Efficient Calculation of Exact Fine Structure Isotope Patterns via the Multidimensional Fourier Transform

Andreas Ipsen

Institute of Mass Spectrometry, College of Medicine, Swansea University, Swansea SA2 8PP, U.K.

ABSTRACT: The isotope patterns of unknown analytes provide information that can be of great value in their identification as part of a mass spectrometry experiment. Determining the range of compounds that are consistent with an empirically observed isotope pattern requires, as an initial step, the calculation of the theoretical isotope patterns of all feasible candidate formulas, and this is not a trivial mathematical task. While algorithms based on the Fourier transform have been used for almost two decades to perform such calculation efficiently, they have hitherto not been able to provide the exact sets of masses and abundances that constitute the fundamental isotope pattern. This article presents a new approach to the treatment of such calculations, which involves arranging and manipulating the isotope patterns of distinct elements as multidimensional data structures. This enables the use of the multidimensional Fourier transform to calculate isotope patterns with an accuracy that is limited only by the errors of floating point arithmetic. The algorithm is both highly efficient and very easy to implement in many programming environments. An open-source implementation of the algorithm in the R programming language will be made publicly available and is also available upon request.



The analytical utility of isotope patterns as a means of establishing the identity of unknown compounds in biological mass spectrometry experiments has become more recognized in recent years,[1,2] alongside an understanding that mass estimates alone are often not sufficient for this task.[3] In a parallel development, the resolution of high-end mass spectrometers is gradually reaching the point where fine structure isotope patterns can be distinguished.[4,5] That is, they are becoming capable of resolving the mass peaks of an analyte, not only by the number of neutrons they contain, but also by how these are distributed among the distinct elements that make up the analyte, for a fixed neutron count. This information is of great value to analysts as it can be used to place further constraints on the possible identities of unknown analytes.[6]

But to determine whether a putative molecular formula is consistent with an observed isotope pattern, the theoretical isotope pattern of the formula must first be calculated. Over the past 50 years, numerous algorithms have been proposed for this task, as has been discussed in a recent review.[7] A substantial subset of these[8−12] do not compute the full isotopic fine structure but make the simplifying assumption that isotopologues containing the same numbers of neutrons may be grouped together since their mass differences are generally very small.[13] This greatly reduces the computational and memory demands of such algorithms, and the approximation is entirely acceptable for most current instruments. However, it is likely to become increasingly problematic as mass spectrometers with very high resolution, capable of distinguishing parts of the isotopic fine structure, become more widely available.

An important class of algorithms for computing isotope patterns are those based on the use of the discrete Fourier transform, which exploit the properties of the fast Fourier transform to perform the required calculations with great computational efficiency. Such methods may make the above assumption that isotopologues with an equal neutron count can be grouped together, and thereby provide approximate isotope patterns.[8] But more commonly, Fourier-based methods are used to calculate "profile-mode" isotope patterns, that is, rather than calculate the underlying masses and abundances associated with each distinct isotopologue they sample the superposition of mass peaks that these would induce over a set of m/z values, given a user-specified mass peak width. A number of variations on this approach have been developed,[14−19] including a recent algorithm that makes use of the 2-dimensional Fourier transform to significantly reduce the computational demands by separating the treatment of the isotopic fine structure from that of isotopologues with distinct neutron counts.[20]

While profile-mode Fourier-based methods can in principle resolve the underlying isotopic fine structure to an arbitrary degree by decreasing the mass peak width and increasing the number of m/z values sampled,[7] this is generally not a practical approach because of its memory requirements. Currently, the

Received: January 9, 2014
Accepted: April 28, 2014
Published: May 19, 2014

   dx.doi.org/10.1021/ac500108n | *Anal. Chem.* 2014, 86, 5316−5322

$$
\begin{array}{lccccccccc}
 & {}^{13}C_0 & {}^{13}C_1 & {}^{13}C_2 & {}^{13}C_3 \ \cdots\ {}^{13}C_{i-1} & {}^{13}C_i & {}^{13}C_{i+1} \ \cdots\ {}^{13}C_{N-1} & {}^{13}C_N \\
\boldsymbol{C}_1 = & \left(P(^{12}C), \right. & P(^{13}C), & 0, \cdots & & & & \cdots 0 & \left. \right) \\
\boldsymbol{C}_2 = & \left(P(^{12}C)^2, \right. & 2P(^{12}C)P(^{13}C), & P(^{13}C)^2, & 0, \cdots & & & \cdots 0 & \left. \right) \\
\vdots & & & & & & & & \\
\boldsymbol{C}_i = & \left(P(^{12}C)^i, \right. & iP(^{12}C)^{i-1}P(^{13}C), \cdots & & \cdots iP(^{12}C)P(^{13}C)^{i-1}, & P(^{13}C)^i, & 0, \cdots & \cdots 0 & \left. \right) \\
\vdots & & & & & & & & \\
\boldsymbol{C}_N = & \left(P(^{12}C)^N, \right. & NP(^{12}C)^{N-1}P(^{13}C), \cdots & & & & \cdots NP(^{12}C)P(^{13}C)^{N-1}, & P(^{13}C)^N & \left. \right)
\end{array}
$$

Figure 1. Set of N vectors, all of length $N + 1$, that provide convenient and consistent data structures for storing and manipulating the isotope patterns of C_1, ..., C_N. The first entry of each vector corresponds to the isotopic variant composed purely of ^{12}C, while subsequent entries corresponds to variants that are increasingly composed of ^{13}C as indicated above the vectors in red.

most efficient methods of computing the underlying isotopic fine structure are not via the use of the Fourier transform but through the recursive properties of the polynomials that describe it.[21] While modern recursive methods can provide the exact isotopic fine structure in an efficient manner,[22−24] a procedure for doing so via the Fourier transform would clearly be desirable both as a practical tool and as a conceptual advance in addressing this problem. Such an algorithm, which uses a geometric structure called a *k-simplex* to describe the distribution of neutrons among the isotopologues of groups of distinct elements, is presented below.

THEORY

As is the case with certain other algorithms[21,24] for the calculation of the exact fine structure isotope pattern of a molecular formula, the current method is initially applied to each group of distinct elements whose abundances are calculated separately. That is, to calculate the isotope pattern of, say, $C_{254}H_{338}N_{65}O_{75}S_6$, the isotope patterns of each of C_{254}, H_{338}, N_{65}, O_{75}, and S_6 are first calculated separately. The isotopic abundances of the whole molecule can then be obtained through the outer product of the distinct elemental abundances, while the corresponding masses can be obtained from the corresponding "outer sum" of masses. Thus, if the vectors

$$u = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{254} \end{bmatrix} \text{ and } v = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{338} \end{bmatrix} \tag{1}$$

contain the isotopic abundances of C_{254} and H_{338}, respectively, their outer product will yield the matrix

$$w = u \otimes v = \begin{bmatrix} u_0v_0 & u_0v_1 & \cdots & u_0v_{338} \\ u_1v_0 & u_1v_1 & & \\ \vdots & & \ddots & \vdots \\ u_{254}v_0 & & \cdots & u_{254}v_{338} \end{bmatrix} \tag{2}$$

which contains all the isotopic abundances of $C_{254}H_{338}$. Taking the outer product of this matrix with the vectors containing the isotopic abundances of N_{65}, O_{75}, and S_6 will then yield a 5-dimensional array containing all the isotopic abundances of $C_{254}H_{338}N_{65}O_{75}S_6$. The corresponding masses are obtained in an analogous manner, the only differences being that the vectors used should contain the masses of the isotopic variants of C_{254}, H_{338}, N_{65}, O_{75}, and S_6 and their entries should be summed rather than multiplied in eq 2. Regardless of the manner in which the isotope patterns are computed, these calculations (or a close variation on them) will be required, and in many cases they will constitute the most computationally demanding step of the algorithm, because of the large number of terms involved.

Elements with Two Stable Isotopes. The nature of the present algorithm will depend on the number of stable isotopes of the element to which it is applied. The simplest nontrivial case is for the class of elements, such as carbon, that have two stable isotopes.

Suppose there are N carbon atoms in the molecular formula under consideration. Then, the preliminary goal is to calculate the probability that an arbitrary sample of N such atoms is composed of n ^{13}C atoms and $(N - n)$ ^{12}C atoms, based on the natural abundances of the two isotopes, $P(^{12}C)$ and $P(^{13}C)$. This probability should be calculated for n ranging from 0 to N to determine the full isotope pattern of C_N.

The formalism used to describe the data structures needed for this problem can be made relatively intuitive. Let $\boldsymbol{C}_1$ be a vector of length $N + 1$, whose first two entries, $\boldsymbol{C}_1[0]$ and $\boldsymbol{C}_1[1]$, are the natural abundances of ^{12}C and ^{13}C with the remaining entries being 0. Let the vectors $\boldsymbol{C}_2$, ..., $\boldsymbol{C}_N$ be defined analogously so that they contain the isotopic abundances of C_2, ..., C_N. As has previously been proposed,[25,26] the nonzero terms of each of these vectors can be expressed through the expansion of the polynomial $(P(^{12}C) + P(^{13}C))^i$ as shown in Figure 1.

Since $\boldsymbol{C}_i[0]$ refers to the first entry of $\boldsymbol{C}_i$, $\boldsymbol{C}_i[n]$ provides the abundance of the isotopic variant with n ^{13}C atoms and $(i - n)$ ^{12}C atoms. Because of the manner in which the convolution operation will be applied to these vectors, it is useful to interpret the $\boldsymbol{C}_i$ as being periodic vectors such that $\boldsymbol{C}_i[n] = \boldsymbol{C}_i[n + N + 1]$.

As has been noted by Meija[27] in the context of isotopic gross structures, the binomial expansion that describes the isotope pattern of an element with two isotopes is related to Pascal's triangle. More specifically, the binomial coefficients in $\boldsymbol{C}_i$ match the ith row of Pascal's triangle, when following the convention that the top row of the triangle is the zeroth row. In addition, the arrangement of the nonzero terms of the vector $\boldsymbol{C}_i$ can be compared to the points of a discrete 1-simplex of length $i + 1$ (which is simply a set of $i + 1$ points arranged in one direction). Although this relationship is trivial in the present case, it can be

dx.doi.org/10.1021/ac500108n | Anal. Chem. 2014, 86, 5316−5322

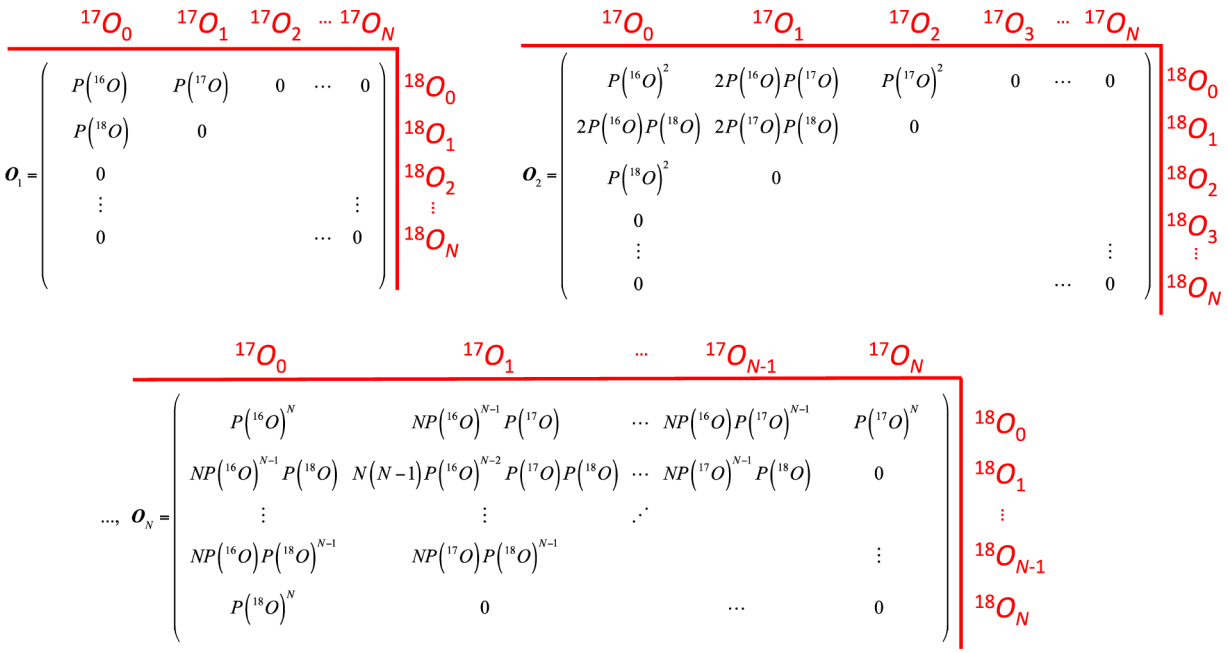


Figure 2. Isotope patterns of O_1, ..., O_N can be stored in the upper left triangles of $(N + 1) \times (N + 1)$ matrices to facilitate their mathematical manipulation. The first entry of each matrix corresponds to the isotopic variant composed purely of ^{16}O. Entries of higher columns have more ^{17}O isotopes, while entries of higher rows have more ^{18}O isotopes, as indicated in red.

generalized for elements with more than two isotopes as will be discussed in the next subsection.

By definition, the vector $\mathbf{C}_N$ contains the isotope pattern of C_N, which is the desired quantity. The key to using the Fourier transform to compute $\mathbf{C}_N$ is to note that it can be obtained from the N-fold convolution of $\mathbf{C}_1$ with itself. The convolution of two (periodic) vectors of length $N + 1$, $\mathbf{U}$, and $\mathbf{V}$, is given by

$$(\mathbf{U} \otimes \mathbf{V})[n] = \sum_{k=0}^{N} \mathbf{U}[k]\mathbf{V}[n - k] \tag{3}$$

But if these vectors are defined as $\mathbf{U} = \mathbf{C}_1$ and $\mathbf{V} = \mathbf{C}_i$, this convolution is

$$\begin{aligned}(\mathbf{C}_1 \otimes \mathbf{C}_i)[n] &= \sum_{k=0}^{N} \mathbf{C}_1[k]\mathbf{C}_i[n - k] \\ &= P(^{12}\mathrm{C})\mathbf{C}_i[n] + P(^{13}\mathrm{C})\mathbf{C}_i[n - 1] \\ &= P(^{12}\mathrm{C})\left[\frac{i!}{(i - n)!n!}P(^{12}\mathrm{C})^{i-n}P(^{13}\mathrm{C})^{n}\right] \\ &+ P(^{13}\mathrm{C})\left[\frac{i!}{(i - n + 1)!(n - 1)!}P(^{12}\mathrm{C})^{i-n+1}P(^{13}\mathrm{C})^{n-1}\right] \\ &= \frac{(i + 1)!}{(i - n + 1)!n!}P(^{12}\mathrm{C})^{i+1-n}P(^{13}\mathrm{C})^{n} \\ &= \mathbf{C}_{i+1}[n]\end{aligned} \tag{4}$$

so that, starting from $\mathbf{C}_1$, each of $\mathbf{C}_2$, ..., $\mathbf{C}_N$ can be generated by repeated convolution with $\mathbf{C}_1$.

But by the convolution theorem,[28] the convolution of $\mathbf{C}_i$ and $\mathbf{C}_1$ can also be computed via the discrete Fourier transform through

$$\mathbf{C}_1 \otimes \mathbf{C}_i = F^{-1}\{F\{\mathbf{C}_1\}F\{\mathbf{C}_i\}\} \tag{5}$$

where the Fourier transformed vectors are multiplied component-wise: a practice that will be maintained for all Fourier transformed vectors and arrays hereafter.

Consequently, it is clear that by repeated use of the convolution theorem the N-fold convolution of $\mathbf{C}_1$ with itself can be obtained from

$$\mathbf{C}_N = F^{-1}\{F\{\mathbf{C}_1\}^N\} \tag{6}$$

That is, the full set of isotopic abundances of C_N can be obtained by taking the Fourier transform of $\mathbf{C}_1$, raising the resulting entries to the power of N, and taking the inverse Fourier transform of the output. Note that the $(N - 1)$ 0s in the initial vector $\mathbf{C}_1$, are required to give the output vector the appropriate length and to avoid any aliasing errors.

The algorithm may so far be regarded as a variation on the conventional Fourier-based methods, wherein the origin is shifted to the mass of the lowest-mass isotopic variant and the equidistant interval on which the Fourier transform operates is the mass difference of the two isotopes of the element examined. However, with the current method, the abundances of the distinct isotopologues are associated strictly with their exact masses, which are calculated separately, with the mass of the isotopologue with $(N - n)$ ^{12}C atoms and n ^{13}C atoms being given by the simple expression:

$$\begin{aligned}&M(^{12}\mathrm{C} = N - n, {}^{13}\mathrm{C} = n) \\ &\quad = (N - n)m_a(^{12}\mathrm{C}) + nm_a(^{13}\mathrm{C})\end{aligned} \tag{7}$$

where $m_a(^{12}C)$ and $m_a(^{13}C)$ are the atomic masses of ^{12}C and ^{13}C.

Elements with Three Stable Isotopes. The situation is more complicated for elements, such as oxygen, that have three stable isotopes. Since the mass differences associated with the distinct isotopes are not integer multiples of one another, the above procedure cannot be extended by simply adding the additional isotopic abundances in the initial vector ($\mathbf{C}_1$ in the previous section). Rather, a second dimension must be added to the system in order to account for the full set of distinct isotopic variants that can be produced. Nevertheless, the

dx.doi.org/10.1021/ac500108n | Anal. Chem. 2014, 86, 5316−5322

procedure for calculating the isotope patterns remains closely analogous to that described in the previous subsection.

If there are N oxygen atoms in the molecular formula under consideration, then all the abundances that can result from every possible combination of the three stable isotopes with abundances $P(^{16}O)$, $P(^{17}O)$, and $P(^{18}O)$, must be determined. As before, intuitive data structures, $\boldsymbol{O}_1$, $\boldsymbol{O}_2$, ...,, $\boldsymbol{O}_N$, can be defined, that contain the isotope patterns of O_1, O_2, ..., O_N. Their terms can be expressed through the expansion of the polynomial $(P(^{16}O)+P(^{17}O)+P(^{18}O))^i$ and can be contained within $(N + 1) \times (N + 1)$ matrices as shown in Figure 2.

From these expressions it is apparent that the terms of the isotope pattern can be regarded as corresponding to the points of a discrete 2-simplex (a discrete triangle), which in this case are stored in a top left triangle of the matrices. Each corner of the simplex corresponds to isotopic variants that consist solely of ^{16}O, ^{17}O, or ^{18}O, and the compositions of terms in between can be determined from their positions relative to these corners. For example, the entry in the second column and second row of $\boldsymbol{O}_N$ corresponds to the abundance of $^{16}O_{N-2}{}^{17}O_1{}^{18}O_1$, which is $N(N-1)P(^{16}O)^{N-2}P(^{17}O)^1P(^{18}O)^1$. While the powers to which the isotopic abundances are raised in each term correspond to the isotopic composition, the trinomial coefficients match the Nth layer of Pascal's 3-simplex (Pascal's tetrahedron), if its top layer is regarded as the zeroth layer.

This interpretation generalizes straightforwardly to higher dimensions for elements with more than three isotopes. If there are I isotopes, the isotope pattern can be stored in a discrete $(I - 1)$-simplex (which is the notion of a discrete triangle generalized to the $(I - 1)$th dimension), and the Nth "layer" of Pascal's I-simplex[29] provides the multinomial coefficients for all possible permutations of the isotopologues of N such atoms. While isotope patterns are often explained with reference to the multinomial expansion, this simplex-based interpretation can make the relationships of the terms somewhat easier to visualize.

Just as multiple convolutions of the vector $\boldsymbol{C}_1$ with itself can be used to generate the isotope pattern of C_N, the analogous operation for matrices can be used to generate the isotope pattern of O_N from $\boldsymbol{O}_1$. If $\boldsymbol{U}$ and $\boldsymbol{V}$ are $(N + 1) \times (N + 1)$ periodic matrices, their convolution is given by

$$(\mathbf{U} \otimes \mathbf{V})[n_1, n_2] = \sum_{k_2=0}^{N} \{ \sum_{k_1=0}^{N} \mathbf{U}[k_1, k_2]\mathbf{V}[n_1 - k_1, n_2 - k_2]\} \tag{8}$$

and based on arguments similar to those used previously, it can be shown that

$$\mathbf{O}_i \otimes \mathbf{O}_1 = \mathbf{O}_{i+1} \tag{9}$$

Moreover, the convolution theorem generalizes to two dimensions,[30] so that the above convolution can be obtained through

$$\mathbf{O}_i \otimes \mathbf{O}_1 = {F_{(2)}}^{-1}\{F_{(2)}\{\mathbf{O}_i\}F_{(2)}\{\mathbf{O}_1\}\} \tag{10}$$

Here, $F_{(2)}$ denotes the 2-dimensional discrete Fourier transform, which can easily be obtained by applying the conventional 1-dimensional Fourier transform to each row of $\boldsymbol{O}_1$, and then to each column of the result (the order is arbitrary). As before, it is clear that by repeated use of the 2-dimensional convolution theorem, the N-fold convolution of $\boldsymbol{O}_1$ with itself, provides the isotopic abundances of O_N, through

$$\mathbf{O}_N = {F_{(2)}}^{-1}\{F_{(2)}\{\mathbf{O}_1\}^N\} \tag{11}$$

As before, the masses of O_N are calculated separately from the abundances, and are obtained through the expression:

$$M(^{16}O = N - n_1 - n_2,\ ^{17}O = n_1,\ ^{18}O = n_2) = (N - n_1 - n_2)m_a(^{16}O) + n_1 m_a(^{17}O) + n_2 m_a(^{18}O) \tag{12}$$

which is trivial to calculate for all isotopic variants.

General Case. For elements with four or more isotopes, the procedure is entirely analogous, as the simplex model of the isotope patterns, the Fourier transform, and the convolution theorem all generalize straightforwardly to higher dimensions. Thus, the algorithm can be summarized as follows when the full isotopic abundance pattern of N copies of the element E, with I stable isotopes is to be calculated:

(1) Define $\boldsymbol{E}_1$ to be an $(I - 1)$-dimensional array, where each side is of length $N + 1$, where the I entries in, and around, the base corner contain the natural abundances of E, and all other entries are 0.

(2) Calculate

$$\mathbf{E}_N = F_{(I-1)}^{-1}\{F_{(I-1)}\{\mathbf{E}_1\}^N\} \tag{13}$$

where $F_{(I-1)}$ is the $(I - 1)$-dimensional discrete Fourier transform. $\boldsymbol{E}_N$ will be of the same dimensions as $\boldsymbol{E}_1$, but will contain the isotopic abundance pattern of E_N in the entries that comprise the discrete $(I - 1)$-simplex of side length $N + 1$, whose edges match those of the base corner of $\boldsymbol{E}_N$. All other entries will be zero.

(3) Calculate the corresponding masses through

$$M(^{(1)}E = N - \sum_{i=2}^{I} n_{i-1}\,{}^{(2)}E = n_1, \ldots {}^{(I)}E = n_{I-1}) = (N - \sum_{i=2}^{I} n_{i-1})m_a(^{(1)}E) + \sum_{i=2}^{I} n_{i-1} m_a(^{(i)}E) \tag{14}$$

where the quantities $^{(1)}E$, ..., $^{(I)}E$ enumerate the I stable isotopes of E.

(4) The results may be combined with those obtained from different elements by calculating the outer product of the abundances and the "outer sum" of the masses.

In practice, it is advisible to discard extremely low-abundance isotopic variants when calculating the isotopic abundance patterns of larger molecules, since the memory requirements may otherwise become unmanageable. This is true irrespective of the algorithm used to calculate the abundances. Such "pruning" may initially be done separately for each elemental group, prior to calculating the outer product of their isotopic abundances, as it is typically only following their combinations that memory requirements become very demanding. In the author's implementation of the present algorithm, a threshold, (e.g., 10^{-8}) is specified by the user, and as the consecutive outer products and sums are taken, a running evaluation is made of terms that will fall below this limit and these are filtered out. More generally the most appropriate pruning algorithm will depend on the programming language used and on the type of pruning required, for example, whether it is sufficient to discard

dx.doi.org/10.1021/ac500108n | Anal. Chem. 2014, 86, 5316−5322

low-abundance isotopic variants, whether a minimum proportion of the total isotopic abundance should be obtained, or whether the isotopic variants obtained should be within a specific mass range.

PERFORMANCE COMPARISON

The time performance of an implementation of the present algorithm in the R programming language, called *ecipex*, was evaluated against that of *Isotope Calculator*,[24] which is a recursive method written in C++. Another modern recursive method, *isoDalton*[22] was not included in the comparison as it is implemented in a commercial programming language (MATLAB). However, since MATLAB is an interpreted language and since the algorithm on which *isoDalton* is based involves more intermediate calculations, it is not likely to be as fast as *Isotope Calculator*.

The comparison of *ecipex* and *Isotope Calculator* is somewhat problematic as they are written in different languages, with C++ generally being considerably faster than R (also an interpreted language). Furthermore, to minimize memory requirements the algorithm on which *Isotope Calculator* is based does not calculate the full isotope pattern in memory. Like the present algorithm, it initially calculates the isotope patterns of the distinct elemental groups that a formula consists of, but *Isotope Calculator* only outputs the final combined isotope pattern to disk rather than to memory. Since writing to disk is a very time-consuming operation, the time measured for *Isotope Calculator* was only that of the initial time taken to "load" a molecule into memory. This should favor its performance for formulas consisting of two or more distinct elements, since the terms of the outer product and "outer sum" are not calculated.

The programs were run on a 2.7 GHz MacBook Pro with 16GB RAM, running Windows 8.1 via Boot Camp. The threshold, below which isotopic variants were discarded, was set to 10^{-8} for *ecipex* while no such threshold was specified for *Isotope Calculator* since it only applies the pruning as the results are being written to disk. For *ecipex*, the pruning has very little effect on the runtime when only a single elemental group is considered, however when multiple distinct elements are involved, lowering the pruning threshold can induce a considerable increase in runtime due to the large number of terms produced in the outer product and outer sum. The runtimes of the two programs for a number of formulas, some of which have been used in previous isotope calculator comparisons,[9] are listed in Table 1.

Clearly, the performance of *ecipex* is markedly faster for all formulas except Ca_{20}. The runtime of *Isotope Calculator* generally scales better as the number of stable isotopes is increased, but elements with many stable isotopes are relatively rare in biology and they are typically only present in very low copy numbers within individual compounds. The two largest formulas tested, $C_{17600}H_{26474}N_{4752}O_{5486}S_{197}$ and $C_{23832}H_{37816}N_{6528}O_{7031}S_{170}$, cause *Isotope Calculator* to return an error, presumably because of their size.

Table 1. Runtimes of *Isotope Calculator* and *ecipex* for Selected Molecular Formulas[a]

molecular formula	stable isotopes	*Isotope Calculator* (C++) (s)	*ecipex* (R) (s)
$C_{1000000}$	2	12.50	1.63
O_{2000}	3	29.78	4.66
S_{200}	4	26.26	8.21
Ca_{20}	6	2.43	4.65
$C_{2023}H_{3208}N_{524}O_{619}S_{20}$	2, 2, 2, 3, 4	4.80	0.21
$C_{2934}H_{4615}N_{781}O_{897}S_{39}$	2, 2, 2, 3, 4	7.73	0.62
$C_{5047}H_{8014}N_{1338}O_{1495}S_{48}$	2, 2, 2, 3, 4	16.08	2.26
$C_{8574}H_{13378}N_{2092}O_{2392}S_{77}$	2, 2, 2, 3, 4	42.25	10.27
$C_{17600}H_{26474}N_{4752}O_{5486}S_{197}$	2, 2, 2, 3, 4	NA	78.81
$C_{23832}H_{37816}N_{6528}O_{7031}S_{170}$	2, 2, 2, 3, 4	NA	112.64

[a]The numbers of stable isotopes of each of the elements are also listed in the order that they appear in the formulas.

DISCUSSION

As demonstrated in the above comparison, the algorithm presented here can be made very fast, and its accuracy is only limited by the errors associated with floating point arithmetic. Such errors are not likely to be of much consequence as they will usually be orders of magnitude lower than typical deviations from current estimates of the standard natural isotopic abundances.[31] The algorithm's speed relative to that of recursive alternatives will generally depend on the molecular formulas under consideration and on the precise implementation details. It is arguable that the present algorithm is not ideal in its memory-efficiency due to the 0s output for elements with three or more isotopes, requiring $(N + 1)^{I-1}$ terms to be raised to the power of N, whereas an efficient recursive method would typically involve the calculation of $(N + I - 1)!/(N!(I - 1)!)$ terms, with some additional flexibility for pruning them. The computational complexity of the multidimensional fft will be $O((N + 1)^{I-1}\log((N + 1)^{I-1})) = O((I - 1)(N + 1)^{I-1}\log(N + 1))$, which also exceeds the number of terms in the multinomial expansion. However, the performance comparison in the previous section suggests that over the range of elements and atomic copy numbers typically encountered in biology these asymptotic scaling properties can be largely suppressed by other aspects of the algorithms' implementations and interplay with the operations of modern CPUs. Moreover, the present algorithm is relatively easy to parallelize even for a single elemental group, since it can be decomposed into sets of 1-dimensional Fourier transforms that can be applied independently for each dimension of $\boldsymbol{E}_1$, as can the exponentiation of the resulting values. An efficient parallel implementation of a recursive method would be more problematic because of the inherently sequential nature of the calculations. The present algorithm is furthermore likely to perform considerably better than recursive alternatives when implemented in programming languages that require expressions to be vectorized in order for them to be computed efficiently, since doing so is often impractical for the recursive methods. Finally, there may be scope for some improvements in the computational efficiency of the present algorithm through the use of more specialized Fourier transforms that are better tailored to the simplex character of isotope distributions, though this would in any case require a more complicated implementation.

The present algorithm also has properties that facilitate the calculation of multiple isotope patterns. Note that while step 2 of the algorithm provides the abundances of E_N, it can be used to calculate the abundances of E_i, for any $i < N$, simply by raising the Fourier-transformed array to the power of i instead of N. This obviates the need to define separate arrays, and perform separate initial Fourier transforms when the isotopic abundance patterns of multiple E_i need to be calculated. However, there is a certain trade-off in the computational resources this saves versus those added by the use of an array of

dx.doi.org/10.1021/ac500108n | Anal. Chem. 2014, 86, 5316−5322

side length $N + 1$ rather than the minimally required $i + 1$. This property is therefore most useful when calculating the isotope pattern for a set of E_i that can easily be divided into a small number of groups that are composed of comparable numbers of atoms. For example, the isotope patterns of C_9, C_{10}, C_{999}, and C_{1000}, could all be calculated by using a single Fourier-transformed $\boldsymbol{C}_1$ vector of length 1001, but it would be faster to define an additional $\boldsymbol{C}_1$ vector of length 11, for the calculation of C_9 and C_{10}. A further consideration is that many implementations of the fft are fastest when applied to vectors whose lengths are highly composite, i.e. have many factors, and slowest when their lengths are large prime numbers. Depending on the fft library used, it may therefore be advisable to increase the side lengths of $\boldsymbol{E}_1$ beyond the minimum although this will make the exponentiation of the Fourier-transformed array slower.

Modern algorithms for computing the Nth power of a number employ methods such as repeated squaring or more general procedures known as addition-chain exponentiation[32] to reduce the number of multiplications required to $O(\log(N))$. For example, the exponentiations in $F\{\boldsymbol{C}_1\}^{15}$ can be calculated with 5 sets of component-wise multiplications through $F\{\boldsymbol{C}_1\}^3((F\{\boldsymbol{C}_1\}^3)^2)^2$, rather than the 14 sets required through repeated component-wise multiplication by $F\{\boldsymbol{C}_1\}$. The former method will calculate $F\{\boldsymbol{C}_1\}^2$, $F\{\boldsymbol{C}_1\}^3$, $F\{\boldsymbol{C}_1\}^6$, and $F\{\boldsymbol{C}_1\}^{12}$ as intermediate results, so that $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, $\boldsymbol{C}_6$, and $\boldsymbol{C}_{12}$ can be obtained via the inverse Fourier transform at a relatively small additional computational cost. An elaborate implementation of the present algorithm might exploit this feature when calculating multiple isotope patterns by determining an efficient addition sequence that provides the required powers of $F_{(I-1)}\{\boldsymbol{E}_1\}$ as intermediate results and applying the inverse Fourier transform to these, where doing so is deemed computationally cheaper than performing the full calculations on smaller arrays.

The widespread availability of comprehensive fft libraries makes the core calculation of the present algorithm very easy to implement in many programming environments. For example if **arrayS** is a 51 × 51 × 51 array containing the natural abundances of sulfur in the four entries in, and around, its base corner and 0s everywhere else, the following expression is sufficient to generate the full set of abundances for S_{50} in the R statistical programming language:

```
fft(fft(arrayS)^50, inverse = TRUE)/51^3
```

The abundances must still be extracted and combined with those of other elements, and the corresponding masses must be calculated, but those are relatively trivial computational tasks. Since a multidimensional fft can easily be obtained from the conventional 1-dimensional fft, languages lacking direct support for the former procedure are only at a slight disadvantage.

CONCLUSION

Fourier-based methods for calculating isotope patterns have been highly influential since their introduction by Rockwood in the 1990s because of the computational efficiency that they enable. Through the algorithm presented here, such methods can now for the first time be employed to calculate the full isotopic fine structure of a molecule, without necessitating any fundamental approximations. The new algorithm is easy to implement and has properties that facilitate the calculation of the isotopic abundance patterns of multiple formulas.

Although current instruments are very limited in their ability to measure fine structure isotope patterns reliably, more accurate and higher resolution instruments capable of partially resolving the isotopic fine structure are likely to become more readily available over the coming years. As they do, computational pipelines for the identification of unknown compounds that aim to make full use of the information in the output data will need to incorporate methods for computing and testing the fit of the isotopic fine structures of all feasible formulas. Efficient and accessible procedures for performing the required calculations such as those proposed here may help to address this need.

AUTHOR INFORMATION

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The author wishes to thank Gareth Brenton for valuable discussions. The author is supported by an MRC fellowship in biomedical informatics (Grant No. MR/J013994/1).

REFERENCES

(1) Kind, T.; Fiehn, O. *BMC Bioinf.* **2007**, *8*, 105.
(2) Böcker, S.; Letzel, M. C.; Lipták, Z.; Pervukhin, A. *Bioinformatics* **2009**, *25*, 218−224.
(3) Kind, T.; Fiehn, O. *BMC Bioinf.* **2006**, *7*, 234.
(4) Shi, S. D.-H.; Hendrickson, C. L.; Marshall, A. G. *Proc. Natl. Acad. Sci. U. S. A.* **1998**, *95*, 11532−11537.
(5) Miladinović, S. M.; Kozhinov, A. N.; Gorshkov, M. V.; Tsybin, Y. O. *Anal. Chem.* **2012**, *84*, 4042−4051.
(6) Nagao, T.; Yukihira, D.; Fujimura, Y.; Saito, K.; Takahashi, K.; Miura, D.; Wariishi, H. *Anal. Chim. Acta* **2014**, *813*, 70−76.
(7) Valkenborg, D.; Mertens, I.; Lemière, F.; Witters, E.; Burzykowski, T. *Mass Spectrom. Rev.* **2012**, *31*, 96−109.
(8) Fernandez-de-Cossio Diaz, J.; Fernandez-de-Cossio, J. *Anal. Chem.* **2012**, *84*, 7052−7056.
(9) Claesen, J.; Dittwald, P.; Burzykowski, T.; Valkenborg, D. *J. Am. Soc. Mass Spectrom.* **2012**, *23*, 753−763.
(10) Böcker, S.; Letzel, M. C.; Lipták, Z.; Pervukhin, A. In *Algorithms in Bioinformatics*; Bücher, P.. Moret, B. M. E., Eds.; Lecture Notes in Computer Science; Springer: Berlin, 2006; pp 12−23.
(11) Olson, M. T.; Yergey, A. L. *J. Am. Soc. Mass Spectrom.* **2009**, *20*, 295−302.
(12) Hsu, C. S. *Anal. Chem.* **1984**, *56*, 1356−1361.
(13) Carrick, A.; Glockling, F. *J. Chem. Soc. Inorg. Phys. Theor.* **1967**, 40−42.
(14) Rockwood, A. L. *Rapid Commun. Mass Spectrom.* **1995**, *9*, 103−105.
(15) Rockwood, A. L.; Van Orden, S. L.; Smith, R. D. *Anal. Chem.* **1995**, *67*, 2699−2704.
(16) Rockwood, A. L.; Van Orden, S. L. *Anal. Chem.* **1996**, *68*, 2027−2030.
(17) Rockwood, A. L.; Van Orden, S. L.; Smith, R. D. *Rapid Commun. Mass Spectrom.* **1996**, *10*, 54−59.
(18) Fernandez-de-Cossio, J. *Anal. Chem.* **2010**, *82*, 1759−1765.
(19) Alves, G.; Ogurtsov, A. Y.; Yu, Y.-K. *J. Am. Soc. Mass Spectrom.* **2014**, *25*, 57−70.
(20) Fernandez-de-Cossio, J. *Anal. Chem.* **2010**, *82*, 6726−6729.
(21) Yergey, J. A. *Int. J. Mass Spectrom. Ion Phys.* **1983**, *52*, 337−349.
(22) Snider, R. K. *J. Am. Soc. Mass Spectrom.* **2007**, *18*, 1511−1515.
(23) Li, L.; Kresh, J. A.; Karabacak, N. M.; Cobb, J. S.; Agar, J. N.; Hong, P. *J. Am. Soc. Mass Spectrom.* **2008**, *19*, 1867−1874.
(24) Li, L.; Karabacak, N. M.; Cobb, J. S.; Wang, Q.; Hong, P.; Agar, J. N. *Rapid Commun. Mass Spectrom.* **2010**, *24*, 2689−2696.
(25) Yamamoto, H.; McCloskey, J. A. *Anal. Chem.* **1977**, *49*, 281−283.
(26) Brownawell, M. L.; San Filippo, J. *J. Chem. Educ.* **1982**, *59*, 663.
(27) Meija, J. *J. Chem. Educ.* **2006**, *83*, 1761.

(28) Brigham, E. O. *The Fast Fourier Transform: An Introduction to Its Theory and Application*; Prentice-Hall: Englewood Cliffs, NJ, 1974.
(29) Woods, D.; Kohlenberg, M. J. *Two-Year Coll. Math. J.* **1973**, *4*, 38−43.
(30) Sundararajan, D. *The Discrete Fourier Transform: Theory, Algorithms and Applications*; World Scientific: Singapore, 2001.
(31) Coplen, T. B.; Bohlke, J. K.; De Bievre, P.; Ding, T.; Holden, N. E.; Hopple, J. A.; Krouse, H. R.; Lamberty, A.; Peiser, H. S.; Revesz, K.; Rieder, S. E.; Rosman, K. J. R.; Roth, E.; Taylor, P. D. P.; Vocke, R. D.; Xiao, Y. K. *Pure Appl. Chem.* **2002**, *74*, 1987−2017.
(32) Gordon, D. M. *J. Algorithms* **1998**, *27*, 129−146.