RAPID COMMUNICATIONS IN MASS SPECTROMETRY
Rapid Commun. Mass Spectrom. 2010; **24**: 2689–2696
Published online in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/rcm.4666

Memory-efficient calculation of the isotopic mass states of a molecule

Long Li[1,2], N. Murat Karabacak[1], Jennifer S. Cobb[1], Qi Wang[1], Pengyu Hong[2]* and Jeffrey N. Agar[1]*

[1]Department of Chemistry and Volen Center for Complex Systems, Brandeis University, Waltham, MA 02454
[2]Department of Computer Science, National Center of Behavioral Genomics, Brandeis University, Waltham, MA 02454

Received 3 March 2010; Revised 30 May 2010; Accepted 8 June 2010

Our previous work postulated a transition concept among different isotopic mass states (i.e., isotopic species) of a molecule, and developed a hierarchical algorithm for accurately calculating their masses and abundances. A theoretical mass spectrum can be generated by convoluting a peak shape function to these discrete mass states. This approach suffers from limited memory if a level in the hierarchical structure has too many mass states. Here we present a memory efficient divide-and-recursively-combine algorithm to do the calculation, which also improves the truncation method used in the previous hierarchical algorithm. Instead of treating all of the elements in a molecule as a whole, the new algorithm first 'strips' each element one by one. For the mass states of each element, a hierarchical structure is established and kept in the memory. This process reduces the memory usage by orders of magnitude (e.g., for bovine insulin, memory can be reduced from gigabytes to kilobytes). Next, a recursive algorithm is applied to combine mass states of elements to mass states of the whole molecule. The algorithm described above has been implemented as a computer program called Isotope Calculator, which was written in C++. It is freely available under the GNU Lesser General Public License from http://www.cs.brandeis.edu/~hong/software.html or http://people.-brandeis.edu/~agar. Copyright © 2010 John Wiley & Sons, Ltd.

The calculation of theoretical spectra of molecules is a fundamental area of mass spectrometry. It is crucial for assigning unknown spectra and quantifying isotopically labeled molecules.[1–3] Fourier transform mass spectrometry (FTMS) instruments have resolving powers in the 10^5 to 10^7 range[4–6] and thus can resolve the isotopic fine structure of molecules, including biomolecules such as proteins and lipids. This enables small differences in the elemental compositions of large molecules to be detected. For example, deamidation (which increases the mass by 0.9848 Da) of a 20 kDa protein can be quantified using the isotopic envelope and accurate mass determined by FTMS.[7] The increasing popularity of these FTMS instruments demands algorithms that account for isotopic fine structure when simulating the isotopomer distribution of molecules.

Many methods have been developed for isotopic fine structure calculations. Two popular methods include Yergey's polynomial-based stepwise method[8] and Rockwood's Fourier transform method.[9,10] Most other methods only calculate the nominal mass distribution (no isotopic fine structure).[11,12] Typically, isotopic gross structure appears as a single peak in an experimental spectrum. The peaks of isotopic fine structures appear when the resolving power of an instrument is high enough (see Shi *et al.*[4] for an example of fine structures of proteins at the resolving power of $m/\Delta m \approx 8\,000\,000$) and for such cases methods of calculating only nominal mass distribution are insufficient, and in fact deleterious. Here *peak* is used to refer to the signal in experimental mass spectra, which has a certain width. Probably as a result of calculation speed, most standalone and web-based programs have implemented nominal mass-only algorithms, such as IsoPro[13] and Isotopica.[14] For a brief review of all of these methods, see Li *et al.*[15]

To overcome both memory problems and the accuracy problems associated with prior polynomial-based methods, our previous work[15] postulated a transition concept among different isotopic mass states (i.e., isotopic species) of a molecule, and developed a hierarchical algorithm for accurately calculating their masses and abundances. This transition concept is based on an analogy between the gross and fine isotopic structure of molecules with the gross and fine atomic structures of atomic physics. In atomic physics, only a given number of energy states exist, and a set of rules define a hierarchical relationship between these energy states (e.g., if $n = 0$, then $l = 0$, and $ml = 0$). Similarly, only a given set of 'mass states' exists, for which we could define a hierarchical relationship.[15] For example, the number of nucleons 'A' of isotopic species, which defines the gross isotopic structure, is analogous to n, the principle quantum number, such that the purely monoisotopic form of a molecule represents its 'ground state' and is responsible for the monoisotopic peak (referred to as 'A' or 'M' by mass spectrometrists; referred to a 'A' *ut infra*) in a mass spectrum. Replacing a single abundant atom with its less abundant isotope (e.g., ^{12}C for

**Correspondence to*: P. Hong or J. N. Agar, MS 018 or MS 029, Brandeis University, 415 South Street, Waltham, MA 02454, USA.
E-mail: hongpeng@brandeis.edu; agar@brandeis.edu

^{13}C) increases the nominal mass to A+1, i.e., a lower mass state is excited to a higher mass state. In other words, an *isotopic gross structure* of a molecule is a 'virtual' set including all of mass states that have equal nucleon numbers. These individual mass states are referred to as *isotopic fine structures* of that isotopic gross structure. For instance, the molecule carbon monoxide (CO) has four gross structures, i.e., 28, 29, 30 and 31, if we only consider the stable isotopes ^{12}C, ^{13}C, ^{16}O, ^{17}O, ^{18}O and ignore unstable isotopes such as ^{14}C. For example, within the gross structure of 28, there is only one mass state, $^{12}C^{16}O$, and within the gross structure of 29 are two fine structures, $^{13}C^{16}O$ and $^{12}C^{17}O$.

This hierarchical relationship allows the calculation of the isotopomer distribution starting from the monoisotopic mass and moving to higher A, and thus allows previously calculated nominal mass levels to be dumped to physical memory, making the calculation of larger molecules feasible. A theoretical spectrum can be generated by convoluting a peak shape function (e.g., Gaussian function or Cauchy-Lorentz function) to these discrete mass states. Therefore, our method is more accurate than Yergey's method,[8] and, unlike FT methods,[9,10,16,17] the isotopic composition information of each mass state is maintained.

Although our hierarchical algorithm[15] reduces the memory demand by dispersing the whole mass states to different levels in a hierarchical structure, this algorithm still suffers from memory problems for those molecules whose individual gross structures include many fine structures. For example, the maximum number of the mass states in a single level of bovine insulin is 4.1×10^9, and therefore a computer with 2 gigabytes of RAM could not calculate the whole isotopic distribution.

This paper addresses memory problems when accurately calculating the entire isotopic mass states of a molecule. Instead of treating all elements in a molecule as a whole, the new divide-and-recursively-combine algorithm first 'strips' each element one by one. For the mass states of each element, a hierarchical structure is established and kept in the memory. This process reduces the memory usage significantly (e.g., for bovine insulin, memory was reduced from 10^9 bytes to 10^3 bytes). A recursive algorithm is then applied to combine mass states of elements to mass states of the molecule. Although theorists may be interested in the comprehensive calculation of the mass states of a molecule, in practice it is not necessary to consider all these mass states because many of their abundances are too low to be observed experimentally. Therefore, we have also given users the option of employing truncation to cut off the undetectable levels.

DIVIDE-AND-RECURSIVELY-COMBINE ALGORITHM

This algorithm is based on the transition theory postulated in our previous work.[15] Our previous hierarchical algorithm considers the transition among the mass states of the whole molecule. Here our new divide-and-recursively-combine algorithm only considers the transition among the mass states of each element. The mass states of the molecule can be obtained by recursively combining the mass states from each element together.

Hierarchical structure of a single element molecule

Individual isotopic species of a molecule, defined by a common molecular formula, are considered as different mass states. Here we first look at the mass states of molecules which consist of a single element. Each mass state can be represented by a unique *configuration number* [n_1, n_2, ...], where n_1 is the number of the lightest isotope of the element, n_2 is the number of the second lightest isotope of the element, and so on. Specifically, the ground state includes only the lightest isotopes, i.e., [n, 0, 0, ..., 0], where n is the total number of the atoms of that element; and the highest excited state includes only the heaviest isotopes, i.e., [0,0, ..., 0, n]. For other states, if the nucleon number of a mass state is k more than that of the ground state, this mass state is called the k-th excited state.

We define the element's *transition* as the conversion between two adjacent isotopes (suppose the isotopes have been sorted). Based on this transition concept, all of the mass states can form a hierarchical structure by their nucleon numbers (see Fig. 1). The top level (level 0) only contains the ground state; the bottom level only contains the highest excited state. The nucleon number difference between any two adjacent levels is one. For example, the molecule S_2 has nine different mass states and their configuration numbers are: [2,0,0,0], [1,1,0,0], [0,2,0,0], [1,0,1,0], [0,1,1,0], [0,0,2,0], [1,0,0,1], [0,1,0,1], [0,0,1,1] and [0,0,0,2], if we only consider the stable isotopes ^{32}S, ^{33}S, ^{34}S and ^{36}S. The three different transitions (^{32}S to ^{33}S, ^{33}S to ^{34}S and ^{34}S to ^{36}S) and the hierarchical structure have been demonstrated in Fig. 1.

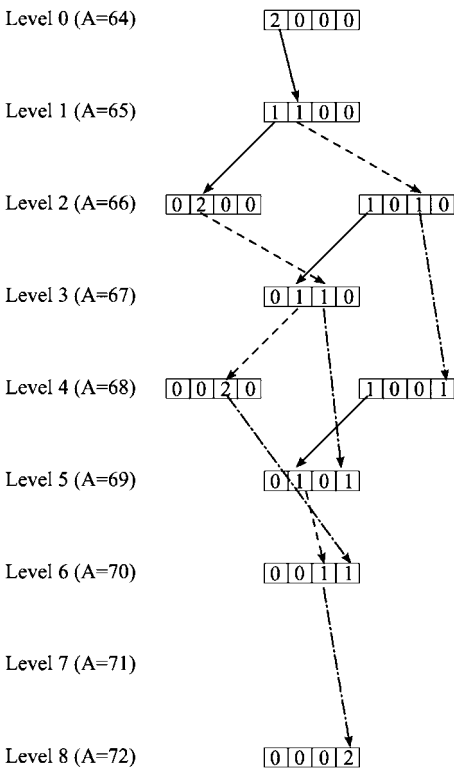


Figure 1. The hierarchical structure of the S_2 molecule. The solid lines, dashed lines, dash-dotted lines denote the transition of ^{32}S to ^{33}S, ^{33}S to ^{34}S and ^{34}S to ^{36}S, respectively. Note: there is no mass state in the level 7.

DOI: 10.1002/rcm

This hierarchical structure allows the mass and the abundances of each mass state to be accurately and quickly computed. The mass and the abundances of the ground state can be calculated as $n^* m_1$ and P_1^n, respectively, where m_1 and p_1 are the mass and the abundance of the lightest isotope of the element. Starting from level 1, the following iterative formulas are used to calculate the mass and the abundance of new transitioned mass state ([..., n_i - 1, $n_{i+1}+1$, ...]) from the previous transitioning mass state ([..., n_i, n_{i+1}, ...]) when considering the transition from the i^{th} isotope to the $(i+1)^{th}$ isotope:

$$m([\ldots, n_i - 1, n_{i+1}+1, \ldots]) = m([\ldots, n_i, n_{i+1}, \ldots]) + m_{i+1} - m_i;$$

$$p([\ldots, n_i - 1, n_{i+1}+1, \ldots]) = p([\ldots, n_i, n_{i+1}, \ldots]) * n_i * p_{i+1}/(n_{i+1} * p_i).$$

Construction of mass states of a molecule

We repeat the process above to calculate the corresponding hierarchical structure of each element of a molecule. If we pull out one mass state from each elemental hierarchical structure, and combine them together, a new molecular mass state forms. The mass of the new molecular mass state is the sum of masses of these elemental mass states, and the abundance of the new molecular mass state is the product of abundances of these elemental mass states.

In our previous work,[15] we showed there also exists a hierarchical structure for a molecule, see Fig. 2 for the example of carbon monoxide. Instead of using molecular transitions as in previously,[15] a recursive algorithm is employed to calculate the mass and abundance of each molecular mass state.

Here we first use CO as an example to demonstrate the recursively combination process and then generalize it to any molecule. As stated earlier, if we want to obtain the mass states of CO, we need grasp a mass state from the hierarchical structure of C and a mass state from the hierarchical structure of O, and combine them together. For example, if we want to obtain the mass states of CO in level 1, the possible levels of C, from which we grasp a mass state, are 0 and 1. Once we decide to grasp a mass state of level 0 of C, the possible level of O is level 1; and the possible level of O is level 0 for level 1 of C. Figure 3 lists all of combinations.

Now we generalize this process for any molecule. We choose the notation that the level starts from zero. Suppose the first element has $(L_{E1}+1)$ levels, the second element has

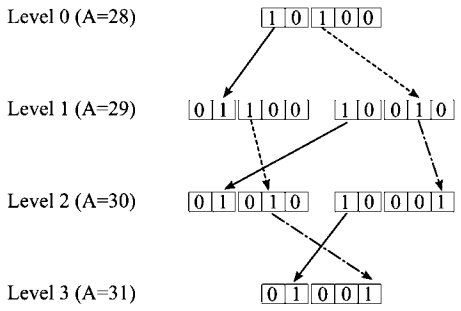
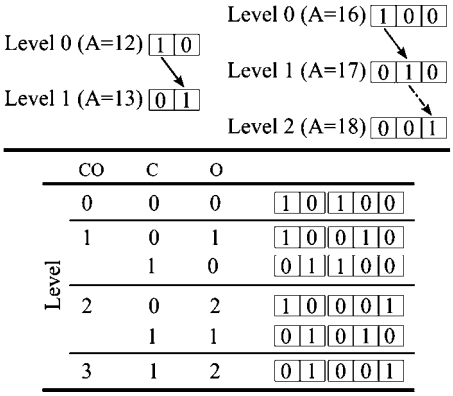


Figure 2. The hierarchical structure of the carbon monoxide molecule. The solid, dashed line and dash-dotted lines stand for the transitions of ^{12}C to ^{13}C, ^{16}O to ^{17}O and ^{17}O to ^{18}O, respectively.



Level	CO	C	O	
	0	0	0	1 0 1 0 0
	1	0	1	1 0 0 1 0
		1	0	0 1 1 0 0
	2	0	2	1 0 0 0 1
		1	1	0 1 0 1 0
	3	1	2	0 1 0 0 1

Figure 3. The divide-and-recursively-combine algorithm on the CO molecule. Only the hierarchical structures of C_1 and O_1 are kept in memory, and the mass states of CO are calculated by recursively combining the mass states of C_1 and O_1 together.

$(L_{E2}+1)$ levels, and so on. The number of whole molecular levels is $L_M = (L_{E1} + L_{E2} + \ldots + L_{Ek} + 1)$ for a molecule which consists of k elements. For any level L ($0<=L<=L_M$) of the molecule, there are many combinations from the elements. For example, let the first element be level L and other elements be level 0, or the second element be level L and other elements be level 0.

The recursive algorithm can be applied here to obtain all combinations of elements. Let us start from the first element. Given a molecular level L, the level range for the first element needs to be determined, designed as $[L_{1min}, L_{1max}]$. If we treat the elements from the second element to the last one as a whole, the possible minimum and maximum levels are 0 and $(L_{E2}+L_{E3}+\ldots+1)$, respectively. Thus, there are two situations for the first element. If $0<=L<=(L_{E2}+L_{E3}+\ldots+1)$, then each level of the first element is possible to combine with other elements, i.e., the range is $[0,\min(L_{E1}, L)]$; if $L > (L_{E2}+L_{E3}+\ldots+1)$, only those levels greater than $L - (L_{E2}+L_{E3}+\ldots+1)$ in the first element can combine with other elements, i.e., the range is $[L_{1min}, L_{1max}] = [L - (L_{E2}+L_{E3}+\ldots+1), \min(L_{E1}, L)]$.

For any level L_1 within the range $[L_{1min}, L_{1max}]$, we need to find the level range for the second element. Imagining now there are only (k – 1) elements and the problem is converted to find the level range of the 'first' element (i.e., E_2) among the (k – 1) elements. It is the same problem to find the range of the first element E_1. This recursive process can be repeated until the last element E_k, see the recursive function **navigateMassLevels** in the following pseudo code.

The above recursive process knows the specific levels of each element. Suppose we get one combination, $(L_1, L_2, ..L_k)$, of a molecular level, L. The problem then becomes how to combine the mass states in L1, L2, etc., to molecular mass states. Usually, there are two or more mass states in each level of elemental hierarchical structure. Again, we still need to use the recursive algorithm to combine these together.

In the level L_1 of the first element, there are S_{L1} elemental mass states; in the level L_2 of the second element, there are S_{L2} elemental mass states; and so on. For the first element, we navigate from the first elemental mass state to the last mass state. Once a mass state of the first element is chosen, we navigate the mass states in the second element. This process

DOI: 10.1002/rcm

continues until the last element, see the recursive function **navigateMassStates** in the following pseudo code. Obviously, the recursive algorithm is the best choice here. It is difficult to use a loop to do this because the number of elements in a molecule is variable.

Here is the pseudo code for the algorithm:

```
; this pseudo code is used to navigate the mass state
Function navigateMassStates (element Ex)
  Global currentLevel [L1, L2, …, Lk]
  Global currentState [S1, S2, …, Sk]
  if element Ex is last element Ek
for Sk = 1 to SLk
  calculate mass and abundance of state [S1, S2, …, Sk]
end
  else
    for Sx = 1 to SLx
  navigateMassStates (element Ex+1)
end
  end
end

; this pseudo code is used to navigate the level
Function navigateMassLevels (element Ex, x2kMoleculeLevel LM)
  Global currentLevel [L1, L2, …, Lk]
  if element Ex is last element Ek
    Lk = LM
navigateMassStates (E1);always start from the first element
  else
if 0<=LM<=(LEx+1+LEx+2+…+LEk+1)
  [Lx,min, Lx,max] = [0, min(LEx,LM)]
else
  [Lx,min, Lx,max] = [LM-(Lx+1+Lx+2+…+Lk+1), min(LRx,LM)]
end
for Lx = Lx,min to Lx,max
      navigateMassLevels (element Ex+1, level LM - Lx)
    end
  end
end
```

Finally, the theoretical spectrum is generated by convoluting a peak shape function to the discrete molecular mass states.[15] The most often used peak shape functions for such purpose are the Gaussian function and the Cauchy-Lorentz function. The contribution of each mass state (its mass is m_k and abundance is p_k) to the whole theoretical spectrum can be

DOI: 10.1002/rcm

calculated by:

$$f(m;\ \sigma) = p_k \exp\left(-\frac{(m-m_k)^2}{2\sigma^2}\right) \quad (1)$$

$$f(m;\ \gamma) = \frac{\gamma^2 p_k}{\gamma^2 + (m-m_k)^2} \quad (2)$$

where σ and γ are the parameters of the Gaussian and Cauchy-Lorentz functions, respectively, which are used to control the width of the peak. Their full width at half maximum are $\sigma(\mathrm{In}\ 256)^{1/2}$ and 2γ, respectively. Therefore, their simulated resolving power defined by $R = m/\Delta m_{50\%}$ are $m_k/(\sigma(\mathrm{In}\ 256)^{1/2})$ and $m_k/2\gamma$, respectively. The final form of the formula to calculate the theoretical isotopic envelope is:

$$f_{total}(m) = N\sum_k p_k \exp\left(-\frac{(m-m_k)^2 R^2 \mathrm{In}\ 256}{2m_k^2}\right) \quad (3)$$

$$f_{total}(m) = N\sum_k \frac{p_k m_k^2}{m_k^2 + 4R^2(m-m_k)^2} \quad (4)$$

where N is the normalization to experimental spectra.

EXPERIMENTAL

Human Cu/Zn superoxide dismutase (SOD1) was expressed and purified from *Saccharomyces cerevisiae* strain EGy118Δ‴ SOD1 (lacking yeast SOD1) transformed with human SOD1. A single colony was added to 3 mL YPD media (1% yeast extract, 2% bactopeptone, 2% glucose, pH 7) at 30°C and shaken overnight. Successive dilutions of 1:20 was made into MonoExpress cell growth media (12C 99.95%, 14N 99.97%, Cambridge Isotope Laboratories, Andover, MA, USA) and shaken at 30°C. Yeast cell cultures were harvested by centrifugation. The cell pellet was resuspended in lysis buffer (200 mM tris(hydroxymethyl)aminomethane (Tris) buffer, pH 8.0, 0.1 mM disodium ethylenediaminetetraacetate (EDTA), 50 mM sodium chloride) and was lysed.

SOD1 was purified using an in-house packed anti-hSOD1 affinity column with the antibodies generated as previously reported[18] and immobilized onto POROS beads (POROS-AL 20 μm i.d., Applied Biosystems, Foster City, CA, USA) following the manufacturer's instructions. Elution of SOD1 from the antibody column was performed with 5% acetic acid and the eluent was used directly for mass spectrometry. SOD1 was directly infused into the electrospray ionization (ESI) source of a hybrid quadrupole Fourier transform ion cyclotron resonance (FT-ICR) mass spectrometer (apex-qe 94, 9.4 tesla, Bruker Daltonics, Billerica, MA)

RESULTS AND DISCUSSION

Using this new algorithm, we calculated the theoretical spectra of the 5736.6 peak of protonated bovine insulin ($C_{254}H_{378}N_{65}O_{75}S_6$) at four different resolutions, see Fig. 4. The masses and the abundances of the isotopes used in the calculation are listed in the Table 1. The result is consistent with previous calculations from Werlen[19] and Rockwood *et al.*[9]

We also demonstrate the utility of this novel algorithm by calculating the theoretical envelope of human SOD1 protein in two different situations: ^{13}C and ^{15}N in natural abundance

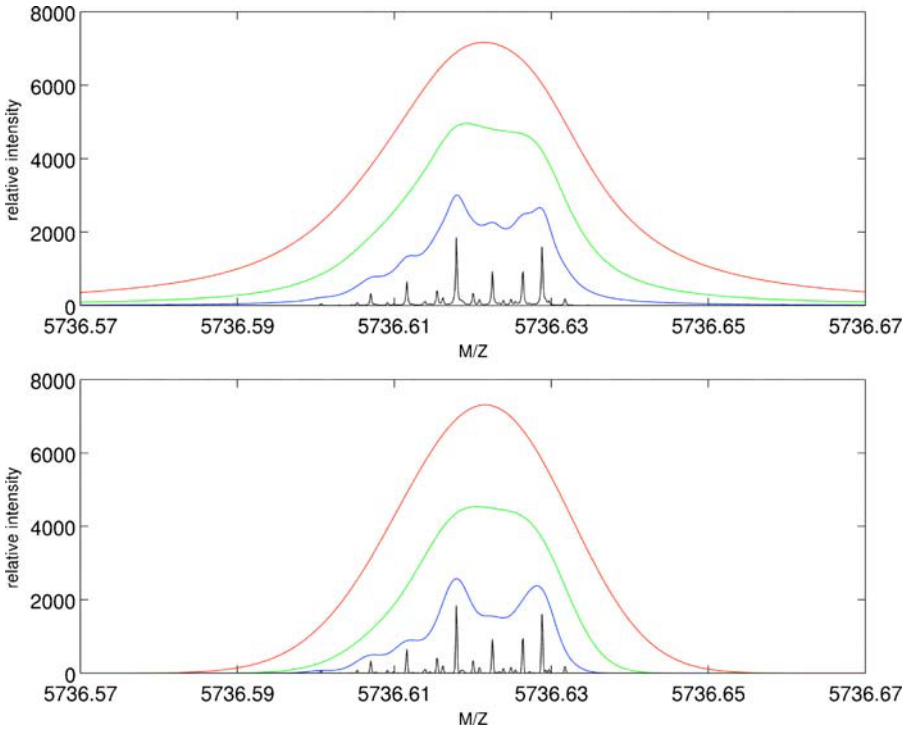


Figure 4. The theoretical envelopes of the M+6 peak at four different resolutions. The top panel is calculated using the Cauchy-Lorentz function (Eqn. (4)) and the bottom panel is calculated using the Gaussian function (Eqn. (3)). For all of the calculations, the normalization N in Eqns. (3) and (4) is 100 000; the resolving power R from top to bottom in each panel are 300 000, 600 000, 1 500 000, and 27 000 000, respectively.

DOI: 10.1002/rcm

Table 1. Masses and abundances of H, C, N, O, and S used in our calculations. All data were downloaded[25]

Isotope	Mass	Abundance
^{1}H	1.00782503	0.99985
^{2}H	2.01410178	0.00015
^{12}C	12	0.9893
^{13}C	13.00335484	0.0107
^{14}N	14.00307401	0.99632
^{15}N	15.0001089	0.00368
^{16}O	15.99491462	0.99757
^{17}O	16.991315	0.00038
^{18}O	17.9991604	0.00205
^{32}S	31.97207069	0.9493
^{33}S	32.9714585	0.0076
^{34}S	33.96786683	0.0429
^{36}S	35.96708088	0.0002

and ^{13}C and ^{15}N depleted. The agreement between theoretical calculated isotopic envelope and the experimental data is good but not excellent, see Fig. 5. The difference likely resulted from either inaccurate isotope statistics used in the calculation (e.g., $^{12}C/^{13}C$ abundance ratios can be altered by metabolic processing, and moreover the $^{12}C/^{13}C$ abundance ratios in our starting materials were estimates that were not confirmed by isotope ratio mass spectrometry); or the experiment itself (e.g., $\sim 10^6$ ions are sampled for a given experiment and $\sim$20 ions are required for peak detection, which results in some statistical variation in the sampled mass states). Considering the high level of signal to noise of our experimental data, inaccurate relative isotope abundance statistics were likely the major contributor.

Decreased memory usage

The new divide-and-recursively-combine algorithm saves the memory in a significant magnitude. The number of all mass states of E_n is:

$$(n + I - 1)!/n!(I - 1)!,$$

where n is the number of atoms, and I is the number of isotopes of that element.[20] For example, the formula of bovine insulin is $C_{254}H_{377}N_{65}O_{75}S_6$, and only about 8k bytes memory are needed to save the mass and abundance information of the hierarchical structures of carbon (255 mass states), hydrogen (378), nitrogen (66), oxygen (2926) and sulfur (84). In contrast, our previous hierarchical algorithm needs more than 10^9 bytes of memory to keep the maximum level.[15] Theoretically, the divide-and-recursively-combine algorithm requires $\sim 10^7$ bytes memory when calculating a molecule near 500k Da in the IPI Human database (downloaded on July 9, 2008),[21] and about 5 Gigabytes memory when calculating the maximum protein of $\sim$3.8M Da. *Therefore, this algorithm is capable of calculating the complete isotopomer distribution of any known protein using a desktop computer*. In studies of biological molecules, metals are often found in biomolecules, e.g., metalloproteins. Although the numbers of their isotopes are more than those of C, H, N, O and S, the numbers of their atoms are small. For example, Zn has five stable isotopes. Even 10 atoms of Zn only require 10^3 bytes memory.

Improvement of truncating negligible levels

Since only a small number of levels contribute to the observed experimental spectrum, other levels can be neglected. For example, although there are 871 total levels, the first 12 main levels of bovine insulin represent 99% of the abundance of the entire isotopic distribution. Therefore, our previous hierarchical algorithm adopts a truncation method to cut off those negligible levels.[15] Starting from level 0, all mass states are calculated until level L, at which the cumulative abundance reaches a threshold, e.g., 0.99. It has been shown that the peak intensity of the ground state (monoisotopic species) becomes less and less with increased molecular weight.[22] The natural isotopic distribution of SOD1 in Fig. 4 is an example. Therefore, some levels on the left side of the whole isotopic distribution can also be truncated. This means we can set a two-level threshold, L_{left} and L_{right}, to cut off those negligible levels whose level number is less than L_{left} or greater than L_{right}. For example, the mass states from level 17 to level 47 represent 99% abundance of all mass states of the protein nuclear factor 1 (IPI ID: IPI00749495.1). The total probability of any level from level 0 to level 16 and the levels greater than 47 is too low to be detected. Because of the divide-and-recursively-combine algorithm, it is possible to only calculate these main levels of level 17 to level 47, instead of from level 0 to level 47. In principle, the calculation can be started from any level between level 17 and level 47. The easiest implementation is to begin from the level which has the maximal abundance mass state. It is easy to determine the maximum-mass-state level of each element when calculating its hierarchical structure. The sum of these level numbers is the maximum-mass-state level of the molecule. Then the levels on the left and right to the maximum-mass-state level are calculated. If the sum of abundances of mass states in the current calculated left-most level is greater than in the current calculated right-most level, then move to calculate the left level to the current calculated left-most level. Otherwise, move to calculate the right level to the current calculated right-most level. This process is repeated until the cumulative abundance reaches the threshold.

Yergey's algorithm also used the pruning technique to deal with the memory problem. Each time an atom is added, only those isotopic species with abundance above a user-defined threshold are kept. Unfortunately, the deletion of a low-abundance isotopic species at an early step can result in the deletion of one or more high-abundance (greater than the threshold) isotopic species in subsequent steps. Therefore, although the masses and the abundances of the remaining isotopic species are exact, those missing isotopic species can result in significant distortion of the isotopic envelope, especially for large molecules.[10] From the view of our hierarchical structure, those missing isotopic species which result in the distortion are within the major levels. However, our truncation technique is different from the pruning technique. Our algorithm only truncates the minor levels, which do not affect the whole spectrum. For the major levels, any species are retained even its abundance is small. Therefore, there is no distortion problem in our algorithm.

DOI: 10.1002/rcm

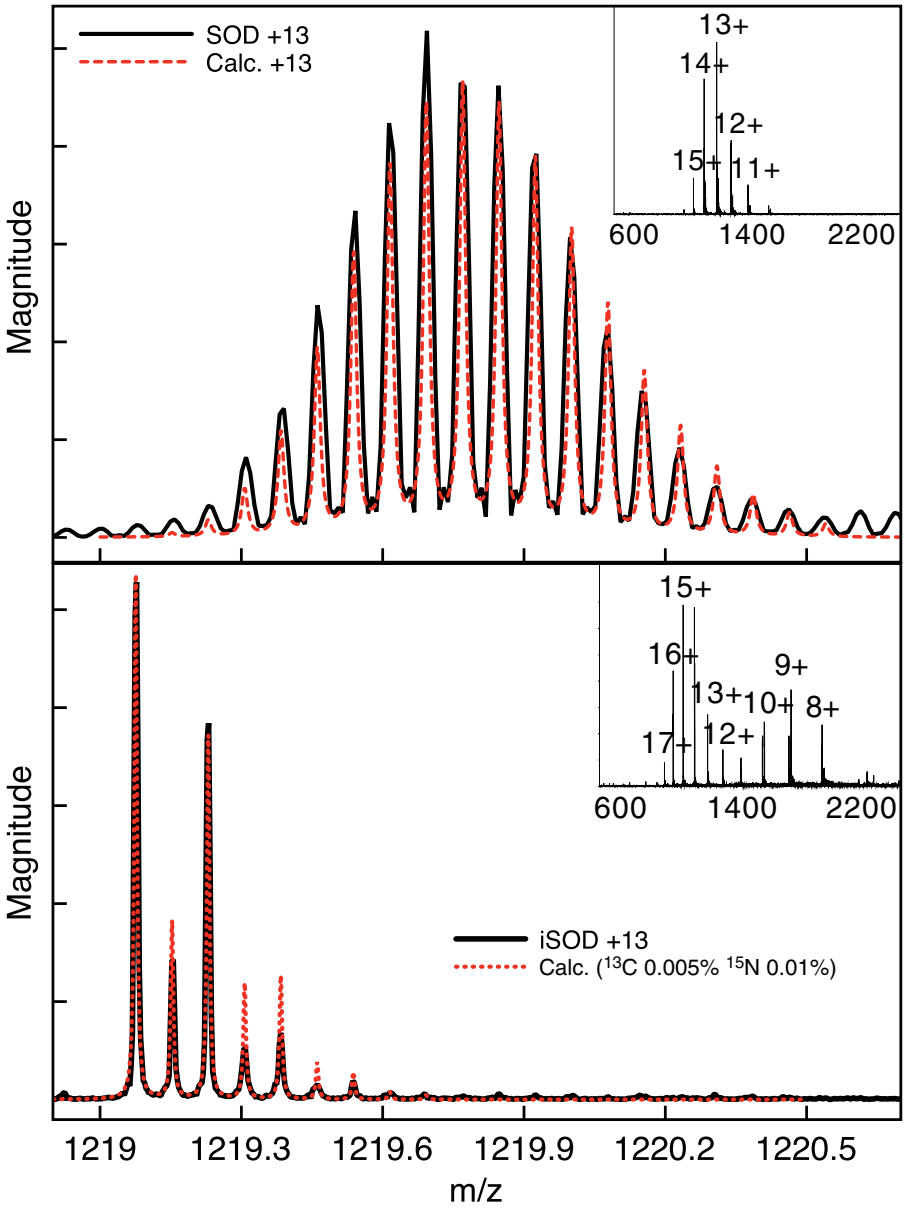


Figure 5. The calculated isotopic envelope of human SOD1 protein versus the experimentally determined envelope. Mass spectrum of the +13 charge state of human SOD1 protein (with a native single disulfide bond) was calculated from the chemical formula $C_{681}H_{1100}N_{204}O_{224}S_4$ having natural isotopic abundances (top, dashed red line filled) and having its ^{13}C and ^{15}N isotopes diluted (bottom). Plotted in black is the FT-ICR mass spectrum. Inset shows the entire mass spectrum. Both the *m/z* values and isotopic envelope fit well to the spectrum of SOD1 with natural isotopic abundance. The extent of isotopic dilution was not known accurately (the manufacturer provides only an upper limit of ^{13}C and ^{15}N) and thus fitted to 0.005% ^{13}C and 0.01% ^{15}N using the calculations from the algorithm. The resolving powers used were based on values reported by DataAnalysis software (Bruker Daltonics, Billerica, MA, USA) which were 80 000 for the top and 180 000 for the bottom spectrum. *m/z* values of the spectrum were internally calibrated, magnitude values of the experimental spectra were kept constant and the simulated magnitudes were multiplied by a coefficient to match the experimental value with the simulated magnitude of the theoretically most abundant isotopic peak.

Higher resolution, less calculation time

When calculating the whole theoretical spectrum, an array of mass points is first created over the whole mass range of the spectrum, the intensities for each mass point are set to zero. The size of the array depends on the user's choice. For example, suppose the mass range is 100 *m/z*, and the interval is 0.0001, then the size is 1 000 000. Then each time when a new molecular mass state is obtained, its contribution is calculated using Eqn. (1) or (2) and added to the whole spectrum. The calculation of a single mass state always starts

DOI: 10.1002/rcm

from the mass points near its mass (m_k), then continues to its left or right mass points. Once the contribution of one mass point is less than a value, see 0.0000001, the calculation to the left or right stops. This process has an important characteristic: higher resolution, less calculation time. The reason is the following: for a fixed point array, the higher resolution, the less mass points are included to calculate in the Gaussian function or Cauchy-Lorentz function. In addition, unlike Rockwood's Fourier transform-based methods, our mass array has nothing to do with the resolution. However, as a minimum requirement, the interval between the points should be less than the interval of the experimental data. Otherwise, the comparison between the theoretical calculation and experimental data will be difficult.

Implementation of the algorithm

The algorithm described above has been implemented as a computer program called Isotope Calculator, which was written in C++. It is freely available under the GNU Lesser General Public License.[23] This new divide-and-recursively-combine algorithm is based on our previous hierarchical approach,[15] and improves the efficiency of memory usage. The comparison of computation time between the hierarchical approach and other methods was discussed in our previous work.[15] However, in addition to enabling the isotopomer distributions of larger molecules, the new algorithm is generally faster than the previous hierarchical approach. For example, on a desktop with Intel Core2 Duo CPU@2.0GHz, 2 gigabytes of RAM, the new algorithm needs 0.9 s at the resolving power $R = 10^5$ and 0.08 s at the resolving power $R = 10^7$ to calculate the theoretical envelope of bovine insulin ($C_{254}H_{377}N_{65}O_{75}S_6$); the previous hierarchical algorithm needs 3.26 s and 0.78 s, respectively.[15] Our program can output the isotopic composition information of each mass state, including the level in which it belongs, the number of each elementary isotope, the mass and the abundance of each mass state. This information can be used to accurately calculate the nominal isotopic peaks which is the probablility-weighted sum of all the mass states in the same level. It should be pointed out that another algorithm[24] (a stand-alone program called 'qmass'[11]) can also calculate the accurate nominal mass by calculating the isotopic composition of nominal isotopic peaks, but does not deal with the isotopic composition of the individual mass state.

Several issues have not been addressed here, including the scaling properties of the computational time as a function of resolution; the scaling properties when this method is extended to higher molecular weight compounds or high isotopic complexity; the stack overflow when using recursive algorithm.

Acknowledgements

This work was supported by Brandeis Faculty Fund to PH, and the National Institutes of Health Grant R01NS065263 to JA.

REFERENCES

1. Waanders LF, *et al. J. Am. Soc. Mass Spectrom.* 2007; **18**: 2058.
2. Ong SE, Mann M. *Nat. Chem. Biol.* 2005; **1**: 252.
3. Du Y, *et al. Anal. Chem.* 2006; **78**: 686.
4. Shi SD, *et al. Proc. Natl. Acad. Sci. USA* 1998; **95**: 11532.
5. Brustkern AM, *et al. J. Am. Soc. Mass Spectrom.* 2008; **19**: 1281.
6. Wilkins CL. *TrAC Trends Anal. Chem.* 2007; **26**: 65.
7. Robinson NE, *et al. Rapid Commun. Mass Spectrom.* 2006; **20**: 3535.
8. Yergey J. *Int. J. Mass Spectrom. Ion Phys.* 1983; **52**: 337.
9. Rockwood AL, *et al. Rapid Commun. Mass Spectrom.* 1996; **10**: 54.
10. Rockwood AL, *et al. Anal. Chem.* 1995; **67**: 2699.
11. Rockwood AL, Haimi P. *J. Am. Soc. Mass Spectrom.* 2006; **17**: 415.
12. Olson MT, Yergey AL. *J. Am. Soc. Mass Spectrom.* 2009; **20**: 295.
13. IsoPro, 3.1. Available: http://sites.google.com/site/isoproms/.
14. Fernandez-de-Cossio J, *et al. Nucleic Acids Res.* 2004; **32**: W674.
15. Li L, *et al. J. Am. Soc. Mass Spectrom.* 2008; **19**: 1867.
16. Rockwood AL. *Rapid Commun. Mass Spectrom.* 1995; **9**: 103.
17. Rockwood AL, VanOrden SL. *Anal. Chem.* 1996; **68**: 2027.
18. Taylor DM, *et al. J. Biol. Chem.* 2007; **282**: 16329.
19. Werlen CR. *Rapid Commun. Mass Spectrom.* 1994; **8**: 976.
20. Snider RK. *J. Am. Soc. Mass Spectrom.* 2007; **18**: 1511.
21. Kersey PJ, *et al. Proteomics* 2004; **4**: 1985.
22. Yergey J, *et al. Anal. Chem.* 1983; **55**: 353.
23. Available: http://www.cs.brandeis.edu/~hong/software.html or http://people.brandeis.edu/~agar.
24. Rockwood AL, *et al. J. Am. Soc. Mass Spectrom.* 2004; **15**: 12.
25. Available: http://physics.nist.gov/cgi-bin/Compositions/stand_alone.pl?ele=&all=all&ascii=html&isotype=some.

DOI: 10.1002/rcm