research articles Journal of proteome research

Code Developments to Improve the Efficiency of Automated MS/MS Spectra Interpretation

Rovshan G. Sadygov,* Jimmy Eng, Eberhard Durr, Anita Saraf, Hayes McDonald, Michael J. MacCoss, and John R. Yates, III

Department of Cell Biology, SR-25, The Scripps Research Institute, North Torrey Pines Road, La Jolla, California 92037

Received December 10, 2001

We report the results of our work to facilitate protein identification using tandem mass spectra and protein sequence databases. We describe a parallel version of SEQUEST (SEQUEST-PVM) that is tolerant toward arithmetic exceptions. The changes we report effectively separate search processes on slave nodes from each other. Therefore, if one of the slave nodes drops out of the cluster due to an error, the rest of the cluster will carry the search process to the end. SEQUEST has been widely used for protein identifications. The modifications made to the code improve its stability and effectiveness in a high-throughput production environment. We evaluate the overhead associated with the parallelization of SEQUEST. A prior version of software to preprocess LC/MS/MS data attempted to differentiate the charge states of ions. Singly charged ions can be accurately identified, but the software was unable to reliably differentiate tandem mass spectra of +2 and +3 charge states. We have designed and implemented a computational approach to narrow charge states of precursor ions from nominal resolution ion-trap tandem mass spectra. The preprocessing code, 2to3, determines the charge state of the precursor ion using its mass-to-charge ratio (*m*/*z*) and fragment ions contained in the tandem mass spectrum. For each possible charge state the program calculates the expected fragment ions that account for precursor ion *m*/*z* vlues. If any one of the numbers is less than an empirically determined threshold value then the spectrum corresponding to that charge state is removed. If both numbers are higher than the threshold value then +2 and +3 copies of the spectrum are kept. We present the comparison of results from protein identification experiments with and without using 2to3. It is shown that by determining the charge state and eliminating poor quality spectra 2to3 decreases the number of spectral files to be searched without affecting the search results. The decrease reduces computer requirements and researcher efforts for analysis of the results.

Keywords: mass spectrometry • protein identification • database search • charge determination

1. Introduction

Mass spectrometry (MS) in conjunction with database searching has become a powerful tool for analyzing complex protein mixtures.[1,2] Protein mixtures can be digested using site-specific endoproteases such as trypsin. The complex mixture of peptides is then separated and introduced into the mass spectrometer. The peptides are then ionized using electrospray ionization (ESI). Mass spectra of the ionized peptides are recorded using triple-quadrupole, ion-trap, or quadrupole time-of-flight mass spectrometers. Experiments can be set up to yield two types of data. In single MS analysis, masses of ionized peptides from digested protein mixture are recorded (protein fingerprint). In tandem mass spectrometry (MS/MS), the first mass analyzer separates ions with a specific mass-to-charge ratio, m/z. The peptides (called parent peptides) with the selected m/z dissociate in collisions with atoms of an inert gas. The fragment ion m/z values are then separated and detected. Thus, two sets of information are obtained in the tandem mass spectrometry experiment: mass-to-charge ratio of the parent peptide and mass spectrum of its fragment ions.

Database searching software[3−5] finds the identity of proteins in a mixture assuming that they are present in the database. The software compares the experimental tandem mass spectrum with theoretical spectra created from amino acid sequences of database proteins and identifies the sequence that best fits the tandem mass spectrum. The rapid expansion of protein databases and improvements in the sensitivity of mass spectrometers has increased the scope and complexity of the protein mixtures that can be analyzed by the above approaches. These developments pose challenges for database searching software. Experiments can routinely generate tens of thousands of tandem mass spectra. Each one of these spectra needs to be correlated with the amino acid sequences of a protein database that can contain tens of millions of amino acids. The precursor ion mass of an experimental spectrum can match up to 1 million amino acid sequences in these databases depending on the mass tolerance used in the search. Theoreti-

10.1021/pr015514r CCC: $22.00

cal spectra of all these sequences need to be compared with the experimental spectrum to determine the identity of the peptide expressed in the mass spectrum. The problem becomes even more involved for experiments using ESI process. When mass spectrometers are used that produce nominal resolution mass spectra it is difficult to uniquely determine charge states of multiply charged parent peptides. Therefore, we assume a spectrum represents a peptide ion corresponding to charge states of +2 or +3 and then calculate the corresponding molecular weight for each charge state. An experimental spectrum is then created for each calculated molecular weight. Obtaining quality tandem spectra for peptide ions with charge states greater than +3 charge is rare and thus ignored. Clearly, determination of the charge state would halve the number of spectral files and reduce the computational efforts for these experiments.

The growing magnitude of computational analysis places stringent requirements on the efficiency of database searching software and the quality of the experimental data. The software needs to be efficient in use of computer resources and robust for searches of large number of spectral files. For spectral files from experiments using ESI, preprocessing is necessary to determine charge states of the multiply charged parent peptides and to eliminate bad quality data. Two requirements of a preprocessing program are to reduce the number of spectral files and not to affect the protein identification results.

The purpose of this work is 2-fold. We made the peptide identification program, SEQUEST-PVM, more robust. The program has been modified to remove the indefinite interdependence of slave nodes on each other. Our new code for charge determination, 2to3, narrows the charge states of the multiply charged parent peptides and eliminates spectra of low quality. Thus, 2to3 reduces the number of spectral files in the database search and alleviates the computational load. The reduction is done in a way that does not affect protein identification results and is demonstrated on protein identification experiments performed with the *S. cerevisiae* 26S proteosome[6] and human lens.[7] In the next section, we describe the sample material. Section 3 explains the parallelization aspects of SEQUEST-PVM and changes made to the program to make it fault tolerant. Section 4 describes the underlying principles of 2to3 and gives examples of its performance for different types of experimental data. Section 5 summarizes and concludes the paper.

2. Experimental Data

We used two sets of data that represent common experiments performed in the laboratory, 26S proteosome from S. *cerevisiae* and the soluble protein fraction from a human lens sample, to validate 2to3. Results from the 26S proteosome analysis have been reported previously[6] and are common for the analysis of protein complexes. The mass spectra were searched against a database containing predicted yeast open reading frames (ftp://genome-ftp.stanford.edu/pub/yeast/yeastORFs/orf_trans.fasta.Z) and common contaminants, such as trypsin and keratin. Lens sample data was chosen to represent analysis of a more complex mixture of proteins. The spectra were searched against the human database (ncbi:520 000 entries). These data have also been previously reported.[7] Sample preparations and experimental data collection details can be found in the above references.

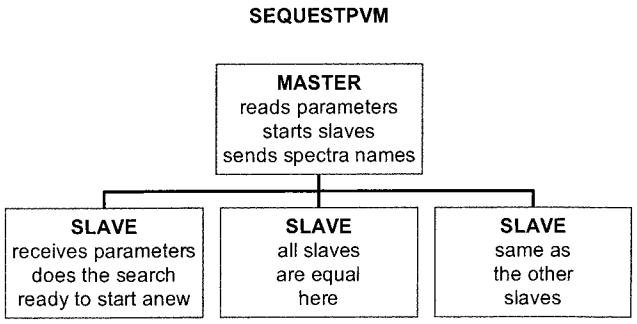

Figure 1. Chart of the master−slave relationship in SEQUEST-PVM.

3. SEQUEST-PVM

SEQUEST correlates tandem mass spectral data of peptides with amino acid sequences from a protein database.[3] Candidate peptides, whose masses are equal (with some accuracy) to the precursor mass, are identified from the database. Sequence analysis is done using two scoring criteria. For preliminary scoring, a score is calculated for each amino acid sequence by matching fragments of the sequence and the ions observed in the experimental spectrum. The score accounts for the intensity of the matched ion peaks, their continuity, and the length of the amino acid sequence. For the second criterion, SEQUEST generates a theoretical mass spectrum for each of the 500 highest preliminary scored peptides. In the theoretical spectrum, y- and b-ions are assigned intensity of 50, their immediate vicinity (within 1 amu) has an intensity of 25. Neutral loss fragments due to water and ammonia are assigned an intensity of 10. The experimental spectrum is divided into 10 mass windows of equal size, and the intensity in the each of these windows is normalized to 50. Then the experimental and theoretical spectra are cross-correlated using a fast Fourier transform technique.[3] A peptide is identified on the basis of the scoring results from the two methods. SEQUEST and its functions have been extensively discussed in the literature.[3,8] We describe here a parallel version of the code, SEQUEST-PVM, which runs on a cluster of computers.

The use of a stand-alone program becomes cumbersome for searches of tens of thousands of spectral files. Running these searches on a single computer against large databases may take months of CPU time. Assuming that a large number of computers are available, the problem becomes how to distribute the computational load efficiently among the computers and do it in a way requiring minimal effort from a user. SEQUEST-PVM solves these problems by utilizing tools of parallel virtual machine (PVM).[9] It will become clear from the discussion below that a single search routine is not parallel but the collective search is.

SEQUEST-PVM consists of two programs, master and slave. Design of these programs is simple and efficient. The master initializes slaves and sends spectral file names to the slaves. The slave program is the actual search code and it runs on slave nodes. When a slave finishes a search it signals the master and the master sends to it a new spectral file. The master plays a role of computational load distributor. The process continues until all spectral files are searched. At that point, the master terminates the search process by terminating slave programs. The master runs on one computer, the slaves run on all computers, Figure 1. To communicate, the master and slaves

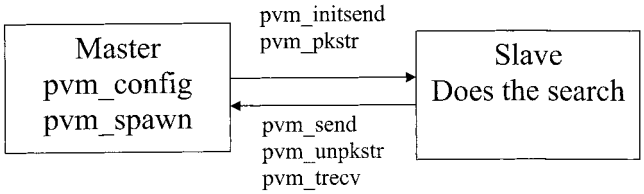

Figure 2. PVM functions and message-passing between the master and slave in SEQUESTPVM.

use tools of PVM. With the exception of the PVM calls, the slave program does not differ much from the stand alone SEQUEST.[3]

PVM is a parallel programming environment for Unix workstations. In SEQUEST-PVM it is mainly used for its message-passing capabilities. PVM has a set of modules for interprocess communication. Interfaces to these modules differ slightly between C and FORTRAN. Since SEQUEST-PVM is written in C, we will refer to PVM interfaces for C. The functions are illustrated in Figure 2 and explained below.

PVM daemon creates a virtual machine out of the cluster of computers. The master in SEQUEST-PVM obtains characteristics of the machine (number of computers and their architectures, hostnames, speeds, etc.) by calling pvm_config(). Then the master spawns child processes (slaves) with the pvm spawn() function. This function tells PVM which program should be started on each computer. At this point, the master broadcasts the parameter set to the slaves. The parameter set includes the address of a protein database, mass tolerances, and amino acid modifications. The broadcasting is done only once.

Once the slaves are initiated and parameters are read, the master program sends spectral file names to the slaves. The slaves do the search and upon completion send a message to the master informing it of readiness to start a new search.

All of the above information—parameters, filenames, and receipt notes—are communicated using PVM's message-passing functions. The process takes several steps and functions. At first, a message buffer is initiated with the call to pvm_initsend(). This function clears the sending buffer and prepares to accept a new message. It takes different arguments in heterogeneous and homogeneous clusters. In homogeneous clusters no message encoding and decoding are necessary and this saves time. In heterogeneous clusters a message is first encoded by the sending processor and then decoded by the receiving processor. The procedure is time-consuming but allows for computers of different architectures to form a single virtual machine.

After pvm_initsend() initiates the message buffer, SEQUEST-PVM uses pvm_pkstr() function to pack a message into the active buffer. This function takes only one argument—a string to be packed.

Once packed and stored in active buffer, the message is sent to the PVM process by the pvm_send() routine. This function takes two arguments, task identification number (tid) of the process to which the message is sent, and a message tag, an integer value identifying the message. pvm_send() is asynchronous. It does not wait for the receiving process to signal a matching receipt.

One problem with SEQUEST-PVM has been its dependence on each node in the cluster for smooth functioning. Thus, if one of the nodes were to halt then the whole search process would be stopped. The problem was due to the pvm_recv() routine used to obtain messages. This function takes two

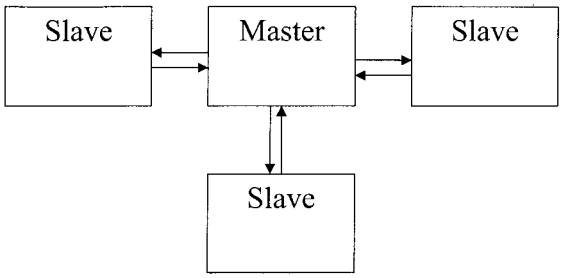

Figure 3. Master−slave relationship in SEQUEST-PVM after interprocess decoupling.

Table 1. Overheads Associated with the Paralellization in SEQUESTPVM[a]

no. of nodes	search time (h)	search time per processor (h)	no. of "outside" calls
1	4.11	4.11	0
3	4.61	1.54	2519
5	4.44	0.89	3110

[a] The clusters consisted of AMD Athlon Computers running LINUX.

arguments, tid of the sending processor and the message. pvm recv() blocks the receiving processor until the message with the tag from the sending processor with the tid has arrived. In other words, until the master is finished communicating with one of the slaves it cannot switch to any other task. This implementation was prone to stoppages and "hang-ups", since a fault in one of the slaves would halt the entire cluster at that point. To avoid this problem and make the code more fault tolerant, we implemented a waiting time determined function—pvm_trecv(). This function behaves just like pvm_recv(), but it has an additional argument—a pointer to a structure timeval. Elements of this structure specify how long the receiving processor will wait for the message before returning empty. The use of this function with some modifications to the message handling structure allowed us to make SEQUEST-PVM tolerant toward the numerical faults in slave nodes. Now the master program waits for specific amount of time for a slave to inform it that the search has finished. If the master does not receive the message within that time interval, that search is terminated and the spectrum is removed from the search. It can be said that pvm_trecv() decouples master−slave communications in SEQUEST-PVM, Figure 3.

Once the message is in the active receive buffer it is unpacked using the pvm_unpkstr() routine. It takes a single argument, which is a string, and to which the message from the active receive buffer is copied.

As noted above, SEQUEST-PVM is very efficient and has very low overhead associated with the parallelization. To demonstrate this, in Table 1 we present search times of a set of data from samples of *S. cerevisiae*. The searches were carried on a virtual machine consisting of one, three, and five computers. As is seen from Table 1, total processor search time is higher when the number of computers in the cluster is increased. One reason for this is the increase in the number of message-passing. However, this is not a monotonic dependence as evidenced from Table 1. Thus, the increase of the cluster size from three to five decreases the overhead time. The percentage of the overhead with respect to the total search time is below 1%.

4. Program To Eliminate Poor-Quality Spectra and To Narrow Charge "Uncertainty"-2to3

To correlate a tandem mass spectrum with an amino acid sequence from a protein database, SEQUEST needs to know the molecular weight of the parent peptide. To accurately calculate the molecular weight of a peptide from an electrospray ionization mass spectrum, the charge state of the ion must be determined. ExtractMS, an LC/MS/MS data preprocessing software program, can uniquely determine charge states of only singly charged parent peptides. For tandem mass spectra with higher charge states, molecular weights presuming +2 and +3 charge states are created for the same spectrum. Charged states higher than +3 are ignored because if formed their tandem mass spectra are rarely of sufficient quality to obtain an unambiguous protein identification. Consequently, each spectrum representing a charge state greater than +1 is searched twice. The actual charge state is inferred from SEQUEST results. In general, the charge state with the better overall SEQUEST score is accepted to be the true charge state. This approach leads to increased search times and additional efforts of analyzing "overhead" spectra. Determination of the peptide charge is important for efficient analysis of mass spectral data. One approach to address this problem has been the use of spectral deconvolution to determine charge states.[10,11] We tested entropy-based deconvolution for experimental data from our laboratory. We did not find this approach useful in our case, because more than one charge state is required for accurate calculation of the molecular weight. Our m/z scan range is [200, 2000], and often only one ion from a charge state appears in the spectrum. We found that determination of the charge state of precursor ion from its mass spectrum and mass-to-charge ratio is more effective. To determine the charge, we develop a dissociation pattern for each charge state of a precursor ion. Then we choose the charge state, which accounts for most of the fragment ions. With this criterion we are also able to eliminate spectra, which are unlikely to fare well in a database search. A computer program called 2to3 implements this approach.

2to3 counts all fragment ions resulting from all possible dissociation schemes of a multiply charged precursor ion. Assuming that there has been no neutral loss, the equation governing all possible fragmentations of a precursor ion can be presented as

$$Pz = \sum_{i=1}^{N} P_i z_i \qquad (1)$$

Where P and z are mass-to-charge ratio and charge of the parent peptide, respectively. P_i and z_i refer to the ith fragment ion, and N is the number of fragments from dissociation of a single parent peptide. In theory, N is equal to 2 for doubly charged precursor ions and to 3 for triply charged ones. But under the ambient experimental conditions, dissociation of a triply charged precursor into three fragment ions is not observed. Rather, it is thought that the triply charged precursors dissociate into two fragments—singly and doubly charged ions. Therefore, N is set equal to 2 for triply charged spectra, too. Total charge is conserved

$$z = \sum_{i=1}^{N} z_i$$

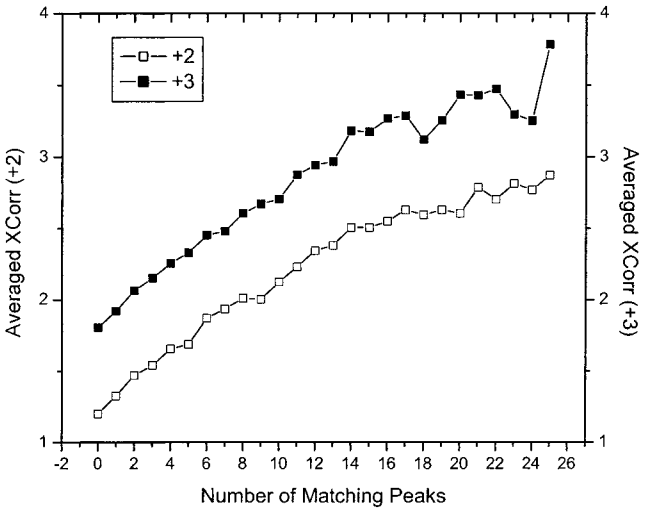

Figure 4. Averaged Xcorr for doubly and triply charged spectra as function of matching peaks for 26S proteosome from *S. cerevisiae*.

2to3 reads the experimental spectrum and counts the number of fragment ion pairs, matching-peaks, satisfying eq 1. The mass tolerance in eq 1 is chosen to be 0.6 amu. An intensity cutoff is used to filter the experimental data. To be considered as valid, a mass-to-charge ratio peak must have intensity not less than 20% of the averaged (without the maximum intensity peak) spectrum intensity. The choice of values for the parameters is based on our empirical observations and extensive testing.

For 2to3 to work optimally, predicted fragment ions need to be in the scanned mass range. Therefore, the best results will be achieved for precursor peptides satisfying following condition

$$P_{\min} \le Pz \le P_{\max}$$

where $P_{\max}$ and $P_{\min}$ are the maximum and minimum scanned m/z values, respectively.

There is a correlation between the number of matching peaks and the spectrum quality as it relates to SEQUEST scores. As noted above, SEQUEST has a two-tiered scoring scheme. The preliminary score accounts for the number of matched ions, their continuity, and peptide length. The cross-correlation score, Xcorr, is a measure of correlation between experimental and theoretical spectra. The theoretical spectrum is constructed from predicted fragmentation of an amino acid sequence. If there are no matching peaks in the experimental spectrum, then it probably means that it will be impossible for any trial peptide to have more than half of its b- and/or y-ions coincide with their experimental counterparts. This experimental spectrum will inevitably have a low preliminary score and is most likely to have a low Xcorr as well, no matter what database is searched. This point is illustrated in Figure 4. The figure depicts average Xcorr as a function of number of matching peak pairs for doubly and triply charged tandem mass spectra obtained from samples of 26S proteosome of *S. cerevisiae*. As it is seen from the figure the average Xcorr value of spectra that have no matching-peaks is about 1.3 for doubly and 1.8 for triply charged spectra. From our experience, to be considered for further analysis a spectrum should have an Xcorr of at least 2.5. The Xcorr values increase with the number of matches. This indicates a correlation between the matching peaks count

Table 2. Comparison of the Search Results and Times before and after Running 2to3

		no. of spectra (+2/+3)	search time[a] (h)	avg Xcorr (+2/+3)	no. of identified proteins[c]	no. of matching spectra
26S proteosome from *S. cerevisiae*	before 2to3	10905/10905	138	2.2/2.3	93	3404
	after 2to3	9516/5461	66	2.4/2.6	93	3380
lens sample	before 2to3	34711/34711	764	2.3/2.5	130	11 740
	after 2to3	29569/18122	567	2.5/2.8	130	11 223

[a] The times reported are the cumulative search times of a cluster consisting of 27 computers (16 1.2 GHz Athlons, 10 533 MHz 21164 ALPHAs, and 1 533 MHz 21264 ALPHA, all running LINUX). [b] The average Xcorr's are those of the highest ranking peptides. [c] The number of non-redundant proteins is reported.

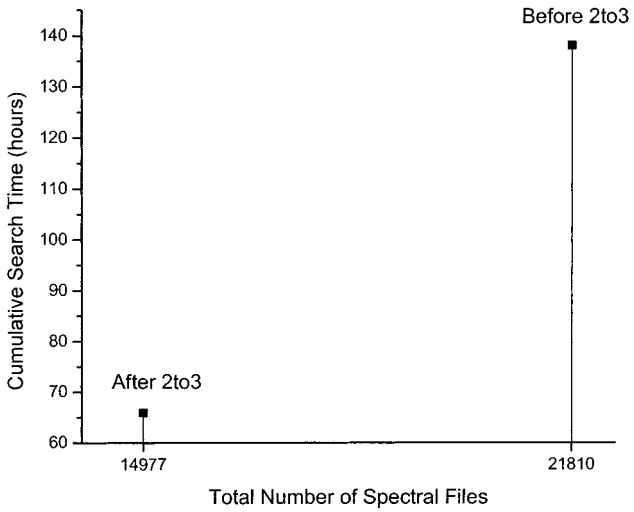

Figure 5. Spectral files and cumulative search times before and after 2to3 for data from 26S proteosome of *S. cerevisiae*.

and the quality of spectrum. Most of the spectral files have the peak count under 25. But some spectra may have as many as 60 pairs of m/z values satisfying eq 1.

As seen in Figure 4, the spectra that have number of matching pairs less than three also have very low Xcorr values. Searching poor quality tandem mass spectra does not help with protein identification since the quality of matches tends to be poor as well. Removing poor quality spectra from search process will reduce the computational load and should not affect the identification results. 2to3 deletes any spectrum (its charge state) that has less than three pairs of m/z peaks satisfying eq 1.

As we show below, removing poor-quality spectra either obtained from the experiment or by wrong charge assignment does not affect protein identification results. If any of the charge states has more than three pairs of matching peaks, 2to3 keeps that charge state. As a result, some spectra still have +2 and +3 "versions", but the overall search time is drastically reduced, and most importantly, the protein identification results are not affected.

In Table 2, we present the results of running 2to3 for tandem mass spectra from samples of *S. cerevisiae* 26S proteosome and human lens. 2to3 takes only 1 min to run, but the gain in database search time is high, Figure 5. On average, the program eliminates about 20−30% of the spectra, for about half of the remaining spectra the charge state (+2 or +3) is determined uniquely. For the rest of the spectra the charge cannot be assigned uniquely, so both of the charge states are still searched and the correct molecular weight and match is ascertained from the search results. The average Xcorr is increased for both +2 and +3 spectra, which is another indication that the quality of the searched spectra has increased.

5. Conclusion

We have made changes to a peptide identification program, SEQUEST-PVM, to improve its fault tolerance. The changes effectively decouple search processes on the slave nodes and master-slave communications from each other. As a result, in the modified program each search process is independent from the rest of the searches.

We have described the computational implementation of an approach to narrow charge states of tandem mass spectra and identify low quality spectra. The program, 2to3, uses parent peptides' mass-to-charge ratio and its mass spectrum to estimate the charge state. It counts the number of fragment ion pairs that explains the dissociation of the precursor ion without a neutral loss. We show that there is a relationship between the number and Xcorr values calculated with SEQUEST. The charge state is determined based on this correlation. The state that has less than the threshold value of the ion pairs is eliminated. 2to3 requires very little time run, but elimination of the large number of low quality spectra may save up to %50 of the search time. The threshold values, and data filtering parameters are determined such that no protein identification is missed after 2to3. The paper analyzes results of running 2to3 in two different data sets, samples from *S. cerevisiae* 26S proteosome and human lens.

References

(1) Yates, J. R., III. Database searching using mass spectrometry data. *Electrophoresis* **1998**, *19*, 893−900.
(2) Patterson, S. D.; Aebersold, R.; Goodlett, D. R. Mass spectrometry-based methods for protein identification and phosphorylation site analysis. In *Proteomics: from protein sequnce to function*; Dunn, S. R. P. a. M. J., Ed.; Springer-Verlag: New York, 2001.
(3) Eng, J. K.; McCormack, A. L.; Yates, J. R., III. An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *J. Am. Soc. Mass Spectrom.* **1994**, *5*, 976−989.
(4) Perkins, D. N., et al. Probability-based protein identification by searching sequence database using mass spectrometry data. *Electrophoresis* **1999**, *20*, 3551−3567.
(5) Zhang, W.; Chait, B. T. ProFound: An Expert System for Protein Identification Using Mass Spectrometric Peptide Mapping Information. *Anal. Chem.* **2000**, *72*, 2482−2489.
(6) Verma, R., et al. Selective degradation of ubiquitinated Sic1 by purified 26S proteasome yields active S phase cyclin-Cdk. *Mol. Cells* **2001**, *8*, 439−448.
(7) MacCoss, M. J., et al. Shotgun Identification of Protein Modifications from Protein Complexes and Lens Tissue. *Proc. Natl. Acad. Sci., U.S.A.*, submitted.
(8) Yates, J. R.; I., McCormack, A. L.; Eng, J. Mining Genomes with MS. *Anal. Chem.* **1996**, *68*, 534A−540A.
(9) Geist, A., et al. *PVM: Parallel Virtual Machine*; MIT Press: Cambridge, MA, 1994.
(10) Mann, M.; Meng, C. K.; Fenn, J. B. Interpreting Mass Spectra of Multiply Charge Ions. *Anal. Chem.* **1989**, *61*, 1702−1708.
(11) Reinhold, B. B.; Reinhold, V. N. Electrospray Ionization Mass Spectrometry: Deconvolution by an Entropy-Based Algorithm. *J. Am. Soc. Mass Spectrom.* **1992**, *3*, 207−215.

PR015514R